

# Analisis *Order Picking* dengan Menggunakan Metode *Routing Heuristics* di Gudang PT. GMS

Agung Chandra

Fakultas Teknik, Jurusan Teknik Industri, Universitas Mercubuwana  
Jl. Raya Meruya Selatan no.01, Kembangan, Jakarta Barat 11650  
Email: agungchandra\_07@yahoo.co.uk

Received 2 July 2015; Accepted 1 October 2015

---

## Abstract

*Order picking is the most labour and costly activities both in manual system and automated system. So, it is necessary to improve efficiency and productivity. One way to do is to reduce travel distance or travel time. Time can be reduced by implementing routing heuristics method which has the shortest distance, and shorter distance means less energy consumption and fit to greener warehousing concept. The research at PT. GMS, Jakarta shows the routing method – Midpoint, The Largest Gap, and Combined – dynamic programming are the most efficient method*

**Keywords:** *order picking, travel time, routing heuristics, greener warehousing*

---

## 1. PENDAHULUAN

Biaya logistik mempunyai peranan penting dalam perusahaan. Biaya logistik memiliki porsi sebesar 10% dari total penjualan. Gudang, termasuk transportasi dan inventori merupakan salah satu pengendali dari total biaya logistik (Dukic & Opetuk, 2008). Ada 5 aktivitas utama dalam gudang yakni *receiving, put away, storage, order picking, dan shipping* (Bartholdi & Hackman, 2008). *Order Picking* merupakan aktivitas yang paling tinggi biayanya dalam pergudangan dan bisa mencapai 55% dari total biaya operasi pergudangan, sehingga dianggap sebagai prioritas utama dalam peningkatan produktivitas (Tompkins et al, 2003), bahkan bisa mencapai 65% dari total biaya operasional gudang (Theys et al, 2010). *Order Picking* juga merupakan proses yang banyak menggunakan tenaga kerja baik dalam pergudangan yang menggunakan sistem manual ataupun sistem automasi (De Koster, et al, 2007).

*Order Picking* merupakan proses pengambilan barang dari lokasi simpan berdasarkan permintaan pelanggan (Dukic & Opetuk, 2008) dan mencakup proses mendapatkan jumlah yang tepat dan produk yang tepat (De Koster, et al, 2007).

Dalam proses *picking*, pesanan pelanggan / *customer orders* dikonversi menjadi *pick orders*. Waktu untuk melakukan *picking* dibagi menjadi waktu untuk melakukan *travel*, waktu untuk melakukan proses *picking*, dan waktu yang tersisa / *remaining time*. Waktu

untuk *travel* merupakan waktu yang berhubungan dengan perpindahan antar lokasi yang harus dikunjungi dimana barang disimpan dan diambil. Sedangkan waktu *picking* merupakan waktu yang berhubungan dengan pengambilan barang / *grabbing the items*, termasuk juga meletakkan barang ke troli, memeriksa *pick list* dan membaca *pick list* / lokasi barang yang akan diambil berikutnya. Waktu yang tersisa meliputi aktivitas untuk meletakkan troli, menunggu untuk penugasan berikutnya, sosial, dan lain – lain. Pengurangan jarak dan waktu *travel* mempunyai pengaruh penting terhadap total waktu untuk melakukan *order picking* (De Koster, et al, 1999). Karena komponen waktu *travel* merupakan kontribusi utama terhadap waktu untuk melakukan *picking*, maka dapat dikatakan bahwa *travel time* yang berlebihan merupakan suatu pemborosan (Bartholdi & Hackman, 2008).

Untuk sistem *order picking* yang manual, waktu *travel* akan naik seiring dengan jarak *travel*, sehingga jarak *travel* sering dipertimbangkan sebagai tujuan utama dalam merancang dan mengoptimasi gudang (De Koster, et al, 2007). Metode yang paling sering digunakan untuk mengefisienkan *order picking* yang berfokus mengurangi waktu *travel* dapat dikategorikan menjadi 3 grup yakni: *routing, storage and batching* (Roodbergen, 2006). Metode *routing* untuk menentukan urutan dan route *travelling* dan digunakan untuk meminimalkan jarak *travel*; sedangkan metode *storage* digunakan untuk mengalokasikan barang ke lokasi

simpan berdasarkan aturan – aturan tertentu; *order batching* adalah mengelompokkan pesanan pelanggan menjadi satu *picking order* (Dukic & Opetuk, 2008). Strategi *routing* yang baik bisa mengurangi waktu *travel* sebanyak 30% (De Koster & Van der Poort, 1999). Dengan berkurangnya jarak berarti pula berkurangnya juga konsumsi energi, yang disebut *greener warehousing* (Dukic et al., 2010).

Kondisi gudang di PT. GMS, proses *order picking*-nya masih menggunakan sistem manual (*employing human*) yang dilakukan oleh operator dan menggunakan *pick cart*, namun belum pernah dilakukan analisis mengenai waktu atau jarak yang ditempuh pada saat melakukan proses *picking*, sehingga penulis bertujuan untuk menganalisis metode *order picking* yang paling meminimalkan jarak *travel* dengan menggunakan *routing methods* dan bisa mendapatkan waktu yang terpendek. Dengan demikian *service level* terhadap pesanan pelanggan bisa lebih cepat dan produktivitas warehouse bisa ditingkatkan.

## 2. METODE PENELITIAN

### 2.1 Metode Routing

*Routing* merupakan masalah untuk menentukan urutan yang optimal untuk melakukan pengambilan item yang diminta atau yang dipesan secepat mungkin. Pada prakteknya, masalah *routing order pickers* di gudang bisa diselesaikan dengan menggunakan *heuristics* karena dianggap lebih mudah dan logis. Hal ini disebabkan karena adanya beberapa kekurangan dari metode optimal *routing* pada prakteknya (De Koster, et al, 1999) seperti algoritma optimal tidak ada pada setiap *layout*, *optimal route* tidak logis untuk *order picker* dan algoritma optimal yang standard tidak bisa memperhitungkan masalah *aisle*, hal ini berbeda jika menggunakan metode *heuristics*.

*Order picker* memulai *order picking* pada depot dan menerima daftar pengambilan barang / *pick list*, dan harus mengambil barang pada lokasinya dan kemudian kembali ke depot, dan biasanya *picker* menggunakan *picking carts* (Gue, 2008). Penulis pada penelitian ini menggunakan metode *heuristics*: S shape / *Traversal Strategy*, *Return Method*, *Midpoint method*, *The Largest Gap Strategy*, *Combined / Composite heuristics* dengan menggunakan *dynamic programming* (De Koster, et al, 1999), dan penulis menghitung jarak *travelling* pada perpindahan / pergerakan secara horizontal saja, berbagai jenis metode *heuristics* terdapat pada Gambar 1.

Asumsi untuk penelitian ini adalah barang pesanan dapat dilakukan dalam satu route (De Koster and Van der Poort, 1998), atau disebut juga *discrete picking*, *picking aisle* pada kondisi lurus dan memiliki arah yang sama (Dukic, 2008), *order picker* bergerak secara satu dimensi dan frekuensi akses terdistribusi secara merata di semua lokasi, serta jumlah barang per order lebih banyak dibandingkan dengan jumlah *aisle* yang ada proses *picking* (Sadovsky & Hompel, 2011).

Pada penelitian ini, *order picking system* yang digunakan adalah *system picking conventional*. Ilustrasi untuk metode *heuristics* adalah sebagai berikut:

S shape / *Traversal Strategy*: *aisle* tanpa *pick* tidak dilalui, dan *aisle* yang ada sedikitnya satu *pick* maka akan dilalui, terkecuali jumlah *aisle*-nya ganjil, maka pada *aisle* yang terakhir dilalui menggunakan metode *Return*. *Return method*: *picker* masuk dan keluar dari setiap *aisle* yang sama, hanya *aisle* yang ada *pick* yang dilalui. *Midpoint method*: membagi gudang dalam 2 area, depan (*front*) dan belakang (*back*). Bagian depan dilalui dari depan dan bagian belakang dilalui dari belakang. *The Largest gap* mirip dengan *midpoint strategy*, gap merupakan perpisahan antara 2 *pick* yang berdekatan, antara *pick* pertama dengan *aisle* bagian depan atau *pick* terakhir dengan bagian belakang. Jika gap terbesar ada di antara 2 *pick* yang berdekatan maka *order picker* melakukan route kembali dari kedua ujung *aisle*. Dengan demikian gap terbesar merupakan bagian dari *aisle* dimana *order picker* tidak melewatinya (Celik & Sural, 2012). Sedangkan *combined / composite method* menggunakan *dynamic programming* (Dukic & Oluic, 2007).

### 2.2 Skema Penelitian

Penelitian dilakukan dengan membandingkan beberapa metode *routing* yang tersedia untuk menghitung jarak tempuh yang dilakukan oleh seorang *picker* dalam mengeksekusi *pick list / order list of customer*. *Pick list* merupakan item barang yang harus di-*pick* pada lokasi bin. Penghitungan jarak tempuh mengacu pada Gambar 1. Dari beberapa metode *routing*, maka dipilih metode yang menghasilkan jarak tempuh paling pendek, secara singkat langkah – langkah penelitian dapat dilihat pada Gambar 2.

### 2.3 Kalkulasi Jarak

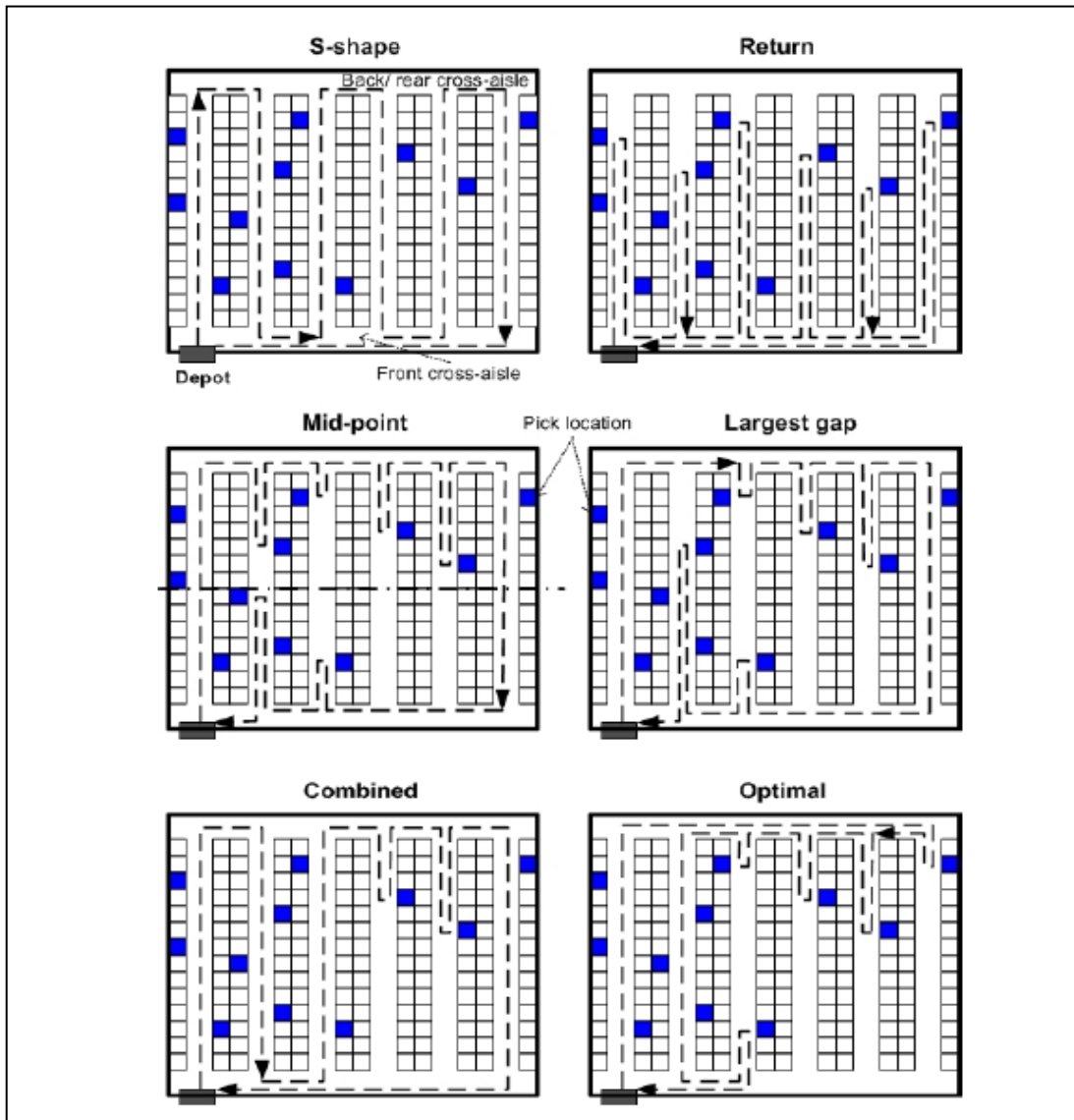
Untuk menghitung jarak yang terjadi pada sistem *picking* konvensional dirumuskan sebagai berikut (Sadovsky & Hompel, 2011):

$$S_n = S_B + S_W + S_A \quad (1)$$

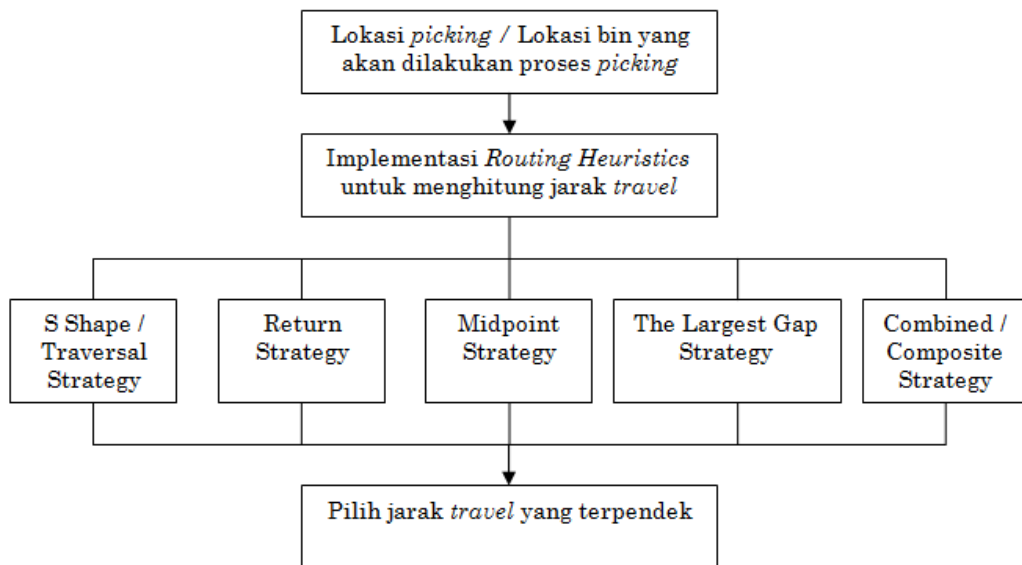
Dimana:

- $S_n$  merupakan jarak yang terjadi pada sistem *picking* konvensional
- $S_B$  merupakan jarak dasar (*basic distance*)
- $S_W$  merupakan jarak yang ada di antara *aisle* (*within aisle distance*)
- $S_A$  merupakan jarak *across aisle* (*across aisle distance*)

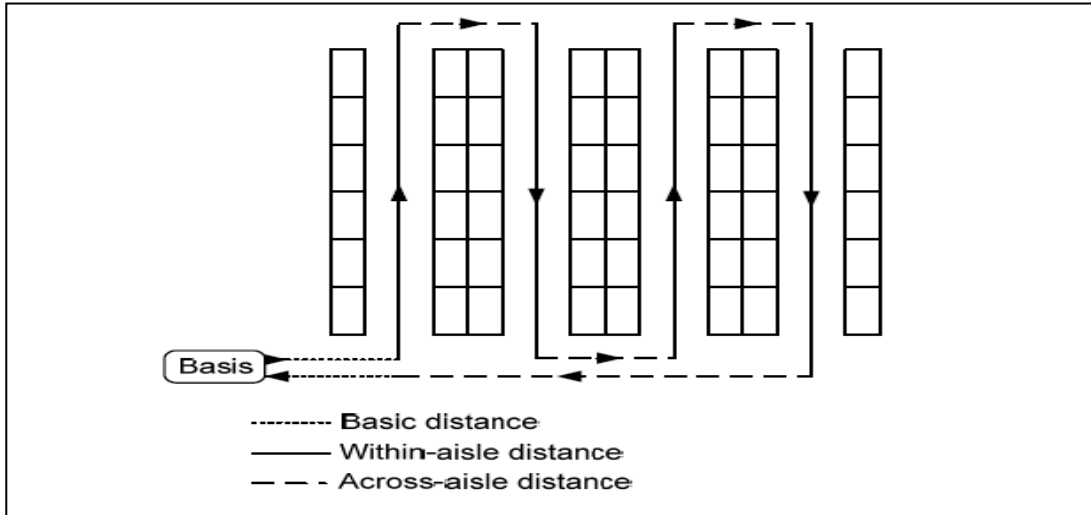
Penjelasan  $S_n$ ,  $S_B$ ,  $S_W$ ,  $S_A$  bisa dilihat pada Gambar 3. Kemudian ada beberapa parameter yang berhubungan dengan jarak seperti pada Gambar 4.



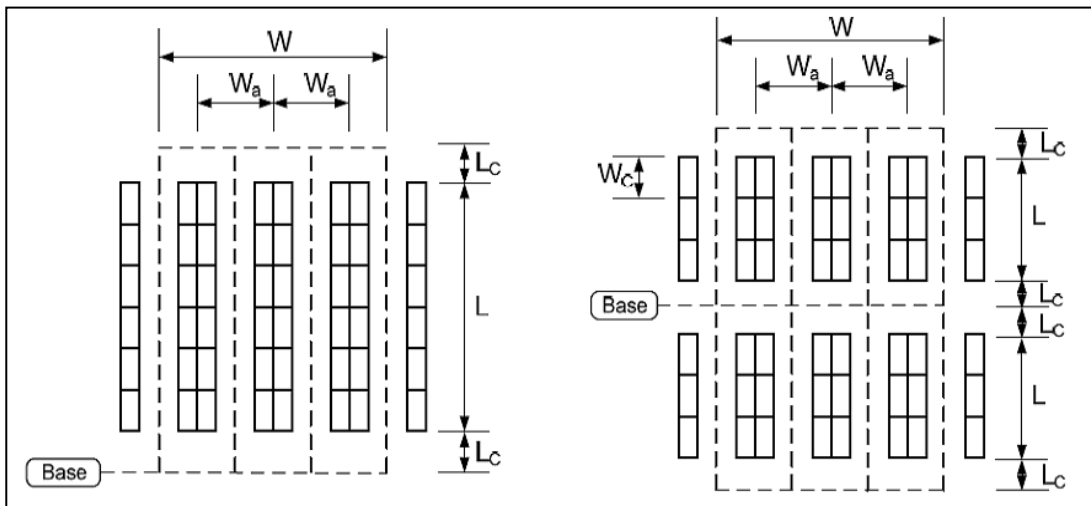
Gambar 1. Beberapa Metode Routing



Gambar 2. Skema Penelitian



Gambar 3. Jarak (*distance*) dalam *order picking storage*



Gambar 4. Parameter dalam layout single block (kiri) dan *double block* (kanan)

Berdasarkan Gambar 4,  $W_a$  = lebar *aisle*;  $W_c$  = lebar lokasi bin;  $L$  = panjang *aisle*;  $L_c$  = panjang konstan *aisle* depan.

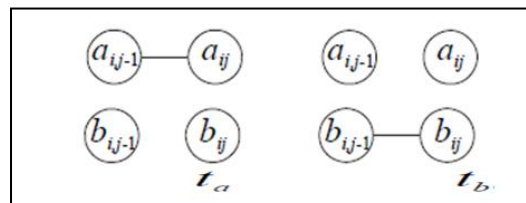
*Combined / Composite Strategy* dengan menggunakan *Dynamic Programming Method* (Roodbergen & De Koster, 2001). Metode *dynamic programming* akan digunakan oleh *order picker* pada blok tunggal  $i$  ( $i = 1, \dots, k$ ). *Route* ini akan dimulai pada *subaisle* yang paling kiri yang memiliki item 1 dan berakhir pada *subaisle* yang paling kanan yang memiliki item  $r$ .

$L_j$  merupakan *partial route* yang akan melewati semua lokasi *pick* dalam *subaisle* 1 sampai  $j$ , dan *partial route* dibedakan menjadi 2 kelompok:

$L_j^a$  merupakan *partial route* yang berakhir pada bagian belakang *subaisle*  $j$

$L_j^b$  merupakan *partial route* yang berakhir pada bagian depan *subaisle*  $j$

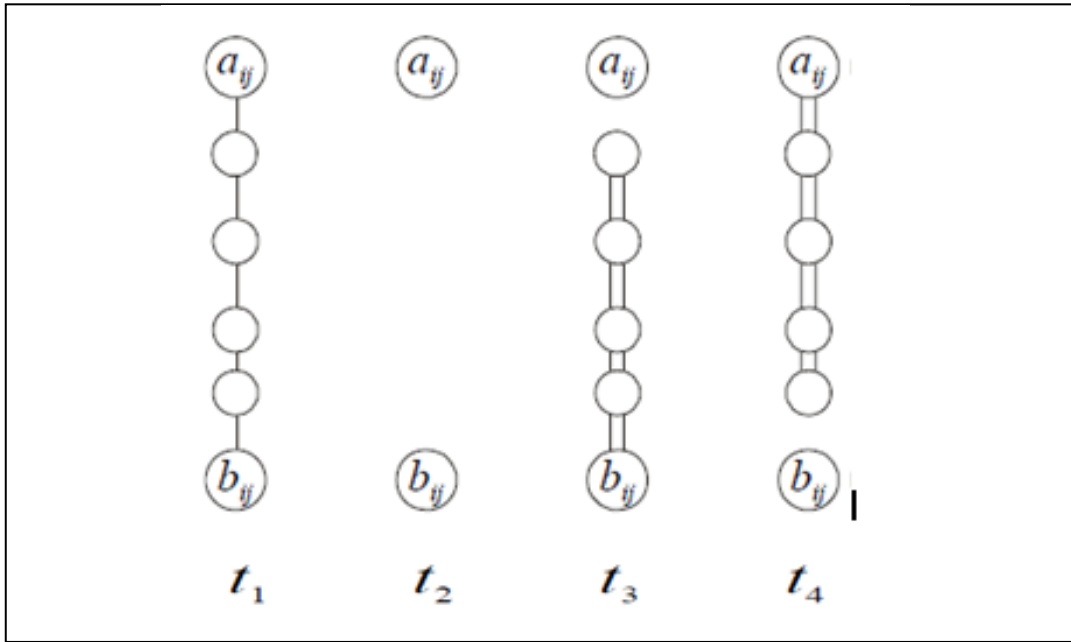
Ada 2 cara dari titik *subaisle*  $j-1$  ke  $j$ , dan ini diilustrasikan pada Gambar 5.



Gambar 5. Transisi  $t_a$  dan  $t_b$

Dimana  $t_a$  merupakan transisi bagian belakang blok; sedangkan  $t_b$  transisi bagian depan blok.

Ada 4 cara untuk melakukan proses *picking* di *subaisle*  $j$  seperti yang diilustrasikan pada Gambar 6. Berdasarkan Gambar 6,  $t_1$  merupakan transisi yang melewati *subaisle* seluruhnya,  $t_2$  merupakan transisi yang tidak melewati *subaisle*,  $t_3$  merupakan transisi yang masuk dan keluar dari depan block dan  $t_4$  merupakan transisi yang masuk dan keluar dari belakang block.



Gambar 6. Transisi metode *heuristics*

**Langkah ke-1**

Jika blok *i*, merupakan blok yang terjauh dari depot, maka bisa dimulai dengan :

$L_\ell^a$  mulai pada node  $b_{i1}$  dan selesai pada node  $a_{i1}$  dan transisinya  $t_1$

$L_\ell^b$  mulai dan berakhir pada node  $b_{i1}$  dan transisinya  $t_3$   
atau

$L_\ell^a$  mulai dan berakhir pada node  $a_{i1}$  dan transisinya  $t_4$

$L_\ell^b$  mulai dari node  $a_{i1}$  dan berakhir pada node  $b_{i1}$  dan transisinya  $t_1$

$a_{ij}$  merupakan bagian belakang *subaisle* *j* dalam blok *i*.

$b_{ij}$  merupakan bagian depan *subaisle* *j* dalam blok *i*

**Langkah ke-2**

Untuk setiap *subaisle* *j* yang berurutan ( $1 < j < r$ ), maka jika *subaisle* *j* berisi item akan berlaku Persamaan (2). Tetapi jika *subaisle* *j* tidak terdapat item, maka berlaku Persamaan (3).

**Langkah ke-3**

Untuk *subaisle* yang terakhir (*subaisle* *r*), maka ditentukan dengan Persamaan (4). Rumus ini akan digunakan untuk menyelesaikan *order picking route*. *Combined heuristics* ini merupakan penggabungan antara metode *S shape* dan *Return* (Dukic et al, 2010).

$$L_j^a = \begin{cases} L_{j-1}^a + t_a + t_4 & \text{if } c(L_{j-1}^a + t_a + t_4) < c(L_{j-1}^b + t_b + t_1) \\ L_{j-1}^b + t_b + t_1 & \text{otherwise} \end{cases}$$

$$L_j^b = \begin{cases} L_{j-1}^b + t_b + t_3 & \text{if } c(L_{j-1}^b + t_b + t_3) < c(L_{j-1}^a + t_a + t_1) \\ L_{j-1}^a + t_a + t_1 & \text{otherwise} \end{cases} \tag{2}$$

$$L_j^a = L_{j-1}^a + t_a$$

$$L_j^b = L_{j-1}^b + t_b \tag{3}$$

$$L_r^b = \begin{cases} L_{r-1}^b + t_b + t_3 & \text{if } c(L_{r-1}^b + t_b + t_3) < c(L_{r-1}^a + t_a + t_1) \\ L_{r-1}^a + t_a + t_1 & \text{otherwise} \end{cases} \tag{4}$$

### 3. HASIL DAN PEMBAHASAN

Kondisi *layout* gudang PT. GMS untuk dilakukan *order picking* digambarkan seperti terlihat pada Gambar 7. Dari Gambar 7, kotak yang berwarna hitam merupakan lokasi *picking*.

Dimensi lokasi bin adalah 1x1 meter, jumlah *aisle* utama (*main aisle*) ada 3 dimana tiap lebar *aisle* 2.5 meter, panjang *aisle* ( $L$ ) = 4 meter, sedangkan panjang konstan depan *aisle* ( $L_c$ ) = 0.75 meter, jarak antara *aisle* pertama dengan terakhir ( $W$ ) = 9 meter, dan jarak dasar (*basic distance* =  $S_B$ ) = 1.25 meter, jarak *within aisle* =  $S_W$  = 5.5 meter, jarak *across aisle* =  $S_A$  = 9 meter.

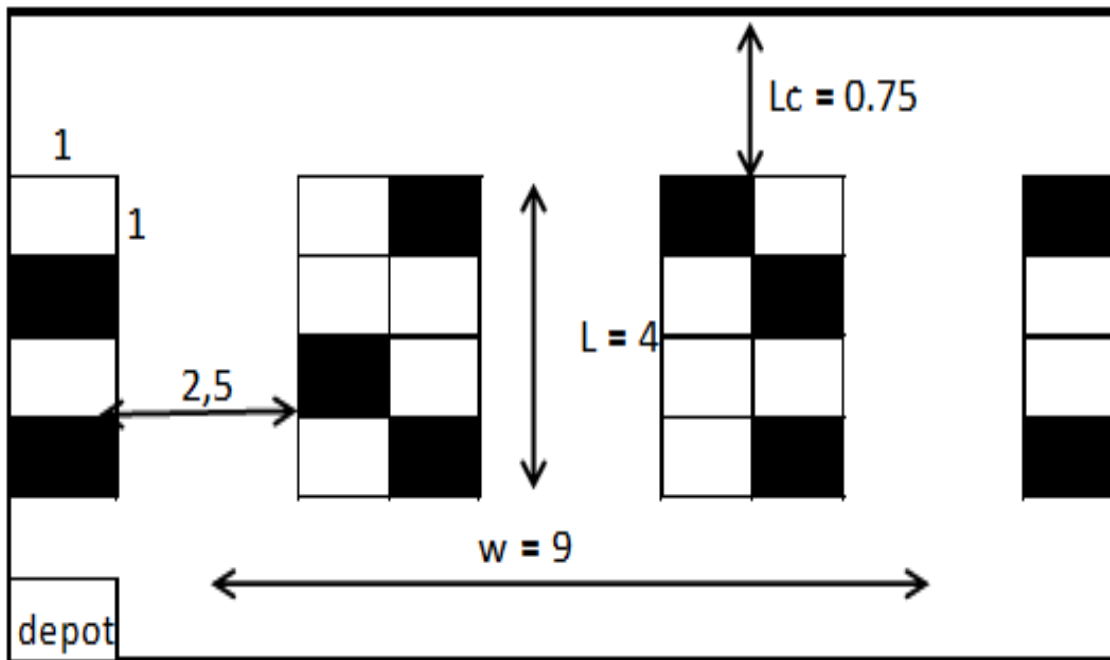
Hasil perhitungan untuk tiap metode *heuristics* adalah sebagai berikut:

- Metode S Shape / Traversal =  $1.25 + 5.5 + 4.5 + 5.5 + 4.5 + (2 \times 4.75) + 9 + 1.25 + 0.5 = 41.5$  m
- Metode Return =  $1.25 + (2 \times 3.75) + 4.5 + (2 \times 4.75) + 4.5 + (2 \times 4.75) + 9 + 1.25 + (3 \times 0.5) = 48.5$  m

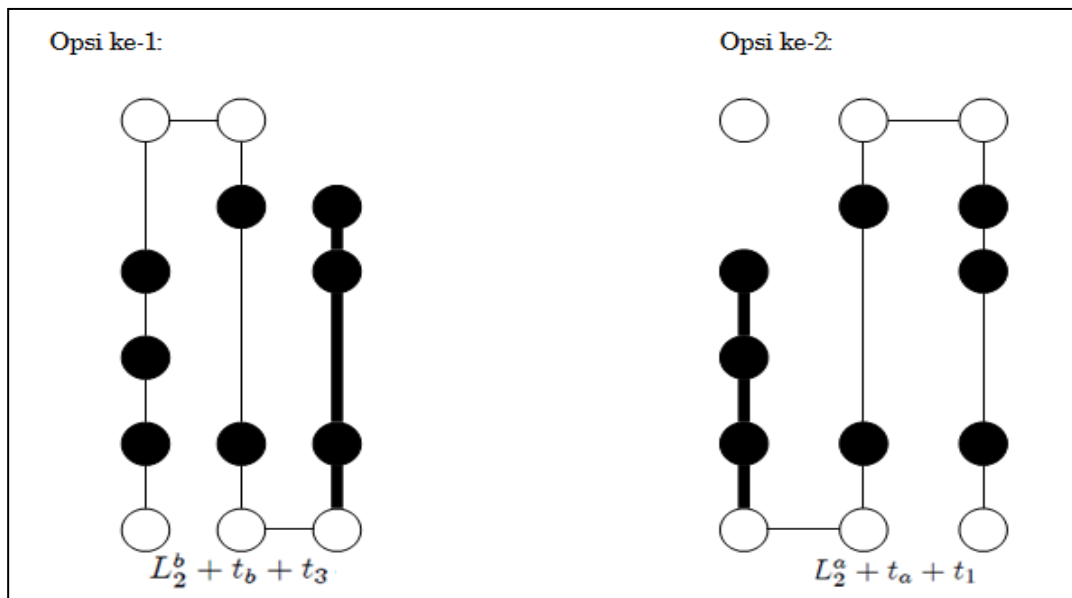
- Metode Midpoint =  $1.25 + 5.5 + 4.5 + (2 \times 1.75) + 4.5 + 5.5 + 4.5 + (2 \times 1.75) + 4.5 + 1.25 + (2 \times 0.5) = 39.5$  m
- Metode The Largest Gap = 39.5 meter
- Metode *Combined Heuristics* dengan menggunakan *dynamic programming*. Hasil kalkulasi untuk dua opsi (Opsi ke-1 dan Opsi ke-2) disajikan pada Gambar 8.

Kedua opsi tersebut merupakan jarak terpendek dari *dynamic programming* yang terlihat pada Gambar 8. Metode *combined heuristics* ini menghasilkan jarak:

- a. Opsi ke-1:  
 $1.25 + 5.5 + 4.5 + 5.5 + 4.5 + (2 \times 4.75) + 9 + 1.25 + 0.5 = 41.5$  m
- b. Opsi ke-2:  
 $1.25 + (2 \times 3.75) + 0.5 + 4.5 + 5.5 + 4.5 + 5.5 + 9 + 1.25 = 39.5$  m



Gambar 7. *Layout Order Picking*



Gambar 8. Hasil kalkulasi *dynamic programming*

Dari kelima metode tersebut, jarak terdekat adalah 39.5 meter yang dihasilkan oleh metode *Midpoint*, *The Largest Gap* dan *Combined Heuristics*. Tabel jaraknya disajikan pada Tabel 1.

Tabel 1. Rekap Metode *Routing*

Metode	Jarak (meter)
<i>S shape / Traversal</i>	41.5
<i>Return</i>	48.5
<i>Midpoint</i>	39.5
<i>The Largest Gap</i>	39.5
<i>Combined</i>	39.5

Dengan demikian di gudang PT.GMS, proses *picking* bisa dilakukan dengan metode *routing Midpoint* atau *The Largest Gap* ataupun *Combined Heuristics*. Dibandingkan dengan metode *S shape / Traversal*, maka lebih efisien sebesar 4.8%, sedangkan dibandingkan dengan metode *Return*, maka lebih efisien 18.5%.

Faktor lain yang bisa dianalisis lebih lanjut untuk mempercepat proses *picking* di PT. GMS adalah menggunakan *picker* lebih dari satu orang, akan tetapi bisa menimbulkan *congestion* (kemacetan) pada saat mereka bertemu di *aisle* atau lokasi *picking* yang sama. Jika terjadi kemacetan maka konsekuensinya bisa memperlambat proses *picking*, namun hal ini bisa dianalisis lebih lanjut dengan model antrian. Faktor lain yang bisa dianalisis lagi adalah dengan menggunakan *picking strategy: zone picking*, strategi ini dapat mengurangi kemacetan dan mempercepat waktu *travel* (De Koster, et al, 2011).

#### 4. KESIMPULAN

Pada penelitian di gudang PT. GMS saat ini metode *routing* yang paling efisien dalam melakukan proses *picking* secara system manual adalah dengan menggunakan metode *Midpoint*, *The Largest Gap* dan *Combined - dynamic programming* dengan jarak tempuh 39.5 meter. Metode ini lebih pendek 4.8% dibandingkan dengan metode *S Shape* dan 18.5% dibandingkan dengan metode *Return*.

Penelitian lebih lanjut bisa dilakukan dengan memasukkan faktor kemacetan (*congestion / blocking*) dan penggunaan metode *batching* atau *zoning*.

#### 5. DAFTAR PUSTAKA

1. Bartholdi, J.J, and S.T. Hackman, Warehouse and Distribution Science, Georgia Institute of Technology, 2008, available on line: [www.warehouse-science.com](http://www.warehouse-science.com)
2. Celik, M. and H. Sural, The Order Picking Problem in Fishbone Aisle Warehouses, July 19, 2012.
3. De Koster, R.E, T.L. Duc, N. Zaerpour, Determining the Number of Zones in a Pick-and-Sort Order Picking System, *International Journal of Production Research*, 2011.
4. De Koster, R., E. Van der Poort, Routing Orderpickers in a Warehouse: A comparison between optimal and heuristic solution, *IIE Transaction* 30, 1998, p.469 – 480.
5. De Koster, R., K.J. Roodbergen, R. Van Vourden, Reduction of Walking Time in the Distribution Center of De Bijenkorf, *New Trends in*

- Distribution Logistics*, Springer, Berlin, 1999, p. 215 – 234.
6. De Koster, R., Le-Duc, T., Roodbergen, K.J., Design and Control of Warehouse Order Picking: A Literature Review, *European Journal of Operational Research* 182 (2), 2007, p. 481 – 501.
  7. Dukic, G., V. Cesnik, T. Opetuk, Order Picking Methods and Technologies for Greener Warehousing, *Strojarstvo* 52 (1), 2010, p.23 - 31
  8. Dukic, G., and C. Oluic, Order Picking Methods: Improving Order – Picking Efficiency, *International Journal of Logistics System and Management* 3 (4), 2007, p. 451 – 460.
  9. Dukic, G., and T. Opetuk, Analysis of Order Picking in Warehouses with Fishbone Layout, *Proceedings of ICIL*, Tel Aviv, Israel, 2008.
  10. Gue, K., and R.D. Meller, Aisle Configuration for Unit Load Warehouse, *IIE Transaction*, 2008.
  11. Roodbergen, K.J., and I.F.A. Vis, A Model for Warehouse Layout, *IIE Transaction* 38 (10), 2006, p. 799 – 811.
  12. Roodbergen, K.J., and R. De De Koster, Routing Methods for Warehouse With Multiple Cross Aisle, *International Journal of Production Research* 39 (1), 2001, p. 1865 – 1883.
  13. Sadowsky, W., and M. Ten Hompel, Calculation of The Average Travel Distance in a Low Level Picker to Part System Considering Any Distribution Function With The Aisle, *Logistics Journal: Reviewed*, ISSN 1860 – 7977, 2011.
  14. Theys, C., O. Braysy, W. Dullaert, B. Raa, Using a TSP Heuristic for Routing Order Picker in Warehouse, *European Journal of Operational Research* 200, 2010, p.755 – 763.
  15. Tompkins, J.A., J.A. White, Y.A. Bozer, J.M.A. Tanchoco, *Facilities Planning*, John Wiley and Sons, New Jersey, 2003.