UNIVERSITAS INDONESIA

Simulasi Crossdock menggunakan software Arena

TESIS

Diajukan sebagai salah satu syarat untuk memperoleh gelar Magister Teknik

FAIZAL

NPM.0906578560

FAKULTAS TEKNIK

PROGRAM STUDI TEKNIK INDUSTRI

DEPOK

JULI 2011

HALAMAN PERNYATAAN ORISINALITAS

Tesis ini adalah hasil karya saya sendiri,

Dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan dengan benar

Nama : FAIZAL

NPM : 0906578560

Tanda Tangan:

Tanggal : Juli 2011

HALAMAN PENGESAHAN

Tesis ini diajukan oleh :

Nama : Faizal

NPM : 0906578560

Program Studi : TEKNIK INDUSTRI

Judul Tesis : Simulasi Crossdock menggunakan software Arena

Telah berhasil dipertahankan dihadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Master 2 Université d’Artois (Perancis) dan Magister Teknik pada Program Studi Teknik Industri, Fakutas Teknik, Universitas Indonesia

DEWAN PENGUJI

Pembimbing : Gilles Goncalvez ( )

Pembimbing : Hamid Allaoui ( )

Penguji : Daniel Jolly ( )

Penguji : Hamid Allaoui ( )

Penguji : Gilles Goncalvez ( )

ARRAS BÉTHUNE DOUAI LIEVIN LENS UNIVERSITÉ D'ARTOIS

Ditetapkan di :

Tanggal :

Mengetahui : Kepala Departemen Teknik Industri UI

(Prof. Teuku Yuri M. Zagloel)

KATA PENGANTAR / UCAPAN TERIMAKASIH

Alhamdulillahirobbil alamin, puji syukur saya panjatkan kepada Tuhan Yang Maha Esa, atas segala berkah, rahmat, petunjuk dari Nya saya dapat menyelesaikan tesis ini. Penulisan tesis ini saya lakukan sebagai langkah memenuhi syarat mencapai gelar Magister Teknik Jurusan Teknik Industri pada Fakultas Teknik Universitas Indonesia. Saya menyadari bahwa tulisan ini merupakan hasil bantuan dan bimbingan dari berbagai pihak dari mula perkuliahan hingga tersusunnya tesis ini. Untuk itu, saya mengucapkan terima kasih kepada:

1. Prof. Teuku Yuri M. Zagloel selaku Ketua Departemen Teknik Industri yang memberikan dukungan positif untuk melanjutkan studi double degree ke Perancis.

2. Prof. Irwan Katili selaku Ketua Program Kerjasama UI-Kemenhub dalam program Double Degree UI-Perancis yang memberikan kesempatan untuk menambah wawasan dan pengetahuan kami di Perancis.

3. Prof.Gilles Goncalvez, DR. Hamid Allaoui, atas bimbingan dan motivasi nya.

4. Tenaga pengajar selama di Universitas Indonesia dan Universitas Artois yang telah banyak memberikan bekal yang sangat berguna untuk tugas akhir ini.

5. Istri ku Puput Yusda Apriliana dan anak-anak kebanggaan dan harapanku dunia akhirat Muhammad Althaf Faizal dan Athallah Ikhwan Faizal yang memberikan dukungan penuh dan doa yang tulus untuk keberhasilan menjalankan amanah menuntut ilmu ini, serta keluarga di Kediri dan Palembang yang banyak membantu dalam segala hal.

6. Serta para sahabat seperjuangan di rantau, dan pihak-pihak yang tidak dapat kami sebutkan satu-persatu yang telah banyak membantu dan mendukung perjuangan study ini.

Pada kesempatan ini pula saya memanjatkan doa dan harapan kepada Allah Subhanahu wata'ala memberikan kebaikan dan balasan yang mulia kepada semua pihak yang telah membantu dan semoga ilmu, hasil, tesis membawa manfaat bagi perkembangan ilmu untuk kebaikan umat.

Bethune-France, 22 Juni 2011

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

Nama : Faizal

NPM : 0906578560

Program Studi : Teknik Industri

Departemen : Teknik Industri

Fakultas : Teknik

Jenis Karya : Tesis

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia Hak Bebas Royalti Noneksklusif (Non-exclusive Royalty-Free Right) atas karya ilmiah saya yang berjudul :

Simulasi Crossdock menggunakan software Arena

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (database), merawat dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Béthune – France

Pada tanggal: Juni 2011

Yang menyatakan,

(Faizal)

ABSTRAK

Nama : Faizal

Program Studi : Teknik Industri

Judul : Simulasi Crossdock menggunakan software Arena Versi 13.5

Crossdock adalah salah satu teknik logistik baru di dalam penanganan material yang mana produk yang datang di pintu inbound langsung di muat di pintu outbound dimana tidak ada penyimpanan kalaupun ada kurang dari satu hari bahkan kurang dari satu jam sehingga bisa menurunkan biaya inventori. Keuntungan yang terlihat jelas dari sistem crossdock adalah menurunkan dua komponen biaya yaitu penyimpanan dan pengambilan barang dari rak. Di dalam sistem crossdock barang datang dari beberapa pemasok di konsolidasikan di crossdock kemudian di distribusikan ke pelanggan. Skenario didalam model crossdock menggunakan simulasi Arena ini adalah terdapat tiga pintu inbound dan tiga pintu outbound dengan rata-rata waktu antar kedatangan satu truk per jam dengan waktu rata-rata pelayanan satu truck per jam. Model di jalankan selama dua belas jam dengan tiga puluh replikasi untuk mendapatakan tingkat kepercayaan > 95 %. Dan hasilnya adalah rata-rata kedatangan truk inbound adalah dua belas dengan tingkat kepercayaan 91 % sementara rata-rata truk yang keluar adalah sebelas dengan tingkat kepercayaan 90 %. Adapun untuk rata-rata tingkat kegunaan pintu inbound adalah 34.4% untuk pintu inbound satu, 30.3 % untuk pintu inbound dua dan 36.8% untuk pintu tiga. Sementara tingkat kegunaan dari forklift adalah 27.5 % untuk forklift satu, 17.8% untuk forklift dua, 19.6% untuk forklift tiga, 13.9% forklift empat, 13.7% untuk forklift lima, and 15.7% untuk forklift enam. Dan untuk waktu rata-rata pemindahan pallet adalah 1.13 menit untuk pallet satu, 1.19 menit untuk pallet 2 dan 1 menit untuk pallet tiga. Dari hasil simulasi diatas dapat disimpulkan bahwa sistem operasi crossdock harus dia rencanakan agar bisa mengoptmalkan penggunaan sumber daya yang di miliki.

Kata Kunci: crossdock, simulasi

ABSTRACT

Name : Faizal

Study Program : Industrial Engineering

Title : Crossdock Simulation with Arena

Crossdock is a new logistics technic in material handling which products directly shipments from inbound trucks to outbound trucks. Crossdock can reduce the cost of inventory with direct delivery product without storing. Some obvious advantages from crossdock First crossdock eliminates two cost- and labor-intensive functions: storage and order picking of a traditional warehouse, while still allowing it to serve receiving and shipping functions. The different between traditional warehouse and crossdock is in crossdock they eliminate storing activity, that's mean product came from plusieurs supplier consolidated inside crossdock and delivery directly to their destination. From our model we have three inbound door and three outbound door while inside the dock we have six forklift to transfer product from inbound door to outbound door. The interraival truck is EXPO (60) minutes and service time in this case time to discharge and load pallet from inbound dock and to outbound dock TRIA (30, 60, 90). Arena simulation is one of the discrete event simulation with the objective is mesure average waiting line of the queue, these mesure change only when entity enters and leaves the system. Our simulation run seven hundred twenty minutes with thirty replication and the result is average truck inbound enter the system 12 with halfwidth 9% and truck out 11 with halfwidth 10%. The average utilization inbound docks is 34.4%, 30.3 % and 36.8%. utilization of forklfit 27.5% for foklift 1, 17.8% for forklift 2, 19.6% for forklift 3, 13.9% forklift 4, 13.7% forklift 5, and 15.7% forklift 6. From the view of simulation we can conclue that The crossdock operation is need to be planned for to optimizing our resources utilization.

Keyword: crossdock, simulation

UNIVERSITAS INDONESIA

UNIVERSITÉ D'ARTOIS

Simulasi Crossdock menggunakan software Arena

TESIS

Diajukan sebagai salah satu syarat untuk memperoleh gelar Magister Teknik

FAIZAL

NPM.0906578560

FAKULTAS TEKNIK

PROGRAM STUDI TEKNIK INDUSTRI

DEPOK

JULI 2011

ACKNOWLEDGEMENTS

First and foremost I would like to thank to **Gilles Goncalvez** for his kindness to allow me studying Logistics and for the support in giving us book *Simulation and modelling with Arena*, from that book we can add our knowldge in Arena simulation.

I would like to extend to **Hamid Allaoui** with your assistance your advice and your motivation we can finish this internship with the subject Crossdock simulation with Arena .

I would like to acknowledge to **Daniel Jolly** due to using your lab and Arena software (licence) for our study.

I would like thank to **M. Catherine Couturier** Responsable Pedagogics CMI who allow me to choce subject (Logistics)

My gratitude is also extended to my wife Apriliana, with him I can keep strong to read, study, until finish

Last but not least I would like to Thank to my Friends Indonesien, and friends Doctorat, sorry for disturbing you as long as our internship

Table of Content

Chapter. 1 Introduction

The Internship is an integral part of the master in Industrial Enginnering Program in Faculte Science Applique (FSA) Universite Artois . This work term should make us valuable insight into professional and industry-oriented side of logistics in practice. To fulfill the academic requirements of the internship we are required to submit an internship report as well as presentation. And our subject is Cross dock with Arena Simulation

Started at 04 April 2011 until 04 July 2011 we have to working 08h per day , 5 day / weeks in Laboratoire Genie Informatique et Automatique de l'Artois (LGI2A).

Purpose of this study was to find out how crossdock operational working from view of simulation and then how to input it in Simulation Arena version 13.5.

1.1 Report Scheme

Chapter 1 . Introduction

In this chapter we describe about our program as a master 2 pro that at semester 2 in the second year we have to do internship about three month. In this chapter too we describe our reseach goal

Chapter 2. Literature review

In this chapter we describe what is crossdock. In order to gain a deep understanding on crossdock we make two question What is Crossdock and Why Crossdock. In this chapter we also do literature study of what is simulation, systems, modeling and discrete event simulation. And we also try to explore what is advantages or strengths of the Arena software version 13.5.

Chapter 3. Model Approach and Assumption

In this chapter we describe how the process flow of our model in order to make more easier in building our model after. And in this chapter we also provides information on the assumption that we use in our model. For example, inter arrival time trucks, number of pallet will discharge, length of our croosdock facility and speed of our forklift.

Chapter 4. the results and analysis

In this chapter we analyze the results of the model that we have made. To get an optimal performace for example minimum total waiting time in the product Inbound and outbound door door we use "Optquest for Arena"

Chapter 5. Conclusion

In this chapter we conclude about the utility of this simulation for various industrial fields.

Chapter 6. Reference and Appendix

1.2. Reseach Methodology

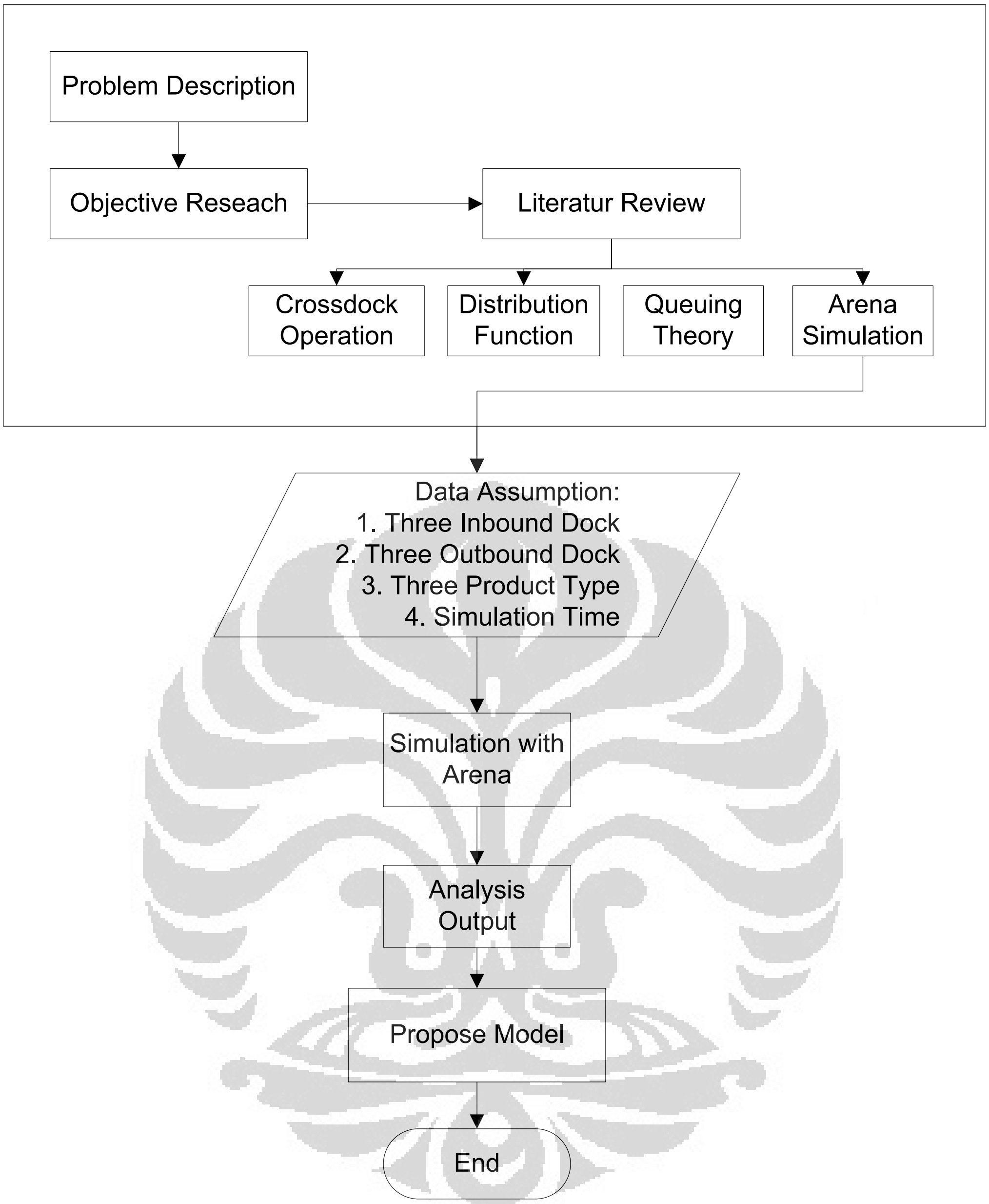


Figure 1.2.1 Reseach Methodology

Chapter. 2 Literatur Review

2.1 Cross dock as part of the logistics

Entering the competitive market nowdays makes the logistics industry seeking strategy to eliminate non value added activity to reduce cost and increse profit. As a consequence costs must be lowered throughout the chain by driving out unnecessary costs and focusing attention on adding value(Haag, S *et.al*, 2006)[2]. Keah Choon Tan, (2000) in his framework of supply chain management said the objective in supply chain management and logistics is to reduced cycle time , inventory and increase customer satisfaction with providing visibility and consolidating distribution centers. Where warehouses is including inside (Tayful Altiok, benjamin melamed, 2007).

Warehouses are an essential component of any supply chain (Jinxiang Gu *et al*. 2007). Becouse it plays a very important role in all transfers of products from the place of origin to ultimate users (Danuta Kisperska-Moron, 1999). Traditional warehouse activities includes receiving, storage, order picking, and shipping. From the whole activities order picking is the most labour-intensive operation in warehouses as almost 55 % of the total warehouse operating expense and the second is storage 20%, receiving 15%, and the rest is shipping operation (Vijay Sangam, 2010, Rene de koster *et.al*, 2006). Crossdocking have a chance to cut almost a half of activity in traditional warehouse.

Crossdock is a new logistics technic in material handling which products directly shipments from inbound trucks to outbound trucks. In the other word Crossdock is optimization in warehousing system. Crossdock can reduce the cost of inventory with direct delivery without storing. The objective of daily cross-dock’s operation plan is to assign the incoming trailers to the inbound door positions and the outgoing trailers to the outbound doors position, in order to minimise handling i.e., the total distance travelled by the freight itself or by the forklifts. (yuval cohen and baruch keren, 2009).

Crossdocking has some obvious advantages. First crossdocking eliminates two cost- and labor-intensive functions: storage and order picking of a traditional warehouse, while still allowing it to serve receiving and shipping functions (Zhengping Li *et.*al, 2007). Handling costs is reduced because it minimizes “the number of touches”. Second, the speed of material movement is faster, cargo normally takes a few days and even months in traditional ware- housing and it only normally takes less than 24 hours in a crossdock. Third, inventory is much lower and the throughput is higher in crossdocking compared to traditional warehousing. In addition, when timing is well coordinated, products can be made available in shorter time windows, thus reducing cycle times. So crossdocking's impact to supply chain is to make it more responsive to customer demand as compared to traditional warehousing.

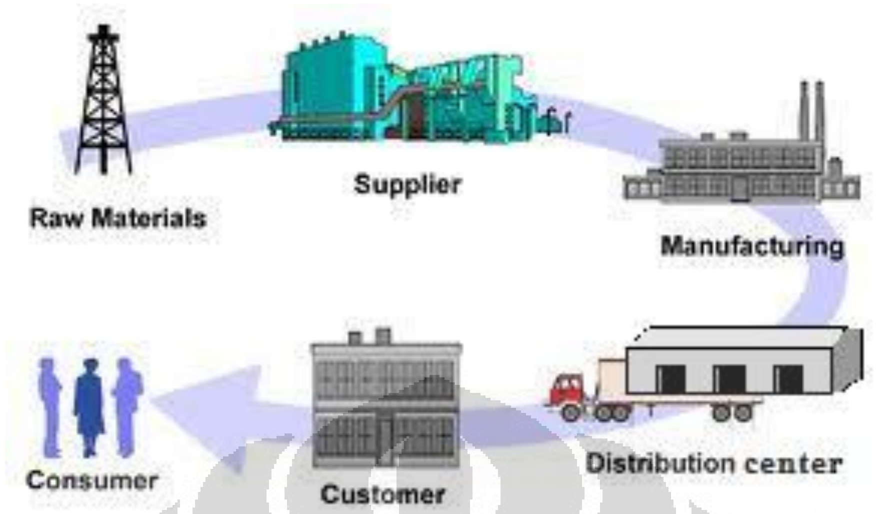


Figure 2.1 Cross dock in Supply Chain System

2.2 The Advantage of Cross dock

(Yan Liu, Soemon Takakuwa 2010) The primary objective of crossdocking is to **eliminate storage**, excessive handling, and lead time while minimizing transportation and storage costs and maintaining a high level of customer service.

Crossdocking's impact to supply chain is to make it more **responsive** to customer demand (Zhengping Li *et.al* 2007). Crossdocking has some obvious advantages. First crossdocking eliminates two cost- and labor-intensive functions: storage and order picking of a traditional warehouse, while still allowing it to serve receiving and shipping functions. Second, the speed of material movement is faster becouse products arrive in Inbound dock has their destination. Third, inventory is much lower and the throughput is higher in crossdocking compared to traditional warehousing. In addition, when timing is well coordinated, products can be made available in shorter time windows, thus reducing cycle times.

Gue and Kang 2001, Crossdock is a way to **reduce inventory holding cost** and transportation cost.

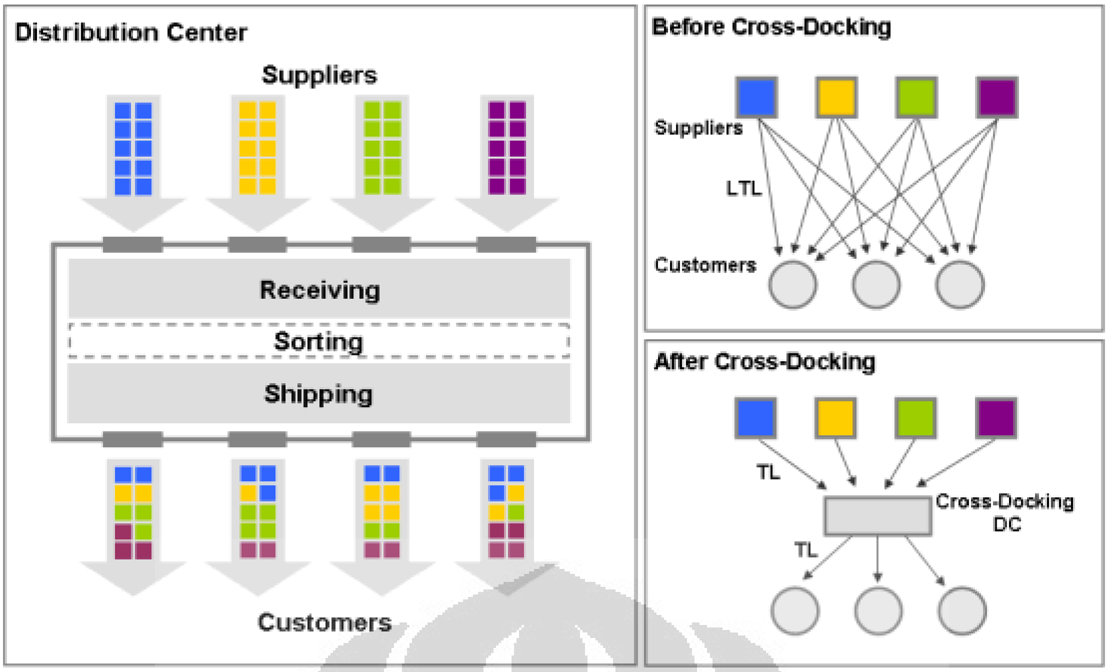


Figure 2.2 Distribution before and after Crossdock

2.4 Queueing Theory

The objective of queuing analysis is to offer reasonably satsifactory service to waiting customers (hamdi taha 2007) which can then used to design the service installation and minimize cost due to waiting time and increase service facility. the point important from the queuing theory is the Poisson and Exponential, becouse it helps identify the situation where queueing applied in our situation.

The poisson process have been used to model external arrivals to a variety systems, where arriving entity make “ independent arrival decisions.” For example truck arrive is not “ coordinated “ with discharge pallet in Inbound door, and truck arrivals are usually independent of each other (tayfur altiok 2007). Meaning we cannot predict when truck will arrive. The poisson pdf is defined as

$$P\{x=k\}=\frac{\lambda^k e^{-\lambda}}{k!}, k=0,1,2,\ldots$$

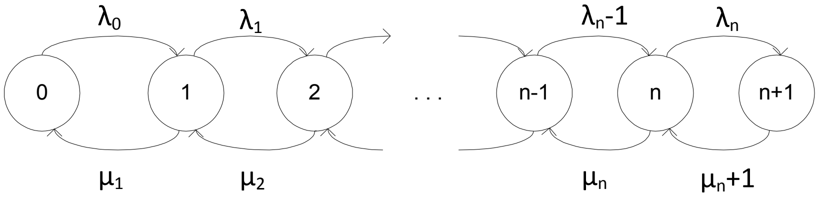


Figure 2.4.1 Poisson queues transition diagram

Where

λ = Average arrival rate
μ = Average service time

There are three components in the system queue is :

1. The Arrival of the population to be served
2. Queues
3. Service facilities

The arrival of entity is represented the interarrival time and the service describe by the service time per entity.

The queue discipline, which represent the order in which entity are selected from queue, is important factor in the analysis of queuing models. The following table is given in queuing system characteristic

Queue characteristic	Symbol	meaning
Interarrival time or service time	D	Deterministic
	M	Exponential
	Oak	Erlang distribution
	G	Other distribution
Queuing Dicipline	FIFO	First In First Out
	LIFO	Last in First Out
	Siro	Service in Random Order
	PRI	The Order of Priority

Table 2.4.1 Queuing System Characteristic

2.4.1 Multiple Server Model

in our model, there are *c* parallel servers. The Arrival rate is λ and the service rate per server is μ.

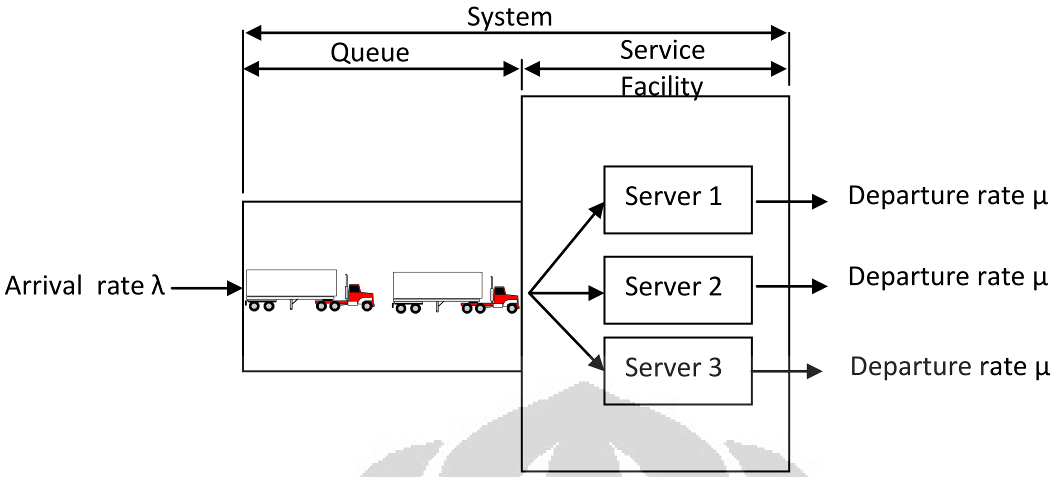


Figure 2.4.2 Schematic of a queuing system with single queue parallel servers M/M/c

The effect of using c (server) parallel is proportionate increase in the facility service rate. The arrival rate is λ and the service rate is μ. Becouse there is no limit on the number in the system, λ_{eff} =λ. Are thus defined

$$\lambda_n = \lambda, n \geq o$$

$$\mu_n = \begin{cases} \dfrac{n\mu, n < c}{c\mu, n \geq c} \end{cases}$$

Thus,

$$\rho_n = \begin{Bmatrix} \dfrac{\lambda^n}{\mu(2\mu)(3\mu)...(n\mu)}\rho 0 = \dfrac{\lambda^n}{n!\mu^n}\rho 0 = \dfrac{\rho^n}{n!}\rho 0, n < c \\ \dfrac{\lambda^n}{\left(\coprod_{i=1}^{c} i\mu\right)(c\mu)^{n-c}}\rho 0 = \dfrac{\lambda^n}{c!c^{n-c}\mu^n}\rho 0 = \dfrac{\rho^n}{c!c^{n-c}}\rho 0, n \geq c \end{Bmatrix}$$

Letting ρ = $\frac{\lambda}{\mu}$, and assuming $\frac{\rho}{c} < 1$, the value of ρ_0 is determined from $\sum \frac{\infty}{n = 0 \, pn = 1}$

Which gives,

$$\rho 0 = \left\{ \sum_{n=0}^{c-1} \frac{\rho^n}{n!} + \frac{\rho^c}{c!} \sum_{n=c}^{\infty} \left(\frac{\rho}{c} \right)^{n-c} \right\}^{-1}$$

$$= \left\{ \sum_{n=0}^{c-1} \frac{\rho^n}{n!} + \frac{\rho^c}{c!} \left(\frac{1}{1 - \frac{p}{c}} \right) \right\}^{-1} , \frac{\rho}{c} < 1$$

The expression for L_q can be determined as follows :

$$Lq = \sum_{n=c}^{\infty} (n-c) \rho_n$$

$$= \sum_{k=0}^{\infty} k p_{k+c}$$

$$= \sum_{k=0}^{\infty} k \frac{\rho^{k+c}}{c^k c!} \rho 0$$

$$= \frac{\rho^{c+1}}{c! c} \rho 0 \sum_{k=0}^{\infty} k \left(\frac{\rho}{c} \right)^{k-1}$$

$$= \frac{\rho^{c+1}}{c! c} \rho 0 \frac{d}{d\left(\frac{\rho}{c} \right)} \sum_{k=0}^{\infty} \left(\frac{\rho}{c} \right)^k$$

$$= \frac{\rho^{c+1}}{(c-1)!(c-\rho)^2} \rho 0$$

Becouse $\lambda_{eff} = \lambda$, $L_s = L_q + \rho$. Thus values of W_s and W_q can be determined by dividing L_s and L_q by λ (Hamdy Taha, operation reseach p.579, 2007)

In our model the interarrival truck is λ = 1 truck / hour and service time is Triangular distribution (30,60,90) / 3 so our average service rate (μ) 1 truck / hour, and we have three c (server) (inbound door facility) and result is :

Notation

λ = Average arrival rate
µ = Average speed of service
P_0 = probability of a service facility is busy or the facility utilization factor
Ls = The average amount in the system (in the queue and being served)
Lq = The Average queue length
Ws = Average time in system
Wq = Average waiting time in queue

Random interarrival and service times are described quantitatively in queueing models by **exponential distribution**, which is defined as (hamdy taha operation reseach p.549, 2007)

$$P(t \leq T) = 1 - e^{-\lambda t}, t > 0$$

Where

λ = Average Arrival in time t

t = time occurance

2.4 Simulation

Simulation refers to a broad collection of methods and applications to mimic the behavior a real system, usually on a computer with appropriate software (Kelton *et al*, 2007). Tayfur Altiok and Benjamin Melamed, 2007 say that Simulation modeling refers to Systems and Models.

System is combination of components that act together to perform a function not possible with any of the individual parts (IEEE).

Model is the enterprise of devising a simplified representation of a complex system with the goal of providing predictions of the system's performance measure of interest (Tayfur altiok, Benjamin Melamed 2007). Simply a model is designed to capture certain behavioral aspects of the modeled system (Morris 1967) in order to gain knowledge and insight into the system's behavior.

2.4.1 type of simulation (hamdy taha, operation reseach, 2007 p. 607)

1. Continous model deals with systems whose behaviour changes *continously* with time. A typical example the study of world population dynamics
2. Discrete models delas with the study waiting lines, with the objective of determining such mesures as the average waiting time and length of the queue. These mesure

change only when a entity enters or leaves the system. The instants at which changes take place occur at specific discrete points in time (arrival and departure events), givin rise to the name **discrete event simulation**.

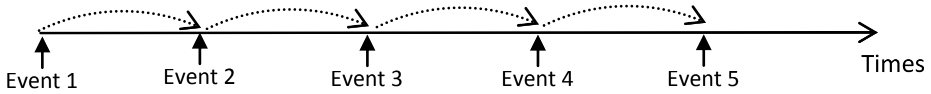


Figure 2.3 Example of occurence of simulation events on the time scale

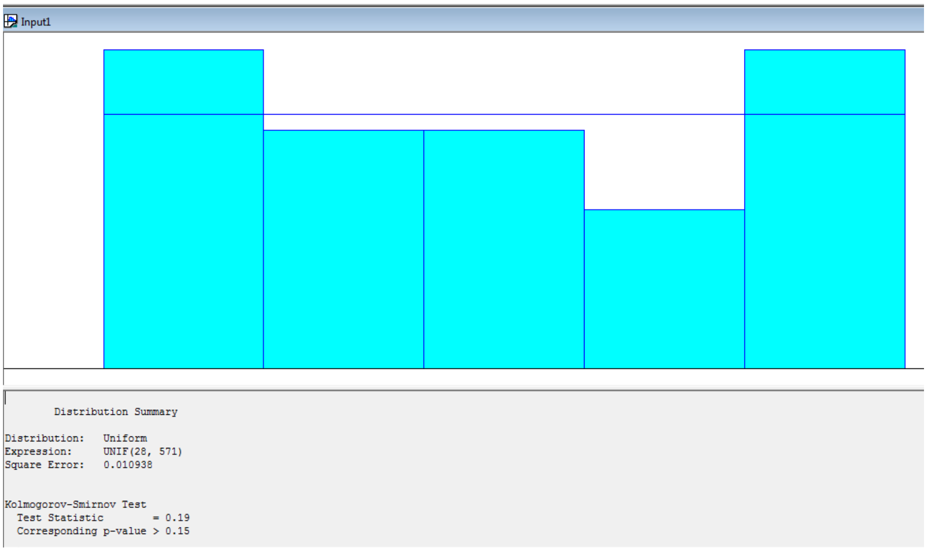
2.5 Arena®

Arena is discrete event simulation produced by Rockwell Software. The software uses graphical user interface that allows modelers to place modules in the workspace to represent differents events or actifities through which entities or objects of the model move and interact. The modules used to create the model used in this reseach include. The engine of Arena® is SIMAN language. Having a simulation language as its engine makes Arena models run extremely fast and makes it possible to model any complex process, such as production process, inventory management or transportation problem. Arena reputation is twenty four (24) year leading the simulation industry, taught in most universities that offer industrial Engineering or systems Engineering. Each year, more than thirty tahousan (30,000) students who have studied Arena for their garduated. (www.arenasimulation.com/user).

Couple of features in the Arena® Simulation.

2.5.2 Input Analyzer

The Input Analyzer is provided as a standard component of the Arena environment. This powerful and versatile tool can be used to determine the quality of fit of probability distribution functions to input data. It may also be used to fit specific distribution functions to a data file to allow you to compare distribution functions or to display the effects of changes in parameters for the same distribution. In addition, the Input Analyzer can generate sets of random data that can then be analyzed using the software’s distribution-fitting features.To run the *Input Analyzer*, double-click on the *Input Analyzer* icon or select the Input Analyzer command from the Tools menu in Arena.



```
        Distribution Summary

Distribution:   Uniform
Expression:     UNIF(28, 571)
Square Error:   0.010938

Kolmogorov-Smirnov Test
  Test Statistic         = 0.19
  Corresponding p-value > 0.15
```

Figure 2.2 screen Input Analyzer for Uniform Distribution

2.5.2 Output Analyzer

The Output Analyzer component of Arena provides an easy-to-use interface that simplifies data analysis and allows you to view and analyze your data quickly and easily.

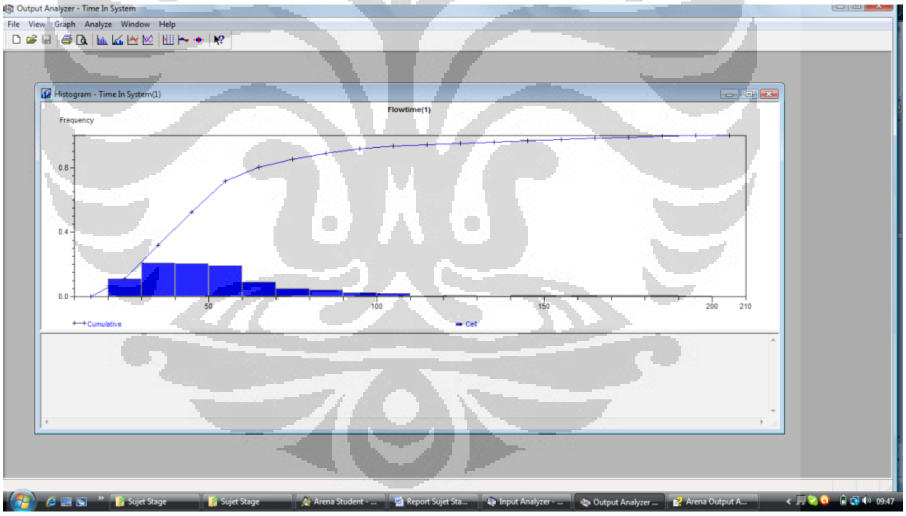


Figure 2.3 Screen Output Analyzer Histogram

2.3.3 OptQuest for Arena

OptQuest's explorer interface displays the hierarchical structure of the components of an optimization problem. OptQuest enhances Arena by automating the search for an optimal strategy (OptQuest Module 2011).

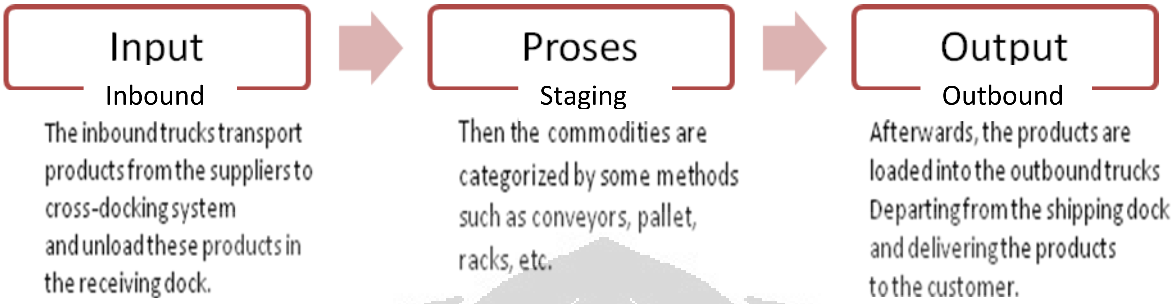
An optimization model in OptQuest for Arena has three major elements: Controls, Response, Constraints and an Objective. Controls are variables or resources in our model such as the number of machines of each type in a job shop. Responses can be used to create constraint expressions and objective expressions. Some condition before using OptQuest for Arena :

1. Model must have variable or resources . If the model has no variables or resources defined this message will appear “The Arena model has no controls defined. The model must have at least one control to run an optimization “. For example, a flowchart type model with only a Create, Delay and Dispose modules can not be optimized.
2. If the Decimal Symbol parameter defined for the machine is set to a comma instead of a period. To check this go to Start/Settings/Control Panel and click on the Regional and Language Options icon and under Regional Options click on the Customize button, there you will be able to define the Decimal Symbol to a period "." instead of a comma ",".

Chapter 3. Model approach and Assumptions

3.1 Model Approach

The crossdock operation is broken down into three major components : Inbound, Staging, Outbound. Retailer submit their order to suppliers. Following are some of the assumptions that are made related to these components.



The inbound trucks transport products from the suppliers to cross-docking system and unload these products in the receiving dock.

Then the commodities are categorized by some methods such as conveyors, pallet, racks, etc.

Afterwards, the products are loaded into the outbound trucks Departing from the shipping dock and delivering the products to the customer.

Figure 3.1 Crossdock system

Inbound

The inbound doors are assigned to fixed locations there is three Inbound door . The trucks are assumed to be docked and ready for unloading. The interarrival trucks EXPO (1) hour. The mix of freights is generated randomly UNIF (10,22) the average service time (discharge pallet) Triangular distribution with mean 1 truck per hour. Each Supplier have one single product to to discharge to the crossdock and then consolidates their orders and load it to truckloads with plusieur product. There workers transfer products to outbound for delivery to individual stores, so that outgoing trucks contain plusieur products to customer. The destination and quantity of product is assigned in staging area. Which means that a given products is assigned to the fix outdoor. One forklift is allocated to one Inbound truck with activity load pallet needs time 0.5 minutes and unload pallet needs time 0.5 minutes , and distance from Inbound to Outbound Area is 75 ft.

Staging

Each destination trailer door has independent staging space of its own. Items are staged on the floor in a single layer and have resources to scanning every pallets came before delivered to outbound door. No racks are used for storage. Medium size of forklift with a weight capacity 5000 lb and velocity 75 ft / minutes with or without freight. One forklift is allocated to one pair of inbound and outbound doors. This practice avoids more than one forklift working on the same trailer resulting in congestion in front of that door. In staging prosess time spent average EXPO (0.5) minutes.

Outbound

The outbound trailer door locations are fixed. An equal outbound demand scenario has been assumed (Taylor and Noble, 2004), where all outbound locations have equal freight flows. The dimensions of the trailer considered are 7.6 ft width, 7.6 ft height and 20 ft depth. Trucks will bring plusieur product that already assigned in staging area.

3.2 Assumption

This model ilustrates cross dock operation, using the notion of station, entity routing among stations, entity moved from inbound to outbound trucks by Forklift, and the control of forklifts movements using distance from inbound door to outbound door and speed's forklift, at which trucks arrive and unload UNIF (10,22) pallets. Facility has three inbound door and three outbound door that have equal probability.

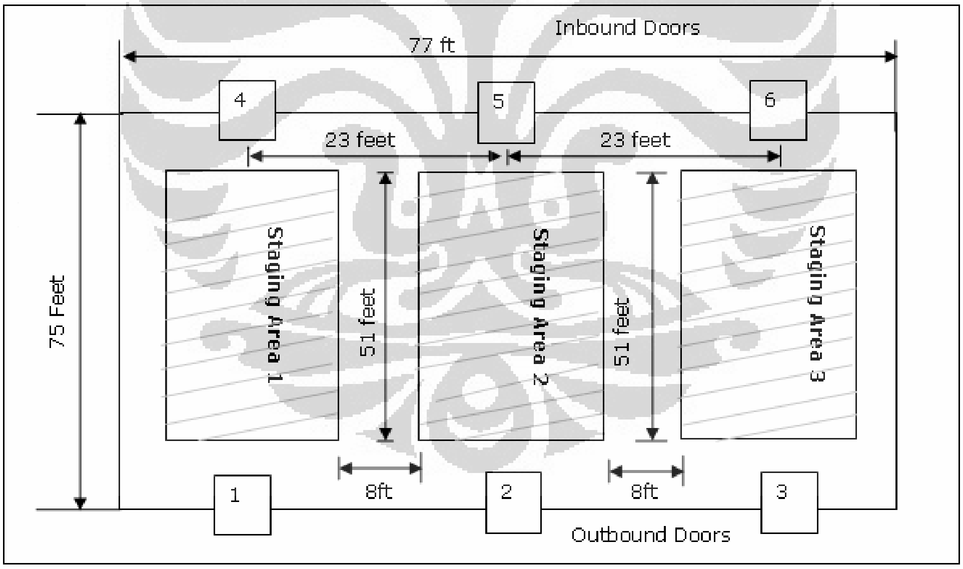


Figure 3.1 Example crossdock dimension (Sumit Nadal, 2005)

3.3 Schema Flow Process

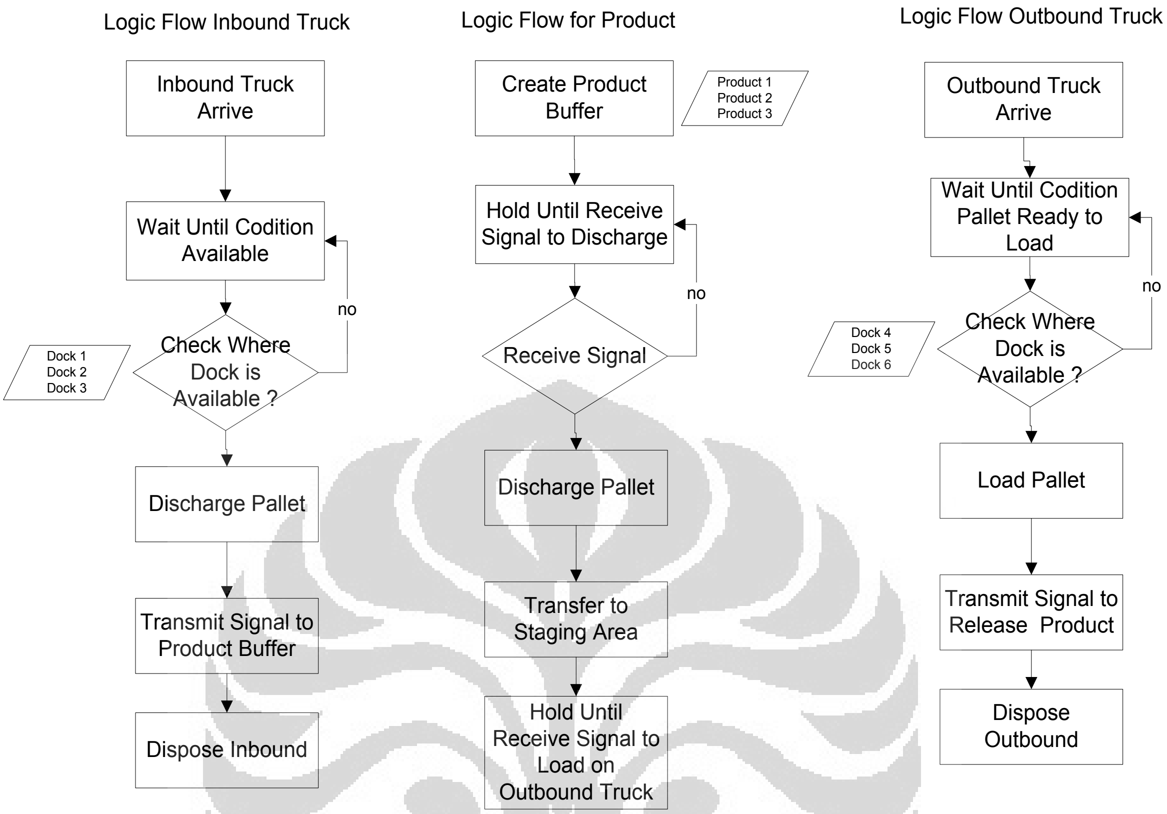
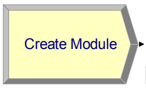

Logic Flow Inbound Truck
Logic Flow for Product
Logic Flow Outbound Truck
Inbound Truck Arrive
Wait Until Codition Available
no
Check Where Dock is Available ?
Dock 1
Dock 2
Dock 3
Discharge Pallet
Transmit Signal to Product Buffer
Dispose Inbound
Create Product Buffer
Product 1
Product 2
Product 3
Hold Until Receive Signal to Discharge
no
Receive Signal
Discharge Pallet
Transfer to Staging Area
Hold Until Receive Signal to Load on Outbound Truck
Outbound Truck Arrive
Wait Until Codition Pallet Ready to Load
no
Dock 4
Dock 5
Dock 6
Check Where Dock is Available ?
Load Pallet
Transmit Signal to Release Product
Dispose Outbound

Figure 3.3. Flow process cross dock operation

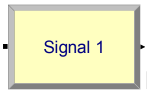
3.4.4 Function of module used

The crossdock model in Arena is built with the following modules as they fall under different components of the crossdock operation :



(Basic Process)

Function : Used as a source to generate new entities, and release them into the model. The entities (Trucks) are created in the **create module**. The time between arrival UNIF (30,60) minutes and connected with connector to signal module

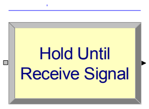


(Advance Process)

Function : Used to send a signal to each hold module in the model where the value of the type parameter is *wait for signal*, in order to release the maximum specified number of entities. **The Signal modules** represents the give the signal when truck arrive it will unload pallet UNIF(10,22)

(Basic Process)

Function : Used as the exit point of entities from a simulation model. Entities arrive at The **Dispose module** are disposed of and removed from the model.

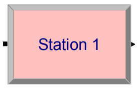


(Advance Process)

Function : Used to hold an entity in a queue to either wait for a signal, wait for a specified condition to become true. The Hold module is waiting signal from signal module to unload pallets with rule Uniform distribution it is defined by two parameters Minimum and maximum (10, 22).

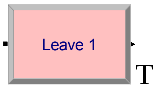
Thus far, entity transfers were implemented via Arena connections (for instantenous transfer) and *Delay* modules (for time-lapse transfer). The *Advance Transfer* template panel provides additional mechanisms of time-lapse transfer of entities among sets of modules or geografhic locations. This section briefly reviews the facilities provided by this template panel.

The *Advanced Transfer* template panel implements a worldview in which entities are transported among *Station* modules. The simplest transfer mechanism uses *Route* modules as dispatch points and *Station* modules as destination points. Additionally, the *Enter* and *Leave* modules may be used to transfer entities into and out of physical or logical locations.



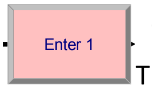
(Advance Transfer)

Fuction : The **Station module** The Station module defines a station (or a set of stations) corresponding to a physical or logical location where processing occurs. If the Station module defines a station set, it is effectively defining multiple processing locations.



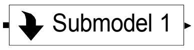
(Advance Transfer)

The **Leave module** The Leave module is used to transfer an entity to a station or module. An entity may be transferred in one of two ways: it can be transferred to a module that defines a station by referencing the station and routing, conveying, or transporting to that station or a graphical connection can be used to transfer an entity to another module.



(Advance Transfer)

The **Enter module** The Enter module defines a station (or a set of stations) corresponding to a physical or logical location where processing occurs. If the Enter module defines a station set, it is effectively defining multiple processing locations.The station (or each station within the defined set) has a matching Activity Area that is used to report all times and costs accrued by the entities in this station. This Activity Area’s name is the same as the station. If a parent activity area is defined, then it also accrues any times and costs by the entities in this station.When an entity arrives at an Enter module, an "unloading" delay may occur and any transfer device used to transfer the entity to the Enter module’s station may be released.



The **Sub Model** The use of submodels in your model not only increases the amount of workspace you have in which to build your model, but it also allows you the ability to better organize your model. Each submodel is represented in its own view, allowing you to partition visually a complex model flowchart into natural, easy-to-manipulate windows. Submodels themselves can contain deeper submodels; there is no limit to the amount of nesting that can occur.

3.3. Building Model Crossdock in Arena® - Modules

The basic from building this model in Arena models are modules. These are the objects that helps define process to be simulated (Kelton, et al.,2004). A schematic representation of the layout of our Cross dock facility is depicted in figure 3.2.

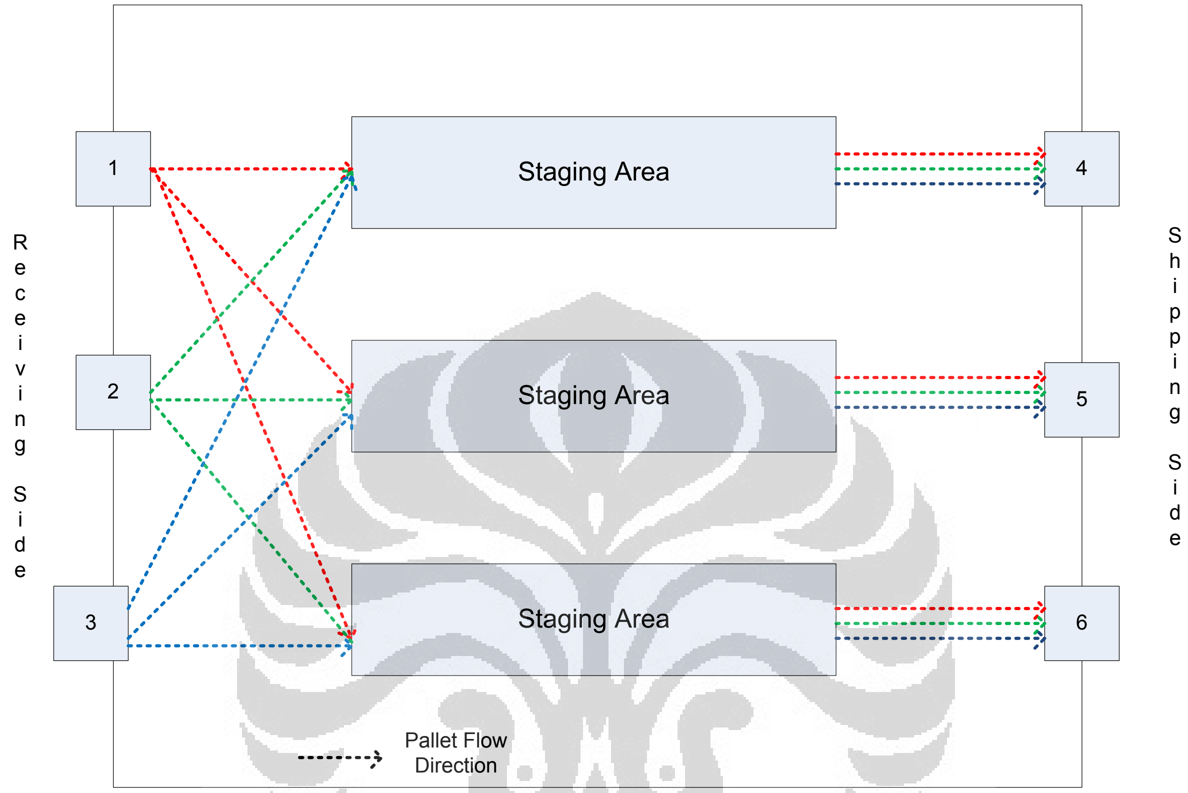


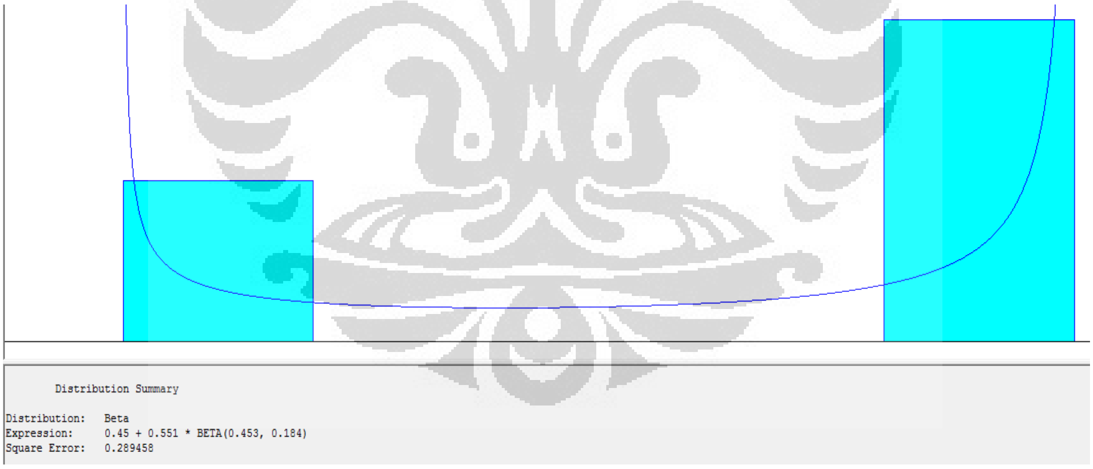
Figure 3.2 Pallet Flow Diagram

3.4 Step by Step in making Crossdock simulation with Arena® Simulation

1. Make observations at the real system, the required equipment:
   a. Stop watch
   b. Paper + stationery
2. Record hours of the arrival of the truck one by one into the system during the hours of operation crossdock
3. Record the time process in the Dock truck during hours of operation

Apply using the distribution fitting *Input Analyzer* for example

08h00-10h00	2	This means that in one hour there is 1 truck
10h00-12h00	2	This means that in one hour there is 1 truck
12h00-14h00	1	This means that in one hour there are 0.5 trucks
14h00-16h00	2	This means that in one hour there is 1 truck
16h00-18h00	2	This means that in one hour there is 1 truck
18h00-20h00	1	This means that in one hour there are 0.5 trucks
Total	10	λ per hour = 1,1,0.5,1,1,0.5



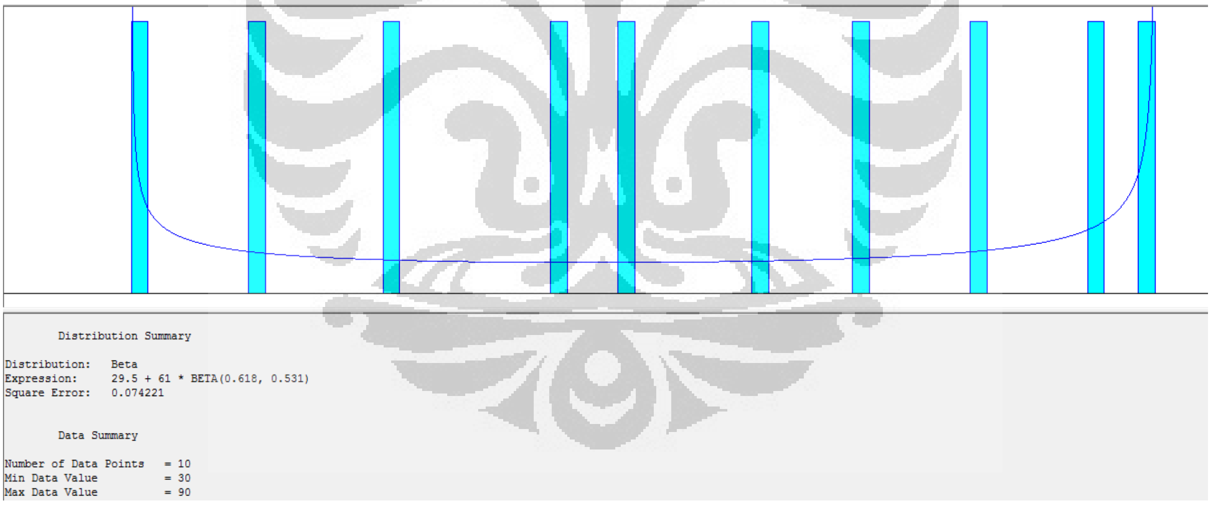
```
     Distribution Summary

Distribution:   Beta
Expression:     0.45 + 0.551 * BETA(0.453, 0.184)
Square Error:   0.289458
```

```
Distribution: Beta
Expression:   0:45 + 0551 * BETA (0453, 0184)
Square Error: 0.289458 → interrarrival rate λ
```

Now do the recording time of service, for example

Truck 1	59''	This means that in 1 hour its service rate 1.01 truck
Truck 2	73''	This means that in 1 hour its service rate 1.22 truck
Truck 3	80''	This means that in 1 hour its service rate 1.33 truck
Truck 4	37''	This means that in 1 hour its service rate 1.62 truck
Truck 5	30''	This means that in 1 hour its service rate 2 truck
Truck 6	90''	This means that in 1 hour its service rate 0.66 truck
Truck 7	45''	This means that in 1 hour its service rate 1.33 truck
Truck 8	55''	This means that in 1 hour its service rate 1.09 truck
Truck 9	87''	This means that in 1 hour its service rate 0.69 truck
Truck 10	67''	This means that in 1 hour its service rate 0.89 truck
Total	623''	μ per hour = 1.01, 1.22, 1.33, 1.62, 2, 0.66, 1.33, 1.09, 0.69, 0.89

59, 73, 80, 37, 30, 90, 45, 55, 87, 67

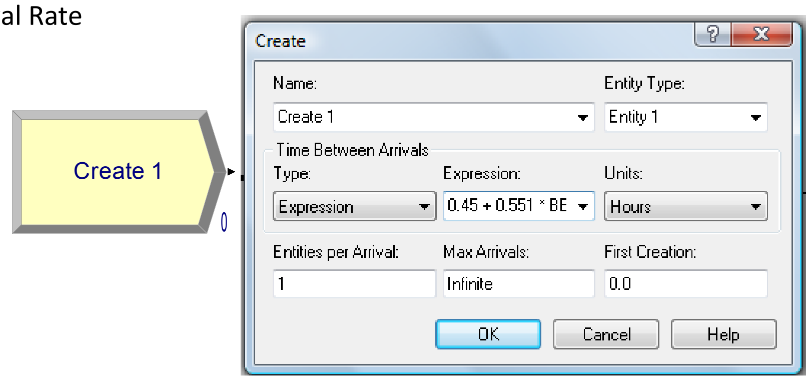


```
        Distribution Summary

Distribution:   Beta
Expression:     29.5 + 61 * BETA(0.618, 0.531)
Square Error:   0.074221

        Data Summary

Number of Data Points   = 10
Min Data Value          = 30
Max Data Value          = 90
```

```
Distribution: Beta
Expression: 29.5 + 61 * BETA (0618, 0531) minutes
Square Error: 0.074221 → μ or delay time (processing time)
```

Furthermore, entering the data into the Arena simulation

Our Interrarrival Rate



Name: Create 1

Entity Type: Entity 1

Time Between Arrivals

Type: Expression

Expression: 0.45 + 0.551 * BE

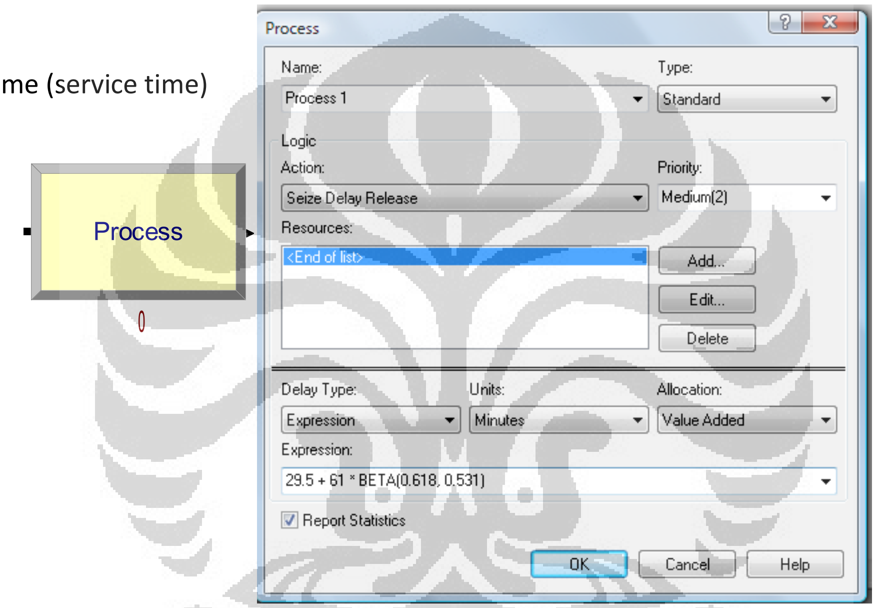
Units: Hours

Entities per Arrival: 1

Max Arrivals: Infinite

First Creation: 0.0

Our Delay Time (service time)



Name: Process 1

Type: Standard

Logic

Action: Seize Delay Release

Priority: Medium(2)

Resources: <End of list>

Delay Type: Expression

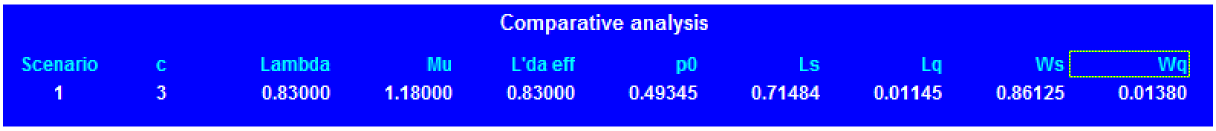
Units: Minutes

Allocation: Value Added

Expression: 29.5 + 61 * BETA(0.618, 0.531)

☑ Report Statistics

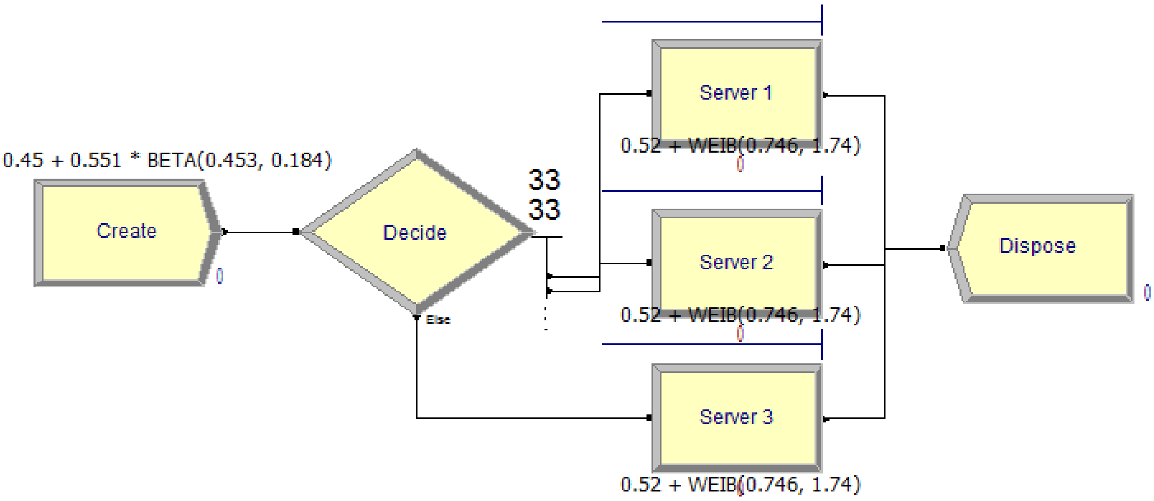
After that we can find the probability of utilization, queue number, time in queue and the total time the entity with the help of software ready to use TORA. With her input the data interarrival rate per hour (1 +1 +0.5 +1 +1 +0.5 / 6) = 0,833 / hour and the service rate (1.01 +1.22 +1.33 +1.62 +2 +0.66 +1.33 +1.09 +0.69 +0.89 / 10) = 1.18 / hour



Comparative analysis

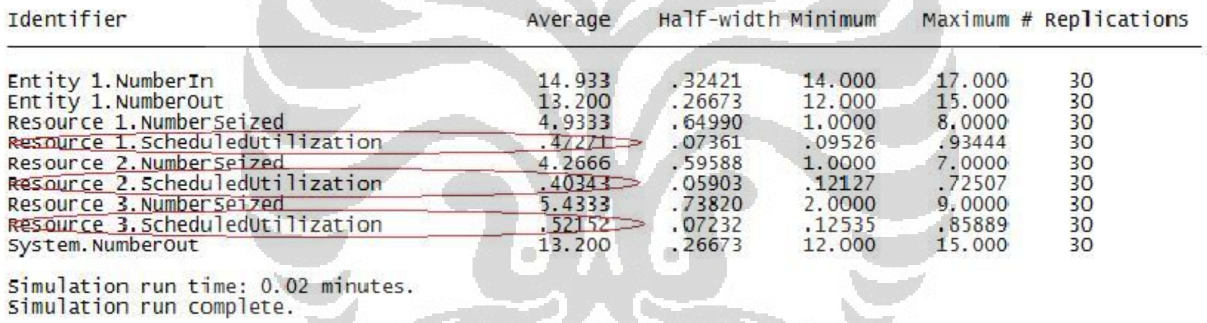
Scenario	c	Lambda	Mu	L'da eff	p0	Ls	Lq	Ws	Wq
1	3	0.83000	1.18000	0.83000	0.49345	0.71484	0.01145	0.86125	0.01380

These results are as a comparison before running the simulation. and this way allows us to validate the model because we had already knew the results (estimates) of our system before the system is actually run.

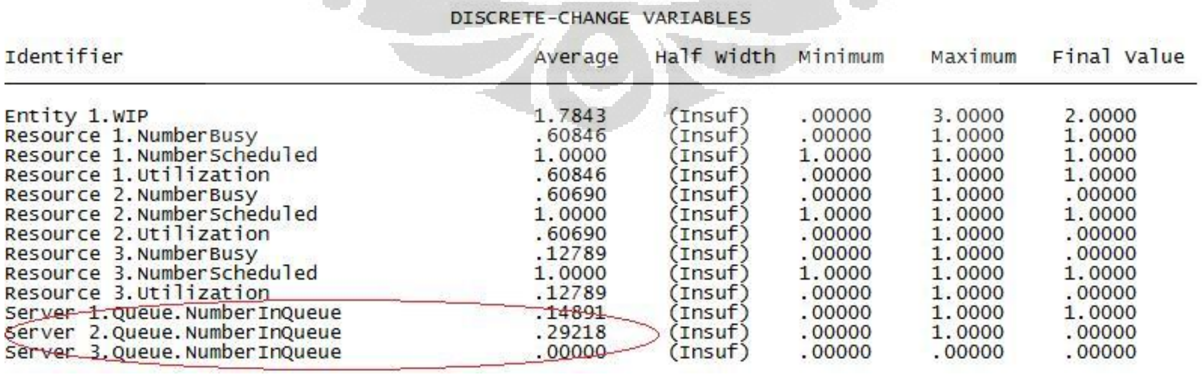


Run

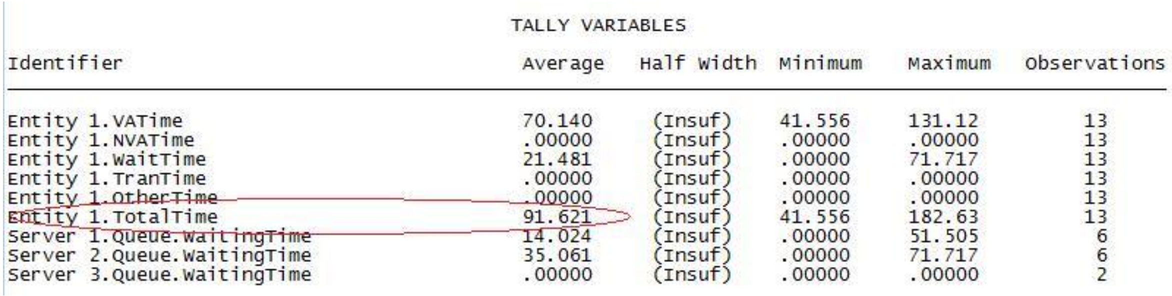
Our simulation result



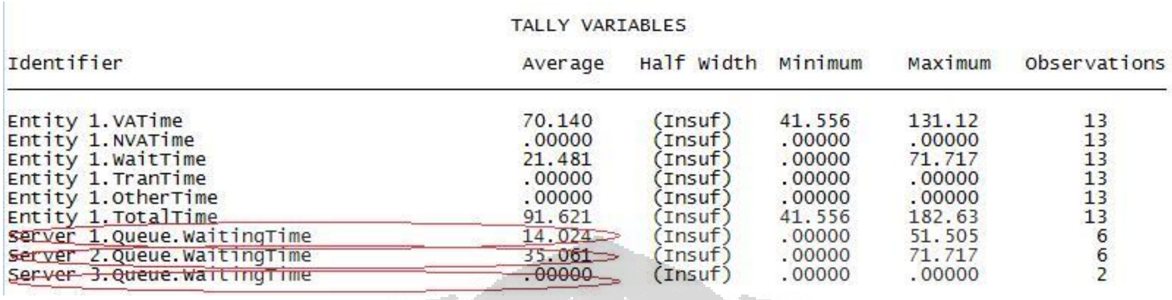
```
Identifier                                  Average    Half-width Minimum    Maximum # Replications

Entity 1.NumberIn                           14.933     .32421     14.000     17.000     30
Entity 1.NumberOut                          13.200     .26673     12.000     15.000     30
Resource 1.NumberSeized                     4.9333     .64990     1.0000     8.0000     30
Resource 1.ScheduledUtilization             .47271     .07361     .09526     .93444     30
Resource 2.NumberSeized                     4.2666     .59588     1.0000     7.0000     30
Resource 2.ScheduledUtilization             .40343     .05903     .12127     .72507     30
Resource 3.NumberSeized                     5.4333     .73820     2.0000     9.0000     30
Resource 3.ScheduledUtilization             .52152     .07232     .12535     .85889     30
System.NumberOut                            13.200     .26673     12.000     15.000     30

Simulation run time: 0.02 minutes.
Simulation run complete.
```



```
                                 DISCRETE-CHANGE VARIABLES

Identifier                                  Average    Half Width  Minimum    Maximum    Final Value

Entity 1.WIP                                1.7843     (Insuf)     .00000     3.0000     2.0000
Resource 1.NumberBusy                       .60846     (Insuf)     .00000     1.0000     1.0000
Resource 1.NumberScheduled                  1.0000     (Insuf)     1.0000     1.0000     1.0000
Resource 1.Utilization                      .60846     (Insuf)     .00000     1.0000     1.0000
Resource 2.NumberBusy                       .60690     (Insuf)     .00000     1.0000     .00000
Resource 2.NumberScheduled                  1.0000     (Insuf)     1.0000     1.0000     1.0000
Resource 2.Utilization                      .60690     (Insuf)     .00000     1.0000     .00000
Resource 3.NumberBusy                       .12789     (Insuf)     .00000     1.0000     .00000
Resource 3.NumberScheduled                  1.0000     (Insuf)     1.0000     1.0000     1.0000
Resource 3.Utilization                      .12789     (Insuf)     .00000     1.0000     .00000
Server 1.Queue.NumberInQueue                .14891     (Insuf)     .00000     1.0000     1.0000
Server 2.Queue.NumberInQueue                .29218     (Insuf)     .00000     1.0000     .00000
Server 3.Queue.NumberInQueue                .00000     (Insuf)     .00000     .00000     .00000
```



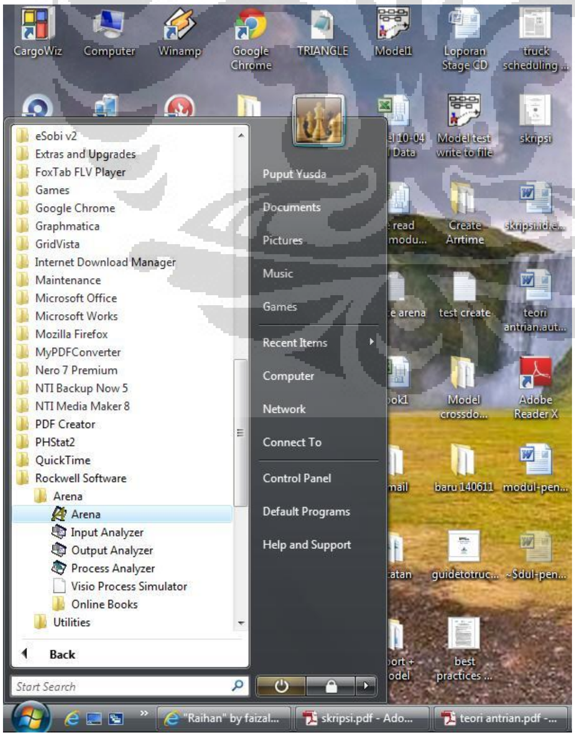
```
                                        TALLY VARIABLES

Identifier                          Average    Half Width   Minimum    Maximum    Observations
_____________________________________________________________________________________________

Entity 1.VATime                     70.140      (Insuf)     41.556     131.12          13
Entity 1.NVATime                    .00000      (Insuf)     .00000     .00000          13
Entity 1.WaitTime                   21.481      (Insuf)     .00000     71.717          13
Entity 1.TranTime                   .00000      (Insuf)     .00000     .00000          13
Entity 1.OtherTime                  .00000      (Insuf)     .00000     .00000          13
Entity 1.TotalTime                  91.621      (Insuf)     41.556     182.63          13
Server 1.Queue.WaitingTime          14.024      (Insuf)     .00000     51.505           6
Server 2.Queue.WaitingTime          35.061      (Insuf)     .00000     71.717           6
Server 3.Queue.WaitingTime          .00000      (Insuf)     .00000     .00000           2
```



```
                                        TALLY VARIABLES

Identifier                          Average    Half Width   Minimum    Maximum    Observations
_____________________________________________________________________________________________

Entity 1.VATime                     70.140      (Insuf)     41.556     131.12          13
Entity 1.NVATime                    .00000      (Insuf)     .00000     .00000          13
Entity 1.WaitTime                   21.481      (Insuf)     .00000     71.717          13
Entity 1.TranTime                   .00000      (Insuf)     .00000     .00000          13
Entity 1.OtherTime                  .00000      (Insuf)     .00000     .00000          13
Entity 1.TotalTime                  91.621      (Insuf)     41.556     182.63          13
Server 1.Queue.WaitingTime          14.024      (Insuf)     .00000     51.505           6
Server 2.Queue.WaitingTime          35.061      (Insuf)     .00000     71.717           6
Server 3.Queue.WaitingTime          .00000      (Insuf)     .00000     .00000           2
```

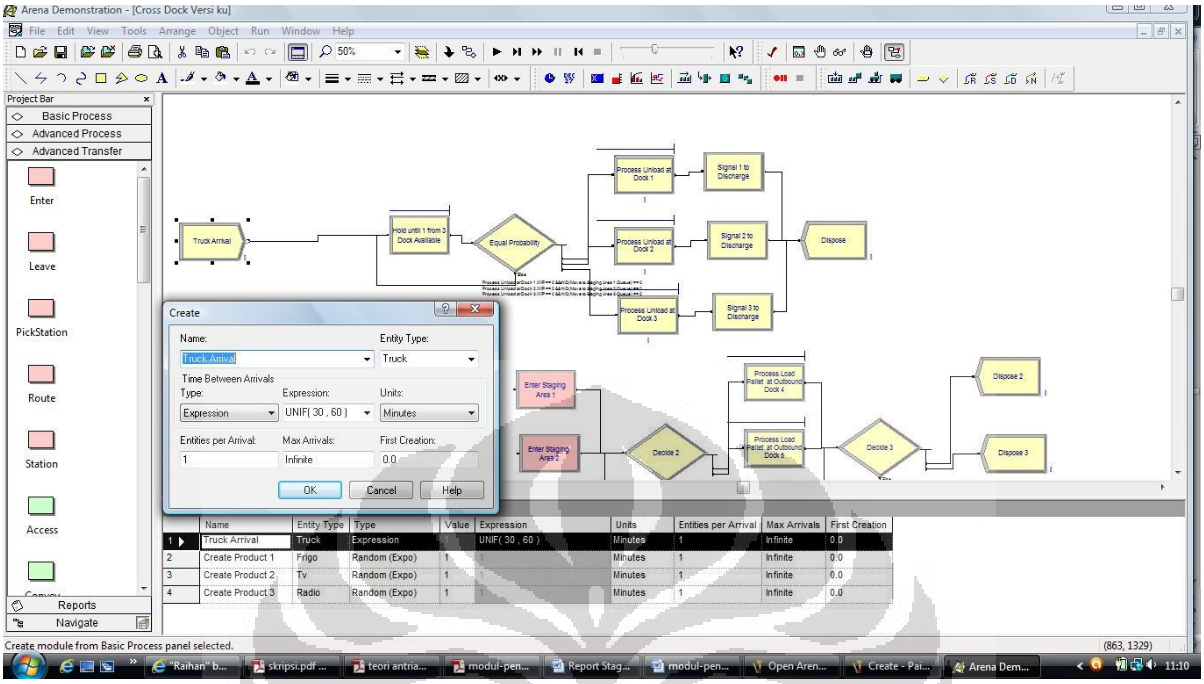
Since we already have a guidance we can modified our model

1. Open Arena Program



3.4.1 *Crete model* Logic for truck Arrival

1. Firstly we *create* truck arrival and defined the time between arrival, for knowing the distribution we can use Input Analyzer, we will discuse about it after this chapter.



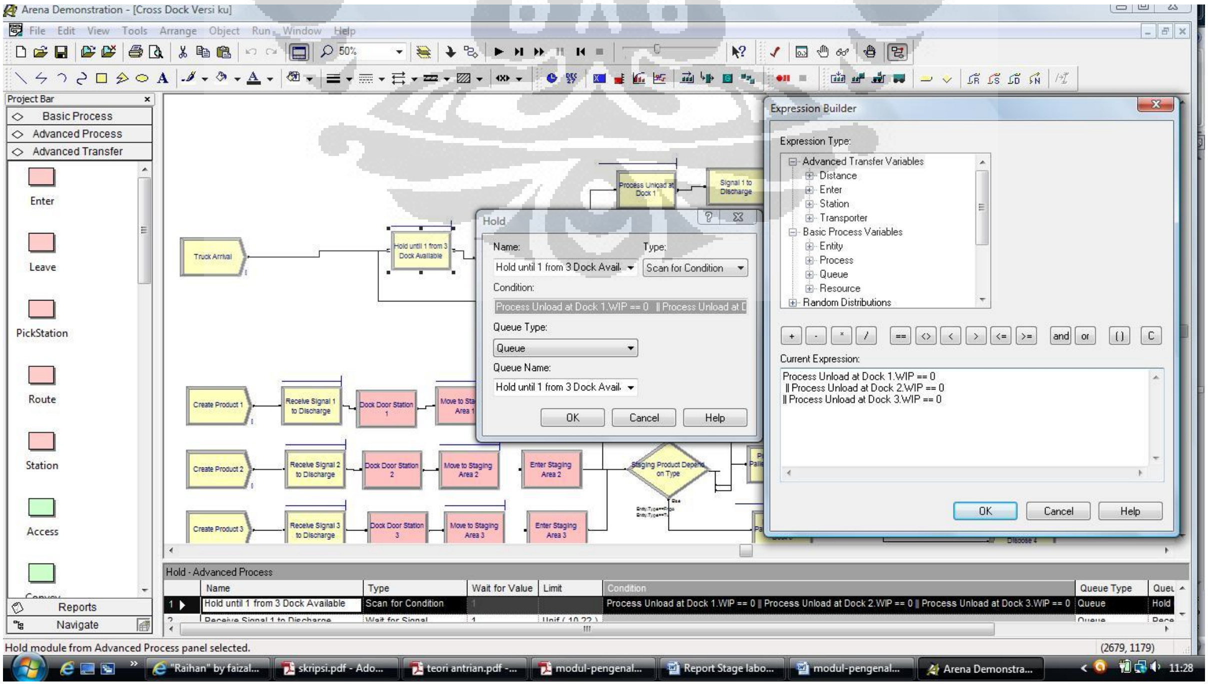
Create

Name: Truck Arrival
Entity Type: Truck
Time Between Arrivals
Type: Expression
Expression: UNIF(30 , 60)
Units: Minutes
Entities per Arrival: 1
Max Arrivals: Infinite
First Creation: 0.0

	Name	Entity Type	Type	Value	Expression	Units	Entities per Arrival	Max Arrivals	First Creation
1	Truck Arrival	Truck	Expression	1	UNIF(30 , 60)	Minutes	1	Infinite	0.0
2	Create Product 1	Frigo	Random (Expo)	1	1	Minutes	1	Infinite	0.0
3	Create Product 2	Tv	Random (Expo)	1	1	Minutes	1	Infinite	0.0
4	Create Product 3	Radio	Random (Expo)	1	1	Minutes	1	Infinite	0.0

Create module from Basic Process panel selected.

1. *Hold Module* with scan for condition this module has fonction to scan the dock door and transporter available condition true if `Process Unload at Dock 1.WIP == 0 || Process Unload at Dock 2.WIP == 0 || Process Unload at Dock 3.WIP == 0`



Hold

Name: Hold until 1 from 3 Dock Avail.
Type: Scan for Condition
Condition: Process Unload at Dock 1.WIP == 0 || Process Unload at D
Queue Type: Queue
Queue Name: Hold until 1 from 3 Dock Avail.

Expression Builder

Current Expression:
Process Unload at Dock 1.WIP == 0
|| Process Unload at Dock 2.WIP == 0
|| Process Unload at Dock 3.WIP == 0

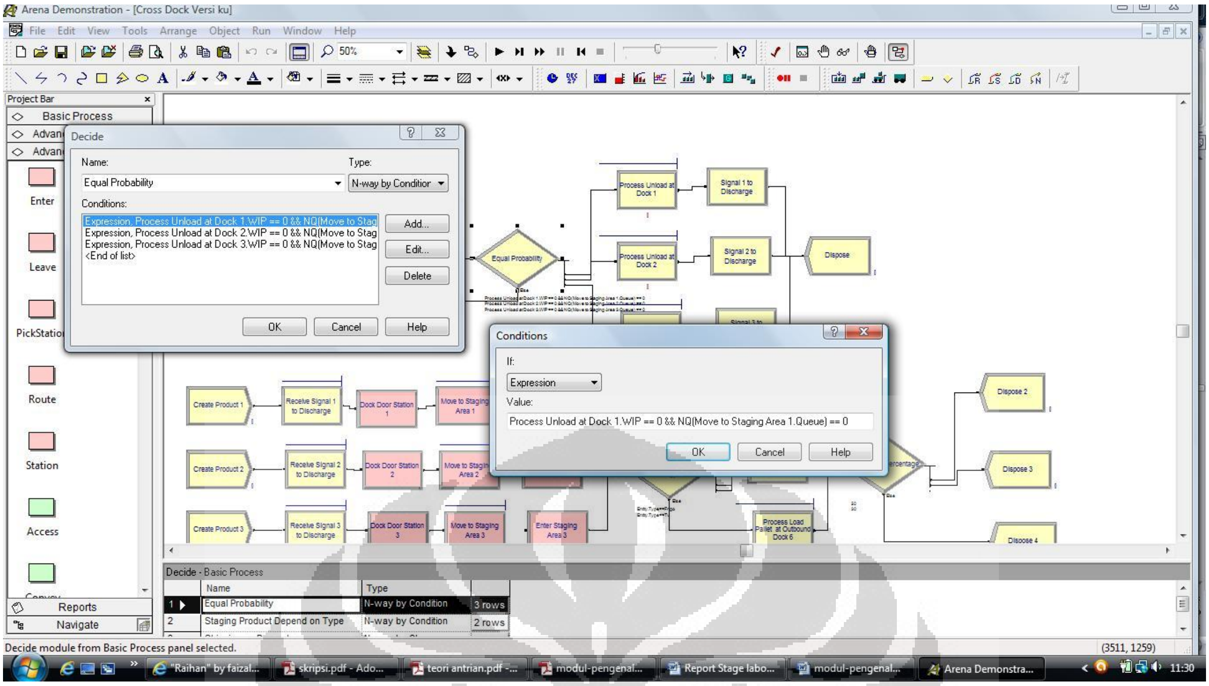
Hold - Advanced Process

	Name	Type	Wait for Value	Limit	Condition	Queue Type	Queu
1	Hold until 1 from 3 Dock Available	Scan for Condition	1		Process Unload at Dock 1.WIP == 0 || Process Unload at Dock 2.WIP == 0 || Process Unload at Dock 3.WIP == 0	Queue	Hold
2	Receive Signal 1 to Discharge	Wait for Signal	1	Unif (10,22)		Queue	Rece

Hold module from Advanced Process panel selected.

2. After that entity (truck) entering *decide module* in this module we choce *N-way by condition* and the condition is `Process Unload at Dock 1.WIP == 0 && NQ(Move to Staging`

`Area 1.Queue) == 0` if the condition oke entity will enter the module true if not it will search the secondly, thirdly



Arena Demonstration - [Cross Dock Versi ku]

Decide

Name: Equal Probability

Type: N-way by Condition

Conditions:

Expression, Process Unload at Dock 1.WIP == 0 && NQ(Move to Stag
Expression, Process Unload at Dock 2.WIP == 0 && NQ(Move to Stag
Expression, Process Unload at Dock 3.WIP == 0 && NQ(Move to Stag
<End of list>

Conditions

If: Expression

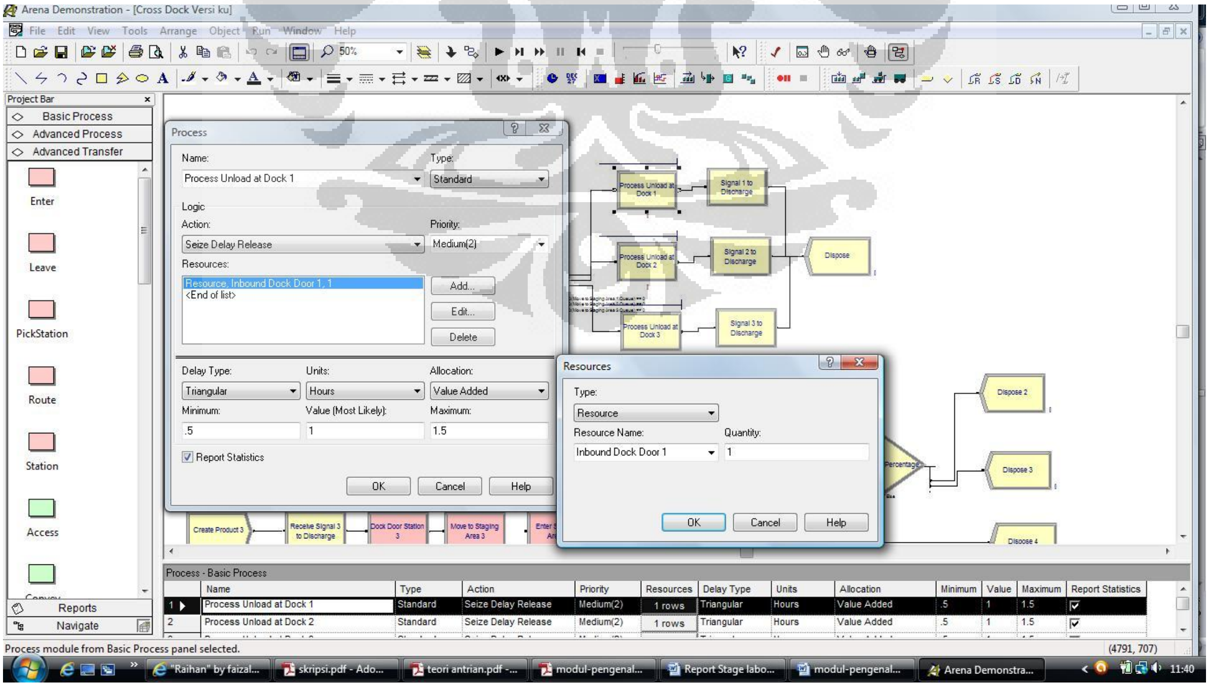
Value: Process Unload at Dock 1.WIP == 0 && NQ(Move to Staging Area 1.Queue) == 0

Decide - Basic Process

	Name	Type	
1	Equal Probability	N-way by Condition	3 rows
2	Staging Product Depend on Type	N-way by Condition	2 rows

Decide module from Basic Process panel selected.

3. Truck processing discharge pallet in *Process Module* in *Delay Type* we assign Triangular this data we analyze from *input analyzer*



Arena Demonstration - [Cross Dock Versi ku]

Process

Name: Process Unload at Dock 1

Type: Standard

Logic

Action: Seize Delay Release

Priority: Medium(2)

Resources:

Resource, Inbound Dock Door 1, 1
<End of list>

Delay Type: Triangular

Units: Hours

Allocation: Value Added

Minimum: .5

Value (Most Likely): 1

Maximum: 1.5

☑ Report Statistics

Resources

Type: Resource

Resource Name: Inbound Dock Door 1

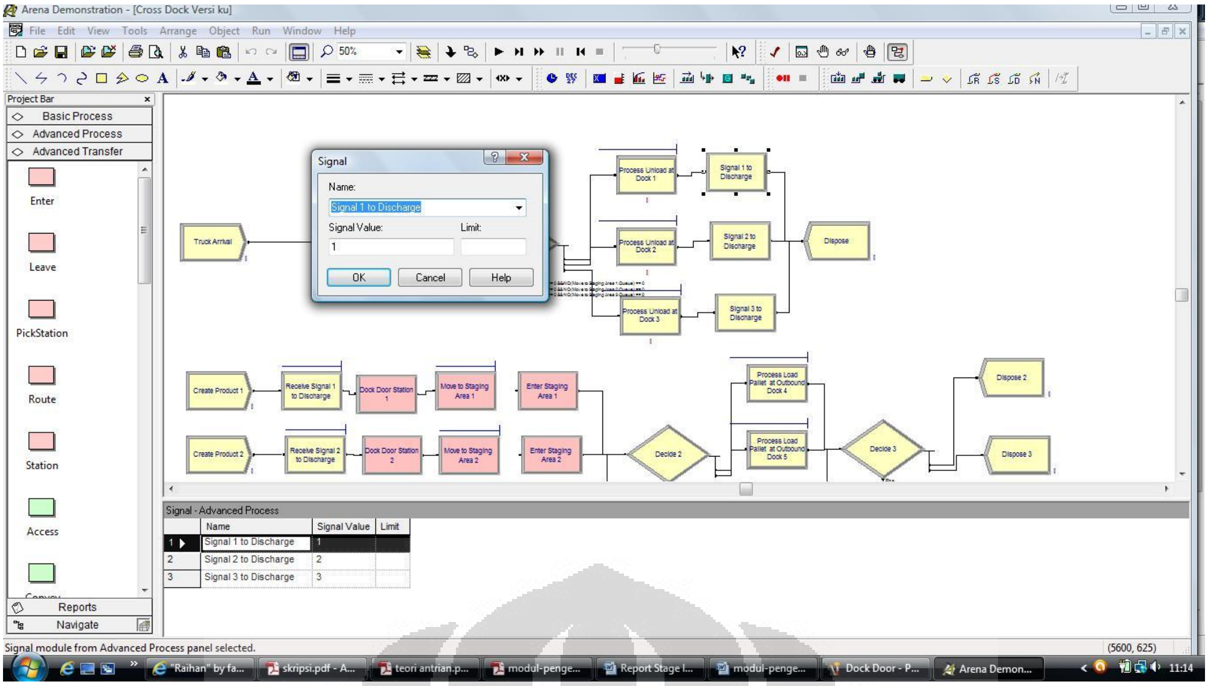
Quantity: 1

Process - Basic Process

	Name	Type	Action	Priority	Resources	Delay Type	Units	Allocation	Minimum	Value	Maximum	Report Statistics
1	Process Unload at Dock 1	Standard	Seize Delay Release	Medium(2)	1 rows	Triangular	Hours	Value Added	.5	1	1.5	☑
2	Process Unload at Dock 2	Standard	Seize Delay Release	Medium(2)	1 rows	Triangular	Hours	Value Added	.5	1	1.5	☑

Process module from Basic Process panel selected.

4. After proces discharging pallet truck entering *Signal Module* to Transmit signal to *Hold module* which holding product until receive signal from trucks. After give signal truck enter *Dispose Module*



Signal

Name: Signal 1 to Discharge

Signal Value: 1

Limit:

OK Cancel Help

Signal - Advanced Process

	Name	Signal Value	Limit
1	Signal 1 to Discharge	1	
2	Signal 2 to Discharge	2	
3	Signal 3 to Discharge	3	

Signal module from Advanced Process panel selected.

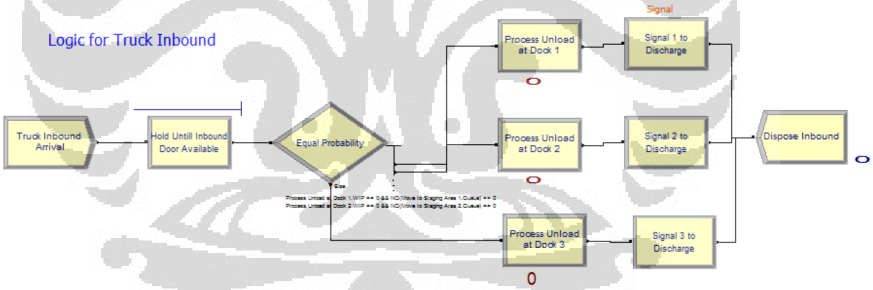
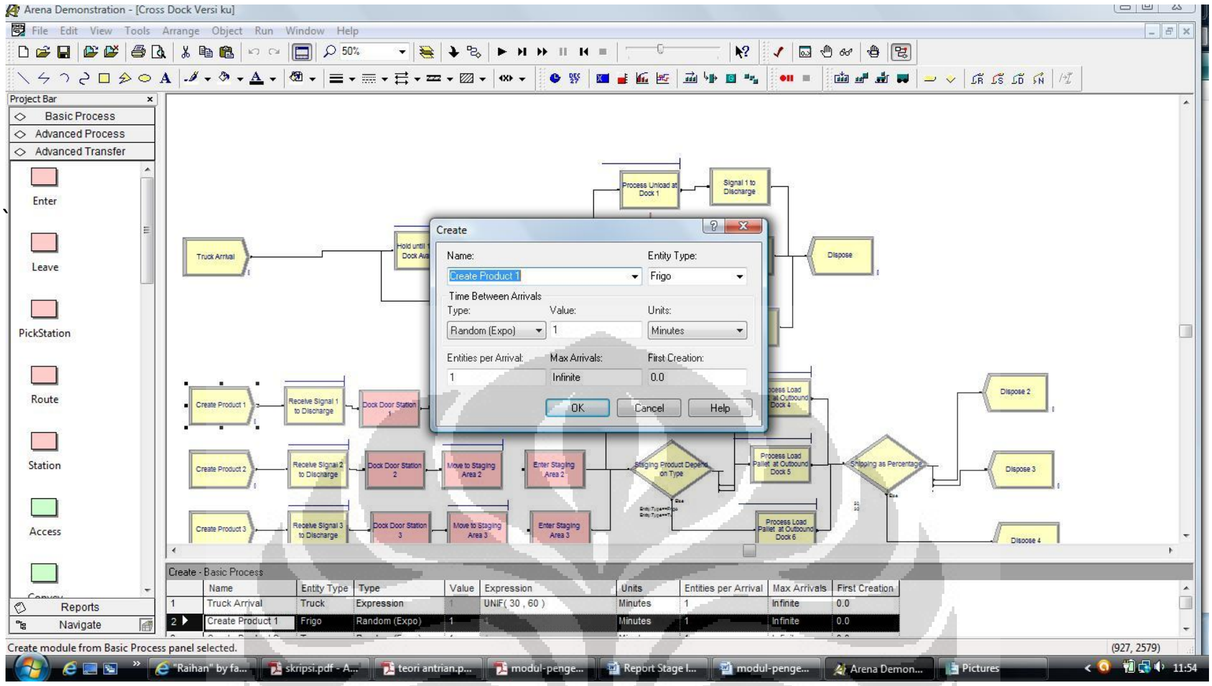


Figure 3.4.1 complete model Inbound Truck

3.4.2 Create Product to Discharge

1. Same way with create truck but in this process product created is assumed infinite but they Hold by *Hold Module* and will release if receive signal to release (we assigne limit UNIF 10,22 pallet)


Arena Demonstration - [Cross Dock Versi ku]
File Edit View Tools Arrange Object Run Window Help
Project Bar
Basic Process
Advanced Process
Advanced Transfer
Enter
Leave
PickStation
Route
Station
Access
Reports
Navigate
Truck Arrival
Process Unload at Dock 1
Signal 1 to Discharge
Dispose
Create Product 1
Receive Signal 1 to Discharge
Dock Door Station 1
Create Product 2
Receive Signal 2 to Discharge
Dock Door Station 2
Move to Staging Area 2
Enter Staging Area 2
Staging Product Depend on Type
Process Load Pallet at Outbound Dock 5
Shipping as Percentage
Dispose 2
Dispose 3
Create Product 3
Receive Signal 3 to Discharge
Dock Door Station 3
Move to Staging Area 3
Enter Staging Area 3
Process Load Pallet at Outbound Dock 6
Dispose 4
"Raihan" by fa... skripsi.pdf - A... teori antrian.p... modul-penge... Report Stage I... modul-penge... Arena Demon... Pictures 11:54

Create

Name: Create Product 1

Entity Type: Frigo

Time Between Arrivals

Type: Random (Expo)

Value: 1

Units: Minutes

Entities per Arrival: 1

Max Arrivals: Infinite

First Creation: 0.0

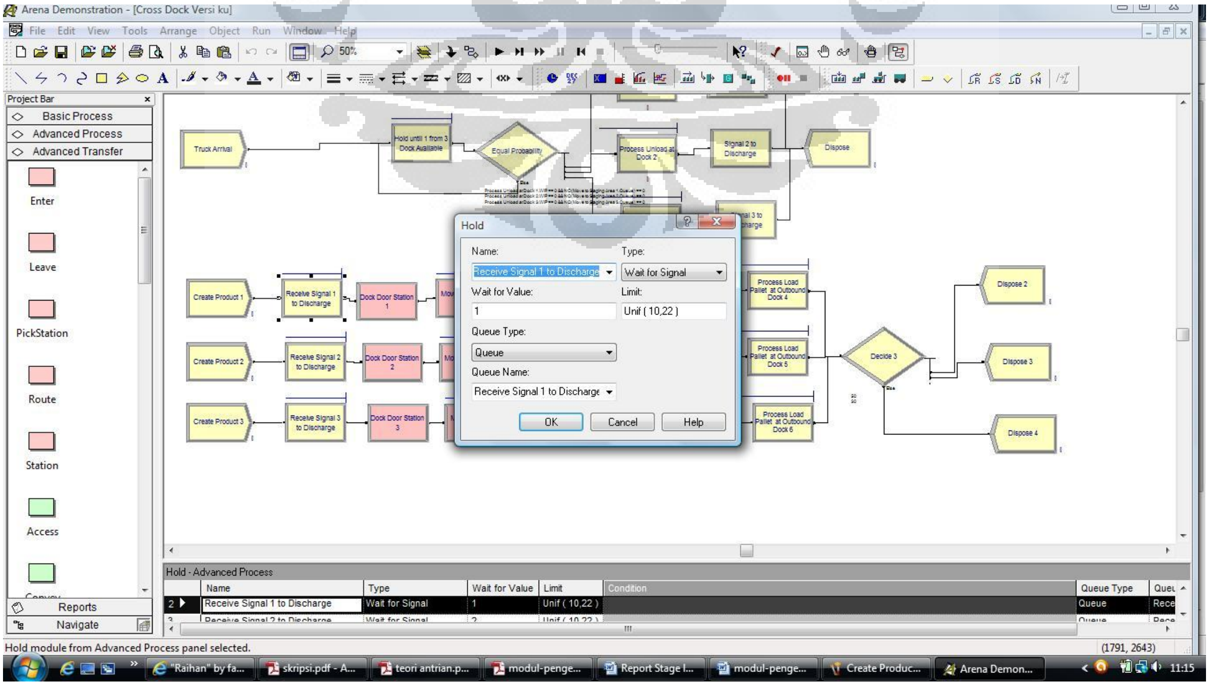
OK Cancel Help

Create - Basic Process

	Name	Entity Type	Type	Value	Expression	Units	Entities per Arrival	Max Arrivals	First Creation
1	Truck Arrival	Truck	Expression	1	UNIF(30 , 60)	Minutes	1	Infinite	0.0
2	Create Product 1	Frigo	Random (Expo)	1	1	Minutes	1	Infinite	0.0

Create module from Basic Process panel selected. (927, 2579)

2. Product *wait for signal* to release


Arena Demonstration - [Cross Dock Versi ku]
File Edit View Tools Arrange Object Run Window Help
Project Bar
Basic Process
Advanced Process
Advanced Transfer
Enter
Leave
PickStation
Route
Station
Access
Reports
Navigate
Truck Arrival
Hold until 1 from 3 Dock Available
Equal Probability
Process Unload at Dock 2
Signal 2 to Discharge
Dispose
Signal 3 to Discharge
Create Product 1
Receive Signal 1 to Discharge
Dock Door Station 1
Create Product 2
Receive Signal 2 to Discharge
Dock Door Station 2
Create Product 3
Receive Signal 3 to Discharge
Dock Door Station 3
Process Load Pallet at Outbound Dock 4
Process Load Pallet at Outbound Dock 5
Process Load Pallet at Outbound Dock 6
Decide 3
Dispose 2
Dispose 3
Dispose 4
"Raihan" by fa... skripsi.pdf - A... teori antrian.p... modul-penge... Report Stage I... modul-penge... Create Produc... Arena Demon... 11:15

Hold

Name: Receive Signal 1 to Discharge

Type: Wait for Signal

Wait for Value: 1

Limit: Unif (10,22)

Queue Type: Queue

Queue Name: Receive Signal 1 to Discharge

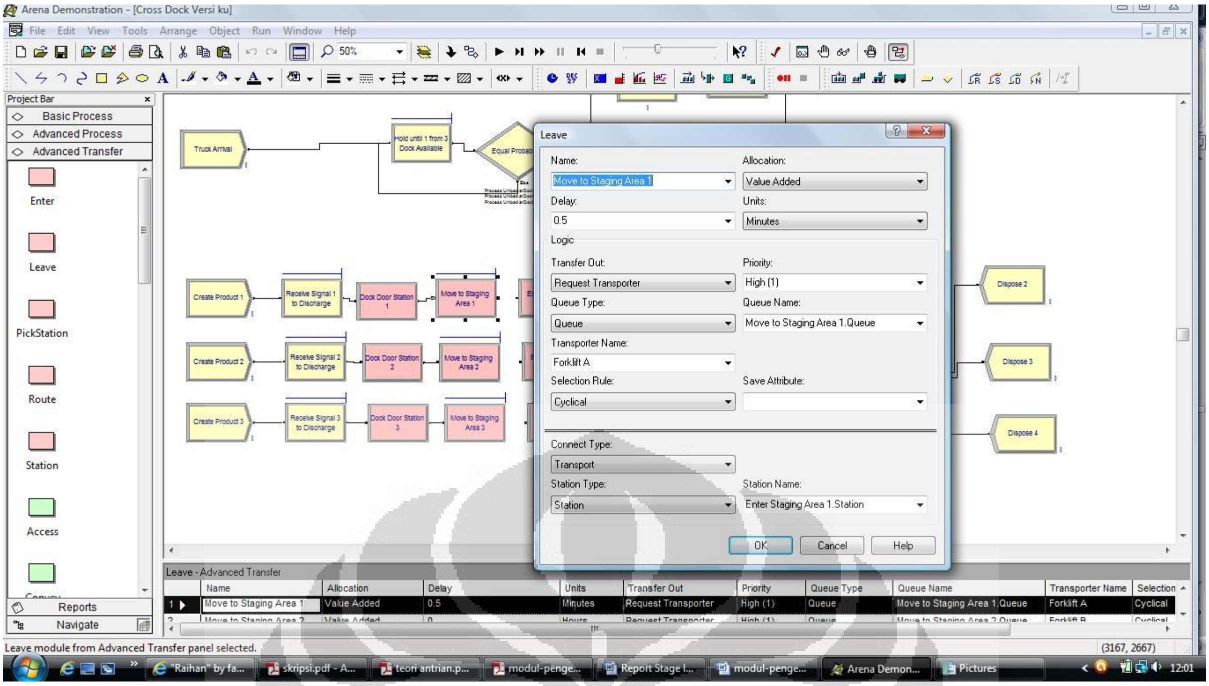
OK Cancel Help

Hold - Advanced Process

	Name	Type	Wait for Value	Limit	Condition	Queue Type	Queu
2	Receive Signal 1 to Discharge	Wait for Signal	1	Unif (10,22)		Queue	Rece

Hold module from Advanced Process panel selected. (1791, 2643)

3. In this module we simulate all product is already infront of dock door and ready to transfer to staging area. Forklift need 0.5 minutes to load and 0.5 minutes Unload pallet onto forklift



Leave

Name: Move to Staging Area 1
Allocation: Value Added
Delay: 0.5
Units: Minutes
Logic
Transfer Out: Request Transporter
Priority: High (1)
Queue Type: Queue
Queue Name: Move to Staging Area 1.Queue
Transporter Name: Forklift A
Selection Rule: Cyclical
Save Attribute:
Connect Type: Transport
Station Type: Station
Station Name: Enter Staging Area 1.Station

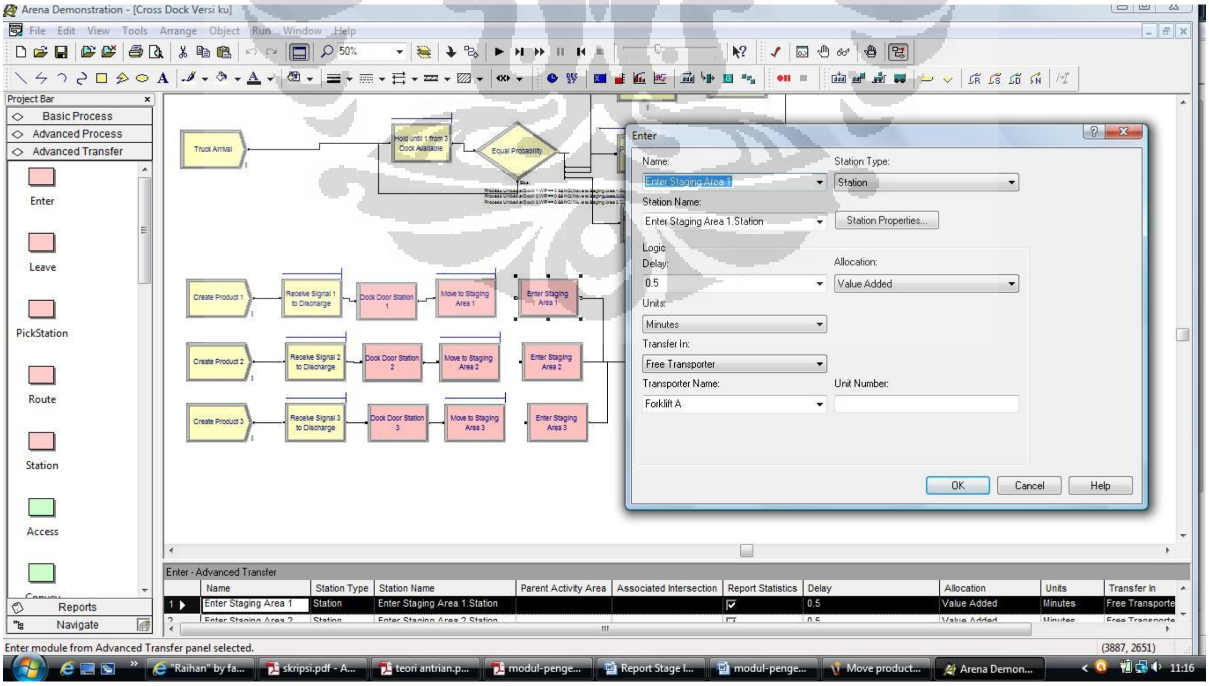
OK Cancel Help

Leave - Advanced Transfer

	Name	Allocation	Delay	Units	Transfer Out	Priority	Queue Type	Queue Name	Transporter Name	Selection
1	Move to Staging Area 1	Value Added	0.5	Minutes	Request Transporter	High (1)	Queue	Move to Staging Area 1.Queue	Forklift A	Cyclical
2	Move to Staging Area 2	Value Added	0	Hours	Request Transporter	High (1)	Queue	Move to Staging Area 2.Queue	Forklift B	Cyclical

Leave module from Advanced Transfer panel selected. (3167, 2667)

4. Product enter *Enter Module* it represent Station staging area where the product will stage depend on their type.



Enter

Name: Enter Staging Area 1
Station Type: Station
Station Name: Enter Staging Area 1.Station
Station Properties...
Logic
Delay: 0.5
Allocation: Value Added
Units: Minutes
Transfer In: Free Transporter
Transporter Name: Forklift A
Unit Number:

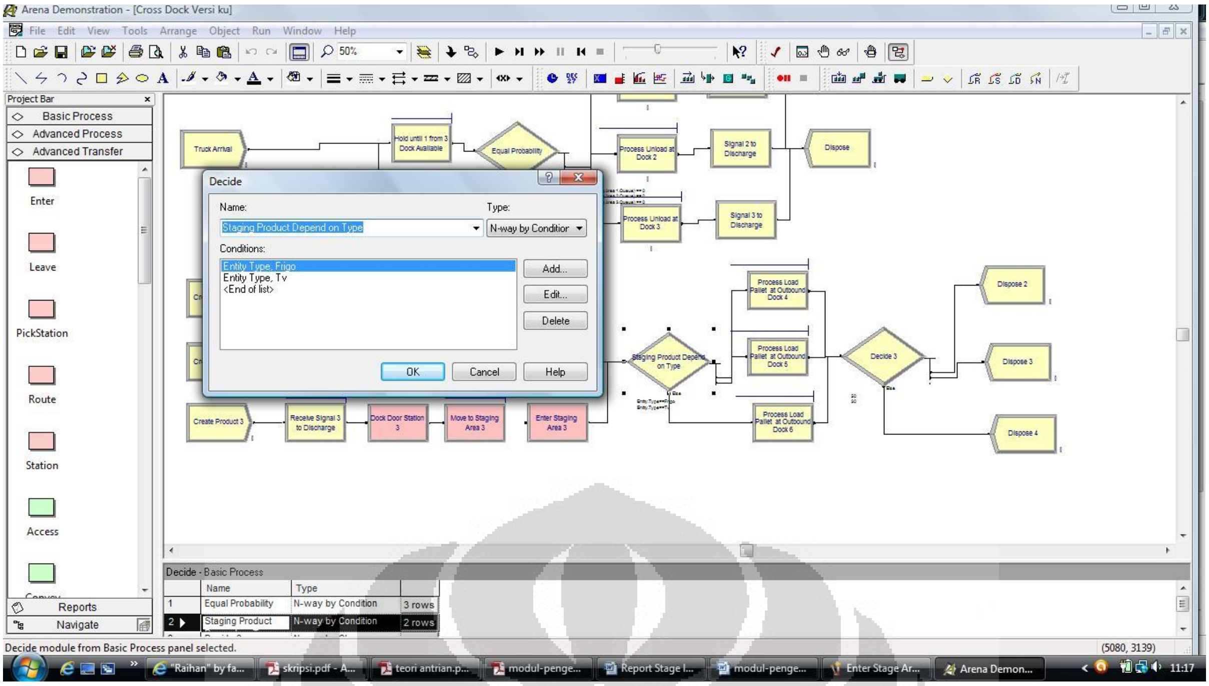
OK Cancel Help

Enter - Advanced Transfer

	Name	Station Type	Station Name	Parent Activity Area	Associated Intersection	Report Statistics	Delay	Allocation	Units	Transfer In
1	Enter Staging Area 1	Station	Enter Staging Area 1.Station			☑	0.5	Value Added	Minutes	Free Transporte
2	Enter Staging Area 2	Station	Enter Staging Area 2.Station			☑	0.5	Value Added	Minutes	Free Transporte

Enter module from Advanced Transfer panel selected. (3887, 2651)

5. In this module the product will stage depends on their type



Decide

Name: Staging Product Depend on Type

Type: N-way by Condition

Conditions:

Entity Type, Frigo
Entity Type, Tv
<End of list>

Add... Edit... Delete

OK Cancel Help

Decide - Basic Process

	Name	Type	
1	Equal Probability	N-way by Condition	3 rows
2	Staging Product	N-way by Condition	2 rows

Decide module from Basic Process panel selected.

(5080, 3139)

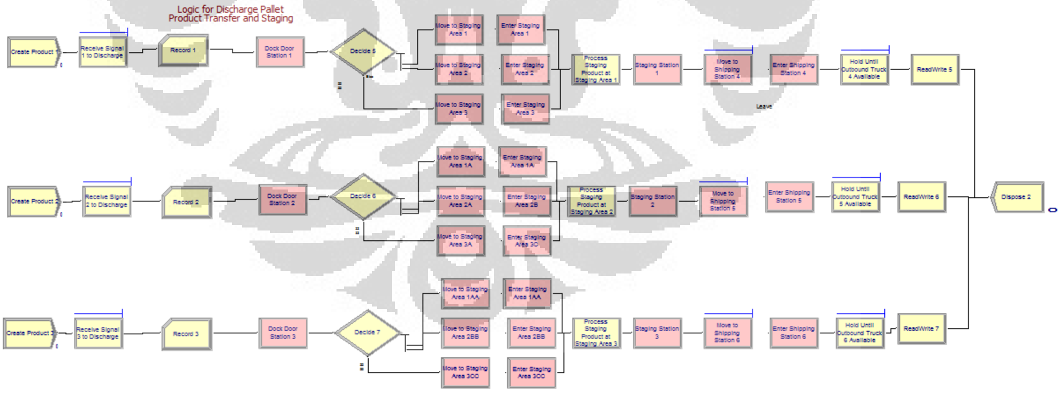
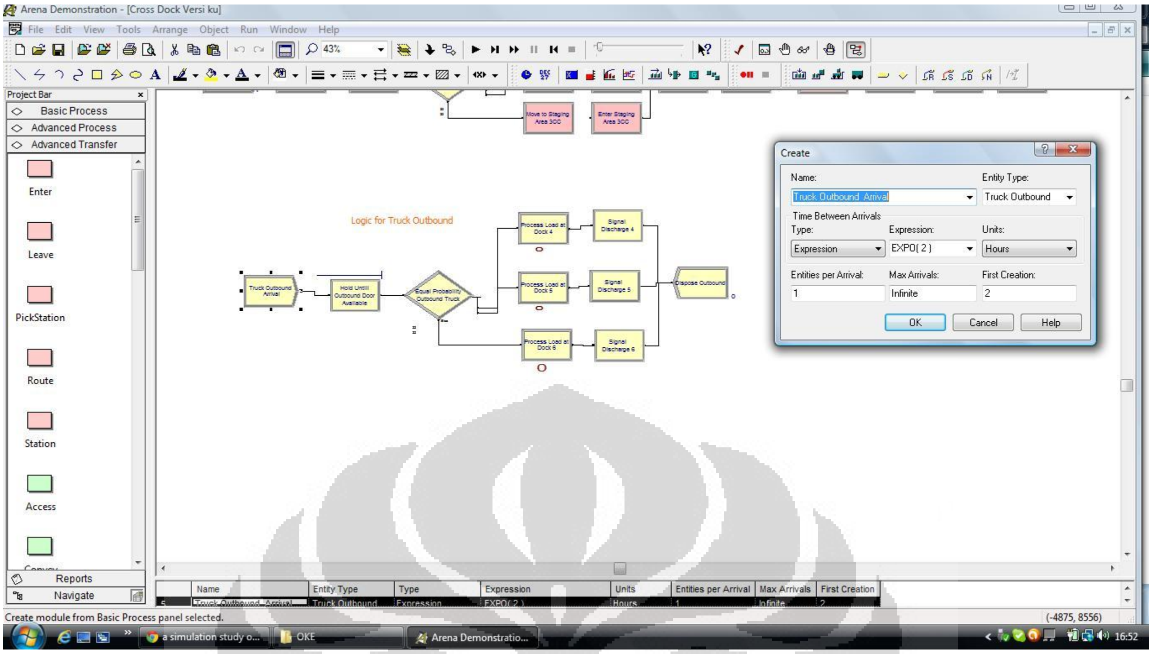


Figure 3.4.2 complete model Product Discharge and Transfer

3.4.3 Trucks Outbound Arrival

1. From the *create* module we make our truck outbound. It will arrive EXPO (2) hour or else λ= o.5 truck outbound / hour. Our first creation is after 2 hour in order to gives time to forklift finish their transfer. From the *hold* module will scan the condition is there any process in Outbound door 4, 5 or 6 if there is one available then the truck outbound will

enter, but if there is no outbound door available truck will wait in hold module until it " pulls " the truck from the queue line.



Arena Demonstration - [Cross Dock Versi ku]

Create

Name: Truck Outbound Arrival

Entity Type: Truck Outbound

Time Between Arrivals

Type: Expression

Expression: EXPO(2)

Units: Hours

Entities per Arrival: 1

Max Arrivals: Infinite

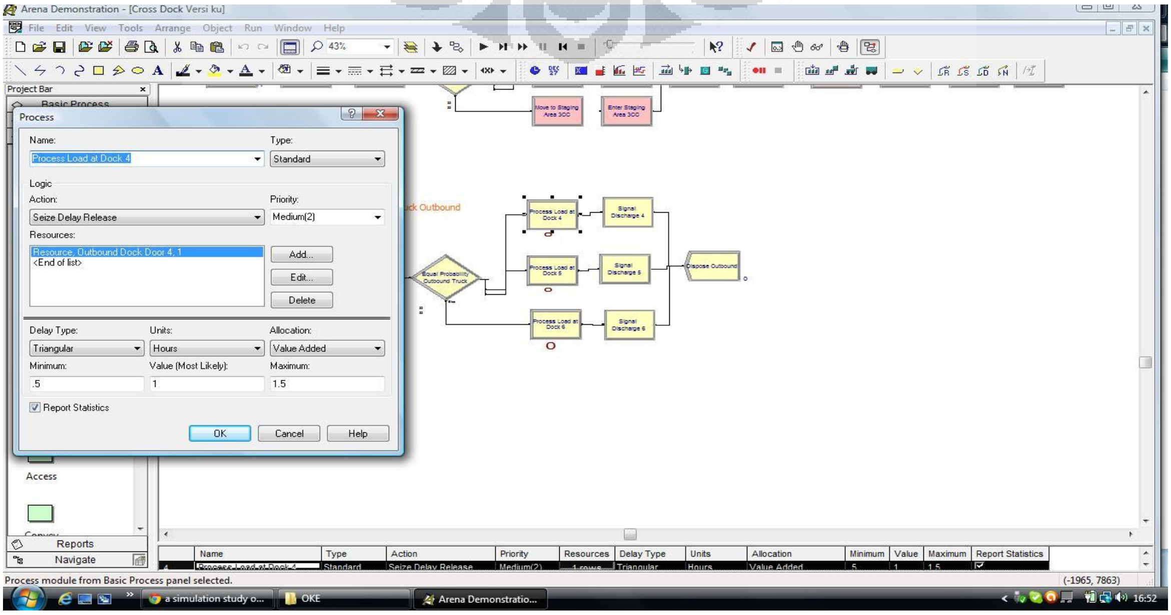
First Creation: 2

	Name	Entity Type	Type	Expression	Units	Entities per Arrival	Max Arrivals	First Creation
5	Truck Outbound Arrival	Truck Outbound	Expression	EXPO(2)	Hours	1	Infinite	2

Create module from Basic Process panel selected.

2. The next process is truck will entering decide module in this module truck will directed to outbound door availabe, we make the condition true if Process Load at Dock 4.WIP == 0 ; true if Process Load at Dock 5.WIP == 0 ; the rest go to outbound 6. Since 0 = idle and 1 = busy. Next step truck will enter the outbound door to load pallets.

3. This step the truck processing to load pallets for the processing time we use Triangular Distribution with *min, mode,max* (0.5,1,1.5) hour per truck outbound, than we have average 1 truck / hour.



Arena Demonstration - [Cross Dock Versi ku]

Process

Name: Process Load at Dock 4

Type: Standard

Logic

Action: Seize Delay Release

Priority: Medium(2)

Resources:

Resource, Outbound Dock Door 4, 1
<End of list>

Delay Type: Triangular

Units: Hours

Allocation: Value Added

Minimum: .5

Value (Most Likely): 1

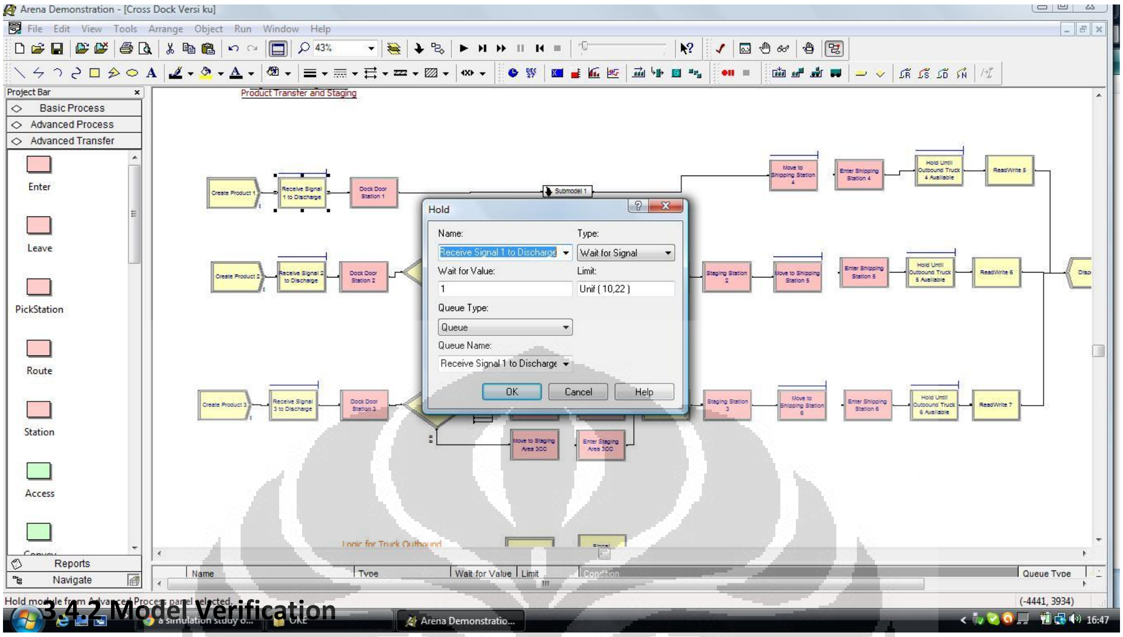
Maximum: 1.5

☑ Report Statistics

	Name	Type	Action	Priority	Resources	Delay Type	Units	Allocation	Minimum	Value	Maximum	Report Statistics
4	Process Load at Dock 4	Standard	Seize Delay Release	Medium(2)	1 rows	Triangular	Hours	Value Added	.5	1	1.5	☑

Process module from Basic Process panel selected.

4. After process load pallet truck will pass signal module to order the product in *hold* module from the Shipping station we assign UNIF (10,22) pallet will load, but in this condition if there is only five pallet in the *Hold* shipping station then outbound truck only carry maximum pallet they have. Then outbound truck dispose.



Name: Receive Signal 1 to Discharge

Type: Wait for Signal

Wait for Value: 1

Limit: Unif (10,22)

Queue Type: Queue

Queue Name: Receive Signal 1 to Discharge

Hold module from Advanced Process panel selected.

(-4441, 3934)

3.4.2 Model Verification

Model verification is the proses ensuring that Arena model behaves in the way it was intended according to the modeling asumption made. (David Kelton *et.al* simulation with arena 3rd edition p. 300). Another way to look at verification process is to consider it as " building the model correctly " (Simulation Handling Book Ch.7)at least two way to make verification from Arena simulation

1. Using **F4** from our keyboard

2. Using *trace* from Command menu

3.4.3 Model Validation

Model validation was defined as the process ensuring that a model represents reality at a given confidence level " building the correct model "(Simulation Handling Book Ch.8). There are two major types of validation of interest to the simulation practitioner. The first of these is face validity. Face validity means that the model, at least on the surface, represents reality we can run our model very slowly to discover error. The second is statistical validity. Statistical validity involves a quantitative comparison between the output performance of the actual system and the model (Law and Kelton, 2000). The simulation practitioner must achieve both types of validity to have confidence that the model is accurate.

3.4.4 Model replication

The number of replication is correlated with confidence interval. Confidence interval is represent percentage of error from our point estimate in our case is (total number out truck). The simulation run is 720 Minutes with 30 replication *" why 30 "* how we can we decide *n*= 30. This is formula to search replication correlated with confidence interval for terminating system

$$X \pm t_{n-1,1-\alpha/2}\frac{s}{\sqrt{n}}$$

n = number replication
$t^2_{n-1,1-a/2}$ = t table

for example our average sample mean (total Number out truck) from 5 replication (*n* = 5) is 13 truck with standar deviation (*s*) 11. 3 we got *halfwidth* 7 truck, than 7/13 = 53,8% error in point estimate 13 truck. If we want to achieve some spesific half-width (*h*) we can use the formule

$$n \cong n_0 \frac{h_0^2}{h^2}$$

Where *n* is the number of initial replication we have and h_0 is the half width we got. In the total number out example above, to reduce the half width from *h_0 = 7 truck* to say , *h* = 3, we'd thus need a total replication

$$n \cong 5\frac{7^2}{3^2} = 27.2 \text{ (approximation)}$$

Chapter 4. Scenario Problem

Model of a cross-dock system that groups and transfers material for further shipment. This facility has three incoming docks and three outgoing docks. Trucks arrive at each of the incoming docks with loads of material on pallets. The interarrival time is (EXPO) 1 hour between truck arrival on each incoming dock (all times are in minutes). Each trucks will have a number of pallets drawn from a UNIF (10,22) that need to be transfered to one of outgoing docks. The processing time for discharge pallet using Triangular distribution with *min,mode,max (0.5,1,1,5)* hour per truck. Each dock have equal probability of any incoming pallet going to any of the three outgoing docks. When trucks arrive, an ressources with troly unloading the pallets at the incoming docks. This activity require time (0.5) load and (0.5) unload for each pallets (all time are in minutes).So the pallets are transferred by Forklift one by one pallets to staging area to assign their quantity and destination. After that pallets are transfered to the outbound dock which are located on the other side of the building this activity require (0.3) load and (0.3) unload. Truck outbound assign interarrival time EXPO (2) hour and first create after 2 hour simulation to give time to pallet fully transfered. From the staging area every pallet assign to their destination dock and quantity, they (trucks) required UNIF (10,22) pallet but this difficult becouse transfer time take much time before arrive at outbound door. In our model we have six forklift with speed average 75 ft/minute. We run our Simulation 720 minutes with 30 replication.

In practice, transshipment has various layouts. In this research the layout is being restricted to three inbound and three outbound door. Eventhough restriction is not realistic (in a real transshipment platform) but can be used as a baseline for other layouts. In this model an incoming truck arrives at inbound door and unloads products for various destinations. If the outgoing truck is going to the fine destination, the products are moved directly to outbound truck (direct transit of products), in the other hand. In studied model, the following assumptions are considered:

- Each trailer leaves the inbound door when it is fully unloaded. On the other side, each trailer leaves the outbound door when it is fully loaded.
- The storage capacity is assumed unlimited.
- All incoming and outgoing semi trailers are available at time zero.
- The total numbers of arriving and departing products are depends on time process.
- Loading, unloading time have same distribution
- transfer time are constant and are not considered.

Chapter 5. Result and Analysis

From 720 minutes simulation we got the result truck inbound and in as a table below with average 12 truck in and 11 truck out with halfwidth 9% and 10 %

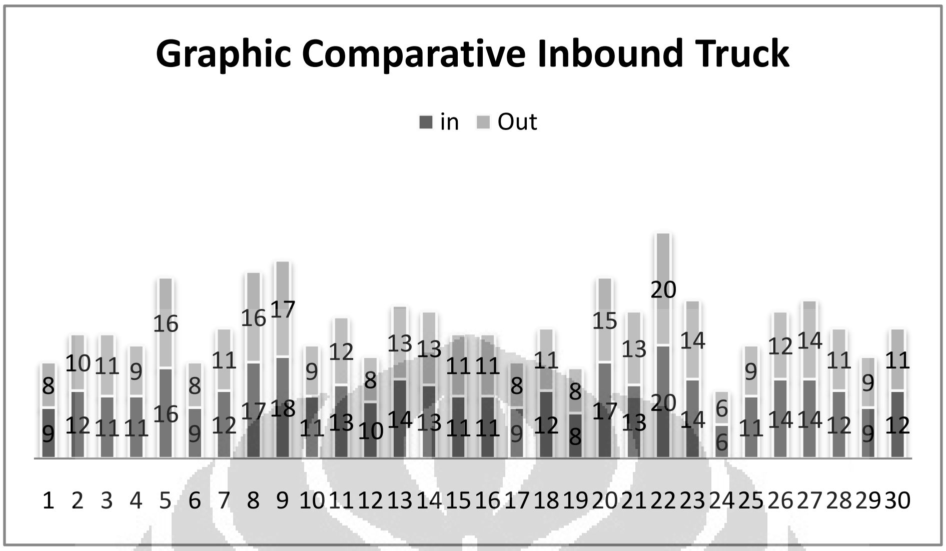


Figure 5.1 grafik comparative between Truck in and Truck out from 30 replication

From the table 5.1 we can see truck arrive at time 00 enter inbound door 1 and exit at the 55 minutes 43 second with number of pallets 20. In this activity we assumed that serve hour (row) is the time from resource complete their discharging pallets

Event	Time in (at Minute)	Dock number	Time Out	Serve Hour (discharge)	Pallet discharge
1	0	1	55''43	55''43	20
2	20''77	2	89''31	68''54	13
3	142''48	2	205''34	62''86	14
4	203''63	3	265''.67	62''.04	21
5	258''17	1	314''79	56''62	11
6	268''.98	2	345''.22	76''.24	11
7	385''.06	2	454''.88	69''82	11
8	385''14	1	434''.30	49''16	18
9	403''77	3	471''21	67''44	10
10	512''.12	1	612''14	100''	18

Table 5.1 event occurrence from Inbound Dock according time simulation

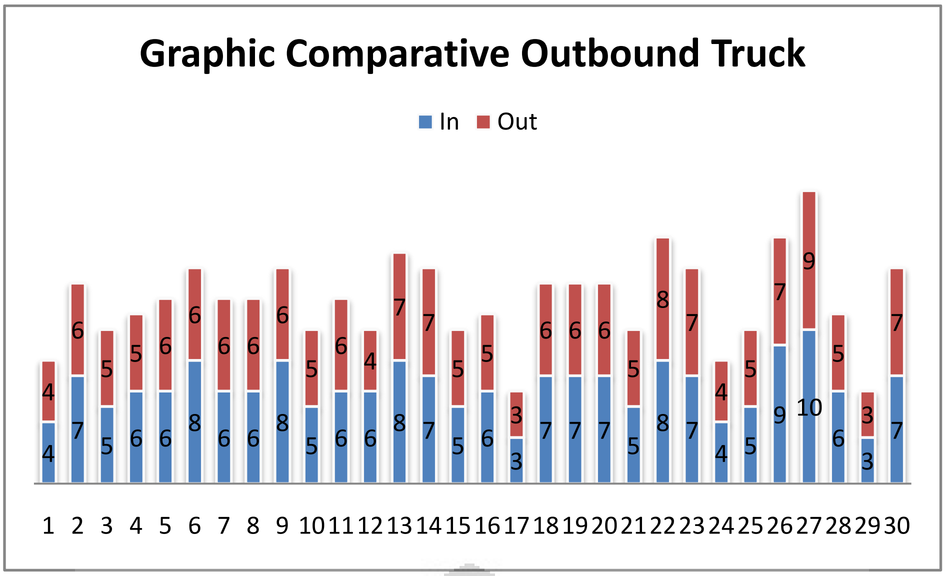


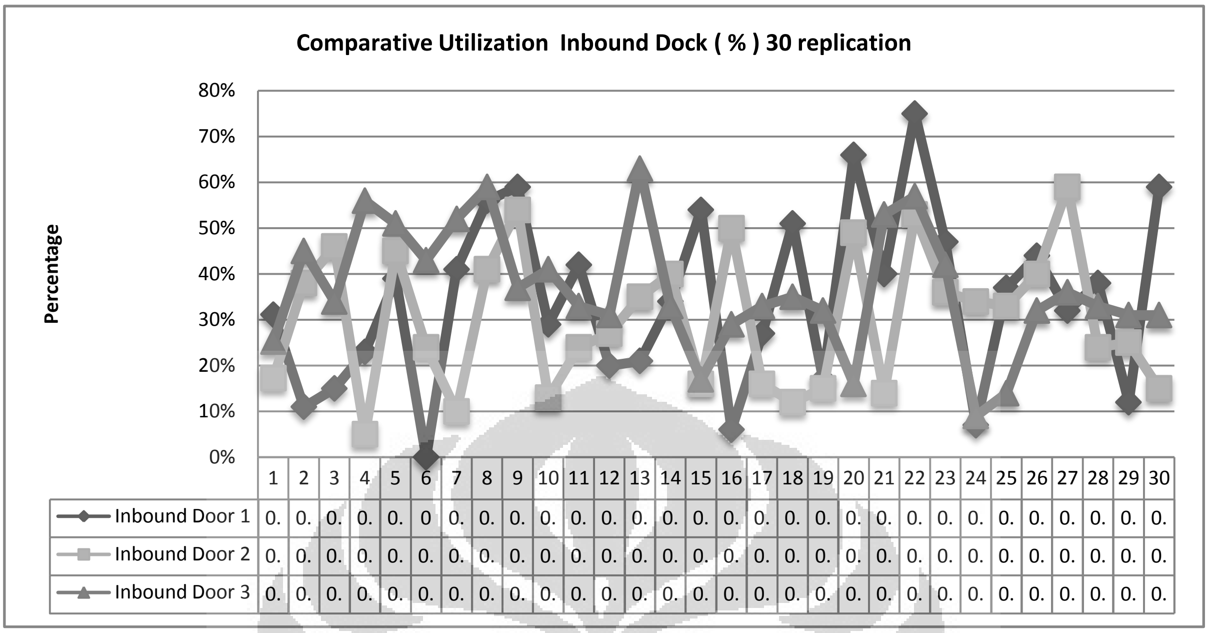
Figure 5.2 grafik comparative between Truck in and Truck out Outbound Door from 30 replication

From the Outbound dock door we have average 6 truck in and 5 truck out with halfwidth 10% and 9%

Event	Time in	Dock Number	Time Out	Serve Hour	Number Pallet Load
1	157''	6	180''.9	24''	8
2	240''	4	291'' 7	51''	22
3	342''	5	293''	51''	14
4	501''56	5	535''.5	34''	19

Table 5.2 event occurrence from Outbound Dock according time simulation

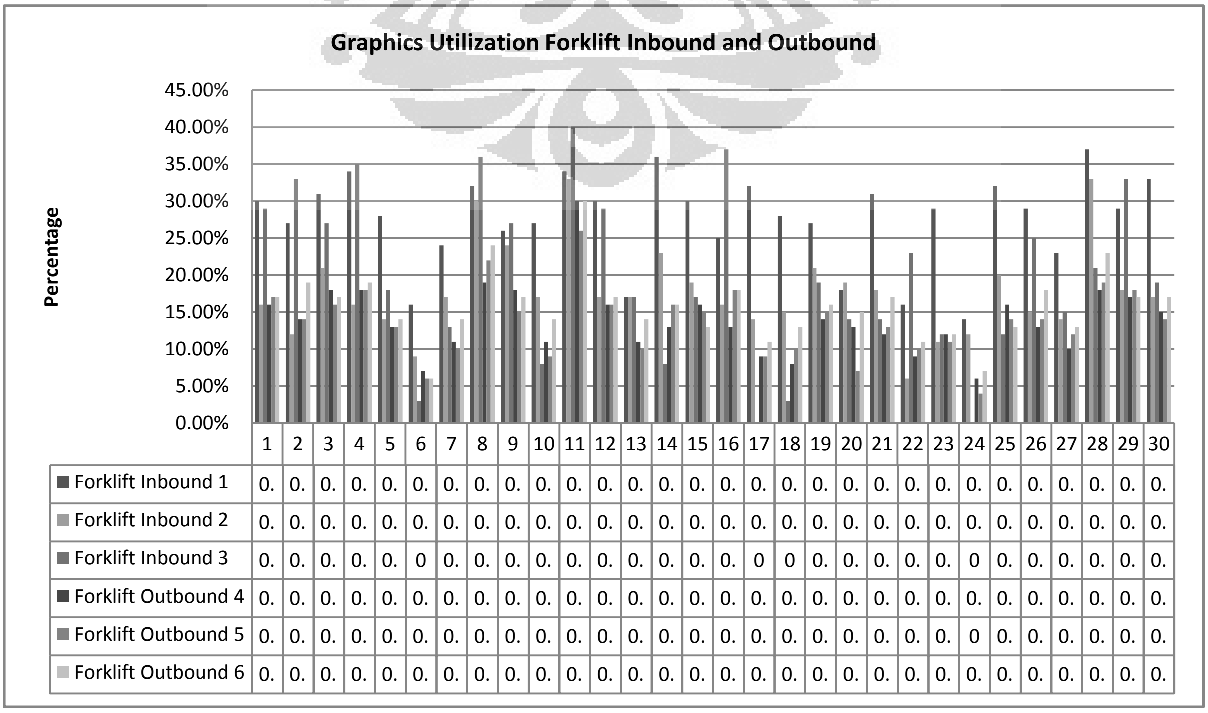
As a result utilization inbound dock from 30 replication we have inbound dock door 1 average 34.4% inbound door 2 have average 30.3 % of utilization and inbound door 3 have average 36.8% from their utilization



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Inbound Door 1	0.	0.	0.	0.	0.	0	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
Inbound Door 2	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
Inbound Door 3	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.

Figure 5.3 total utilization inbound door from 30 replication

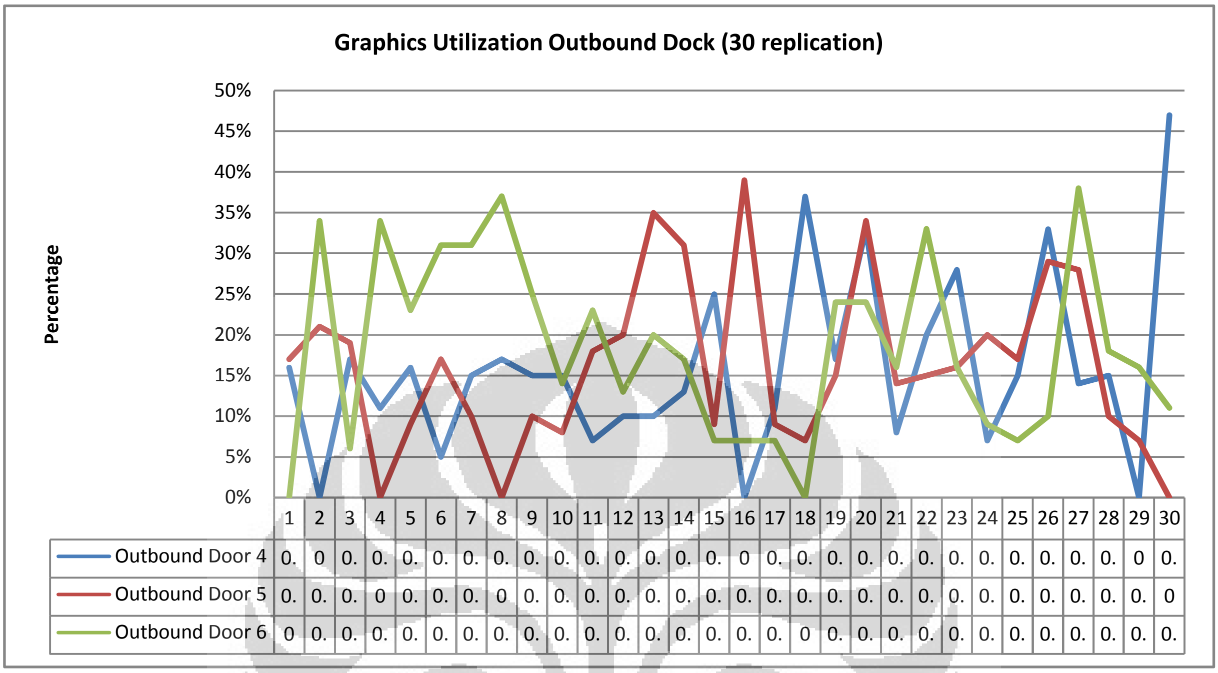
Since we have six resources three forklift inbound and three forklift outbound we have average utilization 27.5 %, 17.8 % ,19.6%, for forklift inbound 1, 2, and 3. 13.9%, 13.7%, 15.7% for forklift outbound 4, 5 and 6. Almost there is no queue inside of our crossdock building.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
■ Forklift Inbound 1	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
■ Forklift Inbound 2	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
■ Forklift Inbound 3	0.	0.	0.	0.	0.	0	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0	0	0.	0.	0.	0.	0.	0	0.	0.	0.	0.	0.	0.
■ Forklift Outbound 4	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
■ Forklift Outbound 5	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0	0.	0.	0.	0.	0.	0.
■ Forklift Outbound 6	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.

Figure 5.4 total utilization Forklift

From outbound dock we have more small utilization percentage than inbound dock. Becouse of in our model truck out bound first creation at time 120 minute to give time for forklift finish transfer pallet from inbound to outbound dock. Outbound bond door 4 have 15.9 % utilization, outbound dock 5 16.1 %, and 18.4 % for outbound door 6



For the total time transfer per pallet (product 1, product 2 and product 3) 1.162 minutes, 1.111 minutes, 0.958 minutes with halfwidth 0. 02498, 0.01337, and 0.1667 with percentage error is 2%, 1% and 17 %.

Chapter 6. Conclusion

We conclue that Arena simulation is one of the solving tool to analyzing problems for example problem the flow process (bottle neck), resources utilization or maybe much more advantage from Arena Simulationand that we not explore yet.

The crossdock operation is need to be planned for to optimizing our resources utilization.

Model Proposed

Since our Inbound and Outbound door is fixed, we still can arrange our resources utilization in order to more visible. Is more simple besouse we already know the interarrival rate and service rate from our resources, than we only have to arrange our forklift “on-call” they stand-bye only call

6.1 Future reseach

There are many possibility to expand this reseach for example we can applied in bigger cross dock let say 36 inbound dock and 36 outbound dock or we can combined our queuing analysis with optmization technique, where we seek the minimization of the sum of the two cost, the cost of waiting and the cost of offering service facility.

References

Altiok ,Tayfur and Benjamin Melamed (2007) " *simulation modeling and analysis with arena* " Academic Press is an imprint of Elsevier, 2007

Bartholdi, John J. and Kevin R. Gue. *The Best Shape for a Crossdock* Transportation Science.Vol. 38, No. 2. May 2004. pp. 235-244. DOI: 10.1287/trsc.1030.0077 *Making the Move to Crossdocking*, Maida Napolitano and the staff of Gross & Associates, 2000 copyright, www.werc.org

Cohen, Yuval and Baruch Keren (2009) " *Trailer door in a synchronous cross-dock operation* " International journal Logistics Systems and Management Vol.5, No.5, 2009

Danuta, Kisperka-Moron (1999) " *Warehousing conditions for holding inventory in Polish supply chain* " International journal Production Economics Science Direct(1999) 123-128

Li Zhengping *et.al* (2008) " *Optimal Decision-making on Product Ranking For Crossdocking / Warehousing Operation* ". Proquest.net/pqdweb

L.Whicker *et.al* (2009) " *Understanding the relationships between time and cost to improve supply chain performance* " International Journal Production Economics Science Direct 121 (2009) 641-650

Kelton W, David.*et.al* (2004) " *Simulation with Arena* " third edition, mc graw hill, 2004

Koster, René de *et.al* (2007) " *Design and control of warehouse order picking: A literature review* " European Journal of operation Reseach Science Direct 182 (2007) 481-501

Steyn, Pieter (2010) " *Program Managing the Supply Chain Portfolio*". PM world Today. Vol XII, Issue VI.

Taha, Hamdy A. (2007) " *Operations Reseach an intriodruction* " Eight Edition, Pearson International Edition, 2007

Wang Jiana-Fu (2009) "*Operational Strategies for Single-Stage Crossdocks*" Proquest, 2009

Yen-Chun Jim Wu and I.C.Huang (2007) " *Operation reseach practice on logistics management in Taiwan:An academic view*" European Journal of Operation Reseach Science Direct 18 2 (2007) 428-435

Y. Wu *et.al.* (2007)" *A simulation Study on Supply Chain Complexity in Manufacturing Industry* "

Appendix 1

I. Siman Language

```
;
;
;     Model statements for module:  BasicProcess.Create 1 (Truck Inbound  Arrival)
;

59$           CREATE,
1,HoursToBaseTime(0.0),Truck:HoursToBaseTime(EXPO(1.5)):NEXT(60$);

60$           ASSIGN:        Truck Inbound  Arrival.NumberOut=Truck Inbound
Arrival.NumberOut + 1:NEXT(58$);

58$           SCAN:
                             Process Unload at Dock 1.WIP == 0 || Process Unload at
Dock 2.WIP == 0 || Process Unload at Dock 3.WIP == 0
                             :NEXT(4$);

;
;
;     Model statements for module:  BasicProcess.Decide 1 (Equal Probability)
;
4$            BRANCH,        1:
                             If,Process Unload at Dock 1.WIP == 0 && NQ(Move to
Staging Area 1.Queue) == 0,0$,Yes:
                             If,Process Unload at Dock 2.WIP == 0 && NQ(Move to
Staging Area 2.Queue) == 0,1$,Yes:
                             Else,2$,Yes;

;
;
;     Model statements for module:  BasicProcess.Process 3 (Process Unload at Dock
3)
;
2$            ASSIGN:        Process Unload at Dock 3.NumberIn=Process Unload at
Dock 3.NumberIn + 1:
                             Process Unload at Dock 3.WIP=Process Unload at Dock
3.WIP+1;
68$           QUEUE,         Process Unload at Dock 3.Queue;
67$           SEIZE,         2,VA:
                             Inbound Dock Door 3,1:NEXT(66$);

66$           DELAY:         HoursToBaseTime(Triangular(.5,1,1.5)),,VA;
65$           RELEASE:       Inbound Dock Door 3,1;
113$          ASSIGN:        Process Unload at Dock 3.NumberOut=Process Unload at
Dock 3.NumberOut + 1:
                             Process Unload at Dock 3.WIP=Process Unload at Dock
3.WIP-1:NEXT(7$);

;
;
;     Model statements for module:  AdvancedProcess.Signal 3 (Signal 3 to
Discharge)
;
7$            SIGNAL:        3:NEXT(3$);

;
;
;     Model statements for module:  BasicProcess.Dispose 1 (Dispose Inbound)
;
3$            ASSIGN:        Dispose Inbound.NumberOut=Dispose Inbound.NumberOut +
1;
116$          DISPOSE:       Yes;

;
```

```
;
;     Model statements for module:  BasicProcess.Process 1 (Process Unload at Dock
1)
;
0$            ASSIGN:        Process Unload at Dock 1.NumberIn=Process Unload at
Dock 1.NumberIn + 1:
                             Process Unload at Dock 1.WIP=Process Unload at Dock
1.WIP+1;
120$          QUEUE,         Process Unload at Dock 1.Queue;
119$          SEIZE,         2,VA:
                             Inbound Dock Door 1,1:NEXT(118$);

118$          DELAY:         HoursToBaseTime(Triangular(.5,1,1.5)),,VA;
117$          RELEASE:       Inbound Dock Door 1,1;
165$          ASSIGN:        Process Unload at Dock 1.NumberOut=Process Unload at
Dock 1.NumberOut + 1:
                             Process Unload at Dock 1.WIP=Process Unload at Dock
1.WIP-1:NEXT(5$);

;
;
;     Model statements for module:  AdvancedProcess.Signal 1 (Signal 1 to
Discharge)
;
5$            SIGNAL:        1:NEXT(3$);

;
;
;     Model statements for module:  BasicProcess.Process 2 (Process Unload at Dock
2)
;
1$            ASSIGN:        Process Unload at Dock 2.NumberIn=Process Unload at
Dock 2.NumberIn + 1:
                             Process Unload at Dock 2.WIP=Process Unload at Dock
2.WIP+1;
171$          QUEUE,         Process Unload at Dock 2.Queue;
170$          SEIZE,         2,VA:
                             Inbound Dock Door 2,1:NEXT(169$);

169$          DELAY:         HoursToBaseTime(Triangular(.5,1,1.5)),,VA;
168$          RELEASE:       Inbound Dock Door 2,1;
216$          ASSIGN:        Process Unload at Dock 2.NumberOut=Process Unload at
Dock 2.NumberOut + 1:
                             Process Unload at Dock 2.WIP=Process Unload at Dock
2.WIP-1:NEXT(6$);

;
;
;     Model statements for module:  AdvancedProcess.Signal 2 (Signal 2 to
Discharge)
;
6$            SIGNAL:        2:NEXT(3$);

;
;
;     Model statements for module:  BasicProcess.Create 2 (Create Product 1)
;

219$          CREATE,
1,MinutesToBaseTime(0.0),Refrigerator:MinutesToBaseTime(EXPO(3)):NEXT(220$);

220$          ASSIGN:        Create Product 1.NumberOut=Create Product 1.NumberOut
+ 1:NEXT(8$);

;
;
;     Model statements for module:  AdvancedProcess.Hold 2 (Receive Signal 1 to
Discharge)
```

```
;
8$            QUEUE,         Receive Signal 1 to Discharge.Queue;
              WAIT:          1,Unif ( 10,22 ):NEXT(53$);

;
;
;     Model statements for module:  BasicProcess.Assign 4 (Assign 4)
;
53$           ASSIGN:        Arrtime P1=tnow:NEXT(10$);
;
;
;     Model statements for module:  AdvancedTransfer.Station 1 (Dock Door Station
1)
;

10$           STATION,       Dock Door 1;
225$          DELAY:         0.0,,VA:NEXT(11$);
;
;
;     Model statements for module:  AdvancedTransfer.Leave 1 (Move to Staging Area
1)
;
11$           DELAY:         0.00,,VA:NEXT(239$);

239$          QUEUE,         Move to Staging Area 1.Queue;
243$          REQUEST,       1:Forklift Inbound 1(CYC);
231$          DELAY:         0.5,,VA:NEXT(233$);

233$          TRANSPORT:     ,Enter Staging Area 1.Station;



;
;
;     Model statements for module:  AdvancedTransfer.Enter 1 (Enter Staging Area 1)
;

12$           STATION,       Enter Staging Area 1.Station;
246$          DELAY:         0.5,,VA:NEXT(248$);

248$          FREE:          Forklift Inbound 1:NEXT(48$);
;
;
;     Model statements for module:  AdvancedProcess.Delay 1 (Process Staging
Product at Staging Area 1)
;
48$           DELAY:         EXPO( 0.5 ),,Other:NEXT(51$);
;
;
;     Model statements for module:  BasicProcess.Assign 3 (Assign Product Type from
Inbound truck 3)
;
51$           ASSIGN:        Type=DISC ( 0.3,1,0.65,2,1,3 ):
                             Entity.Type=Types ( type ):
                             Entity.Picture=Pict ( type ):NEXT(13$);
;
;
;     Model statements for module:  BasicProcess.Decide 2 (Staging Product Depend
on Percentage)
;
13$           BRANCH,        1:
                             With,(33)/100,31$,Yes:
                             With,(33)/100,32$,Yes:
                             Else,33$,Yes;
;
;
;     Model statements for module:  AdvancedTransfer.Station 6 (Outbound Door
Station 6)
;
```

```
33$           STATION,       Outbound Door Station 6;
261$          DELAY:         0.0,,VA:NEXT(38$);
;
;
;     Model statements for module:  AdvancedTransfer.Leave 6 (Move to Shipping
Station 6)
;
38$           DELAY:         0.00,,VA:NEXT(275$);

275$          QUEUE,         Move to Shipping Station 6.Queue;
279$          REQUEST,       1:Forklift Outbound 3(CYC);
267$          DELAY:         0.5,,VA:NEXT(269$);

269$          TRANSPORT:     ,Enter Shipping Station 6.Station;
;
;
;     Model statements for module:  AdvancedTransfer.Station 4 (Outbound Door
Station 4)
;

31$           STATION,       Outbound Door Station 4;
284$          DELAY:         0.0,,VA:NEXT(34$);
;
;
;     Model statements for module:  AdvancedTransfer.Leave 4 (Move to Shipping
Station 4)
;
34$           DELAY:         0.00,,VA:NEXT(298$);

298$          QUEUE,         Move to Shipping Station 4.Queue;
302$          REQUEST,       1:Forklift Outbound 1(CYC);
290$          DELAY:         0.5,,VA:NEXT(292$);
292$          TRANSPORT:     ,Enter Shipping Station 4.Station;
;
;
;     Model statements for module:  AdvancedTransfer.Station 5 (Outbound Door
Station 5)
;
32$           STATION,       Outbound Door Station 5;
307$          DELAY:         0.0,,VA:NEXT(36$);
;
;
;     Model statements for module:  AdvancedTransfer.Leave 5 (Move to Shipping
Station 5)
;
36$           DELAY:         0.00,,VA:NEXT(321$);
321$          QUEUE,         Move to Shipping Station 5.Queue;
325$          REQUEST,       1:Forklift Outbound 2(CYC);
313$          DELAY:         0.5,,VA:NEXT(315$);
315$          TRANSPORT:     ,Enter Shipping Station 5.Station;
;
;
;     Model statements for module:  BasicProcess.Create 3 (Create Product 2)
;
328$          CREATE,
1,MinutesToBaseTime(0.0),Television:MinutesToBaseTime(EXPO(3)):NEXT(329$);

329$          ASSIGN:        Create Product 2.NumberOut=Create Product 2.NumberOut
+ 1:NEXT(14$);
;
;
;     Model statements for module:  AdvancedProcess.Hold 3 (Receive Signal 2 to
Discharge)
;
14$           QUEUE,         Receive Signal 2 to Discharge.Queue;
              WAIT:          2,Unif ( 10,22 ):NEXT(54$);
;
```

```
;
;     Model statements for module:  BasicProcess.Assign 5 (Assign 5)
;
54$           ASSIGN:        Arrtime P2=tnow:NEXT(16$);
;
;
;     Model statements for module:  AdvancedTransfer.Station 2 (Dock Door Station
2)
;
16$           STATION,       Dock Door 2;
334$          DELAY:         0.0,,VA:NEXT(17$);
;
;
;     Model statements for module:  AdvancedTransfer.Leave 2 (Move to Staging Area
2)
;
17$           DELAY:         0.00,,VA:NEXT(348$);

348$          QUEUE,         Move to Staging Area 2.Queue;
352$          REQUEST,       1:Forklift Inbound 2(CYC);
340$          DELAY:         0.000000000000000,,VA:NEXT(342$);
342$          TRANSPORT:     ,Enter Staging Area 2.Station;
;
;
;     Model statements for module:  AdvancedTransfer.Enter 2 (Enter Staging Area 2)
;

18$           STATION,       Enter Staging Area 2.Station;
355$          DELAY:         0.5,,VA:NEXT(357$);

357$          FREE:          Forklift Inbound 2:NEXT(49$);
;
;
;     Model statements for module:  AdvancedProcess.Delay 2 (Process Staging
Product at Staging Area 2)
;
49$           DELAY:         EXPO( 0.5 ),,Other:NEXT(51$);
;
;
;     Model statements for module:  BasicProcess.Create 4 (Create Product 3)
;

366$          CREATE,
1,MinutesToBaseTime(0.0),Radio:MinutesToBaseTime(EXPO(3)):NEXT(367$);
367$          ASSIGN:        Create Product 3.NumberOut=Create Product 3.NumberOut
+ 1:NEXT(19$);
;
;
;     Model statements for module:  AdvancedProcess.Hold 4 (Receive Signal 3 to
Discharge)
;
19$           QUEUE,         Receive Signal 3 to Discharge.Queue;
              WAIT:          3,Unif ( 10,22 ):NEXT(55$);
;
;
;     Model statements for module:  BasicProcess.Assign 6 (Assign 6)
;
55$           ASSIGN:        Arrtime P3=tnow:NEXT(21$);
;
;
;     Model statements for module:  AdvancedTransfer.Station 3 (Dock Door Station
3)
;
21$           STATION,       Dock Door 3;
372$          DELAY:         0.0,,VA:NEXT(22$);
;
;
```

```
;     Model statements for module:  AdvancedTransfer.Leave 3 (Move to Staging Area
3)
;
22$           DELAY:         0.00,,VA:NEXT(386$);
386$          QUEUE,         Move to Staging Area 3.Queue;
390$          REQUEST,       1:Forklift Inbound 3(CYC);
378$          DELAY:         0.000000000000000,,VA:NEXT(380$);
380$          TRANSPORT:     ,Enter Staging Area 3.Station;
;
;
;     Model statements for module:  AdvancedTransfer.Enter 3 (Enter Staging Area 3)
;

23$           STATION,       Enter Staging Area 3.Station;
393$          DELAY:         0.5,,VA:NEXT(395$);
395$          FREE:          Forklift Inbound 3:NEXT(50$);

;
;
;     Model statements for module:  AdvancedProcess.Delay 3 (Process Staging
Product at Staging Area 3)
;
50$           DELAY:         EXPO( 0.5 ),,Other:NEXT(51$);
;
;
;     Model statements for module:  AdvancedTransfer.Enter 4 (Enter Shipping
Station 4)
;
35$           STATION,       Enter Shipping Station 4.Station;
404$          DELAY:         0.5,,VA:NEXT(406$);

406$          FREE:          Forklift Outbound 1:NEXT(25$);
;
;
;     Model statements for module:  AdvancedProcess.Hold 5 (Hold Until Outbound
Truck 4 Available)
;
25$           QUEUE,         Hold Until Outbound Truck 4 Available.Queue;
              WAIT:          4,UNIF ( 10,22):NEXT(52$);
;
;
;     Model statements for module:  AdvancedProcess.ReadWrite 5 (ReadWrite 5)
;
52$           WRITE,         Total time Product,RECORDSET(Total Time P1):
                             TNOW-Arrtime P1:NEXT(24$);
;
;
;     Model statements for module:  BasicProcess.Dispose 2 (Dispose 2)
;
24$           ASSIGN:        Dispose 2.NumberOut=Dispose 2.NumberOut + 1;
415$          DISPOSE:       Yes;
;
;
;     Model statements for module:  AdvancedTransfer.Enter 5 (Enter Shipping
Station 5)
;
37$           STATION,       Enter Shipping Station 5.Station;
416$          DELAY:         0.5,,VA:NEXT(418$);

418$          FREE:          Forklift Outbound 2:NEXT(27$);
;
;
;     Model statements for module:  AdvancedProcess.Hold 6 (Hold Until Outbound
Truck 5 Available)
;
27$           QUEUE,         Hold Until Outbound Truck 5 Available.Queue;
              WAIT:          5,UNIF ( 10,22 ):NEXT(56$);
;
```

```
;
;     Model statements for module:  AdvancedProcess.ReadWrite 6 (ReadWrite 6)
;
56$           WRITE,         Total time Product,RECORDSET(Total Time P2):
                             TNOW-Arrtime P2:NEXT(24$);
;
;
;     Model statements for module:  AdvancedTransfer.Enter 6 (Enter Shipping
Station 6)
;

39$           STATION,       Enter Shipping Station 6.Station;
427$          DELAY:         0.5,,VA:NEXT(429$);
429$          FREE:          Forklift Outbound 3:NEXT(29$);
;
;
;     Model statements for module:  AdvancedProcess.Hold 7 (Hold Until Outbound
Truck 6 Available)
;
29$           QUEUE,         Hold Until Outbound Truck 6 Available.Queue;
              WAIT:          6,UNIF ( 10,22 ):NEXT(57$);
;
;
;     Model statements for module:  AdvancedProcess.ReadWrite 7 (ReadWrite 7)
;
57$           WRITE,         Total time Product,RECORDSET(Total Time P3):
                             TNOW-Arrtime P3:NEXT(24$);
;
;
;     Model statements for module:  BasicProcess.Create 5 (Truck Outbound  Arrival)
;

438$          CREATE,
1,HoursToBaseTime(2),Truck:HoursToBaseTime(EXPO(1.5)):NEXT(439$);
439$          ASSIGN:        Truck Outbound  Arrival.NumberOut=Truck Outbound
Arrival.NumberOut + 1:NEXT(44$);
;
;
;     Model statements for module:  BasicProcess.Decide 4 (Equal Probability
Outbound Truck)
;
44$           BRANCH,        1:
                             With,(33)/100,40$,Yes:
                             With,(33)/100,41$,Yes:
                             Else,42$,Yes;
;
;
;     Model statements for module:  BasicProcess.Process 9 (Process Load at Dock 6)
;
42$           ASSIGN:        Process Load at Dock 6.NumberIn=Process Load at Dock
6.NumberIn + 1:
                             Process Load at Dock 6.WIP=Process Load at Dock
6.WIP+1;
447$          QUEUE,         Process Load at Dock 6.Queue;
446$          SEIZE,         2,VA:
                             Outbound Dock Door 6,1:NEXT(445$);

445$          DELAY:         HoursToBaseTime(Triangular(.5,1,1.5)),,VA;
444$          RELEASE:       Outbound Dock Door 6,1;
492$          ASSIGN:        Process Load at Dock 6.NumberOut=Process Load at Dock
6.NumberOut + 1:
                             Process Load at Dock 6.WIP=Process Load at Dock 6.WIP-
1:NEXT(47$);

;
;
;     Model statements for module:  AdvancedProcess.Signal 7 (Signal  Discharge 6)
;
```

```
47$           SIGNAL:        6:NEXT(43$);
;
;
;     Model statements for module:  BasicProcess.Dispose 5 (Dispose Outbound)
;
43$           ASSIGN:        Dispose Outbound.NumberOut=Dispose Outbound.NumberOut
+ 1;
495$          DISPOSE:       Yes;
;
;
;     Model statements for module:  BasicProcess.Process 7 (Process Load at Dock 4)
;
40$           ASSIGN:        Process Load at Dock 4.NumberIn=Process Load at Dock
4.NumberIn + 1:
                             Process Load at Dock 4.WIP=Process Load at Dock
4.WIP+1;
499$          QUEUE,         Process Load at Dock 4.Queue;
498$          SEIZE,         2,VA:
                             Outbound Dock Door 4,1:NEXT(497$);

497$          DELAY:         HoursToBaseTime(Triangular(.5,1,1.5)),,VA;
496$          RELEASE:       Outbound Dock Door 4,1;
544$          ASSIGN:        Process Load at Dock 4.NumberOut=Process Load at Dock
4.NumberOut + 1:
                             Process Load at Dock 4.WIP=Process Load at Dock 4.WIP-
1:NEXT(45$);
;
;
;     Model statements for module:  AdvancedProcess.Signal 5 (Signal  Discharge 4)
;
45$           SIGNAL:        4:NEXT(43$);


;
;
;     Model statements for module:  BasicProcess.Process 8 (Process Load at Dock 5)
;
41$           ASSIGN:        Process Load at Dock 5.NumberIn=Process Load at Dock
5.NumberIn + 1:
                             Process Load at Dock 5.WIP=Process Load at Dock
5.WIP+1;
550$          QUEUE,         Process Load at Dock 5.Queue;
549$          SEIZE,         2,VA:
                             Outbound Dock Door 5,1:NEXT(548$);

548$          DELAY:         HoursToBaseTime(Triangular(.5,1,1.5)),,VA;
547$          RELEASE:       Outbound Dock Door 5,1;
595$          ASSIGN:        Process Load at Dock 5.NumberOut=Process Load at Dock
5.NumberOut + 1:
                             Process Load at Dock 5.WIP=Process Load at Dock 5.WIP-
1:NEXT(46$);
;
;
;     Model statements for module:  AdvancedProcess.Signal 6 (Signal  Discharge 5)
;
46$           SIGNAL:        5:NEXT(43$);
```

Appendix 2

II. Uniform Distribution

```
               Distribution Summary                              Data Summary

    Distribution: Uniform                           Number of Data Points      = 26
    Expression:   UNIF(9.5, 22.5)                   Min Data Value             = 10
    Square Error: 0.056213                          Max Data Value             = 22
    Chi Square Test                                 Sample Mean                = 16.5
      Number of intervals      = 4                  Sample Std Dev             = 4.75
      Degrees of freedom       = 3
      Test Statistic           = 4.63
      Corresponding p-value    = 0.214
```

```
Histogram Summary

Histogram Range     = 9,5 to 22.5 ( 10,22 )
Number of Intervals = 13
==========================================
Int.   No. of             Probability          Cumulative
No.    Data Pts.   x      Density              Distribution
-----------------------------------------------------------------
-
                           Data   Function       Data   Function

  0      6       10.0     0.231     0.0769       0.231     0.0769
  1      1       11.0    0.0385     0.0769       0.269      0.154
  2      1       12.0    0.0385     0.0769       0.308      0.231
  3      1       13.0    0.0385     0.0769       0.346      0.308
  4      1       14.0    0.0385     0.0769       0.385      0.385
  5      0       15.0     0.000     0.0769       0.385      0.462
  6      1       16.0    0.0385     0.0769       0.423      0.538
  7      1       17.0    0.0385     0.0769       0.462      0.615
  8      3       18.0     0.115     0.0769       0.577      0.692
  9      1       19.0    0.0385     0.0769       0.615      0.769
 10      3       20.0     0.115     0.0769       0.731      0.846
 11      2       21.0    0.0769     0.0769       0.808      0.923
 12      5       22.0     0.192     0.0769        1.00       1.00
```

Appendix 3

Half Width Table

	1 -α/2				
degrees of freedom	0.900	0.950	0.975	0.990	0.995
1	3.078	6.314	12.706	31.821	63.656
2	1.886	2.920	4.303	6.965	9.925
3	1.638	2.353	3.182	4.541	5.841
4	1.533	2.132	2.776	3.747	4.604
5	1.476	2.015	2.571	3.365	4.032
6	1.440	1.943	2.447	3.143	3.707
7	1.415	1.895	2.365	2.998	3.499
8	1.397	1.860	3.206	2.896	3.355
9	1.383	1.833	2.262	2.821	3.250
10	1.372	1.812	2.228	2.764	3.169
11	1.363	1.796	2.201	2.718	3.106
12	1.356	1.782	2.179	2.681	3.055
13	1.350	1.771	2.160	2.650	3.012
14	1.345	1.761	2.145	2.624	2.977
15	1.341	1.763	2.131	2.602	2.947
16	1.336	1.746	2.120	2.583	2.921
17	1.333	1.740	2.110	2.567	2.898
18	1.330	1.734	2.101	2.552	2.878
19	1.328	1.729	2.093	2.539	2.861
20	1.325	1.725	2.086	2.528	2.845
25	1.316	1.708	2.060	2.485	2.787
30	1.310	1.697	2.042	2.457	2.750