

## PEMODELAN ANTRIAN SISTEM PENGAMBILAN PESANAN PRODUK PADA GUDANG MINUMAN RINGAN DENGAN SISTEM RAK *DRIVE-IN*

### *A QUEUE MODELLING OF ORDER PICKING SYSTEM IN BEVERAGES WAREHOUSING WITH DRIVE-IN RACK SYSTEM*

M Zaky Hadi<sup>1)\*</sup>, Taufik Djatna<sup>2)</sup>, dan Sugiarto<sup>2)</sup>

<sup>1)</sup>Program Studi Teknologi Industri Pertanian, Fakultas Teknologi Pertanian, Institut Pertanian Bogor  
Kampus IPB Darmaga, Bogor 16680, Jawa Barat, Indonesia  
E-mail: zaky\_hadi@apps.ipb.ac.id

<sup>2)</sup>Departemen Teknologi Industri Pertanian, Fakultas Teknologi Pertanian, Institut Pertanian Bogor

Makalah: Diterima 31 Juli 2017; Diperbaiki 20 November Juli 2017; Disetujui 2 Desember 2017

#### ABSTRACT

*This paper proposed a queue model for beverage order picking system produced from agricultural commodity in warehouse with drive-in rack system in order to analyze and increase the warehouse performance. A (M/M/C):(GD/N/∞) queue approach was used to build the model with the queue approach. This paper considered total operator must-be-assigned to improve queue time (Wq), queue length, and utilization in the warehouse. Stack, queue, linked list, Colour HSB (Hue, Saturation, Brightness) algorithms were used to represent drive-in rack, product queuing, and product status in the warehouses. The model was implemented in computer simulation using java programming. After build the model, a real case industrial problem was presented to test the model. The model could calculate warehouse performance including low performance in racking, transportation to loading area, and loading process for current number of operators and idle operator in the warehouse process. This result was in line with the industry condition that has over capacity and high product bottlenecks in the warehouse. The model was used to find and analyze total operator-must-assigned to increase the warehouse performance. The computer simulation showed colour changing based on density of product stack in queue area. This colour degradation helped for easier understanding and accelerating product status detection (including hold, ready-to-release, and expired beverage products in the warehouse).*

*Keywords: beverage warehousing, drive-in rack, order picking, queue modelling, queueing theory*

#### ABSTRAK

Paper ini mengusulkan model antrian untuk sistem pengambilan pesanan produk-produk minuman ringan dengan studi kasus minuman ringan hasil olahan komoditas pertanian yang memiliki Batasan umur simpan pada gudang dengan sistem rak *drive-in* dalam rangka menganalisis dan meningkatkan kinerja pergudangan. Suatu pendekatan antrian (M/M/C):(GD/N/∞) digunakan untuk membangun model antrian sistem pengambilan pesanan produk minuman ringan tersebut. Paper ini mempertimbangkan total operator yang harus ditugaskan untuk memperbaiki waktu antrian, panjang antrian, dan utilisasi pada gudang berdasarkan pendekatan antrian yang digunakan. Algoritma tumpukan, antrian, *linkedlist*, pewarnaan HSB (*Hue, Saturation, Brightness*) digunakan untuk merepresentasikan rak *drive-in*, antrian produk, dan status produk pada gudang. Selanjutnya, model antrian tersebut diimplementasikan pada simulasi komputer menggunakan bahasa pemrograman java. Suatu studi kasus masalah industri nyata disajikan untuk menguji model yang dibangun. Berdasarkan studi kasus tersebut, model yang dibangun dapat menentukan kinerja gudang meliputi rendahnya kinerja pada proses *racking*, transportasi produk menuju area pembongkaran, dan proses pembongkaran produk untuk sejumlah operator yang saat ini ditugaskan serta adanya operator menganggur pada proses di gudang. Hasil dari model untuk studi kasus sesuai dengan hasil observasi kondisi gudang industri tersebut yang saat ini mengalami kelebihan kapasitas serta *bottleneck* yang tinggi. Selanjutnya model tersebut digunakan untuk menentukan dan menganalisis jumlah operator yang harus ditugaskan untuk meningkatkan kinerja gudang. Simulasi komputer memperlihatkan aliran warna sesuai dengan kerapatan dan tumpukan produk pada area antrian. Degradasi warna ini membantu pemahaman yang lebih mudah dan peningkatan deteksi status produk pada gudang (meliputi produk yang harus ditahan, siap untuk dirilis, dan produk kadaluarsa di gudang).

Kata kunci : pemodelan antrian, teori antrian, pergudangan minuman ringan, rak *drive-in*, sistem pengambilan pesanan produk

#### PENDAHULUAN

Sistem pengambilan pesanan produk (*order picking system*) adalah suatu sistem operasional pergudangan yang berkaitan dengan

proses mengambil dan mengeluarkan produk dari area penyimpanan sebagai respon untuk memenuhi permintaan yang datang dari aktor selanjutnya (*next sphere*) pada suatu rantai pasok (*order*) maupun

permintaan (*demand*) dari konsumen akhir (Petersen dan Schmenner, 1999; De Koster *et al.*, 2007). Sistem ini sangat penting dipertimbangkan karena pergudangan dan inventori merupakan salah satu penggerak (*driver*) yang cukup penting untuk menjamin jalannya arus barang pada sistem logistik secara responsif dan efisien. Proses pada sistem ini meliputi pengelompokan dan penjadwalan pesanan dari konsumen, pengalokasian stok sesuai pesanan produk, menurunkan produk dari rak ke lantai gudang, mengambil produk dari area penyimpanan serta pemisahan produk-produk yang telah diambil (De Koster *et al.*, 2007).

Tujuan paling umum dari *order picking system* adalah untuk memaksimalkan level layanan dengan batasan sumberdaya seperti ketersediaan tenaga kerja, mesin, area, jumlah gudang dan biaya pergudangan (Goetschalckx dan Ashayeri, 1989). Sistem operasional ini penting dipertimbangkan karena kedatangan produk (dari departemen produksi atau dari supplier) biasanya diterima dan disimpan dalam jumlah unit volume yang besar sedangkan konsumen biasanya memesan dalam jumlah unit yang lebih kecil dengan beragam tipe produk. Pesanan konsumen yang datang pada suatu gudang biasanya terdiri dan dikelompokkan dalam jalur-jalur pesanan dimana setiap jalur pesanan memiliki produk-produk tertentu yang unik dan spesifik atau dalam bentuk *stock keeping unit* (SKU).

Order picking teridentifikasi sebagai aktivitas yang paling penting dan memiliki prioritas yang paling tinggi dalam suatu pergudangan untuk memperbaiki produktivitas karena mengkonsumsi biaya pergudangan tertinggi dibandingkan sistem dan proses-proses lainnya (lihat De Koster *et al.*, 2007; Tomkins, 2003; Gademann dan Van de Velde, 2005). Sebagian besar elemen-elemen kerja Order Picking meliputi perjalanan menuju area penyimpanan produk, mencari produk yang akan diambil, mengambil dan membongkar tumpukan produk dari area penyimpanan, pendokumentasian artikel yang diambil dan pengurutan produk sesuai pesanan (Parikh dan Meller, 2007).

Salah satu masalah dan tantangan sistem pengambilan pesanan produk adalah penyimpanan produk-produk minuman ringan hasil olahan pertanian terkemas yang memiliki batasan umur simpan yang terbatas serta tipe rak *drive-in* yang memiliki sistem alokasi dan pengambilan berbentuk *last-in-first-out* (LIFO). Kompleksitas LIFO dan umur simpan ini menyebabkan banyaknya industri minuman ringan terutama industri minuman ringan olahan hasil komoditas pertanian yang menggunakan sistem rak *drive-in* mengalami kelebihan kapasitas dan tingginya kadaluarsa produk. Murahnya biaya instalasi rak serta dapat menyimpan produk dengan jumlah yang lebih besar dibandingkan dengan tipe rak lainnya menjadikan banyak pergudangan industri menggunakan sistem rak ini.

Riset-riset mengenai sistem pengambilan sistem pesanan produk khusus untuk bentuk gudang tertentu telah banyak dilakukan. Pan *et al.* (2016) mengembangkan model optimasi dan antrian pada sistem gudang *pick-and-pass*. Berglund dan Batta (2012) mengembangkan alokasi penyimpanan untuk gudang dengan tipe operasi *Picker-to-Parts*. Yang *et al.* (2015) mengembangkan alokasi penyimpanan dan pengambilan produk pada gudang dengan sistem *shared-storage* dan alokasi penyimpanan-pengambilan *multi-shuttle*. Namun semua riset tentang operasional pergudangan tersebut tidak mempertimbangkan rak tiga dimensi (*Drive-in*, *Drive-Thru* dan sebagainya) sifat produk seperti batasan umur simpan pada produk olahan pertanian.

Riset ini mengembangkan model dasar berbasis komputer untuk sistem pengambilan pesanan produk pada gudang dengan sistem rak *drive-in*. Pendekatan yang digunakan berbasis teori antrian dan algoritma-algoritma untuk merepresentasikan operasional dan fisik rak *drive-in*. Studi kasus pada suatu gudang minuman ringan olahan komoditas pertanian terkemas (PT X) digunakan untuk menguji model. Produk ini terdiri dari minuman teh terkemas, kopi, jeli, dan olahan susu. Riset ini sangat penting dikarenakan belum adanya pengembangan model sistem rak *Drive-in* dan banyaknya industri pangan terkemas yang menggunakan tipe rak ini.

Paper ini disusun dari beberapa bagian. Bagian pertama disajikan metode riset pemodelan yang meliputi formulasi model antrian produk minuman ringan dengan model  $(M/M/C):(GD/N/\infty)$  untuk penentuan jumlah operator. Selanjutnya algoritma tumpukan, antrian, *linkedlist*, pewarnaan HSB (*hue*, *saturation*, *brightness*) digunakan untuk memodelkan rak *Drive-in*, antrian produk, dan status produk minuman ringan pada gudang. Formulasi antrian produk dan algoritma-algoritma tersebut diimplementasikan dalam bentuk simulasi komputer untuk mempermudah proses analisis. Suatu studi kasus gudang industri digunakan untuk menguji model yang telah dibangun. Bagian kedua disajikan model berbasis komputer dan hasil dan studi kasus dari gudang industri minuman ringan olahan komoditas pertanian yang terdiri dari teh, kopi dan susu terkemas PET dan ditumpuk diatas *pallet*. Bagian ketiga disajikan kesimpulan dan riset-riset lanjutan.

## METODE PENELITIAN

Kerangka kerja pemodelan ini terbagi atas tiga. Bagian pertama adalah observasi awal untuk melihat kondisi gudang secara nyata serta untuk mengumpulkan data awal pada gudang yang akan dijadikan sampel. Selanjutnya bagian kedua, dikembangkan model sistem pengambilan pesanan produk minuman ringan menggunakan konsep antrian  $(M/M/C):(GD/N/\infty)$  dan algoritma-algoritma

untuk merepresentasikan sistem rak *drive-in*. Bagian ketiga prototipe model antrian berbasis komputer diujicobakan pada studi kasus yang dikaji. Tahapan tersebut direpresentasikan pada Gambar 1.

**Rancangan Jumlah Operator Penanganan Bahan Berdasarkan Performansi Antrian Menggunakan Model Antrian (M/M/c) : (GD/N/∞)**

Model antrian pada penelitian ini menggunakan model (M/M/c) : (GD/N/∞) berdasarkan perhitungan distribusi data tipe diskrit, area antrian dan jumlah operator terbatas. Menurut Taha (2007) kedatangan dan keberangkatan produk didefinisikan sebagai berikut:

$$\lambda_n = \begin{cases} \lambda, & 0 \leq n \leq N \\ 0, & n > N \end{cases}$$

$$\mu_n = \begin{cases} \mu n, & 0 \leq n \leq c \\ c\mu, & c \leq n \leq N \end{cases}$$

dimana

$\lambda$  : Tingkat kedatangan produk ke dalam sistem

$\mu$  : Tingkat kepergian produk dari sistem

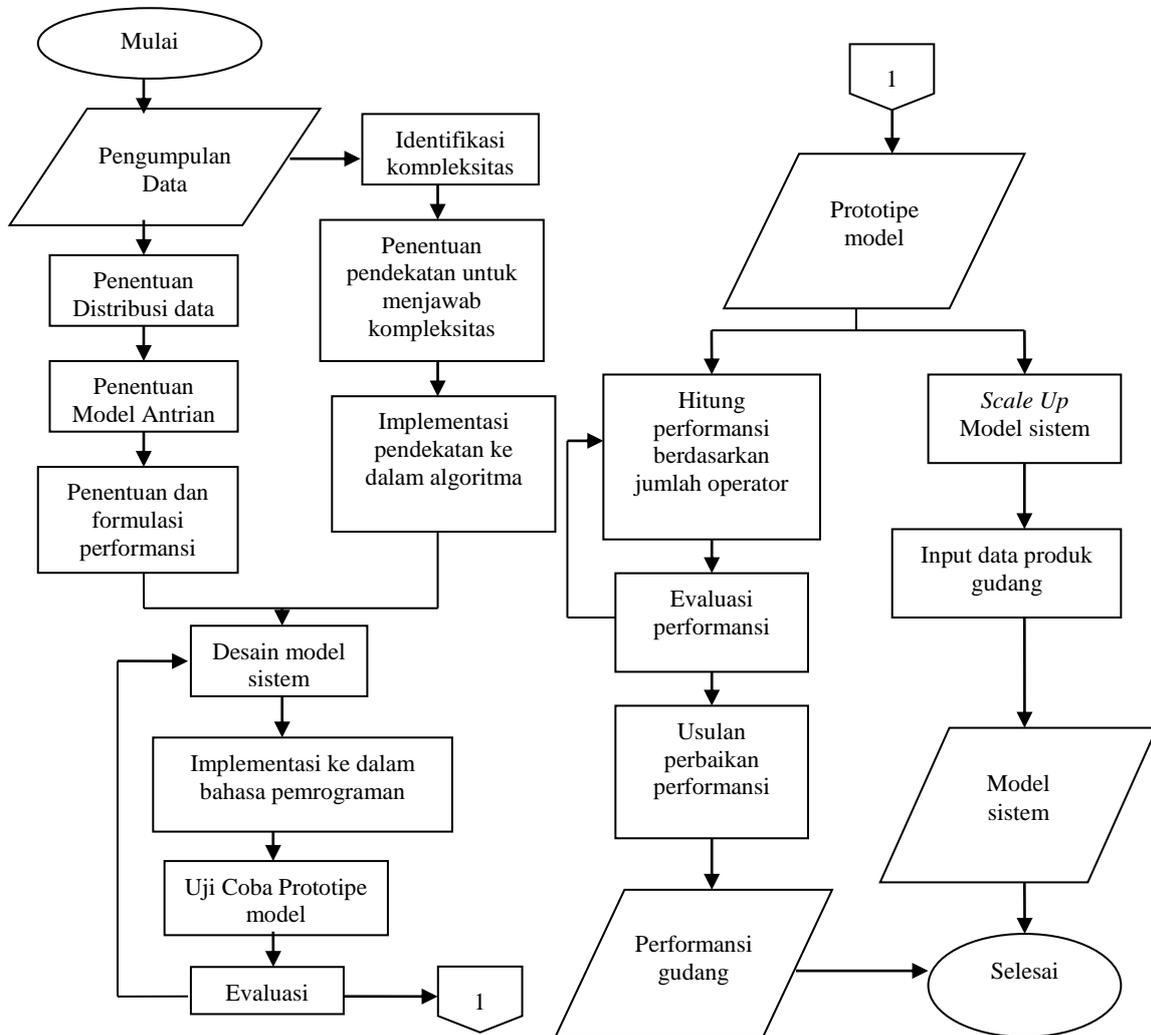
$n$  : Jumlah produk berada dalam sistem

$N$  : Jumlah maksimal produk yang diizinkan berada dalam sistem

Perhitungan performansi antrian (M/M/c) : (GD/N/∞) pada gudang *finish good* meliputi perkiraan banyak produk menunggu pada antrian, perkiraan lama waktu menunggu produk pada area antrian, operator sibuk dan utilitas dengan asumsi kedatangan dan servis produk mengikuti distribusi *Poisson*.

**Formulasi Perkiraan Banyak Produk Menunggu pada Antrian**

Menurut Bhat (2008) rasio antara tingkat kedatangan (*arrival rate*) dan pelayanan (*service rate*) memainkan peranan penting dalam pengukuran performansi sistem antrian. Rasio ini disebut intensitas lalu lintas produk dalam sistem (*traffic intensity*) yang didefinisikan sebagai perbandingan tingkat kedatangan dengan tingkat pelayanan. *Traffic intensity* disebut juga utilitas aliran produk dalam sistem (*utilization of each runaway*).



Gambar 1. Kerangka kerja riset

Utilitas aliran produk dalam sistem padamodel antrian ( $M/M/C$ ) dengan area antrian terbatas berlaku untuk setiap  $c$  server pada sistem antrian sehingga formulasinya *traffic intensity* berlaku untuk setiap  $c$  operator. Menurut Taha (2007) antrian dengan model ( $M/M/c$ ) memiliki dua kondisi berbeda yaitu kondisi  $\rho/c = 1$  dan kondisi  $\rho/c \neq 1$  dengan formulasi perkiraan banyak produk menunggu pada antrian masing-masing kondisi didefinisikan sebagai berikut:

$$P_n = \begin{cases} \frac{\rho^n}{n!} P_0, & 0 \leq n \leq c \\ \frac{\rho^n}{c!c^{n-c}} P_0, & c \leq n \leq N \end{cases}$$

dimana  $P_0$  merupakan probabilitas adanya produk sebanyak 0 (tidak adanya produk) dalam sistem.  $P_0$  didefinisikan sebagai berikut:

$$P_0 = \begin{cases} \left( \sum_{n=0}^{c-1} \frac{\rho^n}{n!} + \frac{\rho^c (1 - (\frac{\rho}{c})^{N-c+1})}{c! (1 - \frac{\rho}{c})} \right)^{-1}, & \frac{\rho}{c} \neq 1 \\ \left( \sum_{n=0}^{c-1} \frac{\rho^n}{n!} + \frac{\rho^c}{c!} (N - c + 1) \right)^{-1}, & \frac{\rho}{c} = 1 \end{cases}$$

Keterangan :

- $L_q$  : Perkiraan banyak produk menunggu pada area antrian
- $N$  : Jumlah maksimal produk yang diizinkan berada dalam sistem
- $n$  : Jumlah produk berada dalam sistem
- $c$  : Jumlah operator
- $P_n$  : Probabilitas kondisi *steady-state*  $n$  produk berada dalam sistem
- $P_0$  : Probabilitas tidak ada produk dalam sistem
- $\rho$  : *Traffic Intensity*

**Formulasi Perkiraan Lama Waktu Menunggu Produk pada Area Antrian**

Menurut Taha (2007) perkiraan lama waktu menunggu produk dipengaruhi oleh kedatangan efektif produk ke dalam sistem. Kedatangan efektif didefinisikan sebagai produk yang datang kedalam sistem saat kapasitas antrian mencukupi. Kedatangan efektif diperoleh dari selisih antara tingkat kedatangan dengan tingkat pelayanan produk yang tidak dapat masuk ke dalam sistem. Tingkat kedatangan produk yang tidak dapat masuk ke dalam sistem diperoleh dari perkalian antara tingkat kedatangan dengan probabilitas  $N$  pelanggan berada dalam sistem. Perkiraan lama waktu menunggu produk pada area antrian dihitung berdasarkan formula *Little's* dengan formulasi sebagai berikut :

$$\begin{aligned} L_s &= \lambda_{eff} W_s \\ L_q &= \lambda_{eff} W_q \\ \lambda_{eff} &= \lambda - \lambda_{lost} = (1 - P_N) \lambda \\ \lambda_{lost} &= \lambda P_N \end{aligned}$$

Dimana:

- $L_s$  : Perkiraan banyak produk menunggu pada sistem
- $L_q$  : Perkiraan banyak produk menunggu pada area antrian
- $\lambda_{eff}$  : Tingkat kedatangan efektif produk ke dalam sistem
- $W_s$  : Perkiraan waktu menunggu produk dalam sistem
- $W_q$  : Perkiraan waktu menunggu produk dalam antrian
- $\lambda_{lost}$  : Tingkat kedatangan produk yang tidak dapat masuk ke dalam sistem
- $N$  : kapasitas maksimum area antrian

**Formulasi Rata-Rata Jumlah Operator Sibuk**

Menurut Taha (2007) selisih antara perkiraan jumlah produk dalam sistem dengan jumlah produk dalam area antrian didefinisikan sebagai rata-rata jumlah operator sibuk. Formulasi rata-rata jumlah operator sibuk didefinisikan sebagai berikut:

$$\bar{c} = L_s - L_q = \lambda_{eff} / \mu$$

Dimana :

- $\bar{c}$  : Rata-rata jumlah operator sibuk dalam sistem
- $L_s$  : Perkiraan banyak produk menunggu pada sistem
- $L_q$  : Perkiraan banyak produk menunggu pada area antrian
- $\lambda_{eff}$  : Tingkat kedatangan efektif produk ke dalam sistem
- $\mu$  : Tingkat kepergian produk dari sistem

**Utilitas Operator**

Utilitas operator merupakan perbandingan antara operator sibuk dengan jumlah operator maksimum. Utilitas operator digunakan untuk mengetahui persentase operator sibuk dengan server menganggur. Formulasi utilitas operator didefinisikan sebagai berikut:

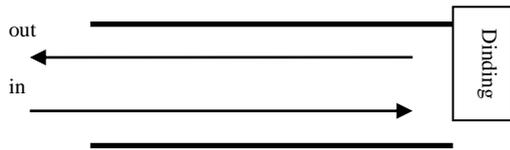
$$Utilitas = \bar{c} / c$$

Dimana

- $\bar{c}$  : Rata-rata jumlah operator sibuk dalam sistem
- $c$  : Jumlah operator maksimum

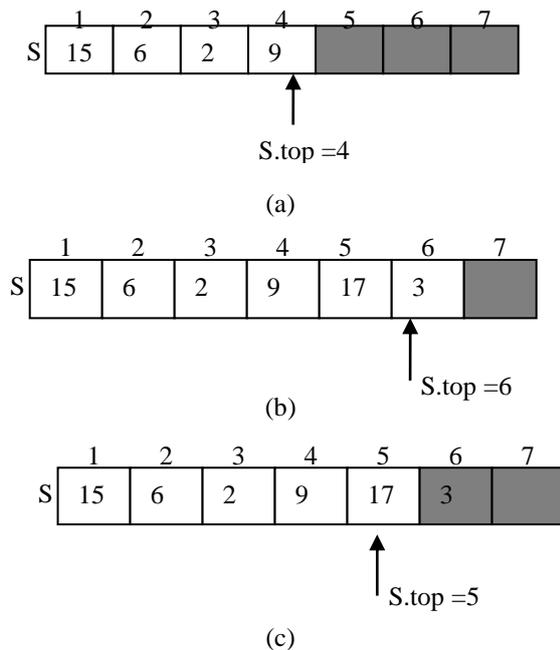
**Konsep Algoritma Tumpukan (Stack)**

Konsep tumpukan digunakan untuk merepresentasikan kompleksitas sistem rak *drive-in*. Aturan pada konsep ini adalah setiap produk yang datang terakhir akan dieksekusi pertama atau *last in first out* (LIFO). Rak *drive-in* adalah rak yang memiliki satu jalan masuk sekaligus sebagai jalan keluar. Material yang disimpan pada rak ini memiliki aturan LIFO. Model rak *drive-in* diilustrasikan pada Gambar 2.



Gambar 2. Ilustrasi model rak Drive-in

Konsep tumpukan direpresentasikan dalam bentuk algoritma tumpukan. Algoritma tumpukan adalah salah satu algoritma struktur atau himpunan data pada Array yang mengimplementasikan aturan LIFO. Operasi *Insert* pada algoritma tumpukan biasanya disebut *Push*, dan operasi *Delete* disebut *Pop*. Menurut Cormen *et al.* (2009) model algoritma tumpukan digambarkan pada Gambar 3.



Gambar 3. Ilustrasi model algoritma tumpukan pada struktur data

Elemen paling atas (*S.top*) pada tumpukan S adalah angka 9. Setelah dilakukan penambahan data 17 dan 3 pada tumpukan, elemen paling atas pada tumpukan S adalah angka 3 (Gambar 3.b). Jika angka 3 pada tumpukan paling atas diambil maka elemen paling atas tumpukan adalah angka 17 (Gambar 3.c). Menurut Cormen *et al.* (2009) operasi tumpukan pada setiap gambar diatas dapat diimplementasikan kedalam beberapa baris kode sebagai berikut:

```

STACK-EMPTY (S)
1  if S.top = 0
2  return TRUE
3  else return FALSE

PUSH (S, x)
1  S.top = S.top + 1
2  S [S.top] = x

POP (S)
1  if STACK-EMPTY (S)

```

```

2  error "underflow"
3  else S.top = S.top - 1
4  return S [S.top + 1]

```

**Konsep Algoritma Antrian (Queue)**

Konsep antrian digunakan untuk merepresentasikan kompleksitas disiplin antrian (FIFO) pengambilan dan operasional pergudangan produk pada gudang. Konsep ini direpresentasikan dalam bentuk algoritma antrian. Algoritma antrian adalah salah satu algoritma struktur atau himpunan data pada Array yang mengimplementasikan aturan FIFO. Operasi *Insert* pada algoritma antrian disebut *Enqueue* dan operasi *Delete* disebut *Dequeue*. Algoritma antrian memiliki bagian kepala (*head*) dan ekor (*Tail*). Ketika elemen data disisipkan (*Enqueue*) maka data akan menempati bagian ekor dari antrian sedangkan elemen data dihapus (*Dequeue*) selalu pada bagian kepala. Menurut Cormen *et al.* (2009) model algoritma antrian digambarkan pada gambar 4.

Gambar 4 menjelaskan suatu susunan (*array*) S yang memiliki 12 penempatan data. Susunan data awal memperlihatkan bagian kepala (*Q.head*) terletak pada posisi ke 7 dan bagian ekor (*Q.tail*) pada posisi ke 12 (Gambar 4.a). Setelah disisipkan data pada posisi ke 12,1, dan 2 bagian kepala tetap pada posisi ke 7 dan bagian ekor berada pada posisi ke 3 (Gambar 4.b). Data diambil pada bagian kepala (posisi 7) sehingga bagian kepala selanjutnya berada pada posisi ke 8. Menurut Cormen *et al.* (2009) algoritma antrian dapat diimplementasikan kedalam beberapa baris kode sebagai berikut:

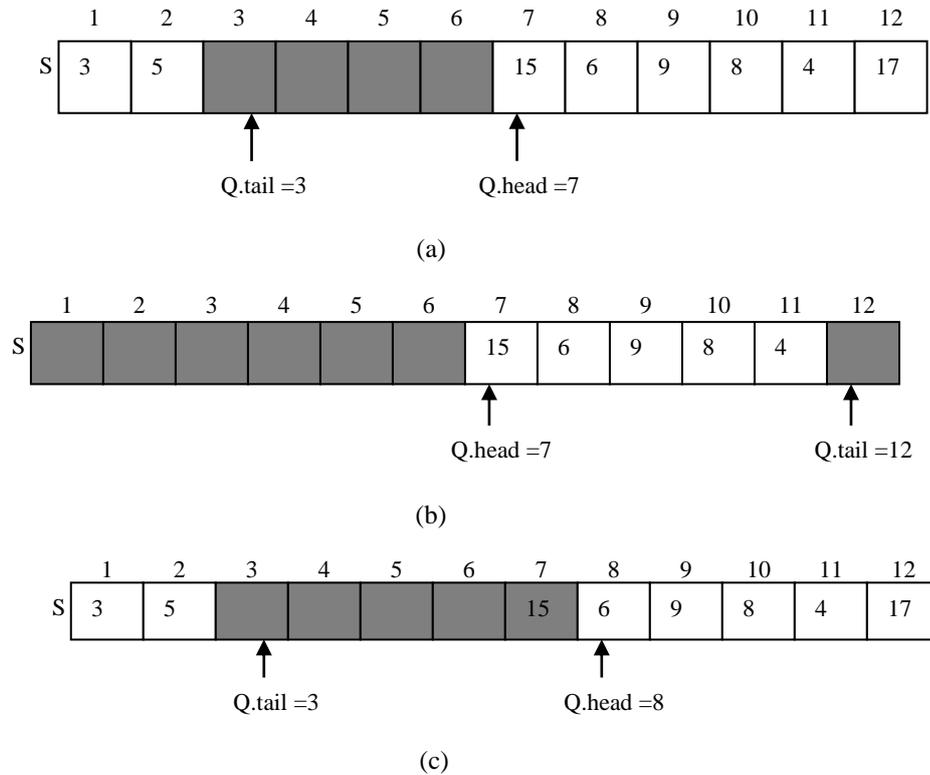
```

ENQUEUE (Q, x)
1  Q[Q.tail]= x
2  if Q.tail == Q.length
3  Q.tail = 1
4  else Q.tail = Q.tail + 1
DEQUEUE (Q)
1  x= Q[Q.head]
2  if Q.head == Q.length
3  Q.head = 1
4  else Q.head = Q.head + 1
5  return x

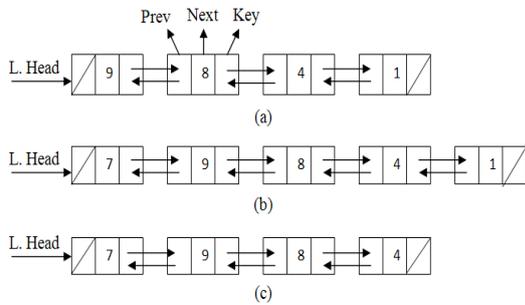
```

Algoritma antrian tersebut dikombinasikan dengan algoritma *LinkedList* karena setiap data pada setiap algoritma antrian memiliki status sebagai *sucessor* dan *predecessor*. Algoritma *Linked List* yang digunakan adalah *doubly linked list* yang digambarkan pada gambar 5.

Gambar 5 (a) menjelaskan algoritma *Doubly Linked List*. Setiap elemen dalam susunan algoritma memiliki atribut-atribut dengan elemen kunci dan *pointers* yang terdiri dari *next* dan *previous*. Atribut *next* pada bagian ekor dan atribut *prev* pada bagian kepala bernilai *Nil*. Jika disisipkan atribut baru pada bagian kepala (angka 7) maka *L.Head* akan berada pada atribut baru tersebut (Gambar 5b), sedangkan jika diambil pada bagian ekor (angka 1) maka ekor akan menjadi angka 4.



Gambar 4. Ilustrasi model algoritma antrian pada struktur model



Gambar 5. Ilustrasi model algoritma *Doubly Linked List*

**Konsep Color HSB (Hue, Saturation, Brightness)**

Konsep *Color HSB* (*hue, saturation, brightness*) digunakan untuk merepresentasikan tumpukan produk pada model sistem perdagangan dengan status warna hijau (H=0,30) jika produk kosong dan merah (H=0,00) jika tumpukan produk penuh pada area rak. Selain itu pewarnaan ini juga digunakan untuk mempermudah mendeteksi status kadaluarsa pada rak *Drive-in*.

**Implementasi ke dalam Bahasa Pemrograman Java**

Keempat perhitungan performansi antrian tersebut dikodekan kedalam bahasa pemrograman java dengan bantuan *Java Modelling Tools Library* (Bertolli *et al.*, 2009). Implementasi ini bertujuan untuk membangun model berbasis komputer. Model berbasis komputer ini selanjutnya digunakan untuk simulasi studi kasus. *Java Modelling Tools library*

adalah *framework* lengkap untuk membantu evaluasi performansi sistem, perencanaan kapasitas, dan studi karakterisasi beban kerja. *Framework* ini ditulis dalam bahasa pemrograman java yang terdiri dari enam fitur yaitu:

1. JSIMgraph, yaitu Simulator model antrian dengan *graphical user interface*
2. JSIMwiz, yaitu Simulator model antrian dengan *wizard-based user interface*
3. JMVA yaitu Analisis nilai rata-rata model antrian
4. JABA, yaitu Analisis asimtot model antrian
5. JWAT, yaitu Analisis beban kerja dari *log* dan data yang digunakan sistem
6. JMCH, yaitu Simulator *Markov chain*

Konsep pewarnaan direpresentasikan dalam bentuk algoritma warna pada pemrograman java. Pemrograman java *Color HSB* merupakan pemrograman yang sudah tersedia sebagai Java SE (*standar edition*) dan merupakan bagian dari *java abstract window toolkit (Java AWT)*. Formulasi aliran warna ini direpresentasikan pada Gambar 6.



Gambar 6. Aliran warna hijau (0,30) menuju merah (0,00) pada *Java Color HSB*

**Alat Bantu Perancangan dan Implementasi Model**

Model dikembangkan menggunakan bahasa pemrograman Java dengan bantuan perangkat lunak komputer yaitu Netbeans IDE 7.3 untuk

implementasi model kedalam bahasa pemrograman java (OC, 2013), Balsamiq Mockups untuk desain antar muka model sistem (BS, 2014), JODA Time Library untuk perhitungan waktu (Colebourne, 2014), *Java Modelling Tools Library* untuk evaluasi performansi sistem (Bertoli *et al.*, 2009), XAMPP untuk pengolahan basis data (Seidler dan Vogelgesang, 2014).

### Studi Kasus pada Pergudangan Minuman Ringan Olahan Komoditas Pertanian Terkemas

Produk minuman ringan yang digunakan sebagai kasus pada penelitian ini adalah produk minuman ringan hasil olahan komoditas pertanian terkemas PET dan ditumpuk diatas *Pallet*. Jenis produk minuman ringan adalah teh manis, kopi, dan olahan susu. Produk-produk olahan pertanian ini tentunya memiliki Batasan umur simpan. Proses pergudangan produk minuman ringan pada gudang *finish good* dilakukan dalam lima tahap yaitu serah terima produk antara pihak gudang dengan produksi, *racking*, produk *Release*, transportasi dari *waiting lines 2* ke area *loading*, dan proses *loading* (bongkar palet). Proses ini dilakukan oleh departemen *Supply Chain Management* yang terdiri dari kepala gudang, *team leader*, dan operator gudang.

Proses serah terima dilakukan oleh admin gudang dengan admin produksi sesuai jumlah produk per palet dan setiap palet yang akan masuk gudang akan diberi status inkubasi (Pengecekan status kualitas produk oleh departemen *quality control*). Selanjutnya produk dibawa oleh operator *hand palet* ke area *waiting lines 1*. Status inkubasi ini berjalan selama delapan hari dengan hasil pengecekan diterima atau *reject*. Jumlah produk per palet adalah 132 kotak dengan jumlah produk masing-masing kotaknya adalah 24 *cup*. Produk dimuat ke palet oleh operator pengemas pada enam *line* produksi.

Proses *racking* dilakukan oleh operator *forklift* ke rak gudang *finish good*. Status produk pada rak adalah inkubasi atau *release* dengan kapasitas maksimal rak adalah 3000 posisi palet. Jenis rak pada proses *racking* adalah rak *drive-in* dengan instalasi rak permanen yang terbagi dalam delapan blok dengan kapasitas masing-masing blok yang berbeda-beda. Instalasi permanen pada rak *drive-in* gudang menyebabkan sulitnya pengaturan tata letak ulang dikarenakan faktor luas area pabrik dan biaya yang cukup besar.

Proses *release* (*picking order*) adalah proses penurunan produk dari rak dengan status produk diterima untuk selanjutnya dilakukan proses bongkar dari palet ke kontainer armada. Sebelum di muat ke armada, produk mengantri di area *waiting lines 2*. Produk yang tidak lolos uji dan kadaluarsa akan diambil oleh operator *forklift* untuk dilakukan penghancuran produk dan proses *treatment* limbah. Tidak ada aturan rute atau scenario *picking order*

pada proses *release* produk ini. Operator mengambil palet pada rak sesuai instruksi dari admin gudang.

Proses selanjutnya adalah transportasi yang dilakukan oleh operator *hand palet* dari area *waiting lines 2* ke area *loading*. Setelah palet sampai pada area *loading* operator memindahkan produk dari palet ke kontainer armada. Produk dicatat oleh operator ekspedisi untuk selanjutnya dibuat surat jalan armada. Armada berkapasitas 4000 kotak produk, dan mentransportasikan produk ke gudang distributor untuk selanjutnya dikirim ke konsumen oleh pihak distributor.

## HASIL DAN PEMBAHASAN

Tahapan implementasi model konseptual dilakukan dengan menerjemahkan formulasi dan algoritma pada analisis kompleksitas dan model antrian kedalam bahasa pemrograman java. Selanjutnya, hasil implementasi ini diuji coba meliputi *debugging* dan validasi dalam bentuk uji coba perhitungan dan representasi sistem pergudangan pada model. Uji coba perhitungan dilakukan dengan membandingkan hasil perhitungan model sistem dengan TORA (Taha, 2007). Uji coba representasi sistem pergudangan dilakukan dengan melihat status warna *bottleneck*, area antrian dan rak pada model sistem. Proses evaluasi dilakukan jika hasil uji coba sistem tidak valid. Tahapan *scale up* dilakukan untuk meningkatkan kapasitas model sistem menjadi 3000 posisi palet yang terdiri dari delapan blok rak. Tahapan selanjutnya pada pengembangan model ini adalah *maintenance* yang dilakukan dengan mengevaluasi *bug* pada model sistem kemudian memperbaikinya.

### Hasil Perhitungan Performansi Antrian Produk untuk Jumlah Operator yang Ditugaskan Saat Ini (*Current Operators*) Sistem Pengambilan Pesanan Produk pada Studi Kasus

Tahapan awal perhitungan performansi antrian sistem pergudangan produk minuman ringan dilakukan setelah model selesai dievaluasi. Data yang dibutuhkan pada proses perhitungan performansi ini adalah kapasitas area antrian, jumlah kedatangan produk dan jumlah *handling* produk. Berdasarkan hasil observasi dan pengukuran langsung di gudang *finish good* didapatkan data rata-rata kapasitas antrian produk, jumlah kedatangan produk dan jumlah *handling* produk untuk setiap prosesnya ditabulasikan pada Tabel 1.

Data ini akan menjadi inputan utama perhitungan performansi antrian pada model sistem pergudangan yang dirancang. Model sistem diberi nama *Pineapple Logistic 1*. Hasil perhitungan performansi antrian dengan jumlah operator awal yang ditugaskan oleh pihak gudang *finish good* menggunakan model sistem yang dirancang ditabulasikan pada Tabel 2.

Tabel 1. Hasil pengukuran kedatangan dan handling produk gudang *finish good*

Nama proses	Kapasitas area antrian produk (Palet)	Jumlah kedatangan produk (Palet/jam)	Jumlah <i>handling</i> produk (Palet/jam operator)
<sup>a</sup> Serah terima	6	36	76
<i>Racking</i>	20	154	50
<i>Release</i>	3000	50	100
Transportasi ke area <i>loading</i>	30	100	25
<i>Loading</i>	25	50	16

<sup>a</sup> Jumlah kedatangan produk pada proses ini adalah input produk dari produksi ke gudang yang terdiri dari enam *line* produksi

Tabel 2. Hasil perhitungan performansi antrian dengan jumlah operator saat ini yang ditugaskan

Nama proses	Jumlah Operator (Operator)	Wq (jam)	Lq (Palet)	Utilitas operator
Serah Terima	2	0	0	1,28
<i>Racking</i>	1	0,05	6	0,23
<i>Release</i>	1	0,01	0	2,4
Transportasi ke area <i>loading</i>	2	0,08	7	0,5
<i>Loading</i>	2	0,24	10	1,75

Berdasarkan tabel diatas terlihat bahwa perhitungan utilitas operator pada proses *racking*, transportasi ke area *loading*, dan *loading* menghasilkan angka lebih dari satu. Menurut Cook (2009) utilitas dalam teori antrian adalah perbandingan antara kedatangan dan pelayanan produk untuk setiap fasilitas pelayanan.

Utilitas diartikan perbandingan antara aktual *output* dengan kapasitas maksimum *output* yang mampu dihasilkan sistem, sehingga utilitas juga dapat diartikan pemanfaatan atau penggunaan suatu sistem. Cook (2009) menjelaskan bahwa jika utilitas suatu sistem lebih dari 1 (100%) maka dapat diartikan input yang dilakukan ke sistem lebih cepat dibandingkan kemampuan menghasilkan output. Area sistem akan menampung input sistem melebihi kapasitas maksimalnya sehingga menyebabkan *bottleneck*. Utilisasi kurang dari 1 mengindikasikan bahwa area penampungan input sistem (*line*) akan stabil dan menuju ke kondisi *steady-state*. *Bottleneck* adalah suatu kondisi dimana sistem menghasilkan *output* lebih sedikit dibandingkan laju input ke dalam sistem dengan area antrian input proses terbatas. *Bottleneck* menyebabkan aliran input terhenti dikarenakan proses sebelumnya tidak dapat beroperasi (Rouse, 2007). Berdasarkan pengertian tersebut dapat disimpulkan bahwa ketiga proses tersebut memiliki operator sibuk lebih besar dari pada operator yang ditugaskan. Tingkat kedatangan produk lebih besar dibandingkan kapasitas *handling* operator sehingga ketiga proses tersebut mengalami *bottleneck*.

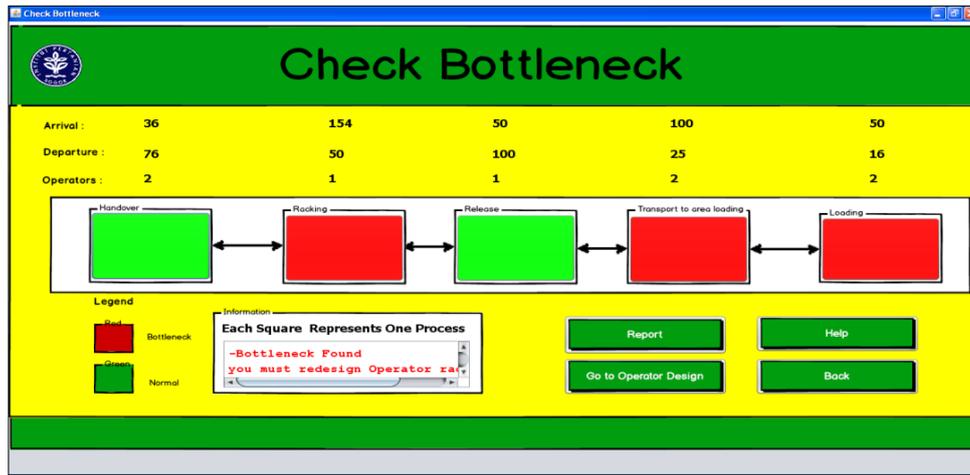
Model sistem yang dirancang mampu mendeteksi kemungkinan terjadinya *bottleneck* berdasarkan tingkat kedatangan produk (*arrival rate*) dan kepergian produk (*departure rate*) pada sistem. *Bottleneck* didesain dengan aturan perkalian jumlah operator dengan *departure rate* untuk proses ke-*n* akan menjadi *arrival rate* pada proses ke *n+1*.

Deteksi *bottleneck* dilakukan dengan menambahkan warna merah pada sistem untuk proses yang mengalami kondisi *bottleneck*. Penggunaan warna ini untuk mempercepat pendeteksian *bottleneck* berdasarkan perhitungan performansi antrian. *Bottleneck* diperlihatkan oleh sistem untuk proses perhitungan performansi antrian dengan jumlah operator yang ditugaskan (*current operator*). Deteksi *bottleneck* digambarkan pada model berbasis komputer pada Gambar 7.

Model sistem yang dirancang mampu mendeteksi kemungkinan terjadinya *bottleneck* berdasarkan tingkat kedatangan produk (*arrival rate*) dan kepergian produk (*departure rate*) pada sistem. *Bottleneck* didesain dengan aturan perkalian jumlah operator dengan *departure rate* untuk proses ke-*n* akan menjadi *arrival rate* pada proses ke *n+1*. Deteksi *bottleneck* dilakukan dengan menambahkan warna merah pada sistem untuk proses yang mengalami kondisi *bottleneck*. Penggunaan warna ini untuk mempercepat pendeteksian *bottleneck* berdasarkan perhitungan performansi antrian. *Bottleneck* diperlihatkan oleh sistem untuk proses perhitungan performansi antrian dengan jumlah operator yang ditugaskan (*current operator*). Deteksi *bottleneck* digambarkan pada model berbasis komputer pada Gambar 7 dan Model sistem tidak mendeteksi adanya *bottleneck* berdasarkan jumlah operator usulan diatas seperti ditunjukkan pada Gambar 8.

### Pemodelan Sistem Pengambilan Pesanan pada Gudang Minuman Ringan

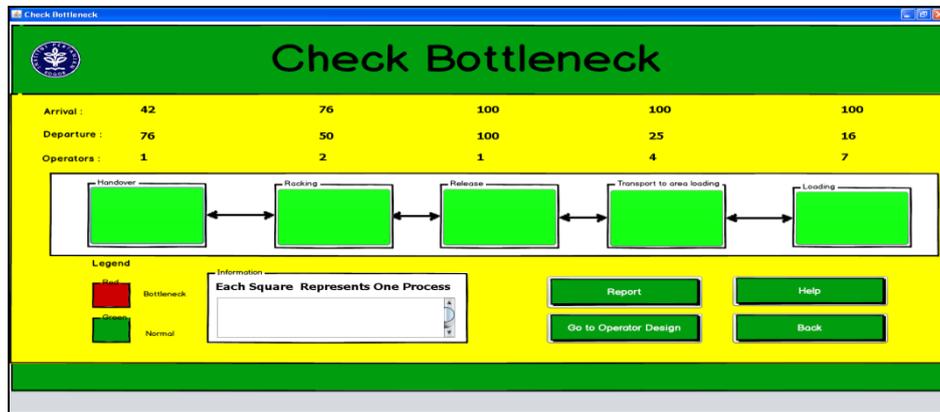
Algoritma *queue*, *stack* dan *color HSB* menghasilkan model sistem yang menjawab kompleksitas permasalahan pada gudang *finish good*. Algoritma antrian menghasilkan model sistem yang merepresentasikan area *waiting lines* seperti Gambar 9.



Gambar 7. Pengecekan *bottleneck* untuk rancangan operator saat ini (*current operator*) di gudang *finish good*

Tabel 3. Hasil perhitungan performansi antrian dengan jumlah operator usulan

Nama Proses	Jumlah Operator (Operator)	Wq (jam)	Lq (Palet)	Utilitas operator
Serah Terima	1	0,01	1	0,54
Racking	2	0,03	2	0,76
Release	1	15,00	1499	1
Transportasi ke area loading	4	0,00	0	0,97
Loading	7	0,03	3	0,85



Gambar 8. Pengecekan *bottleneck* untuk jumlah operator usulan di gudang *finish good* menggunakan sistem *Pineapple Logistic 1.0*



Gambar 9. Visualisasi hasil pemodelan antrian produk pada area *waiting lines* di gudang *finish good* berdasarkan konsep dan algoritma antrian pada sistem *Pineapple Logistic 1.0*

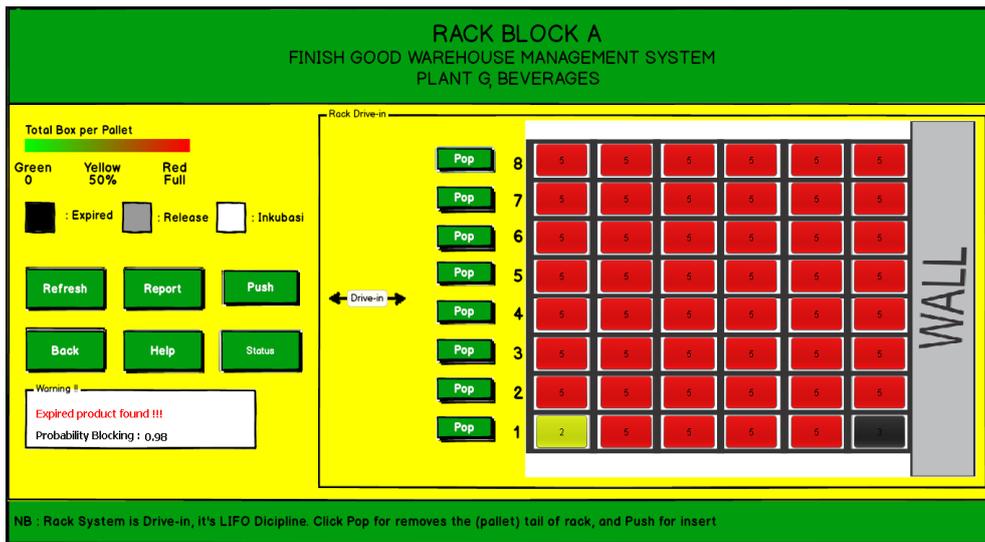
Model algoritma tumpuk merepresentasikan rak *drive-in* dengan sistem rak LIFO. Representasi model rak *drive-in* pada sistem diperlihatkan pada Gambar 10. Warna hitam pada gambar diatas menunjukkan produk kadaluarsa. Aturan produk kadaluarsa adalah 7 bulan setelah produk diinput kedalam sistem. Warna menunjukkan hijau ke merah menunjukkan status tumpukan produk di dalam sistem, sedangkan untuk status produk digudang seperti inkubasi, *release* direpresentasikan dengan warna putih dan abu-abu seperti Gambar 11.

Produk dengan status *release* adalah produk yang telah lolos uji QC sehingga dapat dikeluarkan untuk proses transportasi ke gudang distributor. Produk dengan status tidak lolos uji QC (warna hitam) harus segera dikeluarkan, karena akan menyebabkan gudang menjadi penuh. Berdasarkan hal tersebut sistem dapat mendeteksi status produk secara cepat dan dapat menyelesaikan permasalahan

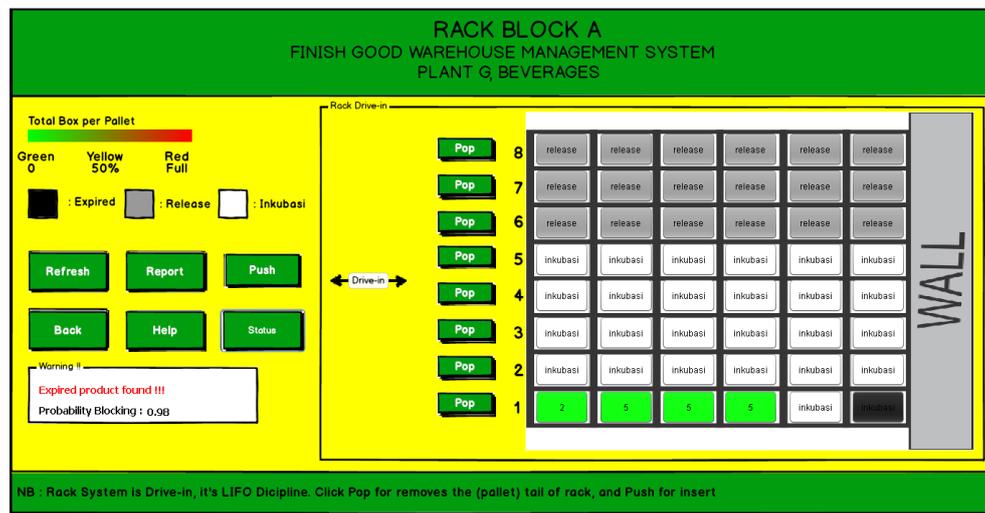
sulitnya mendeteksi status produk karena sistem penyimpanan menggunakan rak *drive-in*.

**Peluang Produk Tidak Dapat Masuk ke dalam Sistem (*Probability Blocking*)**

*Probability blocking* merupakan kemampuan yang ditambahkan pada sistem *Pineapple Logistic 1.0* untuk mengetahui kemungkinan produk tidak dapat masuk ke sistem (antrian atau rak) yang disebabkan kapasitas sistem penuh. Menurut Bhat (2008) *probability blocking* didefinisikan sebagai suatu kondisi pada model antrian dengan jumlah operator dan area antrian terbatas dimana produk tidak dapat masuk ke dalam sistem yang disebabkan sistem sibuk atau tidak mampu lagi menampung input. Pendekatan perhitungan dilakukan dengan metode *transition diagram* untuk *bivariate markov process* (Bhat, 2008).



Gambar 10. Model sistem rak *drive-in* dan status tumpukan produk pada sistem *Pineapple Logistic 1.0*



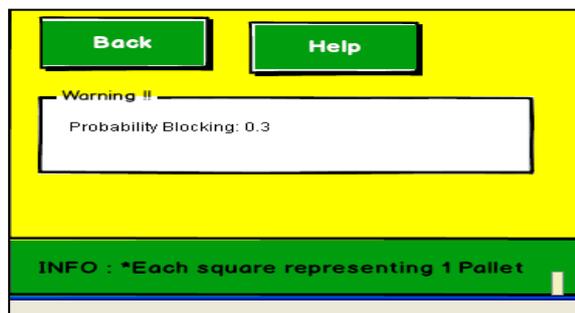
Gambar 11. Model sistem rak *drive-in* dan status produk pada sistem *Pineapple Logistic 1.0*

*Probability blocking* pada model sistem pergudangan ini dilakukan dengan pendekatan penelusuran (*searching*). Teknik ini membandingkan antara penelusuran status tumpukan palet pada rak dan area antrian dengan kapasitas maksimum yang dapat diproses sistem. Teknik penelusuran dilakukan dengan menggunakan bahasa *query* untuk mencari dan menghitung status tumpukan produk pada basisdata. Model perhitungannya didefinisikan sebagai berikut:

*Probability blocking* =

$$\frac{\text{SearchingQuery ( status tumpukan produk pada data)}}{\text{Maksimum kapasitas sistem}}$$

Contoh hasil *probability blocking* diperlihatkan pada *print screen* tampilan sistem *waiting lines 1* dengan jumlah produk mengantri pada sistem 792 kotak pada Gambar 12.



Gambar 12. Hasil identifikasi *probability blocking* pada sistem area *waiting lines 1*

## KESIMPULAN DAN SARAN

### Kesimpulan

Paper ini memodelkan sistem antrian pengambilan pesanan produk minuman ringan pada gudang dengan sistem rak *drive-in* dengan studi kasus pada minuman ringan olahan komoditas pertanian terkemas PET yang terdiri dari teh, kopi dan susu. Pendekatan yang digunakan adalah antrian (M/M/C):(GD/N/∞). Algoritma tumpukan, antrian, *linkedlist*, pewarnaan HSB (*Hue, Saturation, Brightness*) digunakan untuk merepresentasikan rak *Drive-in*, antrian produk, dan status produk pada gudang. Selanjutnya, model antrian tersebut diimplementasikan pada simulasi komputer menggunakan bahasa pemrograman java. Suatu studi kasus masalah industri nyata disajikan untuk menguji model yang dibangun. Berdasarkan studi kasus tersebut, model yang dibangun dapat menentukan kinerja gudang meliputi rendahnya kinerja pada proses racking, transportasi produk menuju area pembongkaran, dan proses pembongkaran produk untuk sejumlah operator yang saat ini ditugaskan serta adanya operator menganggur pada proses di gudang. Model juga

dapat menentukan status produk berdasarkan *hold*, siap rilis dan kadaluarsa di gudang. Namun sistem dan model komputer yang dibangun pada penelitian ini memiliki kekurangan yaitu belum mempertimbangkan alokasi penyimpanan, rute, dan analisis belum mencakup keseluruhan strategi operasional sistem pengambilan produk pada operasional pergudangan.

### Saran

Riset ini adalah riset awalan untuk sistem pengambilan pesanan produk pada gudang minuman ringan dengan sistem rak *drive-in*. Riset-riset lanjutan harus dilakukan untuk menyempurnakan model meliputi alokasi penyimpanan, rute, pengambilan pesanan *batch*, serta implementasi dalam bentuk kasus yang lebih kompleks dan analisis yang lebih komprehensif terkait integrasi level operasional pergudangan.

## DAFTAR PUSTAKA

- Berglund P dan Batta R. 2012. Optimal placement of warehouse cross-aisles in a picker-to-part warehouse with class-based storage. *IIE Transactions*. 44(2):107-120.
- Bertoli M, Gasale G, dan Serazzi G. 2009. *JMT: performance engineering tools for system modeling*. ACM SIGMETRICS Performance Evaluation Review, Volume 36 Issue 4, New York, US, ACM Pres.
- Bhat N. 2008. *An Introduction to Queueing Theory*. Boston; Birkhouser
- [BS] Balsamiq Studio. 2014. Balsamiq Mockups. Tersedia pada [balsamiq.com/products/mockups](http://balsamiq.com/products/mockups).
- Cook JD. 2009. Server utilization. tersedia pada <http://www.johndcook.com> [8 Juni 2014].
- Colebourne J. 2014 Joda-Time. Tersedia pada <http://www.joda.org/joda-time/> [5 April 2014].
- Cormen TH, Leiserson CE, Rivest RL, Stein C. 2009. *Introduction to Algorithms*, 3<sup>rd</sup>. Massachusetts: MIT Pers.
- De Koster R, Le-Duc T, dan Roodbergen K J. 2007. Design and control of warehouse order picking: A literature review. *European Journal of Operational Research*. 182(2): 481-501.
- Gademann N dan Velde S. 2005. Order batching to minimize total travel time in a parallel-aisle warehouse. *IIE Transactions*.37(1): 63-75.
- Goetschalckx M dan Ashayeri J.1989. Classification and design of order picking. *Logistics World*. 2(2): 99-106.
- [OC] Oracle Technology. 2013. Netbeans IDE 7.3. Tersedia pada <https://netbeans.org/community/releases/73/>. [ 11 Juli 2013].
- Pan JCH, Shih PH, Wu MH, Lin JH. 2016. A storage assignment heuristic method based

- on genetic algorithm for a pick-and-pass warehousing system. *Computers & Industrial Engineering*. 81: 1-13.
- Parikh PJ dan Meller RD. 2007. Modeling pick-face blocking in an order picking system. in iie annual conference. Proceedings (p. 866). *Institute of Industrial and Systems Engineers (IISE)*.
- Petersen CG dan Schmenner RW. 1999. An evaluation of routing and volume-based storage policies in an order picking operation. *Decision Sciences*. 30(2): 481-501.
- Rouse. 2007. Definition of Bottleneck. Tersedia pada <http://searchenterprise.wan.techtargget.com/definition/bottleneck> [8 Juni 2014].
- Seidler K dan Vogelgesang K. 2014. Apache Friend-XAMPP. Tersedia pada <http://apachefriend.org>. [25 Maret 2014].
- Taha AH. 2007. *Operation Research*. 7<sup>Ed</sup>.USA; Pearson Prentice Hall.
- Tompkins JA, White JA, Bozer YA, Tanchoco JMA. 2010. *Facilities Planning*. New York: John Wiley & Sons.
- Yang P, Miao L, Xue Z, Qin L. 2015. An integrated optimization of location assignment and storage/retrieval scheduling in multi-shuttle automated storage/retrieval systems. *Journal of Intelligent Manufacturing*. 26(6):1145-1159.