© 2005 – ongoing JATIT & LLS
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

MULTI STEP AUTOMATIC ORDER FULFILLMENT MODEL IN GENERAL TRADING COMPANY

PURBA DARU KUSUMA
School of Electrical Engineering, Telkom University, Indonesia
E-mail: purbodaru@telkomuniversity.ac.id

ABSTRACT

Purchase order fulfillment is a common activity in many companies, especially in general trading company. Meanwhile, this process is never easy, especially for companies that execute this process manually. Related to it, the most common method in purchase order fulfillment is first come first served (FCFS) model. Although this model is very popular, many companies may face classical problem in matching stock with orders: under supplied or over supplied. Besides that, companies also face other problem in failing order fulfillment. Based on this problem, the objective of this research is proposing new order fulfillment model so that the mismatch between order and fulfillment can be reduced. In this work, we propose three purchase order fulfillment model. In this work, our proposed models are developed by combining several common models, such as: FCFS, smallest quantity, highest price, and bidding models. The hypothesis is that by combining the existing popular FCFS model with other models, this mismatch problem can be reduced. Every model in this work is developed based on multi step fulfillment to explore the advantages of every basic model that is used in it. We use general trading company because in this company type, there is not any raw material in its inventory. Based on the simulation result, our proposed models have better performance in increasing the number of successful transactions than the existing models at the same condition. The proposed FCFS-bidding model performs the best in total revenue aspect. Meanwhile, these three proposed models perform better than the existing FCFS model when the average order quantity is high although this advantage is not significant.

Keywords: *Purchase order, First Come First Served (FCFS), Bidding, Smallest Quantity, Highest Price.*

1. INTRODUCTION

Purchase order fulfillment is a common activity in many companies, especially for routine order-delivery company or general trading company. For client, purchasing process is a part of supply chain management [1]. Although this process is common, the fulfillment is never easy. To execute all orders, company must allocate its product in certain quantity of product in its inventory so the stock is safe to fulfill the incoming orders or demand. When the quantity is too small, there are purchase orders that cannot be fulfilled. In many cases, customer can give penalties to its supplier if the supplier fails to meet customer's requirement [2]. In the other side, if the quantity is too big, there will be over supplied condition so that company spends too much resource but the final product needs more time in the inventory before they are delivered to the customers in the future purchase orders [3].

One performance aspect in order fulfillment is lead time. There are several definitions for lead time. Lead time can be defined as time interval between order creation and order fulfillment or completion. Other definition of lead time is the time between start and finish of production [5]. Many researches have done in order to reduce lead time [4,5]. The other term that has similar definition is waiting time. In taxi business, especially in online taxi, waiting time is time from a passenger creates an order until the taxi arrives to pickup him [6-10]. In taxi business, this process meets similar problem. If the taxi fleet is under supplied, the waiting time will be longer or fail orders rate will rise. Based on these researches [6-10], in taxi business, waiting time should be maintained as low as possible.

In many companies, order or purchase order is still fulfilled manually. In this process, company allocates several personels to fulfill purchase orders. When a purchase order comes, these personels will check to the inventory whether the quantity is enough so that the order can be fulfilled. Besides this method, batch approach also can be implemented. In batch approach, orders that come are collected until certain amount or time interval. After that, the orders are fulfilled.

The most common model in purchase order fulfillment is first come first serve (FCFS) model. This model is very popular and it is implemented in many areas, such as in computer process [11-14]. Krishna [11] used this method in big data processing in MapReduce framework. Husain, et al [12] modified this method in disc schedulling work. Umar and Pujiyanta [13] used FCFS in scheduling MPI job in the grid system. FCFS is also used in CPU job scheduling algorithm [14]. This method is also popular in customer order fulfillment in manufacturing system [15-17]. In this method, order that comes earlier will be fulfilled earlier. This method is very fair. The problem is when the current order has not been fulfilled yet then orders after this current order cannot be fulfilled although the stock in the inventory is enough to fulfill these orders. In this case the total waiting time or lead time will arise. Meanwhile, Seidman found that FCFS method can be unstable in manufacturing system [18].

Based on this problem, the main reseach question is what model besides FCFS that can be used to solve this purchase order fulfillment problem. The second question is how can this model is implemented automatically. Based on these research questions, the objective of this research is developing or proposing new order fulfillment model that is better than FCFS model in increasing the number of successful transaction and optimizing the stock. The proposed order fulfillment model should also reduce the mismatch between order and fulfillment. Besides the research question and research objective, the hypothesis of this work is that by combining the existing FCFS model and other models, this mismatch problem can be reduced.

This paper is organized as follows. In section one, we explain the background, research question, and the research purpose. In section two, we explain and make illustration about common problem in current purchase order fulfillment method, especially in FCFS method. In section three we propose the new purchase order fulfillment model. In section four, we explain the implementation of this model into simulation application. In section five, we discuss the simulation result and the performance comparion among models. In section six, we explain the findings, novelties, and contribution. In section seven, we explain the research limitations. In section eight, we conclude the result and propose future research potentials.

2. PROBLEM IN COMMON PURCHASE ORDER FULFILLMENT METHOD

In this section, we explain the condition and problem that rises in the common purchase order fulfillment. Generally, first come first served (FCFS) model is very popular and is used in many companies or processes. This model is also used in many scheduling process.[11-14] In this model, order that comes earlier will be prioritized rather than order that comes later as long as the system can proceed or executes this order. The formal method of the FCFS model is shown in Equation 1.

$$o_{sel} = o \mid \min(t_{arr}(o)) \wedge n_{q\_order}(o) \leq n_{stock} \quad (1)$$

In Equation 1, o_{sel} is order that is selected to be proccesed and t_{arr} is the arrival time of the order. In Equation 1, it is shown that order with the lowest arrival time will be prioritized. Besides, the order quantity (n_{q_order}) must be equal to or lower than the number of stock (n_{stock}) so that the order can be fulfilled.

The illustration of the basic FCFS model is as follows. Suppose that there are 10 customers {c_1, c_2, c_3, …, c_{10}} that order product to the company A. The orders detail is shown in Table 1. In Table 1, it is shown that total requested quantity is 40 units. Meanwhile, the company has only 30 units of the requested product. So, there will be unfulfilled orders.

Table 1. Customers' Attributes

Company	n_{q_order} (unit)	t_{arr} (date)	p_{buyer} (rupiah)
c_1	1	2	15,000
c_2	4	4	14,500
c_3	7	7	15,000
c_4	4	11	16,000
c_5	5	3	13,000
c_6	3	6	13,400
c_7	2	5	14,000
c_8	5	10	14,200
c_9	6	15	15,100
c_{10}	3	12	15,200

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

When the company A uses FCFS model, the fulfillment result is as follows. The fulfilled orders are: c_1, c_5, c_2, c_7, c_6, and c_3. So, there are only six orders that can be fulfilled. There are four orders that cannot be fulfilled. Meanwhile, the total quantity of these orders is 27 units. Because of the initial stock is 30 units, there will be 3 remaining stocks.

The other method is prioritizing the order with bigger quantity. The reason of this method is to minimize the number of orders so that with the similar quantity, company can focus on less number of customers. This method is formalized in Equation 2. In Equation 2, it is shown that the selected order will be the order with the maximum order quantity. Similar to the FCFS model, the order quantity must be equal to or lower than the stock quantity.

$$o_{sel} = o \mid \max\left(n_{q\_order}(o)\right) \wedge n_{q\_order}(o) \leq n_{stock} \quad (2)$$

When the company uses bigger quantity first, the fulfilled orders are: c_9, c_{10}, and c_3. The reason is as follows. c_9 and c_{10} are two orders with the biggest quantity. Total number of these two orders is 27 units. After these two first orders, the remaining stock is three units. Although the order with the third biggest quantity is c_4 with the quantity is 11 units, this order cannot be fulfilled because the quantity is higher than the remaining stock. So, the most possible unfulfilled order is c_3 where the quantity is 3 units.

Based on this result, it is shown that there are three orders that are fulfilled. Fortunately, total quantity of these orders is 30 units. So, there is not any remaining stock so that it is better than the FCFS model in optimizing stocks.

The third common model is highest price prioritization model. In this model, the order with higher transaction price will be prioritized. This model is formalized in Equation 3. As it is shown in Equation 3, the selected order is the order where the price (p_{order}) is the highest one. Similar to the previous models, the quantity of the order must be equal to or lower than the remaining stock.

$$o_{sel} = o \mid \max\left(p_{order}(o)\right) \wedge n_{q\_order}(o) \leq n_{stock} \quad (3)$$

When the company uses this highest price prioritization model, the result is as follows. The fulfilled orders are: c_9, c_{10}, and c_1. In this situation, there are only three orders that are fulfilled. The third fulfilled order is c_1 rather than c_3 although these two orders have same price. It is because the remaining stock after executing c_9 and c_{10} is three units. So, only c_1 meets the requirement.

Besides those requirements, all of these three models have similar general rule. This rule is formalized by using Equation 4. The selected order must be the member of set O which O is the set of purchase orders that are received by the company. Besides that, the order has not been fulfilled so that redundant fulfillment can be avoided. In Equation 4, the order of the status is symbolized with s(o) and it has two possible values: 0 for unfulfilled order and 1 for fulfilled order.

$$o_{sel} = o \mid o \in O \wedge s(o) = 0 \quad (4)$$

3. PROPOSED MODEL

Based on that condition, in this work, we propose multi step purchase order fulfillment model. There are three models that are proposed. In all of these proposed models, FCFS model is used in the first step. In the first model, we combine the FCFS model with the highest price first model. In the second model, we combine the FCFS model with the smallest quantity and the highest price models. In the third model, we combine the FCFS mode with the bidding model.

The first model is FCFS-highest price combined model. In this model, the process is devided into two steps. FCFS model is used in the first step while the highest price model is used in the second step. These models are executed sequentially. The FCFS-highest price model main algorithm is shown in Figure 1. In the beginning, the order status of order i (s_i) is set 0 and this process occurs for all orders. Procedure FCFS() is used to execute FCFS model. Procedure highest_price() is used to execute the highest price model. Variable n_{trans} is the number of transactions and p_{tot_rev} is the total revenue.

In Figure 1, there is variable $n_{min\_q\_order}$. This variable represents the minimum quantity that must be fulfilled by the company so that the transaction occurs. This basic concept is company may provide product with the quantity is lower than the customer's expectation as long as not lower than the minimum quantity. This minimum quantity is determined by using procedure set_minorder based on Equation 5. In Equation 5, variable r_{q_order}

is the deviation ratio of the order. It represents the willingness of the customer i to give quantity discount. It is a floating point and its value ranges from 0 to 1. Bigger value of it means bigger willingness to give quantity discount.

```
begin
  for i = 0 to n_order-1 to
  begin
    s_i ← 0
    n_min_q_order(i)←set_minorder(i)
  end
  n_trans ← 0
  p_tot_rev ←0
  FCFS()
  highest_price()
end
```

Figure 1. FCFS-highest Price Main Algorithm

$$n_{\min\_q\_order}(i) = (1 - r_{q\_order}(i))n_{q\_order}(i) \quad (5)$$

After all order status is set zero, the next step is running the FCFS model. This model is developed based on Equation 1 and Equation 4. The FCFS model algorithm is shown in Figure 2.

```
begin
 i ← 0
 run ← 1
 while run = 1 do
 begin
  if n_stock ≥ n_min_q_order(i) then
  begin
   p_rev ← p_buyer(i) * n_min_q_order(i)
   p_tot_rev ←p_tot_rev + p_rev
   n_stock ← n_stock - n_min_q_order(i)
   s(i) ← 1
  end
  i++;

  if i ≥ n_buyer or n_stock ≤ 0 then
   run ← 0
 end
end
```

Figure 2. FCFS Process Algorithm

There are several new variables that are used in this algorithm. Variable p_{rev} is the revenue in a single transaction. Variable p_{tot_rev} is the total revenue. Variable run indicates whether the process still runs (1) or stops (0). In this algorithm, it is shown that the process stops only if all customers have been scanned or there is not any remaining stock.

The next process is running the highest price procedure. This procedure is developed based on Equation 3 and Equation 4. The highest price algorithm is shown in Figure 3.

The highest price process is devided into two sub processes: cancidate selection and transaction fulfillment. As it is shown in Figure 3, the candidate that is selected must meet three requirements. First, the order has been fulfilled in the first FCFS process. Order that has not been fulfilled in the first FCFS process cannot be candidate in the highest price process. Second, the remaining stock is higher than or equal to the rest quantity of the order so that the order can be fulfilled completely. Third, the price of the order must be the highest among other candidates. The candidate index is then stored in variable sel.

```
begin
 run ← 1
 while run = 1 do
 begin

  //selecting the candidate
  sel ← -1
  p_max ← 0

  for i = 0 to nbuyer
  begin
   n_dev ← n_q_order(i) - n_min_q_order(i)
   if s(i) = 1 and n_stock ≥ n_dev and
   p_buyer(i) > p_max then
   begin
    sel ← i
    p_max ← p_buyer(i)
   end
  end

  //transaction fulfillment
  if sel > -1 then
  begin
   n_dev ← n_q_order(sel)-n_min_q_order(sel)
   p_rev ← p_buyer(sel) * n_dev
   p_tot_rev ← p_tot_rev + p_rev
   n_stock ← n_stock - n_dev
   s(sel) ← 2
  end
  else
   run ← 0
 end
end
```

Figure 3. Highest Price Process Algorithm

In the transaction fulfillment part, there will be transaction only if there is selected candidate. After transaction of this order is fulfilled then its status value is set 2. The highest price process will stop only if there is not any selected candidate.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The second model is the combination of FCFS model, smallest quantity model, and highest price model. This model is executed sequentially from FCFS, smallest, and then the highest price models. The main algorithm of this model is similar to the FCFS-highest price model. The difference is there is smallest quantity model between the FCFS model and the highest price model. The smallest quantity model follows formula in Equation 2, Equation 6, and Equation 4. Meanwhile, the smallest quantity model algorithm is shown in Figure 4.

$$o_{sel} = o \mid \min(n_{q\_order}(o)) \wedge n_{q\_order}(o) \leq n_{stock} \quad (6)$$

```
begin
 run ← 1
 while run = 1 do
 begin
  // find the lowest stock
  nmax ← nmin_q_order(0)
  for i=1 to nbuyer-1
  begin
   if nmin_q_order(i) > nmax then
    nmax ← nmin_q_order(i)
  end

  //find possible candidate
  sel ← -1
  nmin ← nmax
  for i = 0 to nbuyer-1
  begin
   if s(i)=0 and nstock ≥ nmin_q_order(i)
   and nmin_q_order(i) ≤ nmin then
   begin
    sel ← i
    nmin ← nmin_q_order(i)
   end
  end

  //transaction fulfillment
  if sel > -1 then
  begin
   prev ← pbuyer(sel)*nmin_q_order(sel)
   ptot_rev ← ptot_rev + prev
   nstock ← nstock - nmin_q_order(sel)
   s(sel) ← 1
  end
  else
   run ← 0

 end
end
```

Figure 4. Smallest Quantity Process Algorithm

The third model is the combination between FCFS model and bidding model. The basic concept of the bidding model is that after the FCFS runs, every customer has multiple times of opportunity to acquire the remaining stock. This concept is different to the basic highest price model where customer has only one opportunity to propose price, in this third model, customer can propose multiple price.

The basic concept of bidding is as follows. The remaining stock is devided into packets where every packet has certain quantity called step. The step size is static. Customers whose order quantity has not been completely fulfilled have opportunity to bid for the packet as long as the step is smaller than or equal to to the needed quantity. These candidates submit one bidding price in every single packet offering. For every candidate, his bidding price may be different in every offered packet. The winner is the candidate who submits the highest price.

The bidding price is determined by using Equation 7 and Equation 8. In Equation 7, $p_{bid}(i,t)$ is the bidding price that is submitted by customer i at time t. This price contains two components: static price (p_{buyer}) and dynamic price (p_{var}). Variable p_{buyer} is set in the initial process and is used in FCFS model. Meanwhile, the dynamic price varies during the bidding session. The dynamic price is determined by using Equation 8. In Equation 8, p_{varmin} is the minimum multiplier, p_{varmax} is the maximum multiplier, and $p_{varstep}$ is the price step.

$$p_{bid}(i,t) = p_{buyer}(i) + p_{var} \quad (7)$$

$$p_{var} = rand(p_{var\min}, p_{var\max}).p_{var\,step} \quad (8)$$

The bidding process algorithm is shown in Figure 5. In Figure 5, several new variables are used. n_{step} represents the step size or packet size. n_{cur_q} represents the current order quantity of customer i that has been fulfilled.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

```
Begin
 run ← 1
 while run = 1 do
 begin
  if nstock ≥ nstep then
  begin
   //submiting bidding price
   for i = 0 to nbuyer-1
    set_biddingprice(i)

   //finding the candidate
   sel ← -1
   pmax ← 0
   for i=0 to nbuyer-1
   begin
    ndev ← nq_order(i) – nstep
    if ncur_q(i)≤ndev
    and pbid(i)>pmax then
    begin
     sel ← i
     pmax ← pbid(i)
    end
   end
   //transaction fulfillment
   if sel > -1 then
   begin
    prev ← nstep * pbid(sel)
    nstock ← nstock – nstep
    s(sel) ← 1
    ncur_q(sel) ← ncur_q(sel) + nstep
   end
  end
 end
end
```

Figure 5. Bidding Process Algorithm

4. IMPLEMENTATION

The proposed model is then implemented in purchase order fulfillment simulation application. This application is a web based application. It is developed by using PHP language. In this system, we use MySQL database to store the simulation result.

The scenario of this simulation is as follows. There is a virtual trading company. In the beginning of the simulation, the initial variable of this company is set, includes: initial stock ($n_{stock,0}$), bottom price (p_{bottom}), top price (p_{top}), price deviation (p_{dev}), and step size (n_{step}). After the company initial variables are set then customers or orders are generated.

There are several variables that are related to the customers or orders. The first is the number of customers (n_{buyer}). The order quantity is generated stochastically for every order. This value follows exponential distribution based on the average order quantity ($n_{av\_q\_order}$). Meanwhile, there is modification of this basic exponential distribution because of several reasons. First, the output of the exponential distribution is a floating point number. In the other side, the quantity of the order is integer. So, the output of this random variate must be discretized. Besides that, the exponential distribution ranges from zero to infinite number so that it must be limited. The creation of the quantity of the order is determined by using Equation 9 and Equation 11.

$$n_{variat} = -\log\left(\frac{5{,}100}{100}\right).n_{av\_q\_order} \quad (9)$$

$$n_{q\_orderbase} = \text{int}(n_{variat}) \quad (10)$$

$$n_{q\_order} = \begin{cases} 1, n_{q\_order} = 0 \\ n_{q\_order}, else \end{cases} \quad (11)$$

After the quantity of the order is set, the next process is determining the buyer's price. This price is also generated randomly and its value follows uniform distribution so that price that is proposed by a customer may be different to the price that is proposed by the other customers. The price ranges from the company's bottom price to the top price. In this work, the customer's price is determined by using Equation 12 to Equation 14. In Equation 13, the final order price consists of two components: static part (p_{bottom}) and dynamic part ($p_{dynamic}$). The dynamic price is determined by the random price and the price step (p_{step}). After this buyer's variables are set, the purchase order fulfillment process runs.

$$p_{order}(i) = p_{bottom} + p_{dynamic} \quad (12)$$

$$p_{dynamic} = p_{random} \cdot p_{step} \quad (13)$$

$$p_{random} = rand\left(0, \frac{p_{top} - p_{bottom}}{p_{step}}\right) \quad (14)$$

5. DISCUSSION

After the simulation application is built based on these three proposed models, performance of these models is then observed and is evaluated. This process is fulfilled by running the simulation process and then storing the simulation result. In this evaluation process, there are several adjusted variables or independent variables and there are several observed variables of dependent variables.

Journal of Theoretical and Applied Information Technology
30th April 2019. Vol.97. No 8
© 2005 – ongoing JATIT & LLS

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The adjusted variable is variable which its value is set manually before the simulation runs. Meanwhile, the observed variable is variable which its value is obtained after the simulation process ends and its value depends on the adjusted variable value. In this simulation, the adjusted variables includes: initial stock, top price, bottom price, price deviation, step size, number of buyers, average number of order quantity, and maximum order ratio. Meanwhile, the observed variables include: total revenue, total number of transaction, and remaining stock.

The evaluation and observation is done by evaluating the relation between the adjusted variables and the observed variables. Behavior of the system is evaluated by by observing the observed variables due to one adjusted variable that its value changes gradually. While the value of one adjusted variable changes, other adjusted variable value is static and it is set at its default value. The adjusted default value is shown in Table 2.

There are four test groups in this work. The tests are grouped based on the changing adjusted variables. These changing variables are the initial stock, the number of buyers, average quantity of the order, and maximum order ratio.

Table 2. Default Value of Adjusted Variables

Variable	Value	Unit
$n_{stock}(0)$	500	unit
$p_{s\ top}$	10,000	rupiah
$p_{s\ bottom}$	7,000	rupiah
p_{dev}	2,000	rupiah
n_{step}	1	-
n_{buyer}	50	person
$n_{av\ q\ order}$	10	unit
$r_{max\ order}$	0.5	-

In this work, there are several purchase order fulfillment models that are used. Besides these three proposed models (FCFS-highest price model, FCFS-smallest quantity-highest price model, and FCFS-bidding model), we also implement four common models (FCFS model, biggest quantity model, smallest quantity model, and highest price model). So, in every test group, the output of these seven models is observed so that the performance of these seven models can be evaluated and can be compared to each others.

In the first test group, relation between initial stock and the observed variables is analyzed. In this test group, the initial stock ranges from 100 units to 1,000 units. The step size is 100 units. There are 10 sessions in every initial stock value. The result is shown in Table 3 to Table 5.

Table 3. Relation Between Number of Stock and Total Revenue

n_s (unit)	Total Revenue (rupiah)						
	FCFS	Biggest	Smallest	Highest	FCFS-highest	FCFS-small-high	FCFS-bid
100	850,000	829,300	826,020	964,250	849,800	849,800	849,800
200	1,713,264	1,697,173	1,674,000	1,855,064	1,711,582	1,711,582	1,711,582
300	2,542,255	2,542,636	2,476,045	2,633,509	2,565,727	2,565,727	2,622,127
400	3,105,082	3,117,073	3,088,209	3,144,464	3,138,882	3,138,882	3,304,718
500	3,234,700	3,234,700	3,234,700	3,234,700	3,234,700	3,234,700	3,422,673
600	3,310,427	3,310,427	3,310,427	3,310,427	3,310,427	3,310,427	3,500,200
700	3,457,500	3,457,500	3,457,500	3,457,500	3,457,500	3,457,500	3,665,233
800	3,610,327	3,610,327	3,610,327	3,610,327	3,610,327	3,610,327	3,826,018
900	3,566,358	3,566,358	3,566,358	3,566,358	3,566,358	3,566,358	3,785,275
1,000	3,529,145	3,529,145	3,529,145	3,529,145	3,529,145	3,529,145	3,738,555

Table 4. Relation Between Number of Stock and Number of Transactions

n_s (unit)	Total Number of Transactions (Unit)						
	FCFS	Biggest	Smallest	Highest	FCFS-highest	FCFS-small-high	FCFS-bid
100	13	5	29	14	20	20	20
200	27	11	39	28	36	36	36
300	40	24	46	41	49	49	49
400	48	41	49	48	50	50	50
500	50	50	50	50	50	50	50
600	50	50	50	50	50	50	50
700	50	50	50	50	50	50	50
800	50	50	50	50	50	50	50
900	50	50	50	50	50	50	50
1,000	50	50	50	50	50	50	50

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 5. Relation Between Number of Stock and the Remaining Stock

n_s (unit)	Remaining Stock (Unit)						
	FCFS	Biggest	Smallest	Highest	FCFS-highest	FCFS-small-high	FCFS-bid
100	0	0	3	0	0	0	0
200	0	0	5	0	0	0	0
300	0	0	8	1	0	0	0
400	35	33	38	34	33	33	33
500	123	123	123	123	123	123	123
600	210	210	210	210	210	210	210
700	295	295	295	295	295	295	295
800	376	376	376	376	376	376	376
900	480	480	480	480	480	480	480
1,000	589	589	589	589	589	589	589

Table 3 shows that in the beginning, the increasing of the initial stock makes the total revenue increases too. First, the slope is very high especially when the initial stock is from 100 to 400 units. After that, there is increasing in total revenue but the slope is not high. When the initial stock is bigger than 800 units, the total revenue tends to fluctuate. This condition is rationale because at the beginning, the initial stock cannot fulfill all purchase orders. When the initial stock increases, the number of purchase orders that can be fulfilled increases too. But, after all of purchase orders or most of purchase orders can be fulfilled, there is not any potential to increase total revenue.

The comparison about total revenue performance among models due to the increasing of the initial stock size as shown in Table 3 is as follows. The FCFS-bidding model achieves the highest total revenue among other models. In the beginning, the gap in total revenue is not significant. This gap goes wider due to the increasing of the total revenue. Meanwhile, models that implements highest price model (highest price model, FCFS-highest price model, and FCFS-smallest quantity-highest price model) get advantage. But, the total revenue between these models and other models is not significant. The gap is more significant when the initial stock is low and the gap goes smaller when the initial stock increases.

Table 4 shows that the increasing of the initial stock makes the number of transactions increases. At the beginning when the initial stock is low, the number of transactions is low too. Then, the number of transactions increases due to the increasing of the initial stock. After the initial stock reaches certain size, the number of transactions tends to stagnant. In this test, the initial stock threshold is 500 units. After that, the number of transactions is stable in 50 transactions. This condition is rational because when the initial stock is low, not all orders can be fulfilled. Then, the number of transactions increases when the initial stock increases. But, after all of transactions can be fulfilled, there is not any unfulfilled transactions left so that the number of transactions is stagnant.

Comparing to each others, after the initial stock threshold is achieved, the performance in total number of transactions related to the initial stock size is equal among models. Meanwhile, when the initial stock is lower than the initial stock threshold, there is variation in performance among models. The biggest quantity model performs the lowest number of transactions. All of three proposed models perform similar result. The smallest quantity model performs the highest number of transactions. The FCFS model and the highest price model perform similar result and their position is moderate.

In the remaining stock aspect, Table 5 shows that when the initial stock size goes higher, the remaining stock increases too. When the initial stock is low, in most models, the remaining stock is zero because most of or all of stock is absorbed by the purchase order. Meanwhile, when the initial stock size goes higher, this stock tends to be able to fulfill the purchase order so that the remaining stock increases too.

Comparing among models, after the initial stock threshold is achieved, the remaining stock in all models tends to be equal. Before that, the smallest quantity model tends to produce the highest remaining stock compared to other models. Meanwhile, the other models perform equal in remaining stock aspect.

In the second test group, the relation between number of buyers and the observed variables is evaluated. In this test group, the number of buyers ranges from 10 customers to 100 customers. The step size is 10 customers. There are ten simulation sessions in every number of customers value. The result is shown in Table 6 to Table 8.

Table 6. Relation Between the Number of Buyers and Total Revenue

n_{buyer} (person)	Total Revenue (rupiah)						
	FCFS	Biggest	Smallest	Highest	FCFS-highest	FCFS-small-high	FCFS-bid
10	700,127	700,127	700,127	700,127	700,127	700,127	732,573
20	1,330,036	1,330,036	1,330,036	1,330,036	1,330,036	1,330,036	1,393,355
30	2,030,518	2,030,518	2,030,518	2,030,518	2,030,518	2,030,518	2,149,609
40	2,814,273	2,814,273	2,814,273	2,814,273	2,814,273	2,814,273	2,979,400
50	3,203,264	3,203,264	3,203,264	3,203,264	3,203,264	3,203,264	3,394,773
60	4,067,164	4,074,209	4,014,645	4,088,218	4,092,427	4,092,427	4,309,318
70	4,264,709	4,257,791	4,151,582	4,374,645	4,307,864	4,307,864	4,440,309
80	4,303,864	4,311,736	4,190,091	4,486,009	4,337,418	4,337,418	4,415,864
90	4,278,382	4,244,927	4,197,127	4,495,309	4,270,164	4,270,164	4,290,518
100	4,292,945	4,334,945	4,197,200	4,605,073	4,303,627	4,303,627	4,304,091

Table 7. Relation Between the Number of Buyers and Number of Transactions

n_{buyer} (person)	Total Number of Transactions (Units)						
	FCFS	Biggest	Smallest	Highest	FCFS-highest	FCFS-small-high	FCFS-bid
10	10	10	10	10	10	10	10
20	20	20	20	20	20	20	20
30	30	30	30	30	30	30	30
40	40	40	40	40	40	40	40
50	50	50	50	50	50	50	50
60	58	51	59	58	60	60	60
70	60	37	66	60	70	70	70
80	64	35	73	61	79	79	79
90	65	32	80	65	88	88	88
100	64	28	85	60	88	88	88

Table 8. Relation Between Number of Buyers and the Remaining Stocks

n_{buyer} (person)	Remaining Stock (Units)						
	FCFS	Biggest	Smallest	Highest	FCFS-highest	FCFS-small-high	FCFS-bid
10	417	417	417	417	417	417	417
20	342	342	342	342	342	342	342
30	261	261	261	261	261	261	261
40	170	170	170	170	170	170	170
50	122	122	122	122	122	122	122
60	27	26	32	27	26	26	26
70	1	0	11	1	0	0	0
80	0	0	11	0	0	0	0
90	0	0	7	0	0	0	0
100	0	0	10	0	0	0	0

Table 6 shows that if the number of buyers is below the threshold, total revenue will increase due to the increasing of the number of buyers. Meanwhile, after the threshold is surpassed, the total revenue tends to fluctuate. This condition is rationale because as long as the initial stock can fulfill the purchase order, the increasing of the purchase orders will turn to revenue so that the total revenue will increase. Meanwhile, after the threshold is surpassed, it means that the initial stock cannot fulfill the additional orders, the increasing of the purchase orders will not turn to additional revenue because this additional order cannot be fulfilled. In Table 6, the threshold is detected in 70 buyers.

Comparing among models, there is not any significant difference in total revenue. At the same number of buyers, the total revenue is almost equal.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Meanwhile, the FCFS-bidding model provides a little bit higher revenue.

Table 7 shows the dynamic in number of transactions due to the increasing of the number of buyers. Basically, for up to 50 buyers, the number of transactions increases due to the increasing of the number of buyers. Besides, the number of transactions is equal to the number of buyers. It means that all purchase orders can be fulfilled.

After that, the responses are various among models. In three proposed models, until the number of buyers is 70 buyers, the number of transactions is equal to the number of buyers. Then, the number of transactions is still higher than before but it is lower than the number of buyers. It means that after 70 buyers, there will be unfulfilled orders. In FCFS model and highest price model, the total number of transactions tends to fluctuate. It means that after the threshold, the increasing of the number of buyers does not affect to the number of transactions. In the smallest quantity model, the number of transactions still increases but its value is below the value that is produced by the proposed models. The biggest quantity model performs anomaly. The number of transactions tends to decline.

Table 8 shows that if the number of buyers is lower than the threshold, the remaining stock decreases due to the increasing of the number of buyers. This condition is rational because if the number of buyers increases, as long as the orders can be fulfilled, the remaining stock will be reduced. After that, the remaining stock tends to stagnant because there is not any available stock.

Comparing among models, for up to 50 buyers, at the same number of buyers, the remaining stock is equal. Meanwhile, when the number of buyers is 50 buyers, the remaining stock is little bit various. After that, most of remaining stock is zero. But, the remaining stock in the smallest model decreases but its value is still positive.

In the third test group, the relation between the average quantity of the order and the observed variables is evaluated. In this test group, the average quantity ranges from 6 units to 15 units. The result is shown in Table 9 to Table 11.

Table 9. Relation Between the Average Quantity per Order and Total Revenue

$n_{av\_q\_order}$ (unit)	Total Revenue (rupiah)						
	FCFS	Biggest	Smallest	Highest	FCFS-highest	FCFS-small-high	FCFS-bid
6	1,873,773	1,873,773	1,873,773	1,873,773	1,873,773	1,873,773	1,987,682
7	2,338,173	2,338,173	2,338,173	2,338,173	2,338,173	2,338,173	2,479,982
8	2,967,409	2,967,409	2,967,409	2,967,409	2,967,409	2,967,409	3,144,782
9	3,137,327	3,137,327	3,137,327	3,137,327	3,137,327	3,137,327	3,327,955
10	3,673,309	3,672,809	3,662,682	3,673,582	3,674,882	3,674,882	3,896,309
11	3,665,027	3,662,955	3,636,991	3,668,055	3,667,345	3,667,345	3,885,636
12	4,028,209	4,034,573	3,982,009	4,058,209	4,057,118	4,057,118	4,258,782
13	4,135,009	4,158,855	4,017,373	4,235,882	4,202,773	4,202,773	4,360,636
14	4,136,800	4,148,300	3,996,109	4,203,327	4,176,027	4,176,027	4,339,827
15	4,277,745	4,262,100	4,114,927	4,391,155	4,298,964	4,298,964	4,401,555

Table 10. Relation Between the the Average Quantity per Order and Number of Transactions

$n_{av\_q\_order}$ (unit)	Total Number of Transactions (Units)						
	FCFS	Biggest	Smallest	Highest	FCFS-highest	FCFS-small-high	FCFS-bid
6	50	50	50	50	50	50	50
7	50	50	50	50	50	50	50
8	50	50	50	50	50	50	50
9	50	50	50	50	50	50	50
10	50	49	50	50	50	50	50
11	50	49	50	50	50	50	50
12	48	42	49	49	50	50	50
13	45	30	47	46	50	50	50
14	46	35	48	45	50	50	50
15	43	27	46	43	50	50	50

Table 11. Relation Between the Average Quantity per Order and the Remaining Stocks

$n_{av\_q\_order}$ (unit)	Remaining Stock (Units)						
	FCFS	Biggest	Smallest	Highest	FCFS-highest	FCFS-small-high	FCFS-bid
6	278	278	278	278	278	278	278
7	222	222	222	222	222	222	222
8	152	152	152	152	152	152	152
9	132	132	132	132	132	132	132
10	68	68	69	68	68	68	68
11	64	64	67	64	64	64	64
12	27	26	35	28	26	26	26
13	13	10	25	12	10	10	10
14	12	11	27	12	11	11	11
15	2	1	18	2	1	1	1

Table 9 shows that if the average order quantity less than the threshold, the total revenue increases due to the increasing of the average order quantity. After that, the total revenue tends to fluctuate and it means that the increasing of the average order quantity does not affect to the total revenue. In the beginning, the total revenue increases fast. Then, the increasing speed goes low.

Comparing among models, most of models produce similar total revenue. Meanwhile, the FCFS-bidding model produces a little bit higher total revenue. The gap is consistent in any average quantity orders.

Table 10 shows that these three proposed models produces better result in the number of transactions due to the increasing of the average order quantity. In these three proposed models, the number of transactions is maximal for any average order quantity. It means that all purchase orders can be fulfilled although they are not fully fulfilled. Meanwhile, in the FCFS model, the smallest model, and the highest price model, the number of transactions is maximal until the average order quantity is 11 units. After that, the number of transactions decreases. In the biggest quantity model, the number of transactions is maximal until the average order quantity is 9 units. After that, the number of transactions decreases.

Table 11 shows that the remaining stock goes lower due to the increasing of the average order quantity. In the beginning, the decreasing speed is fast. Then, the decreasing speed goes lower. For the average order quantity is up to 9 units, the remaining stock is equal for all models. After that, the remaining stock tends to equal for all models execpt the smallest quantity model. The smallest quantity model tends to produce the highest remaining stock if the average order quantity goes higher. This gap goes wider due to the increasing of the average order quantity.

In the fourth test group, the relation between the maximum order ratio and the observed variables is evaluated. In this test group, the ratio ranges from 0.1 units to 0.9. The step size is 0.1. There are ten simulation sessions in every ratio value. The result is shown in Table 12 to Table 14.

Table 12. Relation Between Maximum Order Ratio and Total Revenue

r_{max_order}	Revenue (rupiah)		
	FCFS-highest	FCFS-small-high	FCFS-bid
0.1	3,420,945	3,420,945	3,498,727
0.2	3,566,455	3,566,455	3,679,636
0.3	3,379,845	3,379,845	3,512,555
0.4	3,338,300	3,338,300	3,510,464
0.5	3,644,045	3,644,045	3,852,036
0.6	3,562,955	3,562,955	3,790,200
0.7	3,673,900	3,673,900	3,949,518
0.8	3,505,818	3,505,818	3,819,618
0.9	3,377,282	3,377,282	3,703,527

Table 12 shows that the total revenue tends to fluctuate due to the increasing of the maximum order ratio. It means that the maximum order ratio does not affect to the total revenue. Comparing among proposed models, the revenue of the FCFS-highest price model and the FCFS-smallest quantity-highest price model is equal. This fact shows that the smallest quantity method that is inserted between the FCFS model and the highest price model does not affect to the revenue. Meanwhile, the FCFS-bidding model produces higher total revenue rather than two other proposed models.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 13. Relation Between Maximum Order Ratio and Total Number of Transactions

r_{max_order}	Total Number of Transactions		
	FCFS-highest	FCFS-small-high	FCFS-bid
0.1	50	50	50
0.2	50	50	50
0.3	50	50	50
0.4	50	50	50
0.5	50	50	50
0.6	50	50	50
0.7	50	50	50
0.8	50	50	50
0.9	50	50	50

Table 13 shows that in any maximum order ratio, the total number of transactions is maximum or equal to the number of buyers. This fact shows that the maximum order ratio does not affect to the number of transactions. It is also because basically, by using these proposed models, the total number of transactions tends to be maximal.

Table 14. Relation Between Maximum Order Ratio and Remaining Stock

r_{max_order}	Remaining Stocks (unit)		
	FCFS-highest	FCFS-small-high	FCFS-bid
0.1	99	99	99
0.2	83	83	83
0.3	103	103	103
0.4	109	109	109
0.5	72	72	72
0.6	82	82	82
0.7	75	75	75
0.8	86	86	86
0.9	104	104	104

Table 14 shows that the remaining stock fluctuates due to the increasing of the maximum order ratio. It shows that maximum order ratio does not affect to the remaining stock. Among proposed models, the remaining stock is equal in any models for any maximum order ratio.

6. FINDINGS AND CONTRIBUTION

Based on the analysis above, there are several findings in this work. These findings can be contributions of this work. These findings are also related to the purpose of this work. These findings are acquired based on the performance of models in observed parameters.

It is shown that these three proposed models have better performance in increasing number of successful transactions rather than other common existing models, especially FCFS [15-17]. This advantage is more significant when the initial stock is far less than the total orders quantity. When the gap between initial order and total order quantity gets lower, this advantage becomes less significant. This fact occurs in all adjusted variables: number of buyers, initial stock, and average order quantity. This fact also shows that this work has met the research purpose in developing models that can improve the number of successful transactions or in other word is reducing fail transactions. Many works said that increasing the number of succesful transactions or order fulfillment can maintain customer loyalty and satisfaction [19,20].

In reducing remaining stock aspect, generally, these three proposed models performance is similar to the existing models, especially when they are compared with the highest price model. Meanwhile, when the average order quantity arises, these proposed models perform a little bit better than the current FCFS model [15-17] and perform much better than the smallest quantity model.

In increasing total revenue, only FCFS-bidding model performs better than other models, including the existing model and other two proposed model. Unfortunately, the FCFS-highest price model and FCFS-smallest quantity-highest price model performance is similar to the existing models. This condition occurs in all adjusted variables: number of buyers, initial stock, and average order quantity.

7. RESEARCH LIMITATIONS

Besides these findings, there are several limitations in this work. These limitations can bes used as basis for for improvement in the future works or it can be future research potentials. First, these models are developed based on single product so that several improvizations or modifications must be improved for multi product order fulfillment model.

Second, these models are developed based on batch system which the initial stock is static and the orders are collected first before fulfilled. The performance may be different if the stock-order condition is dynamic. In this condition, during several period of observation, there are new purchase orders and new product arrivals that come. So, new process simulation must be developed. Third, the environment is general trading company so that modifying these models in the manufacture company will be challenging.

8. CONCLUSION AND FUTURE WORK

This work has proposed three purchase order fulfillment models. Based on the findings, all of these proposed models perform better than the existing models in increasing number of successful transactions so that these new proposed models have met the objective of this research in reducing gap between order and fulfillment. Meanwhile, the advantage in reducing the remaining stock is less significant. Among other models, the FCFS-bidding model produces the highest total revenue.

There are several future research potentials. First, because this work is developed based on single product, improvizations and modifications are needed so that these proposed models can met the characteristics of multi product purchase order fulfillment model. Second, besides batch model that is used in this work, dynamic based purchase order fulfillment model development is also challenging.

REFERENCES:

[1] B. Lawson, P.D. Cousins, R.B. Handfield, K.J. Petersen, "Strategic Purchasing, Supply Management Practices and Buyer Performance Improvement: An Empirical Study of UK Manufacturing Organisations", *International Journal of Production Research*, vol. 47(10), 2009, pp. 2649-2667.

[2] K. Bala, "Supply Chain Management: Some Issues and Challenges- A Review", International Journal of Current Engineering and Technology, vol. 4(2), 2014, pp. 946-953.

[3] K. Grondys, I. Kott, M. Strzelczyk, "The Problem of Excess and Obsolete Inventory Management on the Example of Spare Parts", *Proceeding of FIKUSZ Symposium of Young Researchers*, 2014, pp. 89-96.

[4] J.K. Liker and D. Meier, *Data Mining and Predictive Analytics*, 2nd edition, John Willey & Sons, 2015

[5] A. Jonsson and V. Svensson, "Systematic Lead Time Analysis", Production Engineering Program, Chalmers University of Technology, 2016, master thesis.

[6] P.D. Kusuma, "Multi Parameters Dispatch Model in Taxi Collaboration System", *Journal of Theoretical and Applied Information Technology*, vol. 96(15), 2018, pp. 5042-5053.

[7] P.D. Kusuma, "Auction Based Dispatch Model in Online Motorcycle Taxi System", *Journal of Theoretical and Applied Information Technology*, vol. 96(18), 2018, pp. 6134-6148.

[8] F.A.L. Reis, M.A. Pereira, P.E.M. de Almeida, "Location-Based Service to Reduce the Waiting Time for Taxi Services", *Proceeding of 4th International Workshop on Knowledge Discovery, Knowledge Management and Decision Support*, 2013.

[9] M. Maciejewski, "Benchmarking Minimum Passenger Waiting Time in Online Taxi Dispatching with Exact Offline Optimization Methods", *Archive of Transport*, vol. 30(2), pp. 67-75.

[10] M.S. Rahaman, Y. Ren, M. Hamilton, "Wait Time Prediction for Airport Taxis Using Weighted Nearest Neighbor Regression", *IEEE Access*, November 2018.

[11] M.V. Krishna, "Big Data Processing Using First Come First Served (FCFS) Algorithm", *International Journal of Computer Science and Mobile Computing*, vol. 7(7), 2018, pp. 83-87.

[12] H. Husain, K. Gupta, S. Taneja, "Modified First Come First Serve (MFCFS)", *International Journal of Advanced Computational Engineering and Networking*, vol. 2(11), 2014, pp. 11-15.

[13] R. Umar and A. Pujiyanta, "Development of First Come First Serve-Ejecting Based Dynamic Scheduling (FCFS-EDS) Simulation Scheduling Method for MPI Job in a Grid System", *Journal of Engineering and Applied Sciences*, vol. 12(8), 2017, pp. 1972-1978.

[14] K.S. Kaswan and Amandeep, "A New Technique for CPU Scheduling: Standard Deviation Based", *International Journal of Advanced Research in Computer Engineering & Technology (IJARSET)*, vol. 6(8), 2017, pp. 1278-1282.

[15] D. Lecic-Cvetkovic, N. Atanasov, S. Babarogic, "An Algorithm for Customer Order Fulfillment in a Make-to-Stock Manufacturing System", *International Journal of Computers, Communications and Control*, vol. 5, 2010, pp. 783-791.

[16] R. Pibernik, "Managing Stock-outs Effectively with Order Fulfillment Systems", *Journal of Manufacturing Technology Management*, vol. 17(6), 2006, pp. 721-736.

© 2005 – ongoing JATIT & LLS
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[17] J. Song, S.H. Xu, B. Liu, "Order-Fulfillment Performance Measures in An Assemble-to-order System with Stochastic Leadtimes", *Operation Research*, vol. 47(1), 1999, pp. 131-149.

[18] T.I. Seidman, "First Come, First Served Can be Unstable!", *IEEE Transactions on Automatic Control*, vol. 39(10), 1994, 2166-2171.

[19] S. Thirumalai and K.K. Sinha, "Customer Satisfaction with Order Fulfillment in Retail Supply Chains: Implications of Product Type in Electronic B2C Transactions", *Journal of Operations Management*, vol. 23, 2005, pp. 291-303.

[20] Y. Cao, T.S. Gruca, B.R. Klemz, "Internet Pricing, Price Satisfaction, and Customer Satisfaction", *International Journal of Electronic Commerce*, vol. 8(2), 2003, pp. 31-50.