



## Penerapan Metode *Economic Order Quantity* dan *Reorder Point* Pada Aplikasi Pengelolaan Persediaan Fumigasi

Farel Faleri<sup>1</sup>, Pantjawati Sudarmaningtyas<sup>2</sup>, Valentinus Roby Hananto<sup>3</sup>  
<sup>1,2,3</sup>S1 Sistem Informasi, Fakultas Teknologi dan Informatika, Univeritas Dinamika  
<sup>1</sup>farelfaleri@gmail.com, <sup>2</sup>pantja@dinamika.ac.id\*, <sup>3</sup>valentinus@dinamika.ac.id

### Abstract

Fumigation controls pests by sealing a room using methyl bromide ( $CH_3Br$ ) and oxygen gas. This process requires a gas mask, fumigation plastics, sand snakes, and a certificate. The interview revealed that specific supply shortages cause fumigation delays in 15% of monthly orders. This research aims to develop an inventory control application by applying the economic order quantity (EOQ) and reorder point (ROP) methods to overcome the problem of running out of stock. The EOQ method determines the amount of inventory to order to avoid excess stock, while ROP determines the right time to order inventory. The study involved two types of testing: white box testing and comparing manual calculations with application calculations to determine their consistency using data from March to May. The white box testing results show that the program's internal structure and logic run well. This is supported by evidence that each test case can pass two flow graphs resulting from cyclomatic complexity calculations. In addition, the results of testing the EOQ and ROP method calculations show an insignificant difference between the results of manual calculations and application calculations of 3.87%. This proves that application calculations for EOQ and ROP are running correctly and by the researchers' expectations.

**Keyword:** Reorder Point, Economic Order Quantity, Out of Stock, Fumigation.

### Abstrak

Fumigasi merupakan metode pengendalian hama yang melibatkan penggunaan *gas methyl bromide* ( $CH_3Br$ ) di dalam ruangan yang kedap udara dengan konsentrasi dan waktu tertentu. Untuk melakukan aktivitas fumigasi, beberapa peralatan yang diperlukan antara lain *gas methyl bromide*, gas oksigen, *Cartridge Gas Mask*, Plastik Fumigasi, *Sand Snake*, dan Kertas Sertifikat Fumigasi. Namun, berdasarkan wawancara dengan pihak perusahaan, sering terjadi kehabisan *stock* pada beberapa persediaan. Dalam waktu satu bulan 15% dari total pesanan mengalami penundaan aktivitas fumigasi akibat kekurangan persediaan. Penelitian ini bertujuan membangun aplikasi pengelolaan persediaan dengan menerapkan metode *economic order quantity* (EOQ) dan *reorder point* (ROP) untuk mengatasi masalah kehabisan *stock*. Metode EOQ digunakan untuk menentukan jumlah persediaan yang harus dipesan untuk menghindari kelebihan *stock*, sementara ROP digunakan menentukan waktu yang tepat untuk memesan persediaan. Penelitian ini melibatkan dua jenis pengujian yaitu pengujian *white box* dan perbandingan hasil perhitungan manual dengan aplikasi. Konsistensi perhitungan diuji menggunakan data dari bulan Maret hingga Mei. Hasil pengujian *white box* menunjukkan bahwa alur perhitungan dalam aplikasi berjalan dengan benar dan sesuai harapan. Selain itu, hasil pengujian perhitungan metode EOQ dan ROP menunjukkan selisih yang tidak signifikan antara hasil perhitungan manual dengan perhitungan aplikasi sebesar 3,87%. Hal ini membuktikan bahwa perhitungan aplikasi untuk ROP dan EOQ berjalan dengan benar dan sesuai dengan harapan peneliti.

**Kata kunci:** Reorder Point, Economic Order Quantity, Out of Stock, Fumigasi.

### 1. Pendahuluan

Pengendalian persediaan produk atau bahan baku merupakan hal penting dalam sebuah perusahaan. Persediaan atau produk yang dikendalikan dengan baik akan menjamin kelancaran operasional sehingga perusahaan dapat memenuhi kebutuhan pasar. Dengan demikian secara tidak langsung perusahaan juga memberi pelayanan yang baik kepada pelanggan karena semua pesanan pelanggan dapat terpenuhi. Dalam hal ini para pelanggan dapat merasa dihargai sehingga mereka pun loyal terhadap perusahaan [1]. Pengendalian

persediaan dapat membantu perusahaan mengurangi kemungkinan terjadinya keterlambatan persediaan yang diperlukan dalam operasional perusahaan, mengurangi pemesanan persediaan yang tidak sesuai, selain itu dapat membantu menghindari kenaikan harga persediaan secara mendadak [2]. Permasalahan umum yang sering terjadi dalam pengendalian persediaan adalah *outstock* atau *overstock*.

Perusahaan ini yang bergerak dalam bidang fumigasi menghadapi permasalahan terkait dengan persediaan bahan yang diperlukan untuk proses fumigasi.



Lisensi

Lisensi Internasional Creative Commons Attribution-ShareAlike 4.0.

Berdasarkan data yang diperoleh dalam satu bulan 15% dari total pesanan fumigasi *container* mengalami penundaan karena bahan yang diperlukan untuk fumigasi tidak tersedia.

Penundaan ini dapat mengakibatkan perusahaan mendapat denda dari depo dan *shipper*, yang berdampak adanya pengeluaran tak terduga hingga 10% dari pendapatan perbulan. Jika aktivitas fumigasi tetap dijalankan tanpa menggunakan *safety measure*, perusahaan dapat terkena sanksi dari pemerintah dalam jangka waktu tertentu karena membahayakan fumigator dan pekerja yang berkerja di sekitar area fumigasi. Hal tersebut dapat menimbulkan kerugian pada perusahaan karena tidak adanya pendapatan.

*Economic Order Quantity* (EOQ) merupakan metode yang dipakai untuk mengelola stok atau persediaan. Penggunaan metode ini bertujuan untuk menentukan jumlah barang yang akan dipesan sehingga mampu meminimalkan biaya pemesanan. Penerapan metode EOQ mampu meminimalisasi terjadinya *out of stock* dan *over stock* sehingga operasional perusahaan dapat berjalan lancar [3]. *Reorder Point* (ROP) adalah suatu titik dimana harus dilakukan penambahan persediaan sebelum mengalami kehabisan persediaan (*out of stock*). *Reorder Point* dihitung berdasarkan berapa banyak persediaan yang digunakan dengan lamanya pemesanan persediaan tersebut ditambah dengan *safety stock* persediaan tersebut [4].

Metode EOQ, ROP, dan *safety stock* sudah diimplementasikan pada beberapa penelitian sebelumnya. EOQ, ROP, dan *safety stock* dapat membantu perusahaan dalam menganalisis kebutuhan bahan baku agar proses produksi dapat berjalan sesuai jadwal, hal ini dibuktikan dengan pengujian pada satu bahan baku [5]. Hasil penelitian lain menunjukkan bahwa penerapan metode EOQ memberikan hasil yang lebih optimal dan ekonomis [6]. Hasil penelitian berupa sistem informasi *inventory management* berbasis web yang menerapkan ketiga metode tersebut menunjukkan bahwa sistem informasi yang dibangun dapat membantu pendataan inventori lebih transparan dan meminimalkan *human error* atau manipulasi data, walaupun hasil pengujian *black box* dan *User acceptance test* belum diuraikan secara rinci [4], [7].

Tujuan dari penelitian ini adalah memberikan solusi dengan membangun aplikasi persediaan yang menerapkan metode ROP dan EOQ untuk meminimalkan terjadinya *outstock* atau *overstock*. Pengujian terhadap aplikasi yang dihasilkan dilakukan dengan metode *white box testing* serta membandingkan hasil perhitungan aplikasi dan manual.

## 2. Metode Penelitian

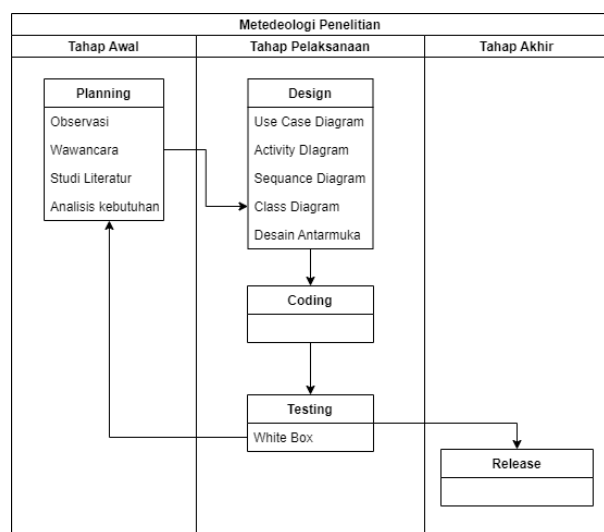
Gambar 1 merupakan tahapan penelitian yang digunakan untuk mencapai tujuan dari penelitian ini. Metode penelitian terdiri atas 3 tahap yaitu, tahap awal,

tahap pelaksanaan, dan tahap akhir. Setiap tahapan mengandung nilai-nilai dari *extreme programming* yaitu *Communication, Courage, Simplicity, Feedback, dan Quality work* [8].

### 2.1. Planning

Tahap *planning* bertujuan untuk mengidentifikasi permasalahan yang muncul dan merumuskan solusi yang sesuai. Beberapa kegiatan yang dilakukan untuk mencapai tujuan tersebut meliputi observasi, wawancara, studi literatur, dan analisis kebutuhan sistem [9].

Pada penelitian ini, dilakukan studi literatur dengan tujuan utama untuk mengidentifikasi permasalahan yang dihadapi oleh perusahaan. Studi literatur ini bertujuan untuk memperoleh pemahaman yang mendalam mengenai masalah yang sedang diteliti serta untuk menggali pengetahuan, metode, dan solusi yang relevan dalam bidang tersebut. Dengan melakukan studi literatur, diharapkan dapat ditemukan berbagai teori yang terkait dengan permasalahan yang sedang dihadapi, yang kemudian dapat digunakan sebagai referensi dalam memberikan solusi yang tepat. Hasil dari studi literatur yang telah dipelajari dan menjadi dasar penelitian ini adalah teori-teori yang terkait langsung dengan pengendalian persediaan yaitu *Economic Order Quantity* (EOQ), *Reorder Point* (ROP), dan *safety stock* (SS).



Gambar 1. Metode Penelitian

*Economic Order Quantity* (EOQ) merujuk pada jumlah pembelian persediaan yang paling ekonomis untuk setiap siklus pembelian. Tujuan dari metode EOQ adalah untuk mencapai persediaan yang minimal, biaya yang rendah, dan meningkatkan kualitas. Dengan merencanakan dan mengimplementasikan EOQ di dalam perusahaan, dapat mengurangi risiko kehabisan stok, mencegah gangguan operasional, dan mengoptimalkan pengelolaan persediaan dengan menghemat biaya. Dengan menerapkan metode EOQ, perusahaan dapat mencapai efisiensi dalam pengelolaan

persediaan, mengurangi biaya penyimpanan, memaksimalkan pemanfaatan ruang gudang, dan menghindari masalah yang disebabkan oleh akumulasi persediaan yang berlebihan [7]. Rumus perhitungan EOQ yang akan digunakan pada aplikasi pengelolaan persediaan pada perusahaan ini, lihat formula 1.

$$EOQ = \sqrt{\frac{2DS}{H}} \quad (1)$$

Untuk menghitung EOQ dibutuhkan beberapa variabel yaitu S sebagai biaya pesan untuk per pesanan dalam satuan (Rp), D sebagai jumlah pemesanan per periode dalam satuan (Kg), H sebagai biaya penyimpanan per unit per periode dalam satuan (Rp).

*Reorder Point* (ROP) adalah titik di mana perlu dilakukan pemesanan persediaan kembali (formula 2) agar kedatangan persediaan yang dipesan dapat tepat waktu, dan *safety stock* mencapai nol. Pemesanan persediaan ini dilakukan secara berulang sesuai dengan kebutuhan[10].

$$ROP = (LT \times D) + SS \quad (2)$$

Untuk menghitung ROP dibutuhkan beberapa variabel yaitu LT sebagai *lead time* dalam satuan (hari), D sebagai rata rata *demand* per periode dalam satuan (Kg), SS sebagai *safety stock* dalam satuan (Kg).

Jumlah pemesanan per periode (D), yang digunakan pada pada rumus EOQ dan ROP, dihitung berdasarkan jumlah permintaan persediaan dibagi dengan periode. Formula 3 adalah notasi dari perhitungan jumlah pemesanan per periode (D).

$$D = \frac{\text{Jumlah Pemintaan Persediaan}}{\text{Periode}} \quad (3)$$

*Safety Stock* merupakan stok cadangan persediaan yang bertujuan untuk mengurangi risiko kekurangan persediaan. Adanya *safety stock* membantu menghindari kerugian yang dapat terjadi ketika persediaan habis. *Safety stock* diperlukan untuk mengatasi ketidakpastian dalam permintaan pelanggan atau keterlambatan dalam pengiriman persediaan. Dengan adanya *safety stock*, perusahaan dapat menjaga kelancaran operasional dan memastikan ketersediaan persediaan yang memadai dalam situasi yang tidak terduga.[5].

$$SS = (\text{Permintaan MAX} - K) \times LT \quad (4)$$

Untuk menghitung *safety stock* dibutuhkan beberapa variabel yaitu LT sebagai *lead time* dalam satuan (hari), K sebagai rata rata permintaan per periode dalam satuan (Kg).

Dua nilai *extreme programming* terdapat dalam tahapan ini adalah *Communication*, dan *Quality work*. *Communication* penting dalam tahap *planning* karena untuk memperoleh data-data yang dibutuhkan dalam penelitian ini dibutuhkan komunikasi antara peneliti dengan pihak perusahaan. *Quality work* penting dalam

tahap ini untuk menghasilkan luaran yang baik dan sesuai harapan.

## 2.2. Design

Tahap analisis dan desain sistem dilakukan dalam tahap ini, di mana hasilnya akan menjadi panduan untuk tahap konstruksi. Tujuan dari tahap ini adalah untuk mencegah ketidaksesuaian dengan persyaratan yang telah ditetapkan.

Tahap ini terdiri dari beberapa bagian, termasuk *use case diagram*, *activity diagram*, *sequence diagram*, *class diagram*, dan *desain* antarmuka[11].

## 2.3. Coding

Pada tahap ini, dilakukan pengembangan sistem menggunakan *framework* Laravel dan MySQL sebagai sistem basis data. Pengembangan sistem mengacu pada rancangan sistem dan *desain* antarmuka yang telah disiapkan.

Pada tahap ini dan tahap sebelumnya, terdapat nilai-nilai yang sama dengan tahap desain. Nilai-nilai tersebut meliputi keberanian (*courage*), kesederhanaan (*simplicity*), dan kualitas kerja (*quality work*). *Courage* penting pada tahap ini karena membutuhkan kepercayaan diri dan keyakinan saat merancang dan mengimplementasikan metode perhitungan dalam aplikasi. Selanjutnya, *simplicity* diperlukan pada tahap ini untuk menghasilkan desain dan alur aplikasi yang *user friendly*, sehingga memudahkan pengguna saat menggunakan aplikasi tersebut. Terakhir, *quality work* berfokus pada pencapaian keseimbangan dalam kualitas kerja, sehingga menghasilkan penelitian yang baik dan sesuai dengan harapan.

## 2.4. Testing

Pada tahap ini, dimulai proses pengujian aplikasi yang telah dikembangkan pada tahap sebelumnya. Pengujian dilakukan menggunakan metode *white box*, dengan tujuan untuk memverifikasi apakah aplikasi beroperasi sesuai dengan rancangan sistem yang telah dirancang pada tahap desain. *White box*, yang juga dikenal sebagai *clear box* atau *glass box*, merupakan metode pengujian perangkat lunak di mana pengujian dilakukan dengan pemahaman yang mendalam tentang struktur internal, desain, dan implementasi dari sistem yang sedang diuji. Dalam konteks pengujian perangkat lunak, *white box* sering digunakan sebagai kebalikan dari *black box testing*, di mana fokusnya adalah pada pemahaman internal sistem yang sedang diuji [12]. Dalam pengujian *white box*, penguji memiliki akses lengkap ke kode sumber, struktur data, aliran eksekusi, dan komponen internal lainnya dari sistem yang sedang diuji. Dalam pengujian *white box*, *flow graph* digunakan sebagai alat bantu untuk menggambarkan visual aliran kontrol dalam program atau kode sumber aplikasi. *Flow graph* mengilustrasikan bagaimana program berpindah dari satu pernyataan ke pernyataan lainnya melalui

percabangan keputusan seperti *sequence, if-else, if-then, while, untils*, dan *case*[13].

*Cyclomatic complexity* merupakan nilai yang dihasilkan dari *flow graph* dengan menghitung jumlah keputusan independen dalam program. Keputusan independen mencakup percabangan keputusan *sequence, if-else, if-then, while, untils*, dan *case*[14].

$$\text{Cyclomatic Complexity } V(G) = E - N + 2 \quad (5)$$

Untuk menghitung *cyclomatic complexity* dibutuhkan beberapa variabel yaitu E sebagai jumlah *edge* (panah) pada *flow graph*, N sebagai jumlah *node* (titik) pada *flow graph*.

*Graph matrix* digunakan untuk menggambarkan hubungan antara simpul-simpul dan sisi-sisi dalam *flow graph*. Matriks ini berisi entri-entri yang menunjukkan keberadaan *edge* yang menghubungkan dua *node* dalam *flow graph*.

Untuk tahapan ini, terdapat dua nilai penting dalam metodologi penelitian, yaitu umpan balik (*feedback*) dan kualitas kerja (*quality work*). *feedback* menjadi penting dalam tahap pengujian karena diperlukan masukan dari pengguna untuk menemukan kekurangan dalam aplikasi dan membantu memperoleh pemahaman yang lebih baik tentang aplikasi tersebut. Selanjutnya, *quality work* menjadi penting dalam tahap pengujian karena penelitian ini harus memiliki kualitas kerja yang seimbang, sehingga setiap tahap penelitian menghasilkan hasil yang baik dan sesuai dengan harapan.

### 3. Hasil dan Pembahasan

#### 3.1. Planning

Tahap ini melibatkan pengumpulan informasi dari berbagai sumber, termasuk observasi, wawancara, studi literatur, dan analisis kebutuhan sistem. Identifikasi masalah yang diperoleh dari hasil wawancara dan observasi menunjukkan terjadinya *out of stock* yang mengakibatkan penundaan proses fumigasi. Identifikasi masalah dan solusi yang ditawarkan disajikan pada Tabel 1.

Tabel 1. Identifikasi Masalah

Identifikasi Masalah	
Masalah yang Ditemukan	Dampak
15% pesanan pelanggan tertunda yang disebabkan dari kurangnya persediaan untuk melaksanakan proses fumigasi dikarenakan kurangnya monitoring persediaan yang diperlukan untuk melaksanakan proses fumigasi.	Terjadinya <i>out of stock</i> pada persediaan yang dibutuhkan untuk melaksanakan proses fumigasi dan banyaknya penundaan pesanan proses fumigasi, lalu jika melakukan proses fumigasi tanpa mematuhi <i>safety measure</i> dapat melanggar <i>safety measure</i> dari pemerintah
Solusi	
Membuat aplikasi pengendalian persediaan menggunakan metode ROP dan EOQ yang bertujuan untuk memberitahukan pemesanan persediaan yang tepat untuk menghindari <i>out of stock</i> dan jumlah persediaan yang dibeli.	

Langkah selanjutnya adalah melakukan proses wawancara untuk mengumpulkan data terkait kendala dan masalah yang terkait dengan pengendalian persediaan gas fumigasi. Dalam tahap ini, wawancara dilakukan secara langsung dengan 5 responden, termasuk 2 pihak internal perusahaan yaitu Direktur/*Top Management* dan Manajer Teknis serta 3 responden dari pihak eksternal. Selain itu, dilakukan studi literatur yang bertujuan untuk memahami metode EOQ sebagai panduan untuk menentukan jumlah barang yang dipesan, serta metode ROP sebagai panduan untuk menambah persediaan sebelum mengalami *out of stock*.

Hasil dari tahap perencanaan berupa kebutuhan fungsional pengguna yang diturunkan dari tugas dan tanggung jawab masing-masing pengguna. Terdapat 3 pengguna yaitu fumigator, staff administrasi, dan manajer. Fumigator merupakan aktor utama yang memiliki 7 fungsional. Staff administrasi hanya memiliki 1 kebutuhan fungsional, sedangkan 5 kebutuhan fungsional dimiliki oleh manajer. Kebutuhan fungsional staff administrasi menjadi dasar untuk kebutuhan fungsional fumigator dan manajer. Fungsional fumigator berisi proses-proses utama terkait EOQ dan ROP, sedangkan manajer lebih fokus pada pencetakan laporan. Uraian lengkap dari pengguna dan kebutuhan fungsionalnya terdapat pada Tabel 2.

Tabel 2. Identifikasi Kebutuhan Fungsional

No	Pengguna	Kebutuhan Fungsional
1.	Fumigator	<ul style="list-style-type: none"> <li>• Mengecek Data Persediaan.</li> <li>• Mengelola Data Penggunaan Persediaan.</li> <li>• Mengelola Data Pemesanan Persediaan</li> <li>• Mengecek Data Pemasukan Persediaan.</li> <li>• Mengecek Data Pesanan</li> <li>• Perhitungan ROP</li> <li>• Perhitungan EOQ</li> </ul>
2	Staff Administrasi	<ul style="list-style-type: none"> <li>• Mengelola Data Pesanan.</li> </ul>
3	Manager	<ul style="list-style-type: none"> <li>• Mencetak Laporan Pesanan.</li> <li>• Mencetak Laporan Persediaan.</li> <li>• Mencetak Laporan Penggunaan Persediaan.</li> <li>• Mencetak Laporan Pemasukan Persediaan.</li> <li>• Mencetak Laporan pemesanan Persediaan</li> </ul>

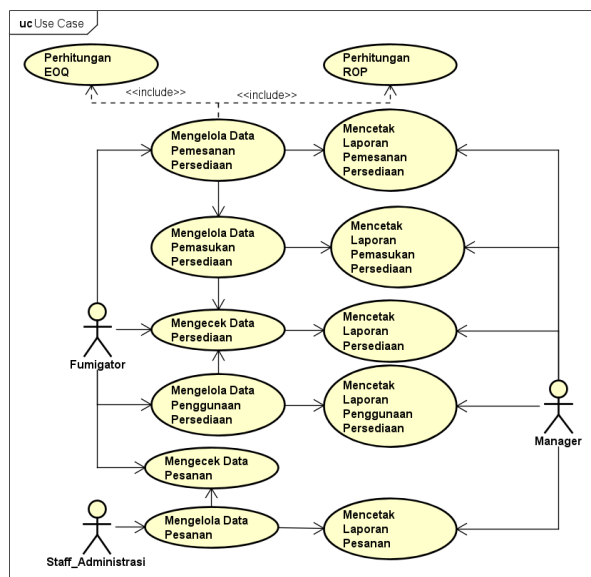
Proses fumigasi melibatkan 2 bahan dan 4 alat. Bahan fumigasi berupa gas *Methyl Bromide* ( $\text{CH}_3\text{Br}$ ) dan oksigen dengan konsentrasi yang berbeda untuk setiap jenis *container*. Sesuai standar yang berlaku, proses fumigasi dilengkapi dengan alat berupa *cartridge gas mask*, plastik fumigasi, dan *sand snake*. Sebagai bukti bahwa *container* sudah difumigasi, maka diterbitkan sertifikat fumigasi [15]. Informasi *bill of material* berupa rincian bahan dan alat yang digunakan dalam proses fumigasi untuk setiap jenis *container* dapat ditemukan pada Tabel 3.

Tabel 3. *Bill Of Material*

No	Jenis Container	Nama Persediaan	Penggunaan
1	Container 20ft	Gas Methyl Bromide (CH <sub>3</sub> Br)	3,5 KG
		Gas Oksigen	2 Tabung
		Cartridge Gas Mask	2 Cartridge
		Plastik Fumigasi	1 Plastik Fumigasi
		Sand Snake	34 Sandsnake
		Kertas Sertifikat Fumigasi	9 Lembar
2	Container 40ft	Gas Methyl Bromide (CH <sub>3</sub> Br)	5,5 KG
		Gas Oksigen	2 Tabung
		Cartridge Gas Mask	2 Cartridge
		Plastik Fumigasi	1 Plastik Fumigasi
		Sand Snake	44 Sandsnake
		Kertas Sertifikat Fumigasi	9 Lembar

### 3.2. Design

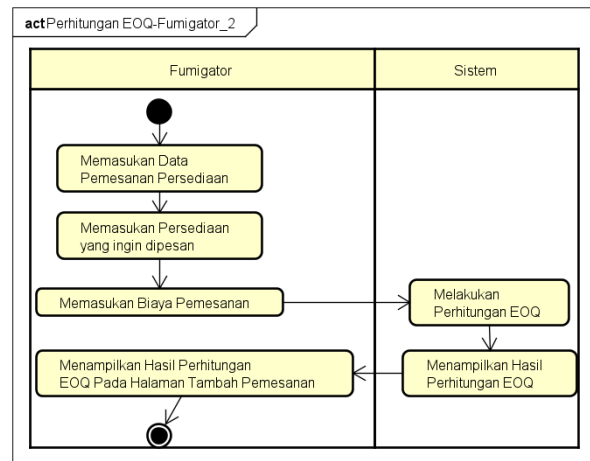
Dalam pembuatan *use case diagram*, digunakan hasil identifikasi kebutuhan sistem yang telah dihasilkan pada tahap sebelumnya. Diagram ini menggambarkan interaksi yang dapat dilakukan oleh admin dalam sistem. Terdapat tiga aktor dalam *use case diagram* ini, yaitu Fumigator, Staff Administrasi, dan Manajer. Aktivitas yang dilakukan oleh Fumigator meliputi pengecekan data persediaan, pengelolaan data penggunaan persediaan, pengecekan data pesanan, pengecekan data pemasukan persediaan, dan pengelolaan pemesanan persediaan. Sedangkan aktivitas yang dilakukan oleh Staff Administrasi meliputi pengelolaan data pesanan. Terakhir, Manajer memiliki kegiatan mencetak laporan persediaan, laporan pemasukan persediaan, laporan penggunaan persediaan, laporan pemesanan persediaan, dan laporan pesanan. *Use case diagram* dapat ditemukan pada Gambar 2.



Gambar 2. *Use Case Diagram*

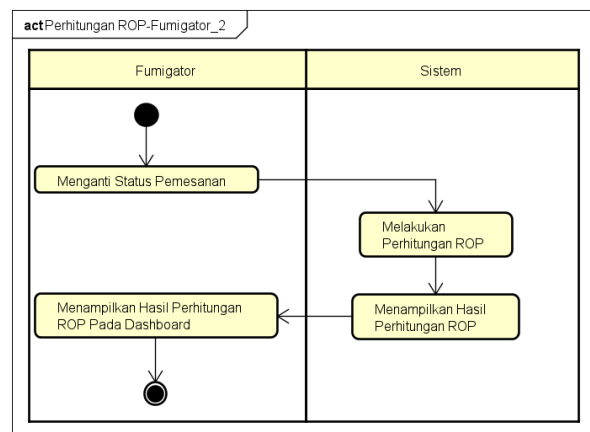
*Use case diagram* akan menjadi dasar untuk pembuatan *activity diagram*. *Activity diagram* mengilustrasikan alur sistem saat pengguna berinteraksi dengan sistem.

Pada *activity diagram*, terdapat aktivitas perhitungan EOQ yang dilakukan oleh fumigator. Dalam aktivitas ini, dimulai ketika fumigator ingin menambah data pemesanan persediaan, setelah fumigator memasukkan biaya pemesanan untuk pemesanan tersebut, sistem akan menerima biaya pemesanan tersebut dan mulai melakukan proses perhitungan EOQ, setelah perhitungan EOQ selesai, sistem akan menampilkan hasil perhitungan EOQ kepada fumigator. Informasi lebih lanjut dapat ditemukan pada Gambar 3.



Gambar 3. *Activity Diagram* Perhitungan EOQ

Pada *activity diagram* perhitungan ROP yang dilakukan oleh fumigator. Dalam aktivitas ini dimulai Ketika fumigator persediaan yang dipesan telah sampai, fumigator akan mengganti status pemesanan, setelah sistem mengetahui bahwa status pemesanan telah diubah, akan mulai melakukan perhitungan ROP, setelah perhitungan ROP selesai, akan ditampilkan kepada fumigator. Rincian lebih lanjut dapat ditemukan pada Gambar 4.

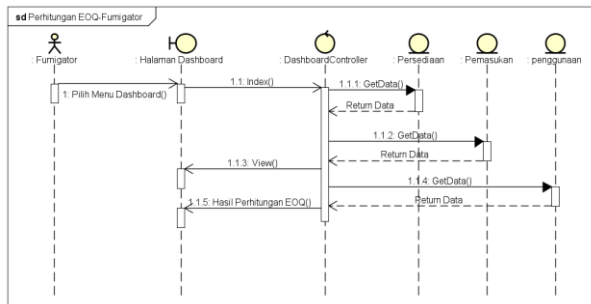


Gambar 4. *Activity Diagram* Perhitungan ROP

Setelah pembuatan *activity diagram*, tahap selanjutnya adalah pembuatan *sequence diagram*. *Sequence diagram*

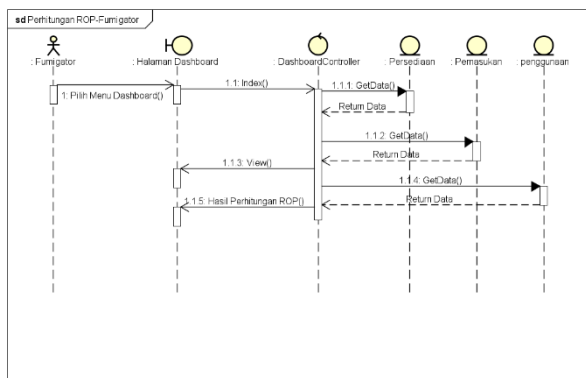
menggambarkan interaksi antara objek-objek yang terkait dalam *use case diagram*. *Sequence diagram* ini mengilustrasikan urutan operasi yang dilakukan oleh objek-objek tersebut dari kiri ke kanan berdasarkan waktu terjadinya dalam pesan yang dikirimkan.

*Sequence diagram* perhitungan EOQ memberikan penjelasan rinci tentang proses yang terjadi dalam *activity diagram* perhitungan EOQ. Diagram ini menggambarkan dengan detail alur yang terjadi dalam proses perhitungan EOQ dan dapat dilihat pada Gambar 5.



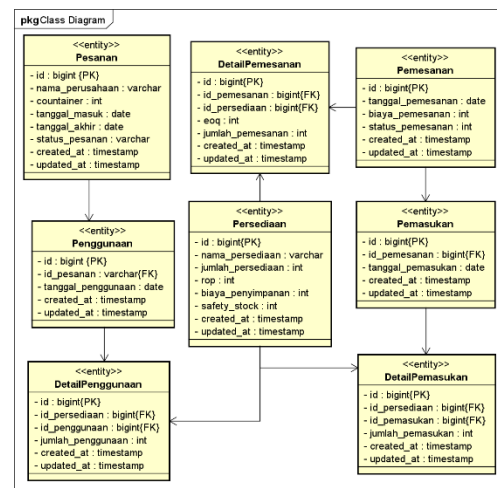
Gambar 5. *Sequence Diagram* Perhitungan EOQ

*Sequence diagram* perhitungan ROP memberikan penjelasan rinci tentang proses yang terjadi dalam *activity diagram* perhitungan ROP. Diagram ini menggambarkan dengan detail alur yang terjadi dalam proses perhitungan ROP dan dapat dilihat pada Gambar 6.



Gambar 6. *Sequence Diagram* Perhitungan ROP

Setelah dibuatnya *sequence diagram*, langkah selanjutnya adalah membuat *class diagram*. *Class diagram* digunakan untuk menggambarkan struktur aplikasi, termasuk tampilan halaman yang akan ditampilkan, *database* yang digunakan untuk menyimpan data, dan *controller* yang diperlukan untuk menjalankan halaman serta manipulasi data yang dilakukan, *class diagram* ini dapat ditemukan pada Gambar 7.



Gambar 7. *Class Diagram*

### 3.3. Coding

Hasil implementasi sistem pada tahap *coding* ditampilkan dalam aplikasi yang telah dibangun.

Persediaan				
Id	Nama Persediaan	Jumlah Persediaan	Safety Stock	ROP
1	Gas Methyl Bromide	50	21	114
2	Gas Oksigen	50	1	5
3	Cartridge Gas Mask	50	3	5
4	Plastik Fumigasi	50	5	7
5	Sand Snake	50	53	65
6	Kertas Sertifikat Fumigasi	50	36	46

Gambar 8. Implementasi Perhitungan ROP

Gambar 8 memperlihatkan implementasi aplikasi untuk perhitungan ROP. Pada aplikasi ini, ketika pengguna melakukan pemesanan melalui halaman pemesanan, lalu pengguna mengubah status pemesanan dari "Dalam Perjalanan" menjadi "Persediaan Telah Sampai", sistem akan memulai perhitungan ROP. Hasil perhitungan ROP dapat dilihat melalui halaman dashboard dan halaman persediaan.

No	Nama Persediaan	EOQ	Jumlah Pemesanan
1	Gas Methyl Bromide	528	528
2	Gas Oksigen	112	112
3	Cartridge Gas Mask	82	82
4	Plastik Fumigasi	93	93
5	Sand Snake	202	202
6	Kertas Sertifikat Fumigasi	187	187

Gambar 9. Implementasi Perhitungan EOQ

Gambar 9 menunjukkan implementasi aplikasi untuk perhitungan EOQ. Dalam aplikasi ini, ketika pengguna ingin memilih persediaan yang akan dipesan, sistem akan memulai perhitungan EOQ hanya untuk persediaan yang dipilih oleh pengguna. Sistem tidak akan

melakukan perhitungan EOQ untuk persediaan yang tidak dipilih. Hasil perhitungan EOQ dapat dilihat melalui halaman pemesanan ketika pengguna ingin memilih persediaan yang akan dipesan.

### 3.4. Testing

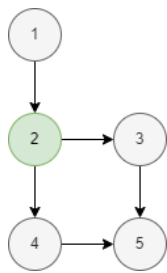
Metode *white box testing* diterapkan untuk memastikan kebenaran alur proses perhitungan ROP dan EOQ pada aplikasi yang dibangun. Pengujian *white box* didasarkan pada *source code* ROP dan EOQ. Gambar 10 menunjukkan *source code* ROP.

```

1 public function changeStatus($id){
2
3     $dbpemesanan = Pemesanan::select('status_pemesanan')->where('id', $id)->first();if
4     ($dbpemesanan->status_pemesanan == 1) {
5
6         $status_pemesanan = 0;}
7
8     else { $status_pemesanan = 1;$details = DB::table('detail_pemesanan as dp')-
9     >join('pemesanan as p', 'dp.id_pemesanan', '=', 'p.id')->where('dp.id_pemesanan', '=',
10     $id)->get();$bulan_tahun = DB::table('penggunaan')
11     >selectRaw('DATE_FORMAT(MAX(tanggal_penggunaan), "%m-%Y") as bulan')
12     >whereRaw('DATE_FORMAT(tanggal_penggunaan, "%m-%Y") < DATE_FORMAT(now(), "%m-%Y")')-
13     >first();$month_before = Carbon::createFromFormat('m-Y', $bulan_tahun->bulan)-
14     >subMonth(2)->format('m-Y');$subdate = Carbon::createFromFormat('d-m-Y', '01' . "-" .
15     $month_before->format('Y-m-d'));$lastdate = Carbon::createFromFormat('d-m-Y', '01' . "-" .
16     $bulan_tahun->bulan->addDay($first->jumlah_hari($bulan_tahun->bulan))->format('Y-m-
17     d');$lastdateparse = Carbon::parse($lastdate);$subdateparse =
18     Carbon::parse($subdate);$jumlah_hari = $lastdateparse-
19     >diffInDays($subdateparse);$avg_date = DB::table('pemesanan as pms')->join('pemesanan as
20     psn', 'pms.id_pemesanan', '=', 'psn.id')->selectRaw('round(avg(DATEDIFF(
21     pms.tanggal_pemesanan, psn.tanggal_pemesanan)) as lead_time')-
22     >where('psn.status_pemesanan', 1)->whereBetween('pms.tanggal_pemesanan', [$subdate,
23     $lastdate])->first();foreach ($details as $detail) {$data = DB::table('detail_penggunaan
24     as dp')->join('penggunaan as p', 'dp.id_penggunaan', '=', 'p.id')->join('persediaan as
25     pd', 'dp.id_persediaan', '=', 'pd.id')->selectRaw('max(dp.jumlah_penggunaan) as max,
26     round(avg(dp.jumlah_penggunaan)) as avg, sum(dp.jumlah_penggunaan) as total')-
27     >whereRaw('pd.id = "' . $detail->id_persediaan . '" AND
28     DATE_FORMAT(p.tanggal_penggunaan, "%m-%Y") >= "' . $month_before . '" AND
29     DATE_FORMAT(p.tanggal_penggunaan, "%m-%Y") <= "' . $bulan_tahun->bulan . '"')-
30     >first();$lead_time = empty($avg_date->lead_time) ? $avg_date->lead_time : 3;$ss =
31     ($data->max - $data->avg) * $lead_time;$d = (int)round($data->total / $jumlah_hari);$rop =
32     ($d * $lead_time) + $ss;$persediaan=where('id', $detail->id_persediaan)->update(['rop'
33     => $rop, 'safety_stock' => $ss]);Pemesanan::where('id', $id)->update(['status_pemesanan'
34     => $status_pemesanan]);return redirect()->back();
35
36 }
    
```

Gambar 10. Source Code ROP

Source code tersebut, selanjutnya diturunkan menjadi *flow graph* sebagaimana terlihat pada Gambar 11. Dari gambar tersebut terlihat bahwa proses perhitungan ROP terdiri atas 5 *edge* (E), 5 *node* (N), dan 1 *predicate* (P).



Gambar 11. Flow Graph ROP

Berdasarkan jumlah *edge* dan *node*, hasil perhitungan *cyclomatic complexity* menggunakan rumus (5) menghasilkan 2 *region*. Setiap *region* memiliki alur *path* sebagaimana disajikan pada Tabel 4.

Tabel 4. Alur Path ROP

Path	Alur Basic Path
Path ke-1	1,2,3,5
Path ke-2	1,2,4,5

*Graph matrix* pada Gambar 12 memvisualisasikan jalur independen dalam *source code* ROP. Total pada *graph matrix* menunjukkan nilai yang sama dengan hasil perhitungan *cyclomatic complexity*.

	1	2	3	4	5	
1		1				1-1=0
2			1	1		2-1=0
3					1	1-1=0
4					1	1-1=0
5						
						1+1=2

Gambar 12. Graph Matrix ROP

Untuk memastikan setiap perhitungan ROP menghasilkan *output* yang diharapkan, maka dilakukan *test case* pada setiap *path*. Hasil *test case* ROP yang disajikan pada Tabel 5, menunjukkan bahwa semua *path* terlewati.

Tabel 5. Hasil Test Case ROP

Path	Skenario Uji	Hasil Yang Diharapkan	Hasil Uji	Hasil Alur
1.	Ketika Status Pemesanan "Dalam Perjalanan"	Tidak Melakukan Perhitungan	Berhasil	[v]Terlewati []Tidak Terlewati
2.	Ketika Status Pemesanan Berubah Menjadi "Persediaan Telah Sampai"	Mendapatkan Hasil Perhitungan ROP	Berhasil	[v]Terlewati []Tidak Terlewati

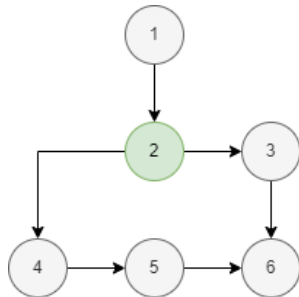
Berdasarkan hasil *test case*, menunjukkan kebenaran dari logika program perhitungan ROP. Hal ini juga didukung dengan tidak adanya perbedaan antara hasil perhitungan *cyclomatic complexity* dengan total dari *graph matrix*.

```

1 public function hitungEOQ(Request $request){
2
3     $request->validate(['biaya_pemesanan' => 'required',
4
5     ]);
6
7     $bulan_tahun = DB::table('penggunaan')-
8     >selectRaw('DATE_FORMAT(MAX(tanggal_penggunaan), "%m-%Y") as bulan')-
9     >whereRaw('DATE_FORMAT(tanggal_penggunaan, "%m-%Y") < DATE_FORMAT(now(), "%m-%Y")')-
10    >first();$temporary_pemesanan = session('temporary_pemesanan');if($request->
11    biaya_pemesanan; $no = 1;$month_before = Carbon::createFromFormat('m-Y', $bulan_tahun->
12    bulan)->subMonth(2)->format('m-Y');foreach ($temporary_pemesanan as $temp) {$data =
13    DB::table('detail_penggunaan as dp')->join('penggunaan as p', 'dp.id_penggunaan', '=',
14    'p.id')->join('persediaan as ps', 'dp.id_persediaan', '=', 'ps.id')-
15    >selectRaw('max(dp.jumlah_penggunaan) as max, round(avg(dp.jumlah_penggunaan)) as avg,
16    sum(dp.jumlah_penggunaan) as total')->whereRaw('ps.id = "' . $temp['id'] . '" AND
17    DATE_FORMAT(p.tanggal_penggunaan, "%m-%Y") >= "' . $month_before . '" AND
18    DATE_FORMAT(p.tanggal_penggunaan, "%m-%Y") <= "' . $bulan_tahun->bulan . '"')->first();
19
20    $persediaan =
21    $persediaan->findorfail($temp['id']);$temporary_pemesanan[$temp['id']]['eq'] = $data-
22    >total > 0 ? round(sqrt((2 * $data->total * $request->biaya_pemesanan) / $persediaan-
23    >biaya_penjimpanan)) : 0;$temporary_pemesanan[$temp['id']]['jumlah_pemesanan'] = $data-
24    >total > 0 ? round(sqrt((2 * $data->total * $request->biaya_pemesanan) / $persediaan-
25    >biaya_penjimpanan)) : 0;$temporary_pemesanan[$temp['id']]['biaya_pemesanan'] =
26    $request->biaya_pemesanan;$session(['temporary_pemesanan' =>
27    $temporary_pemesanan]);$temporary_pemesanan = session('temporary_pemesanan');return
28    redirect()->route('tambah.pemesanan');public function ubahJumlah(Request
29    $request){$request->validate(['jumlah_pemesanan' =>
30    'required']);$temporary_pemesanan =
31    session('temporary_pemesanan');$temporary_pemesanan[$request->id]['jumlah_pemesanan'] =
32    $request->jumlah_pemesanan;session(['temporary_pemesanan' =>
33    $temporary_pemesanan]);return redirect()->route('tambah.pemesanan');
34
35 }
    
```

Gambar 13. Source Code EOQ

Metode *white box* juga digunakan untuk menguji kebenaran logika perhitungan EOQ. Gambar 13 memperlihatkan *source code* EOQ yang dibangun pada penelitian ini. *Flow graph* pada Gambar 14 dibuat berdasarkan *source code* perhitungan EOQ menunjukkan adanya 6 *edge* (E), 6 *node* (N), dan 1 *predicate* (P).



Gambar 14. *Flow Graph* EOQ

Hasil perhitungan *cyclomatic complexity* EOQ diperoleh dua *path* yang menunjukkan bahwa alur perhitungan EOQ hanya memiliki dua kemungkinan *output*. Tabel 6 menyajikan *path* pada setiap kemungkinan tersebut.

Tabel 6. Alur *Path* EOQ

<i>Path</i>	Alur Basic <i>Path</i>
<i>Path</i> ke-1	1,2,3,6
<i>Path</i> ke-2	1,2,4,5,6

Jalur independen *source code* perhitungan EOQ divisualisasikan melalui *graph matrix*. Gambar 15, menunjukkan bahwa total *path* pada *graph matrix* menunjukkan angka yang sama dengan hasil perhitungan *cyclomatic complexity*.

	1	2	3	4	5	6	
1		1					1-1=0
2			1	1			2-1=1
3						1	1-1=0
4					1		1-1=0
5						1	1-1=0
6							
							1+1=2

Gambar 15. *Graph Matrix* EOQ

Tabel 7. Hasil *Test Case* EOQ

<i>Path</i>	Skenario Uji	Hasil Yang Diharapkan	Hasil Uji	Hasil Alur
1.	Tidak Melakukan Pemesanan	Tidak Melakukan Perhitungan EOQ	Berhasil	[v]Terlewati []Tidak Terlewati
2.	Memasukan Biaya Pemesanan	Mendapatkan Hasil Perhitungan EOQ	Berhasil	[v]Terlewati []Tidak Terlewati

Untuk memastikan bahwa tidak ada kesalahan dalam logika perhitungan EOQ, maka dilakukan *test case*

dengan menguji dua *path* dari proses perhitungan untuk memverifikasi hasil *output* pada setiap *path* sesuai dengan harapan. Hasil pengujian pada dua *path* dalam proses perhitungan EOQ disajikan pada **Error! Not a valid bookmark self-reference.**

Berdasarkan hasil *test case* pada Tabel 7, dapat disimpulkan bahwa struktur internal dan logika program perhitungan EOQ berjalan dengan baik. Hal ini juga didukung hasil perhitungan *cyclomatic complexity* dan *graph matrix* yang menunjukkan hasil yang konsisten.

Berdasar hasil *white box testing* yang dilakukan pada proses perhitungan ROP dan EOQ menunjukkan tidak adanya kesalahan pada logika program. Dengan demikian dapat disimpulkan bahwa aplikasi yang dibangun pada penelitian ini dapat digunakan dengan baik dalam konteks perusahaan.

Setelah menyelesaikan *white box testing*, langkah selanjutnya adalah melakukan pengujian perhitungan ROP dan EOQ. Tujuan dari pengujian ini adalah untuk membuktikan bahwa hasil perhitungan dalam aplikasi sesuai dan konsisten dengan perhitungan manual. Hasil rekap data permintaan dalam satu periode terdapat pada Tabel 8.

Tabel 8. Jumlah Permintaan Persediaan Dalam 1 Periode (Maret-Mei)

Nama Persediaan	Total Pembelian Persediaan
Gas Methyl Bromide	2930
Gas Oksigen	160
Cartridge Gas Mask	86
Plastik Fumigasi	50
Sand Snake	370
Kertas Sertifikat Fumigasi	392

Tabel 9 menampilkan ringkasan pembelian persediaan selama periode Maret-April-Mei.

Tabel 9. Rekap Data Permintaan Dalam 1 Periode (Maret-Mei)

Nama Persediaan	Permintaan MAX	Permintaan Rata-Rata	Lead Time	Total Permintaan
Gas Methyl Bromide	96	88,75	3	2794
Gas Oksigen	4	3,71875	3	123
Cartridge Gas Mask	3	2	3	62
Plastik Fumigasi	10	8,285714286	3	58
Sand Snake	120	102,5	3	350
Kertas Sertifikat Fumigasi	70	58,33333333	3	410

Data pada Tabel 9 terdiri atas permintaan maksimum (MAX), permintaan rata-rata, dan *lead time*. Permintaan maksimum (MAX) mengindikasikan permintaan tertinggi dalam periode tersebut, sedangkan permintaan rata-rata merupakan nilai rata-rata permintaan selama periode tersebut. *Lead time* mengukur waktu yang dibutuhkan mulai dari memesan persediaan hingga



persediaan tersebut diterima oleh perusahaan, dan terdapat total permintaan pada periode tersebut. Variabel-variabel ini akan digunakan dalam perhitungan manual ROP dan EOQ.

Perhitungan *manual ROP*, diawali dengan menghitung *demand, safety stock*. Perhitungan *demand (D)* untuk Gas *Methyl Bromide* didasarkan pada total permintaan selama periode Maret sampai Mei yaitu 2794 kg dengan periode 92 hari. Dengan menggunakan rumus (3) maka didapatkan hasil permintaan persediaan Gas *Methyl Bromide* sebesar 30,36956522 kg/hari.

Untuk menghitung *safety stock*, akan menggunakan permintaan maksimum Gas *Methyl Bromide* sebesar 96 kg, permintaan rata-rata per periode (K) sebesar 88,75 kg/hari, dan *lead time (LT)* selama 3 hari. Berdasarkan rumus (4) diperoleh hasil *safety stock* untuk persediaan Gas *Methyl Bromide* adalah sebesar 21,75 kg.

Perhitungan ROP didasarkan pada nilai *demand (D)* Gas *Methyl Bromide* sebesar 30,36956522 kg, *lead time (LT)* Gas *Methyl Bromide* selama 3 hari, dan *safety stock* Gas *Methyl Bromide* sebesar 21,75 kg. Dengan menerapkan rumus (2) diperoleh ROP persediaan Gas *Methyl Bromide* adalah sebesar 112,8586957.

Tabel 10 memuat hasil perhitungan manual ROP untuk semua bahan dan alat yang digunakan pada proses fumigasi. Perhitungan tersebut didasarkan pada variabel-variabel yang disediakan pada Tabel 9.

Tabel 10. Hasil Perhitungan Manual ROP

Nama Persediaan	Safety Stock	ROP
Gas <i>Methyl Bromide</i>	21,75	112,8586957
Gas Oksigen	0,84375	4,854619565
Cartridge Gas Mask	3	5,02173913
Plastik Fumigasi	5,142857143	7,034161491
Sand Snake	52,5	63,91304348
Kertas Sertifikat Fumigasi	35	48,36956522

Berdasarkan hasil perhitungan manual ROP tersebut, dilakukan pengujian untuk membandingkan perhitungan antara aplikasi dan *perhitungan manual*. Gambar 16 menampilkan hasil perhitungan ROP pada aplikasi dengan menggunakan data yang sama seperti perhitungan manual. Dengan membandingkan kedua perhitungan tersebut, dapat dilakukan verifikasi bahwa implementasi hasil perhitungan pada aplikasi sesuai dengan harapan yang diinginkan. Verifikasi ini penting untuk memastikan konsistensi dan kesesuaian hasil perhitungan antara aplikasi dan perhitungan manual.

Nama Persediaan	Jumlah Persediaan	Safety Stock	ROP
Gas <i>Methyl Bromide</i>	50	21	114
Gas Oksigen	50	1	5
Cartridge Gas Mask	50	3	5
Plastik Fumigasi	50	5	7
Sand Snake	50	53	65
Kertas Sertifikat Fumigasi	50	36	46

Gambar 16. Hasil Perhitungan Aplikasi ROP

Berdasarkan hasil perhitungan manual ROP yang terdapat pada Tabel 10 dan hasil perhitungan ROP pada aplikasi yang terdapat pada Gambar 16, dapat dilihat bahwa kedua perhitungan tersebut memberikan hasil yang relatif sama dengan perbedaan sebesar 1,98%. Hal ini membuktikan bahwa implementasi perhitungan ROP pada aplikasi telah berjalan dengan baik dan sesuai dengan harapan yang diinginkan.

Perhitungan EOQ secara manual didasarkan pada data total permintaan persediaan dalam periode bulan Maret-April-Mei yang bersumber dari Tabel 9. Perhitungan manual EOQ terhadap persediaan Gas *Methyl Bromide* menggunakan jumlah pemesanan per periode (D) Gas *Methyl Bromide* sebesar 2794 kg, biaya pemesanan (S) sebesar Rp. 100.000, dan biaya penyimpanan per unit per periode (H) Gas *Methyl Bromide* sebesar Rp. 2.000. Implementasi rumus (1) mendapatkan nilai EOQ persediaan Gas *Methyl Bromide* sebesar 528,5830115.

Dari contoh perhitungan manual EOQ pada persediaan Gas *Methyl Bromide*, dilakukan terhadap semua bahan persediaan yang ada di perusahaan. Hasil perhitungan manual EOQ secara keseluruhan disajikan pada Tabel 11.

Tabel 11. Hasil Perhitungan Manual EOQ

Nama Persediaan	Jumlah Pemesanan Per periode	Biaya Pemesanan	Biaya Penyimpanan	EOQ
Gas <i>Methyl Bromide</i>	2794	100000	2000	528,583
Gas Oksigen	123	100000	2000	110,9054
Cartridge Gas Mask	62	100000	2000	78,74008
Plastik Fumigasi	58	100000	2000	76,15773
Sand Snake	350	100000	2000	187,0829
Kertas Sertifikat Fumigasi	410	100000	2000	202,4846

No	Nama Persediaan	EOQ	Jumlah Pemesanan
1	Gas <i>Methyl Bromide</i>	528	528
2	Gas Oksigen	112	112
3	Cartridge Gas Mask	82	82
4	Plastik Fumigasi	93	93
5	Sand Snake	202	202
6	Kertas Sertifikat Fumigasi	187	187

Gambar 17. Hasil Perhitungan Aplikasi EOQ

Dari hasil perhitungan manual EOQ pada Tabel 11 dan perhitungan EOQ pada aplikasi dalam Gambar 17 terdapat perbedaan sebesar 4,41%. Dengan demikian dapat disimpulkan bahwa kedua perhitungan tersebut

tidak berbeda secara signifikan. Hasil ini menunjukkan bahwa implementasi perhitungan EOQ dalam aplikasi berjalan dengan akurat sesuai dengan ekspektasi yang diharapkan.

Hasil pengujian menggunakan *white box* menunjukkan bahwa struktur internal dan logika program perhitungan ROP dan EOQ berjalan dengan baik. Hal ini didukung oleh hasil *test case* serta kesamaan antara hasil perhitungan *cyclomatic complexity* dengan *graph matrix*. Perbandingan antara perhitungan manual dengan aplikasi menunjukkan perbedaan yang tidak signifikan yaitu sebesar 3,87%. Hal ini membuktikan bahwa penelitian ini telah berhasil menerapkan metode ROP dan EOQ pada aplikasi pengendalian persediaan yang dibangun.

#### 4. Kesimpulan

Berdasarkan hasil penelitian yang dilakukan, dapat disimpulkan bahwa implementasi metode ROP dan EOQ dalam aplikasi pengendalian persediaan pada perusahaan berhasil dilaksanakan. Melalui pengujian dengan metode *white box*, ditemukan bahwa aplikasi yang telah dikembangkan beroperasi dengan baik, akurat, dan sesuai dengan ekspektasi yang diharapkan. Hal ini didukung dengan bukti setiap *test case* dapat melewati dua *flow graph* hasil dari perhitungan *cyclomatic complexity*. Selain itu, melalui pengujian perhitungan ROP dan EOQ, terbukti bahwa perhitungan aplikasi dengan manual menunjukkan hasil yang relatif sama meskipun ada perbedaan sebesar 3,87%. Hal ini mengindikasikan bahwa perhitungan aplikasi berjalan dengan benar dan sesuai dengan tujuan yang ditetapkan oleh peneliti.

#### Ucapan Terimakasih

Peneliti mengucapkan terima kasih kepada Direktur sekaligus pemilik perusahaan karena telah bersedia memberikan waktu, data penunjang penelitian, sekaligus penyemangat bagi penulis dalam menyelesaikan penelitian ini.

#### Daftar Rujukan

[1] R. Ratningsih, "Penerapan Metode Economic Order Quantity (EOQ) Untuk Meningkatkan Efisiensi Pengendalian Persediaan Bahan Baku Pada CV Syahdika," *J. Perspekt.*, vol. 19, no. 2, pp. 158–164, Sep. 2021.

[2] M. A. Swasono and A. T. Prastowo, "Analisis dan Perancangan Sistem Informasi Pengendalian Barang," *JATIKA (Jurnal Inform. dan Rekayasa Perangkat Lunak)*, vol. 2, no. 1, pp. 134–143, 2021.

[3] Z. J. Djalambang, N. Qosim, and H. Hasan, "Analisis Persediaan Beras Pada Toko Bali Yasa Luwuk Banggai," *J. Ekon. Trend*, vol. 9, no. 1, pp. 35–47, Jun. 2021.

[4] D. Mardiaty and Y. Saputra, "Rancang Bangun Inventory System Menggunakan Metode Reorder Point (ROP) Pada Toko Bangunan Irhas Padang," *Zo. J. Sist. Inf.*, vol. 5, no. 1, pp. 163–178, Feb. 2023.

[5] H. Hazimah, Y. A. Sukanto, and N. A. Triwuri, "Analisis Persediaan Bahan Baku, Reorder Point dan Safety Stock Bahan Baku ADC-12," *J. Ilm. Univ. Batanghari Jambi*, vol. 20, no. 2, p. 675, Jul. 2020.

[6] A. A. Sayuti, K. Kusnadi, and H. Hamdani, "Penerapan Metode Economic Order Quantity (EOQ) dan Re Order Point (ROP) untuk Menganalisis Kebutuhan Bahan Baku di PT Otscon Safety Indonesia (OSIN)," *J. Ilm. Wahana Pendidik.*, vol. 8, no. 3, pp. 148–156, 2022.

[7] R. G. Farhad, E. Dewayani, and M. D. Lauro, "Perancangan Dan Pembuatan Aplikasi Inventory Management Berbasis Web Pada PT. X Menggunakan Metode EOQ(Economic Order Quantity)," *J. Ilmu Komput. dan Sist. Inf.*, vol. 11, no. 1, p. 6, Jun. 2023.

[8] N. A. Septiani and F. Y. Habibie, "Penggunaan Metode Extreme Programming Pada Perancangan Sistem Informasi Pelayanan Publik," *J. Sist. Komput. dan Inform.*, vol. 3, no. 3, p. 341, Mar. 2022.

[9] S. S. Letari, A. P. Widodo, T. Soebijono, and H. Bambang, "Analisis Metode Penentuan Rencana Kebutuhan Bahan Baku yang Efektif dan Efisien Studi Kasus PT Rafansa Prima Usaha," *J. Ris. Mhs. Akunt.*, vol. 10, no. 2, pp. 189–204, Oct. 2022.

[10] T. R. Sanni El Randi and D. Meirini, "Analisis Manajemen Persediaan Menggunakan Metode Economic Order Quantity Dan Reorder Point Dalam Pengendalian Persediaan Gas LPG 3 Kg Pada SPBE PT.BCP Cirebon," *Fair Value J. Ilm. Akunt. dan Keuang.*, vol. 4, no. 4, pp. 1263–1279, Nov. 2021.

[11] Y. Handrianto and B. Sanjaya, "Model Waterfall Dalam Rancang Bangun Sistem Informasi Pemesanan Produk Dan Outlet Berbasis Web," *J. Inov. Inform.*, vol. 5, no. 2, pp. 153–160, Sep. 2020.

[12] C. T. Pratata, E. M. Asyer, I. Prayudi, and A. Saifudin, "Pengujian White Box pada Aplikasi Cash Flow Berbasis Android Menggunakan Teknik Basis Path," *J. Inform. Univ. Pamulang*, vol. 5, no. 2, p. 111, Jun. 2020.

[13] H. Gusdevi, S. Kuswayati, M. Iqbal, M. F. Abu Bakar, N. Novianti, and R. Ramadan, "Pengujian White-Box Pada Aplikasi Debt Manager Berbasis Android," *Naratif J. Nas. Riset, Apl. dan Tek. Inform.*, vol. 4, no. 1, pp. 11–22, Jun. 2022.

[14] R. Subagia, R. Alit, and F. A. Akbar, "Pengujian White Box pada Sistem Informasi Monitoring Skripsi Program Studi Informatika," *J. Inform. dan Sist. Inf.*, vol. 01, no. 2, pp. 539–547, 2020.

[15] B. N. Sari, O. Komarudin, T. N. Padilah, and M. Nurhusaeni, "Bill Of Material (BOM) Pada Sistem Inventori Kawasan Berikat Untuk Pelacakan Material Movement," *Ilk. J. Ilm.*, vol. 10, no. 3, pp. 323–330, Dec. 2018.