

**Development and Data Extraction for  
Virtual Manufacturing System**

**THESIS REPORT**

**Proposed as one of the requirements to Obtain a Bachelor Degree in  
Industrial Engineering**



Created by:

Name : Afrilia Dwi R

Student number : 07522129

**INTERNATIONAL PROGRAM  
DEPARTMENT OF INDUSTRIAL ENGINEERING  
FACULTY OF INDUSTRIAL TECHNOLOGY  
UNIVERSITAS ISLAM INDONESIA  
YOGYAKARTA**

**2012**

## DEDICATION PAGE

*This thesis dedicated to Allah.SWT*

*My parents, my brother and my lil' sisters and*

*Someone I love, you're my META*



## MOTTO

Because after a misery there will be an ease. After a misery there will be an ease.

QS. Al-Insyirah: 5-6

Anyone who has never made a mistake has never tried anything new.

Albert Einstein

If you do a mistake, do focus on it, after that you'll know what is the problem of your mistake.

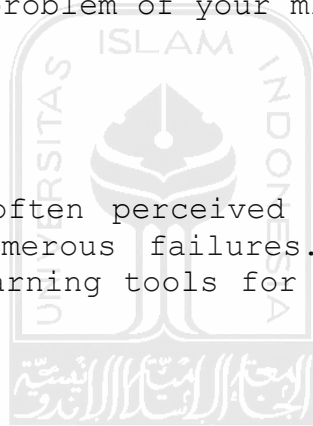
Mario Teguh

While Toyota is often perceived as a success story, we have also had numerous failures. We have used these experiences as learning tools for our people. That's how I was trained.

Aiko Toyoda

To do what you believe is right, to do what you believe is good, and doing these things right then and in that way is a calling from on high. Thus, do it boldly, do as you believe, do as you are.

Eiji Toyoda quoted by Matsubara



## ACKNOWLEDGMENTS

*Assalamu'alaikum Wr. Wb.*

Praises and thanks to Allah SWT, the Lord of hosts who have bestowed grace. There is no god has the right to be worshiped except Him, Allah SWT. Shalawat and Salam to our prophet Muhammad SAW, who have brought all Muslims in this world from the darkness to the brightness like today

Alhamdulillah, this thesis entitled “Development and Data Extraction of Virtual Manufacturing System” is finished. Basically the preparation of this thesis as a prerequisite to achieve Bachelor Degree in the Department of Industrial Engineering, International Program, Faculty of Industrial Technology, Universitas Islam Indonesia.

Step by step in making this thesis has been held, both in conducting research and writing the final report. Surely the author recognize in the writing of this thesis cannot be separated from various parties who have been helped. Therefore, on this occasion the author wants to say thank you as much as possible to:

1. Dean of Faculty of Industrial Technology Ir. Gumbolo Hadi Susanto, MSc.
2. Head of Industrial Engineering Department, Drs. H. M. Ibnu Mastur, MSIE.
3. Muhammad Ridwan Andi Purnomo, ST., M.Sc., Ph.D. as supervisor, who has patiently provided guidance, inspiration and suggestions during the preparation of this thesis. You are so deserving. You are the best motivator.
4. Both my parents are very dear to me, who have given everything I need, love, your prayers so that I could finish this thesis well. Hopefully I can always keep you happy. I love you.
5. My brother, lil' sisters and my *Metta*, your presence has fueled enthusiasm for completing this thesis. My adventurous friends, Lulut, Bagor and Gembrot, which has provided support and enthusiasm
6. Assistance of SIMAN Laboratory, thanks to your questions “kapan lulus?” “kapan pendadaran?” made me have more motivation in completing this thesis
7. All IP 2007, my classmate from the first semester studying at this campus

# TABLE OF CONTENTS

COVER.....	i
DEDICATION LETTER .....	ii
APPROVAL PAGE OF SUPERVISOR .....	iii
APPROVAL PAGE OF EXAMINER COMMITTEE .....	iv
DEDICATION PAGE .....	v
MOTTO .....	vi
ACKNOWLEDGMENT.....	vii
ABSTRACT.....	ix
TABLE OF CONTENTS.....	x
LIST OF FIGURES .....	xii
LIST OF TABLES .....	xiv
LIST OF SYMBOLS .....	xv
CHAPTER I INTRODUCTION	
1.1 Background .....	1
1.2 Problem Formulations.....	3
1.3 Problem Boundaries.....	3
1.4 Research Purposes .....	4
1.5 Research Benefits.....	4
1.6 Research Outline .....	4
CHAPTER II LITERATURES REVIEW	
2.1 Basic Theories.....	6
2.1.1 Virtual Manufacturing.....	6
2.1.2 Virtual Reality.....	9
2.1.3 Virtual Environment .....	12
2.1.4 Robot Arm .....	13
2.1.5 Object-Oriented Programming (OOP).....	18
2.1.6 AutoCAD .....	21
2.1.7 3D Studio Max .....	22
2.1.8 TrueVision3D (TV3D).....	22
2.2 Previous Researches.....	24

## CHAPTER III RESEARCH METODOLOGY

3.1	Research Objects.....	28
3.2	System Development .....	28
3.3	Data Collection .....	29
3.3.1	Data Requirement .....	29
3.3.2	Data Collection Method.....	30
3.4	Analysis Tool .....	30
3.5	Research Flowchart.....	30

## CHAPTER IV SYSTEM DEVELOPMENT

4.1	3-Dimensional Object Modeling.....	34
4.2	Determining the Data .....	44
4.4.1	Data Object Position .....	44
4.4.2	Data Object Movement.....	47
3.3.1	Data Joints.....	52
4.3	System Development .....	52
4.4	Data Extraction .....	56

## CHAPTER V DISCUSSION

5.1	Testing a System Built.....	60
-----	-----------------------------	----

## CHAPTER VI CONCLUSION AND RECOMMENDATION

6.1	Conclusions.....	63
6.2	Recommendations.....	64

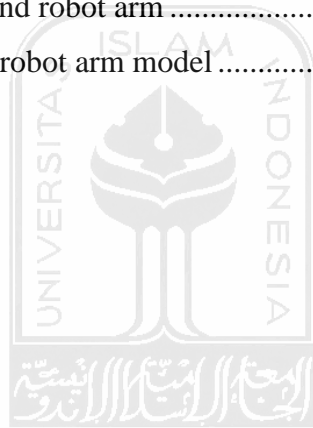
## REFERENCES

## APPENDICES

## LIST OF FIGURES

Figure 2.1 Structure of industrial robot system (Courtesy of Cincinnati Milacron)....	15
Figure 2.2 Irregular Triangular .....	16
Figure 2.3 Several types of joints used in robots .....	18
Figure 2.4 Example of Class .....	19
Figure 3.1 Research Flowchart .....	31
Figure 4.1 Waist.....	35
Figure 4.2 Shoulder.....	35
Figure 4.3 Upper Arm.....	36
Figure 4.4 Fore Arm .....	36
Figure 4.5 Roll Arm.....	37
Figure 4.6 Pitch Hand .....	37
Figure 4.7 Roll Hand.....	38
Figure 4.8 Tool.....	38
Figure 4.9 Robot arm KUKA 15.....	39
Figure 4.10 Joints of Robot arm KUKA 15 .....	39
Figure 4.11 Robot arm in left view .....	40
Figure 4.12 Robot arm's reach distance.....	41
Figure 4.13 Conveyor .....	42
Figure 4.14 NC Drilling .....	42
Figure 4.15 Storage .....	43
Figure 4.16 Work piece.....	43
Figure 4.17 Finish product.....	43
Figure 4.18 Robot arm position .....	44
Figure 4.19 Conveyor position .....	45
Figure 4.20 NC Drilling position .....	45
Figure 4.21 Storage position .....	46
Figure 4.22 Work piece position.....	46
Figure 4.23 Finish product position .....	47
Figure 4.24 Initial position for work piece inside the storage .....	48
Figure 4.25 Work pieces above the first conveyor .....	48
Figure 4.26 Work pieces above the other edge of the first conveyor .....	49

Figure 4.27 Work pieces in NC drilling.....	49
Figure 4.28 First position of finish product in NC drilling.....	50
Figure 4.29 Finish products above the second conveyor .....	50
Figure 4.30 Finish products above the other edge of the second conveyor .....	51
Figure 4.31 Finish products inside the second storage .....	51
Figure 4.32 System Development.....	52
Figure 4.33 <i>Pseudocode</i> load object .....	54
Figure 4.34 Virtual object in VE.....	54
Figure 4.35 <i>Pseudocode</i> to create a room .....	55
Figure 4.36 Virtual Room .....	55
Figure 4.37 <i>Pseudocode</i> to get the matrix.....	56
Figure 4.38 Matrix of first robot arm.....	58
Figure 4.39 Matrix of second robot arm.....	59
Figure 5.1 Example of the robot arm model .....	61





## LIST OF TABLES

Table 2.1 VR benefits in manufacturing applications.....	11
Table 2.2 Review some of previous researches compare to under research.....	27
Table 5.1 Example data joint's length of robot arm .....	60
Table 5.2 Value of $\Theta_A$ and $\Theta_B$ .....	62



## LIST OF SYMBOLS

$L_a$  = Length of a

$L_b$  = Length of b

$L_c$  = Length of c

$TCP$  = Tool Center Point

$WCP$  = Wrist Center Point

$ECP$  = Elbow Center Point

$SCP$  = Shoulder Center Point

$hWCP$  = height of  $WCP$

$r$  = radius



## ABSTRACT

*Virtual manufacturing is knowledge to create an artificial manufacturing consists of virtual environment and virtual models that able to be simulated in accordance with the real system. It is usually used in product design and process design where it's more costless and timeless. This research is to develop virtual manufacturing system in process design including 3D object and data extraction. In developing the 3D object use solid model with 3Ds MAX as software design, and TrueVision3D as the engine with C# as the programming language. The result of this research are 3D object that can be modeled with refer to the real object, virtual manufacturing can be established well, data extraction by using cosines law can be carried out and matrix data on 3D object position can be retrieved to be process further as input data of control system.*

*Keywords: Virtual Manufacturing, Virtual Environment, Object Modeling, Data Extraction*



Virtual manufacturing is knowledge to create an artificial manufacturing consists of virtual environment and virtual models that able to be simulated in accordance with the real system. It is usually used in product design and process design where it's more costless and timeless. This research is to develop virtual manufacturing system in process design including 3D object and data extraction. In developing the 3D object use solid model with 3Ds MAX as software design, and TrueVision3D as the engine with C# as the programming language. The result of this research are 3D object that can be modeled with refer to the real object, virtual manufacturing can be established well, data extraction by using cosines law can be carried out and matrix data on 3D object position can be retrieved to be process further as input data of control system.

Keywords: Virtual Manufacturing, Virtual Environment, Object Modeling, Data Extraction

# CHAPTER I

## INTRODUCTION

### 1.1 Background

The manufacturing sector has become increasingly competitive as markets have become more globalized (Slomp et al., 2005). The globalization of industry and the worldwide competitive economy are forcing leading manufacturing organizations to acquire the ability to develop and produce virtually defect-free products quickly, in response to the opportunities and demands of the changing world market (Chen et al., 2002), and to integrate engineering and production activity. According to Lee et al. (2001), to achieve the integration, a model that covers all the engineering functions, the information flow and the precise characteristics of manufacturing system is needed. This condition led to any manufacturing industry attempt to enhance all levels of decision and control in order to save their time, energy, money and process. In decision and control processes usually carried out a simulation beforehand so that it does not take a long time to get the results. And of course this simulation can be repeated several times until get the desired result with relatively shorter time than directly applied to the real system. One of the concepts that can enhance all levels of decision and control is Virtual Manufacturing (VM).

Generally, VM is one of tools to enhance human ability to predict potential problems and inefficiencies in product functionality and manufacturability before real manufacturing process (Banerjee and Zetu, 2001), so it can assist in decision and

control processes. In addition, according to Banerjee and Zetu (2001), there some areas that can take advantage from development of VM including Product Design, Hazardous Operations Modeling, Production Modeling, Process Modeling, Training, Education, Information Visualization, and Telecommunications and Teletravel.

Lee et al (2001) investigated about the development of VM techniques that can be applied to different levels and activities of manufacturing enterprise from basic production activities, materials processing, plant and facilities layout to higher-order activities of co-ordination and control strategy. The successful launching of a VM system requires multidisciplinary know-how that cuts across a wide spectrum of disciplines in computer and engineering sciences and is more than the traditional simulation of a particular process and operation. However, a sound physical understanding of the process and operation to be simulated is essential and becomes an integral part of any successful VM system to be useful.

Since VM has ability to make rapid changes in products and processes, can minimized production costs and time to market more quickly, but VM still has a few of lack, that are communications efficiency and poor in control. A method to overcome the problem will be a new contribution in this study field. In developing a VM System, the author grouped into four stages of development. The first phase is election and formation of the main models used include extraction of required data, the second stage is develop control algorithms for object movement, the third stages is design the production scheduling, and the last stages is design the production cost.

In this research only perform the first stage of development that is election and formation of the main model and data extraction for that system. The build system consists of a manufacturing environment model, and object model.

## 1.2 Problem Formulations

Based on the background above, the problem can be formulated as follows:

- a. How to develop a VM System including object modeling?
- b. How to extract data for VM System?

## 1.3 Problem Boundaries

To focus the research in achieving the objective, the research needs to be limited in certain aspects. These problem boundaries can be defined as follows:

- a. The system is not discussed about inventory, either raw material or finish product.
- b. The system is not discussed about reject product.
- c. The system is not discussed about detailed information of the object that being processed.
- d. The system is not discussed about cost.
- e. 3-dimensional models were used instead of the original prototype.

## 1.4 Research Purposes

The main objective of this research is to build a VM System and able to extract the data needed, so it'll create a VM System in accordance with the instructions given by user.

## 1.5 Research Benefits

From this research are expected to provide benefits for:

- a. Contribution to scholarly concerns on VM
- b. As a reference in the manufacture and development of VM System

## 1.6 Research Outlines

This thesis is organized systematically into several chapters, with the title of each chapter as follows:

### **CHAPTER II LITERATURE REVIEW**

Contain concepts and basic principles needed to solve research problems, theoretical basis that support a study to be conducted. It also includes a description of the previous research that has correlation with the research conducted.

### **CHAPTER III RESEARCH METHODOLOGY**

This chapter provides information about the research object, model development, and also describes the outline and



flowchart of the research, carried out method, computer program are built, research material used, tools, research procedures and data that will be reviewed as well as the way of analysis used and in accordance with a flowchart have been made.

#### **CHAPTER IV COLLECTING AND PROCESSING DATA**

This chapter describes about how to collect and process the data, analysis and result, include pictures and graphs are obtained. In this chapter is a reference for the discussion of result that will be written in chapter V of the discussion.

#### **CHAPTER V DISCUSSION**

This chapter discuss about the result obtained during the research and compliance with the research objectives so as to generate recommendations.

#### **CHAPTER VI CONCLUSION AND RECOMMANDATION**

This chapter discuss about the conclusion of the analysis made and as well as advice on the result achieved were found during the research so necessary recommendations for further research.p

## CHAPTER II

### LITERATURES REVIEW

Literatures review written in this research are excerpts taken to provide understanding and strengthening of the purpose of writing of the conducted research. The contents of this literature review are the development of research related to the scope of the topic of conducted research and the basic theories that assist in this preparation and completion of this research.

#### 2.1 Basic Theories

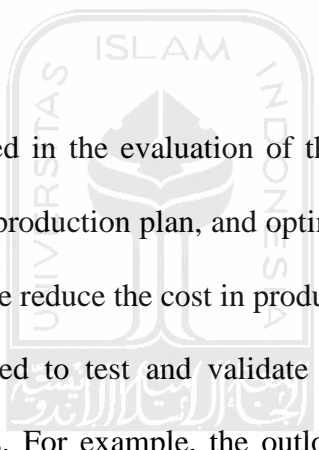
##### 2.1.1 Virtual Manufacturing

The VM concept was first developed at National Bureau of Standards to address specific control problems encountered in the design phase of the automated manufacturing of small batches of machined parts (McLean et al., 1982). It is kind of knowledge and computer-based system technology that integrates manufacturing activities dealing with models and simulation instead of objects and their operations in the real world (Lee et al., 1997, 2001). It is also capable of generating information about the structure, states, and behavior of a manufacturing system as can be observed in a real manufacturing environment (Iwata et al., 1997). In other word, a VM system produces no output such as materials and physical products, but it can produce information about them. The traditional use of VM is for production system design and development (which, from a machine builder perspectives, is similar to product



development), but more advanced applications are support for other life-cycle phases such as operational planning (Ingemansson and Oscarsson, 2006; Solding et al., 2003), service and maintenance (Sundberg et al., 2006), and decommissioning (Klar, 2002).

Lee (2001) tells about the modeling and simulation technologies in VM enhance the production flexibility and reduce the fixed cost since no physical conversion of materials to products is involved. A part from these, VM can be used to reliably predict the business risks and this will support the management in decision making and strategic management of an enterprise. Some typical applications of VM are as follows:

- 
- a. VM can be used in the evaluation of the feasibility of a product design, validation of a production plan, and optimization of the product design and processes. These reduce the cost in product life cycle.
  - b. VM can be used to test and validate the accuracy of the product and process designs. For example, the outlook of a product design, dynamic characteristics analysis, checking for the tool path during machining process, NC program validation, checking for the collision problems in machining and assembly, etc.
  - c. With the use of VM on the internet, it is possible to conduct training under a distributed virtual environment for the operators, technicians and management people on the use of manufacturing facilities. The costs of training and production can thus be reduced.
  - d. As a knowledge acquisition vehicle, VM can be used to acquire continuously the manufacturing know-how, traditional manufacturing

processes, production data, etc. this can help to upgrade the level of intelligence of a manufacturing system.

The benefits derived from VM are as follows:

a. Enhancing the capability of risk measures and control

VM can be used to predict the cost of product development and production as well as provide the information related to the production processes and the process capability. The information is useful for improving the accuracy of the decision made by the designer and the management. The problems in product development and manufacturing processes can also be predicted and resolved prior to the actual production.

b. Shrinking the product development cycle

VM will allow more computer-based product models to be developed and prototyped upstream in the product development process. This will reduce the need for the number of downstream physical prototyped traditionally made to validate the product models and new designs. Thus, the company can reduce its product development time.

c. Enhancing the competitive edge of an enterprise in the market

VM can reduce the cycle time and costs in product development. With the virtual environment provided by VM, the customers can take part in the product development process. The design engineers can response more quickly to the customer queries and hence provide the optimal solution to the customers. The competitive edge of an enterprise in the market can thus be enhanced.

To develop VM technology effectively it is essential that an interface layer be created between virtual reality (VR) software and manufacturing software. Current examples of VR software are based on hardware specifications such as 3D Studio® by Autodesk, Open Inventor™ by SGI, etc. which is geared toward a room-sized VR implementation driven by multiple SGI processor (Banerjee and Zetu, 2001).

### **2.1.2 Virtual Reality**

The idea of creating synthetic interactive environments has been around for years. The first head-mounted display was demonstrated in the 1960s (Sutherland, 1965, 1968) and, in the mid-1970s, Myron Krueger performed the first experiments on what he defined 'artificial reality'. The term VR is credited to Jaron Lanier, who coined it in the early 1990s. Computer scientists create a virtual environment using an interactive visual display that makes the users feel as if they are immersed in a synthetic space (Ellis, 1994; Rosenblum et al., 1995). VR is a rapidly developing computer interface that strives to immerse the user completely within an experimental simulation, thereby greatly enhancing the overall impact and providing a much more intuitive link between the computer and the human participants (Mujber et al., 2004). VR can be broadly defined as the ability to create and interact in cyberspace, i.e. a space that represents an environment which has a lot of similarity to the environment around us. VR is closely associated with an environment commonly known as virtual environment (VE) (Banerjee and Zetu, 2001).

The advances of VR technology in the last decade have provided the impetus for applying VR to different engineering applications such as product design, modeling, shop floor controls, process simulation, manufacturing planning, training,

testing and verification. VR holds great potential in manufacturing applications to solve problems before being employed in practical manufacturing thereby preventing costly mistakes. VR not only provides an environment for visualization in the three-dimensional environment but also to interact with the objects to improve decision making from both qualitative and quantitative perspectives (Tiruvannamalai, 2002).

VR applications in manufacturing have been classified into three groups; operations management, manufacturing processes, and design. VR may play very significant rule in design a new product. VR technology has been applied into two different applications in design; design and prototyping. Operations management has been classified into three categories; planning, simulation and training. And manufacturing processes has been classified into three different areas; machining, assembly, and inspection. The benefits of applying VR technology to those groups are shown in Table 1 (Mujber et al., 2004).

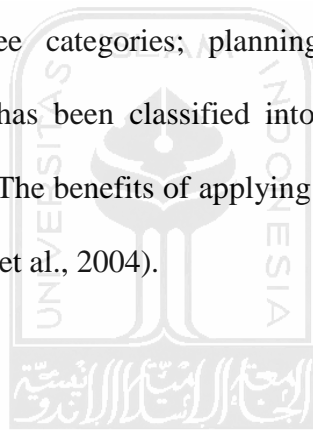


Table 2.1 VR benefits in manufacturing applications

Area	Groups	Benefits
Design	Design	To allow the whole design team to work together in the VE
		To improve visualization of the product by allowing the user to co-exist in the same environment with the product model
		To improved interaction with design in terms of more intuitive model manipulation and functional experimentation
	Prototyping	To reduce significantly the amount of hardware prototypes during conception, design, and evaluation of few products
		To provide a VE for innovating, testing and evaluating of specific characteristics of a candidate design
Operations management	Planning	To improve the understanding of the plans and to support interdisciplinary discussions
		To allow the users to interact and change the model during runtime
		To enable unskilled users to understand and participate in the planning process
		To support the technological as well as the economical modeling of diverse production planning scenarios
	Simulation	To convince the use the simulation tools
		To verify and validate a simulation model
		To enable the user to understand the results
		To provide a VE for communication of results
		To achieve the credibility for the simulation (Matthew, 2000)
	Training	To duplicates an entire manufacturing process to a VE to give trainers their own factory to learn in

		To provide a user with an environment to explore the outcomes of their decision without risk themselves or equipment
		To allow the employees to practices existing and new tasks in safe
Manufacturing processes	Machining	To evaluate the feasibility of a part design and the selection of processing equipment
		To allow the user to study the factors affecting the quality, machining time and costs based on modeling and simulation
	Assembly	To reduce design cycle time, re-design efforts, and design prototypes (S. Jayaram et al., 1997)
		To predict the quality of an assembly, product cycle and costs
		To address assembly and disassembly verification
	Inspection	To model and simulate the inspection process, and the physical and mechanical properties of the inspection equipment
To provide an environment for studying the inspection methodologies, collision detection, inspection plan, factors affecting the accuracy of the inspection process, etc. (Lee et al., 2001)		

### 2.1.3 Virtual Environment

There is an increase in use of 3D VE in architecture, engineering, and construction (AEC) industries. Applications like virtual training environments, virtual prototyping of designs, and joint virtual development of information systems require a valid representation of the real environment. Often, the real environments are industrial facilities such as oil rigs, container terminals, or manufacturing plants (Fumarola and Poelman, 2011). A commonly used definition for VE is an interactive, virtual image



display enhanced by special processing and by non-visual display modalities, such as auditory and haptic, to convince users that they are immersed in a synthetic space. The term immersion refers to the fact that the user gets the feeling that he or she is immersed in the computer environment, the screen separating the user and the computer appears non-existent to the user (Banerjee and Zetu, 2001).

In VE, the information entered must be in accordance with reality, like transferring information about the real objects such as their shapes, sizes, motions, and materials. In order what is made is accordance with the original.

There is no single reason why 3D VE are increasingly popular, but drivers stem from multiple backgrounds. Advantages of 3D VE are found in improving communication (Arayici and Aouad, 2004), increasing insight (Shiratuddin and Thabet, 2007; Woksepp and Olofsson, 2006), supporting collaboration (Bouchlaghem et al., 2004), and supporting decision making (Kam and Fischer, 2004).

#### **2.1.4 Robot Arm**

Industrial robots are beginning now to revolutionize industry. These robots do not look or behave like human beings, but they do the work of humans. Robot are particularly useful in a wide variety of applications, such as material handling, spray painting, spot welding, arc welding, inspection and assembly (Koren, 1983).

There are so many opinions about the definition of industrial robot. According to Koren (1983) definition of the industrial robot is a programmable mechanical manipulator, capable of moving along several directions, equipped at its end with a device called the end-effector, and performs factory work ordinarily done by human

beings. The term robot is used for a manipulator that has a built-in control system and capable of stand-alone operation. The Robot Institute of America (RIA) defines the industrial robot as a reprogrammable multi-functional manipulator designed to move material, parts, tools, or other specialized device through variable programmed motions for the performance of a variety of tasks. The British Robot Association (BRA) has defined the industrial robot as a reprogrammable device with a minimum of four degrees of freedom designed to both manipulate and transport parts, tools or specialized manufacturing implementation through variable programmed motions for the performance of the specific manufacturing task (Nagy and Siegler, 1987). By this definition, however, a washing machine is also a robot; the wash and rinse cycles are programmable and the machine moves material on rotary motions. There for, a definition of an industrial robot must include the following key words: programmable manipulator, end-effector, factory work, stand-alone operation (Koren, 1983), and it must be able to respond to stimuli based on information received from the environment, no matter how restricted that environment may be. The robot will interpret the response either blindly, or by active sensing to bring about all the changes required in its environment (Nagy and Siegler, 1987).

According to Koren (1983) the first industrial robot were installed to replace people in dangerous operations, such as loading and unloading hot parts from processing furnaces, or in hazardous environments, for example, when workers were subjected to long exposures of toxic materials. Today, however, industrial robots are primarily installed to improve productivity in manufacturing operations. The main applications of industrial robots are handling and loading of parts, spray painting, spot

and arc welding, assembly, machining operation such as deburring and grinding, and inspection.

In general, the structure of robot manipulator is composed of a main frame and a wrist at its end. The main frame is frequently denoted as the arm, and the most distal group of joints affecting the motion of the end-effector is referred to collectively as the wrist. This typical structure is shown in Fig. 2-1. The end-effector can be welding head, a spray gun, a machining tool, or a gripper containing on-off jaws, depending upon the specific application of the robot. (Koren, 1983).

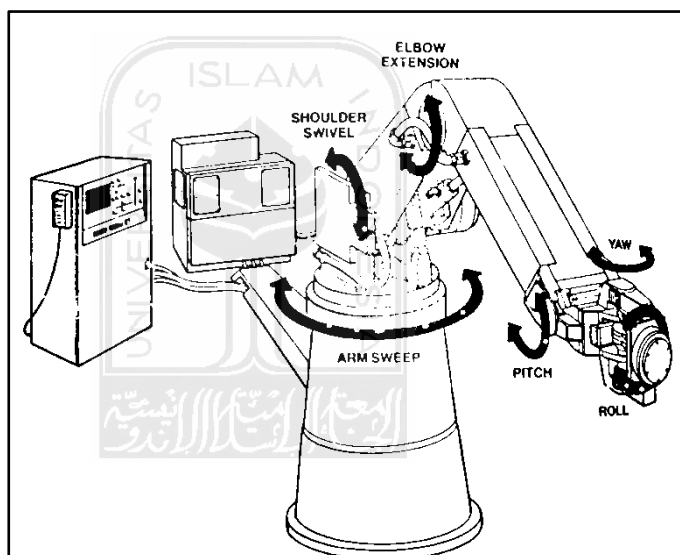


Figure 2.1 Structure of industrial robot system (Courtesy of Cincinnati Milacron)

Degree of freedom is mechanical point on a robot, except in the gripper or tool, at which some form of drive induces motion in a robot parts. The motion can be of a pivoting nature or a reciprocal motion as is produced by a pneumatic or hydraulic cylinder (Asfahl, 1988). Basically the robot needs six axes of motion (or degree of freedom) to reach a point with specific orientation in the space. A different orientation might completely change the position of the robot arm (Koren, 1983). Those six

degree of freedom are Base rotation, Shoulder flex, Elbow flex, Wrist flex, Wrist yaw and Wrist roll.

Although there are exceptions, in most robots the degrees of freedom are in series. According to Asfahl (1988) thus the first degree of freedom in the robot is base rotation imparts motion to all of parts of the robot affected by subsequent degrees of freedom. Conversely, the third degree of freedom (elbow flex), for example, has no effect upon the base movement. It follows that the most sophisticated motion in the entire robot is that of the member driven by the highest degree of freedom. Generally, the robot with the most degrees of freedom can produce the most complex movement, but there are other important factors to consider, such as range and quality of motion within a given degree of freedom.

To find the angle for each joint of the robot arm can use Cosines Law, that are:

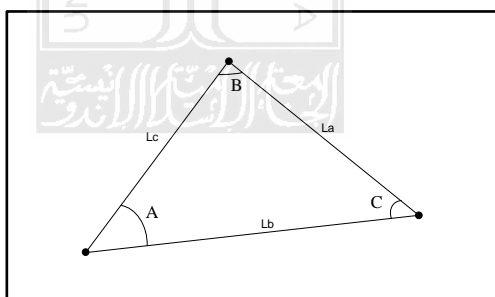


Figure 2.2 Irregular Triangular

$$\cos A = (L_b^2 + L_c^2 - L_a^2) / 2L_b.L_c \dots\dots\dots(1)$$

$$\cos B = (L_a^2 + L_c^2 - L_b^2) / 2L_a.L_c \dots\dots\dots(2)$$

$$\cos C = (L_a^2 + L_b^2 - L_c^2) / 2L_a.L_b \dots\dots\dots(3)$$

Where:  $L_a$  = Length of  $a$

$L_b$  = Length of  $b$

$L_c$  = Length of  $c$

The robot's movements can be divided into two general categories: arm and body motions, and wrist motions. The robot's motions are accomplished by means of powered joints. Three joints are normally associated with the action of the arm and body, and two or three joints are generally used to actuate the wrist. Connecting the various manipulator joints together are rigid members that are called links. In any link-joint-link chain the link that is closest to the base in the chain the input link. The output link is the one that moves with respect to the input link. The joints used in the design of industrial robots typically involve a relative motion of the adjoining links that is either linear or rotational. Linear joints involve a sliding or translational motion of the connecting links. This motion can be achieved in a number of ways (e.g., by a piston, a telescoping mechanism, and relative motion along a linear track or rail). There are at least three types of rotating joint that can be distinguished in robot manipulators. First is rotational joint. In the rotational joint the axis of rotation is perpendicular to the axes of the two connecting links. The second type of rotating joint involves a twisting motion between the input and output links. The axis of rotation of the twisting joint is parallel to the axes of both links. The second type named twisting joint. The third type of rotating joint is a revolving joint which the input link is parallel to the axis of rotation and the output links is perpendicular to the axis of rotation. In essence, the output link revolves about the input link, as if it were in orbit. Below is an illustration of the types of joint (Groover et al., 1986).

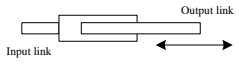
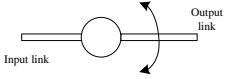
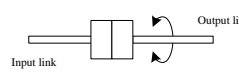
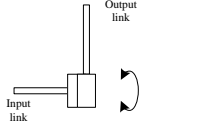
Type	Name	Illustration
L	Linear	
R	Rotational	
T	Twisting	
V	Revolving	

Figure 2.3 Several types of joints used in robots: (a) rotational joint with rotation along an axis perpendicular to arm member axes, (b) rotational joint with twisting action, (c) linear motion joint, usually achieved by a sliding action.

### 2.1.5 Object-Oriented Programming (OOP)

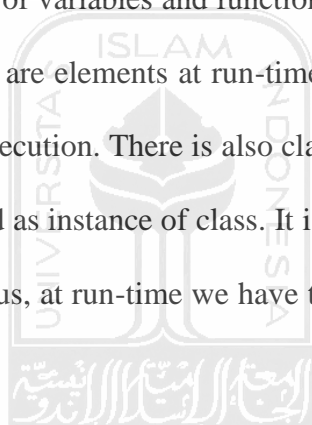
OOP is a programming paradigm that is dominant today, because it can provide the solution principles of modern programming. However, it does not mean that procedural programming is no longer feasible.

OOP was created because of the perceived persistence of the limitations in traditional programming languages. The concept of OOP is, all problem solving is divided into the object. In OOP data and functions that will operate combined into single unit which can be referred to as objects. The process of designing a programming is a process that is not separated from the process that precedes namely the analysis and the process that follows. The discussion about object orientation will not be separated from the concept of objects such as inheritance or decrease, wrap or

encapsulation, and polymorphism. These concepts are fundamental in the orientation of the object that needs to be completely understood and used properly, and avoiding inappropriate use.

In OOP's environment, the programmer defines a static class. At run-time, class will be the object of the instant. There is also the object of which is an instance of a class can always be referenced through the current object, whatever the instant.

Can be defined that class is a data structure of an object, or it is a basic form or blueprint that defines the method of variables common to all objects of some sort. The object itself is a collection of variables and functions resulting from specific templates or so-called class. Objects are elements at run-time that will be created, manipulated, and destroyed when the execution. There is also class is a static definition of the set of objects that may be created as instance of class. It is simply a collection of objects that have similar attributes. Thus, at run-time we have the object. Class is usually made as shown in Figure 2.3



```

Class name_theclass {
//contains a set if functions, variables and other statements
}
In C++
ClassName class {
Public:
Int attribute1;
Int attribute2;
Int operation () {return score_int}
};

```

Figure 2.4 Example of Class

Inheritance is a sentence that must exist in the OOP also called reusable. When using the method of return or replace an existing class, as well as adding new fields and methods instant, then at that moment we are working with inheritance. This concepts is fundamental concept in object orientation and should be used properly, there are several kinds of inheritance in OOP that are single inheritance and multiple inheritance (Hariyadi, 2009).

The first step in OOP is to identify all the objects you want to manipulate and how they relate to each other, an exercise often known as data modeling. Once you've identified an object, you generalize it as a class of objects (think of Plato's concept of the "ideal" chair that stands for all chairs) and define the kind of data it contains and any logic sequences that can manipulate it. Each distinct logic sequence is known as a method. A real instance of a class is called (no surprise here) an "object" or, in some environments, an "instance of a class." The object or class instance is what you run in the computer. Its methods provide computer instructions and the class object characteristics provide relevant data. You communicate with objects - and they communicate with each other - with well-defined interfaces called *messages*.

The concepts and rules used in object-oriented programming provide these important benefits:

- a. The concept of a data class makes it possible to define subclasses of data objects that share some or all of the main class characteristics. Called inheritance, this property of OOP forces a more thorough data analysis, reduces development time, and ensures more accurate coding
- b. Since a class defines only the data it needs to be concerned with, when an instance of that class (an object) is run, the code will not be able to

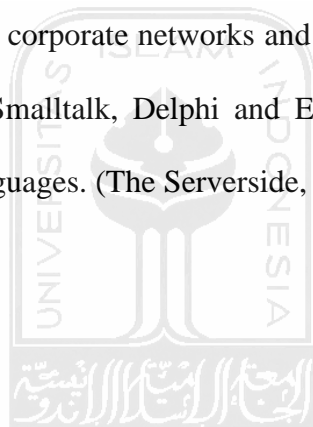


accidentally access other program data. This characteristic of data hiding provides greater system security and avoids unintended data corruption.

- c. The definition of a class is reusable not only by the program for which it is initially created but also by other object-oriented programs (and, for this reason, can be more easily distributed for use in networks).
- d. The concept of data classes allows a programmer to create any new data type that is not already defined in the language itself.

Java, Python, C++, Visual Basic .NET and Ruby are the most popular OOP languages today. The Java programming language is designed especially for use in distributed applications on corporate networks and the Internet. Ruby is used in many Web applications. Curl, Smalltalk, Delphi and Eiffel are also examples of object-oriented programming languages. (The Serverside, 2008)

### 2.1.6 AutoCAD



AutoCAD is a software application for computer-aided design (CAD) and drafting in both 2D and 3D. It is developed and sold by Autodesk, Inc. First released in December 1982, AutoCAD was one of the first CAD programs to run on personal computers, notably the IBM PC. At that time, most other CAD programs ran on mainframe computers or mini-computers that were connected to a graphics computer terminal for each user (Wikipedia, 2011).

### 2.1.7 3D Studio Max

Autodesk 3ds Max, formerly 3D Studio MAX is 3D computer graphics software for making 3D animations. It provides powerful, integrated 3D modeling, animation, rendering, and compositing tools that enable artists and designers to more quickly rump up for production. It was developed and produced by Autodesk Media and Entertainment. It has modeling capabilities, a flexible plugin architecture and can be used on the Microsoft Windows platform. It is frequently used by video game developers, TV commercial studios and architectural visualization studios. It is also used for movie effects and movie pre-visualization (Wikipedia, 2009).

### 2.1.8 TrueVision3D (TV3D)

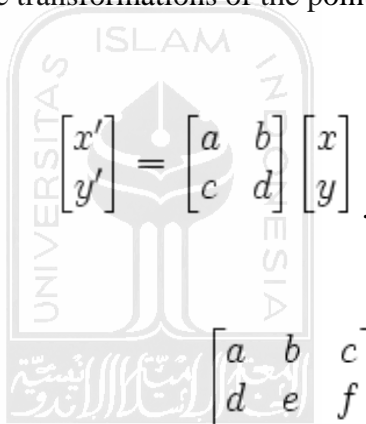
TV3D is a commercial 3D engine first created by Sylvain Dupont in 1999. TV3D engine is written in Visual Basic 6 and C++ and layered on top of the Microsoft DirectX API, currently supporting DirectX version 8. The engine is accessible from a number of programming languages including C++, C#, Delphi and Visual Basic (6 and .NET) (Wikipedia, 2007). It is a complete 3D middleware solution for programmers looking to create anything from next generation games to complex simulations with a shorter amount of time (TrueVision).

TV3D used to develop VE by using many kinds of function, such as function to transfer the object that is LoadXFile, create cylinder with CreateCylinder, lighting with CreateDirectionalLight, call the matrix with GetMatrix, etc. in TV3D there are some matrix that usually used to control the object movement, and one of them is

transformation matrix. This transformation matrix including rotation, scaling and translation

Transformation matrix is a special matrix that can describe 2d and 3d transformations. Transformations are frequently used in linear algebra and computer graphics, since transformations can be easily represented, combine and computed. Transformation matrix can evaluate the transformation that would be performed by multiplying the original array of points. For example in 2d, suppose the transformation

matrix is  $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$ , then the transformations of the points  $\begin{bmatrix} x \\ y \end{bmatrix}$  would be:



$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} .$$

Similarly, to perform 3d transformation  $\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$  on the points  $\begin{bmatrix} x \\ y \\ z \end{bmatrix}$  and it would use

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} .$$

Below is the functional of scaling and rotation:

- a. For scaling  $x' = s_x \cdot x$ ,  $y' = s_y \cdot y$  and  $z' = s_z \cdot z$ . Then the matrix
- $$\begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{bmatrix}$$

form is:

- b. For rotation, there are three different sets of rotation in the 3dimensional transformation matrix, one for each axis that can be rotated around.

X axis rotation: 
$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{bmatrix}$$

Y axis rotation: 
$$\begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix}$$

Z axis rotation: 
$$\begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$
 (Mathforum, 2011)

## 2.2 Previous Researches

Lee et al (2001) investigated about the development of VM techniques that can be applied to different levels and activities of manufacturing enterprise from basic production activities, materials processing, plant and facilities layout to higher-order activities of co-ordination and control strategy. With the use of the VM technologies, many aspects of the activities can be integrated and realized into one system. The manufacturing cost and time-to-market can be reduced.

Chen et al (2007) investigated about development VM for optimize the hybrid layered manufacturing technology. It describes about the fundamental attributes,

including the virtual prototype of a hybrid layered manufacturing facility, behavior models for its spraying and engraving operations, process planning, the model of the virtual manufactured component, and quality evaluation.

Park (2005) investigated about an object-oriented methodology (OOM) to create a virtual flexible manufacturing system (FMS) model that consist of four types of objects: the virtual device model (object model), the transfer handler model (functional model), the state manager model, and the flow controller model (dynamic model).

Slomp et al (2005) investigated about a new operational procedure of creating VM cells periodically, for instance every week, to meet the demand turbulence in markets. It defines a VM cells as a temporary grouping of machines and the minds of workers to families of jobs. This supports the minimization of setup time losses. It also promotes the novel idea of team production.

Chen et al (2002) investigated about the implementation of an Internet-based VM facility for the development and optimization of new manufacturing processes. This facility provides a client with a VM process optimization facility which combines advanced process simulation software with a generic remote machine monitoring system (RMMS).

Amos et al (2008) investigated about the use of VM for press line monitoring and diagnostics. The system offers remote on-line monitoring and diagnostics functions as well as media player-type functions such as replay that allow a service and maintenance expert to analyses disturbances that occur at remote locations.

Offodile et al (2002) investigated integrating information technology (IT) and manufacturing strategies using the VM paradigm. First is traced the advancements in IT-related technologies from the early times to the present day information superhighway and its impact on manufacturing. The developments and growth in IT have impacted manufacturing activities, especially as far as their capacity utilization, inventory turnover and quality, price reduction, market share and return on assets are concerned. Much of these gains has resulted from the use of computer-integrated manufacturing, holonic, and VM in large manufacturing enterprises (LMEs).

Kim et al(2006) investigated about a multichannel visualization module for virtual manufacturing. It creates the immersive VR module which can be related with VM system. The module has proposed the process to use the VM system as the graphic viewer of the immersive VR system in the cluster of PCs, because using the VM system as the graphic viewer for the VR system can avoid data translation process which may make flaws in the translated data. This skipping data translation also causes to reduce the preparation time and ensure the reliability.

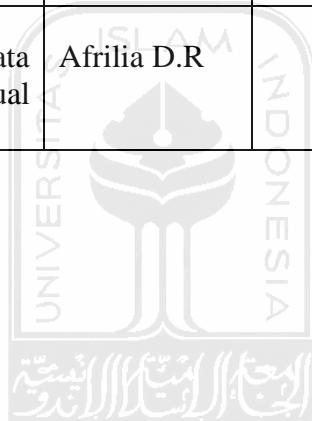
Laender et al (2002) investigated about data extraction by example (DEByE). This is an approach to extracting data from Web sources, based on a small set of examples specified by the user. The novelty is on the fact that the user specifies examples according to a structure of his liking and that this structure is described at example specification time. For the specification of the examples, the user interacts with a tool that developed which adopts nested tables as its visual paradigm.

Heathfield et al (1991) investigate about realizable benefits of using the object-oriented approach in the design and development of clinical decision support systems. These systems seek to build a computational model of some problem domain and

therefore tend to be exploratory in nature. Conventional procedural design techniques do not support either the process of model building or rapid prototyping.

Table 2.2 Review some of previous researches compare to under research.

<b>Research Title</b>	<b>Researcher</b>	<b>3D Object Modeling</b>	<b>Develop VE</b>	<b>Data Extraction</b>
Virtual manufacturing for press line monitoring and diagnostics	Amos H.C. Ng., et al	✓	✓	-
Integrated Virtual Manufacturing Systems for Process Optimization and Monitoring	Lienjing Chen., et al	✓	✓	-
Development and Data extraction of Virtual Manufacturing System	Afrilia D.R	✓	✓	✓



## CHAPTER III

### RESEARCH METHODOLOGY

In this section will discuss about research methodology includes the stages of conducting research that is used as a reference to conduct research in order to stay focus on the goals to be achieved.

#### 3.1 Research Objects

The object of this research is a mini virtual factory includes two virtual robot arms and two virtual conveyors which is used as material handling, besides that there is a virtual NC Drilling that is used to process work piece into virtual finished product.

#### 3.2 System Development

In this research have three steps to develop VM System that are:

1. Develop virtual model

This is the first step in the development of VM System, which is to create 3D model such as two robot arms, two conveyors, an NC Drilling, work piece and finished product.



## 2. Develop virtual environment

The second step is to create virtual environment, which is to make an interactive virtual room and transfer the virtual model to the virtual room, thus forming a mini virtual factory.

## 3. Determine the required data and extract data

The last step is to define the required data, that are data about the details of the model, the position of each object, and object movement which can be extracted and can be used to control the object movement.

### 3.3 Data Collection

#### 3.3.1 Data Requirement

In this research requires some data to support the program created, that are:

##### 1. Data detail model

This data is very important because as the basis for the development of a system, and very influential in determining the data position and data object movement.

##### 2. Data position object

This data is used to determine the position coordinates of each object to be placed within the system built and thus forming a mini virtual factory.

##### 3. Data object movement



This data is used to determine the coordinates of the object movement, from the initial coordinates to the desired coordinate.

### **3.3.2 Data Collection Method**

The method for collecting data is simply to use the Primary Data that is by direct observation in mini virtual factory that have been built.

### **3.4 Analysis Tool**

The analysis tool used to conduct this research is AutoCAD 2008 and 3D Max 2008 as software design to create 3-dimensional object needed, and TrueVision3D as its engine using C# from Microsoft Visual Studio as programming language

### **3.5 Research Flowchart**

This research begins by reviewing literatures from various sources such as reference books, journals, and previous studies. The steps undertaken in the research aims to facilitate in making the preparation of reports. The steps of research can be presented as in Figure 3.1.

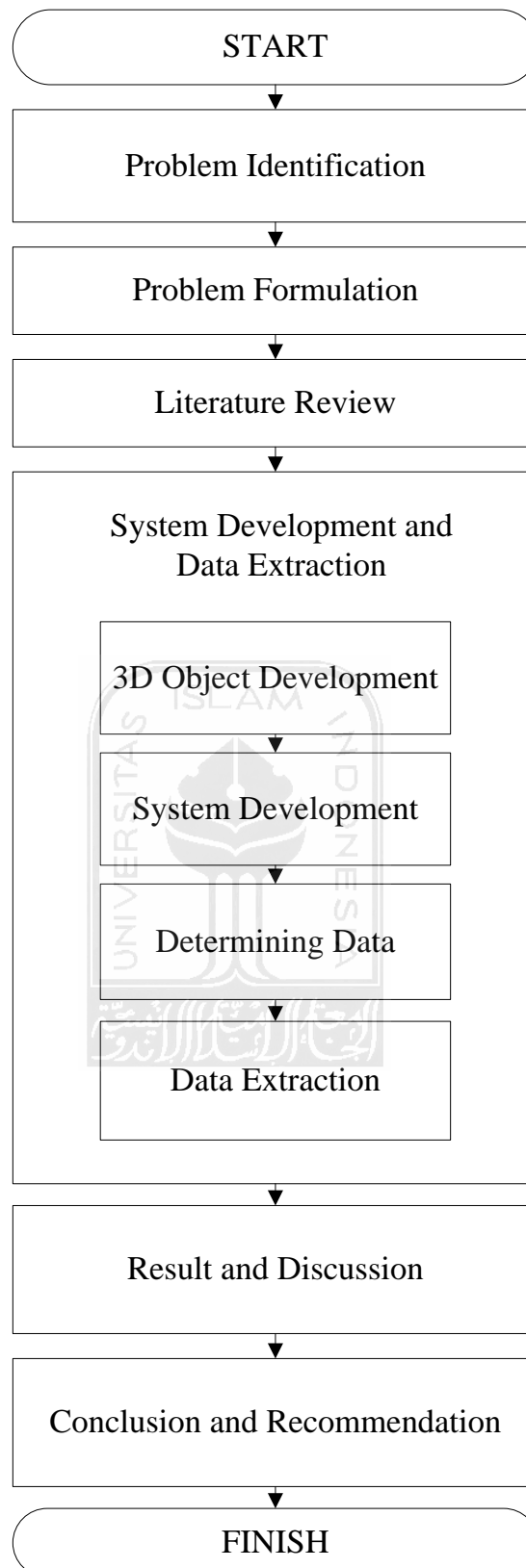


Figure 3.1 Research Flowchart

Explanation of research flowchart are given as follows:

### **1. Problem Identification**

Generally, development of VM System is intended to minimize time, effort or energy, money, and processes in producing a product. But still there are a few of lack in terms of communications efficiency and poor in control.

### **2. Problem Formulation**

Determining critical problem from the research that will be done and solve it using some analytical tools.

### **3. Literature Review**

Literature review is divided into two, deductive and inductive. In this research both these reviews are a comprehensive review of research work, whether published or not from the field of related research. The goal of both reviews is to obtain an overview of the research steps to be taken during the research process.

### **4. System Development and Data Extraction**

#### **a. 3D Object Development**

Making the design in an early stage in develop VM System. Designs that will be need in this research is a robot arm consisting of 7 main parts and an additional part: waist, shoulder, upperarm, forearm, rollarm, pitch hand, roll hand and tool; conveyors, NC Drilling, boxes, work piece and finished product. All designs were designed using

AutoCAD 2008 and 3D Max 2008 are made as closely as possible to the original design.

b. System Development

This section describes the development of VM System. Starting from input the 3-dimensional model that has been made until extracting the required data.

c. Determining the Data

This section will define and explain the data needed, so that the data can be extracted.

d. Data Extraction

This section describes how to extract the required data, that is matrices obtained from the programming.

## 5. Result and Discussion

This section contains the final result of the research which is obtained based on system testing and analysis of test result, and it will be discussed to determine the possibility of deficiency or excess of research results, in order it can make a recommendation for that research result.

## 6. Conclusion and Recommendation

This section is the final step of the research. It contains a summary of the conclusions that explain the achievement of result obtained with suggestions to improve the result of the research. While the recommendations contained about the lack of the research done that will be used as a basis for further research.

## CHAPTER IV

### SYSTEM DEVELOPMENT

This chapter describes the process of development VM System and data extraction to be used to control objects movement.

#### 4.1 3-Dimensional Object Modeling

In the development of VM System, it needs 2 pieces of robot arm KUKA 15, 2 pieces of conveyor, 2 pieces of storage, an NC drilling, a work piece and a finish product. Those models are modeled using CAD software that are AutoCAD 2008 and 3Ds Max 2008.

##### 4.1.1 Robot arm KUKA 15

Robot arms KUKA 15 in this system are as material handling, such as move the object from one to another place. This robot arm has seven main parts and one additional part that able to work independently in accordance with their respective joint. This robot arm has six degrees of freedom. Those seven main parts and one additional part are:

##### a. Waist

Waist is the most basic part of the robot arm. Waist is a crutch for all parts of the robot arm. Waist is fixed, cannot rotate and move. Waist directly related to Shoulder. Figure 4.1 describes the waist.

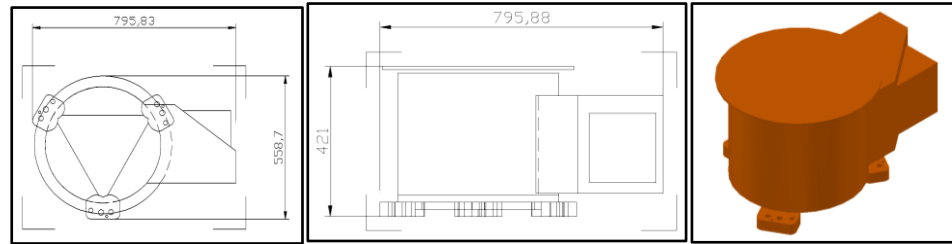


Figure 4.1 Waist (a) top view wireframe, (b) left view wireframe, (c) realistic view.

#### b. Shoulder

Shoulder is the second constituent part after the Waist. Shoulder can move and rotate with twisting according to the desired degrees. In addition to direct contact with Waist, Shoulder also directly related to the Upper arm. Figure 4.2 describes the shoulder.

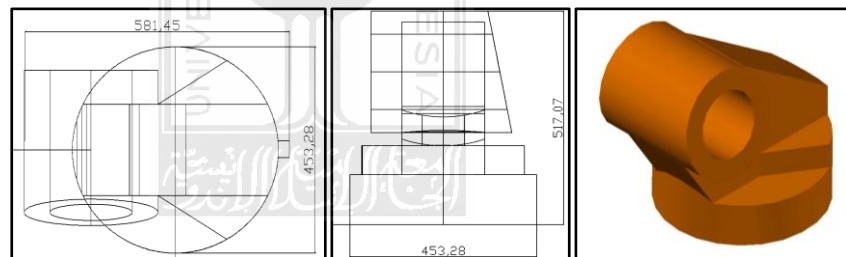


Figure 4.2 Shoulder (a) top view wireframe, (b) left view wireframe, (c) realistic view.

#### c. Upper Arm

Upper arm is the third constituent part after the Waist and Shoulder. Upper arm can move and rotate with revolving according to the desired degrees. In addition to direct contact with the Shoulder, Upper arm also directly related to the Forearm. Figure 4.3 describes the upper arm.

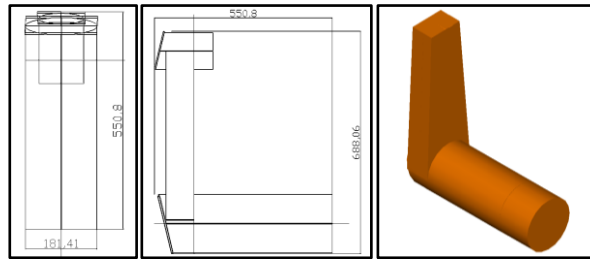


Figure 4.3 Upper Arm (a) top view wireframe, (b) left view wireframe, (c) realistic view.

d. Forearm

Forearm is the fourth constituent part after the Waist, Shoulder and Upper arm. Forearm can move and rotate with rotational according to the desired degrees. In addition to direct contact with Upper arm, Forearm also directly related to the Roll arm. Figure 4.4 describes the forearm.

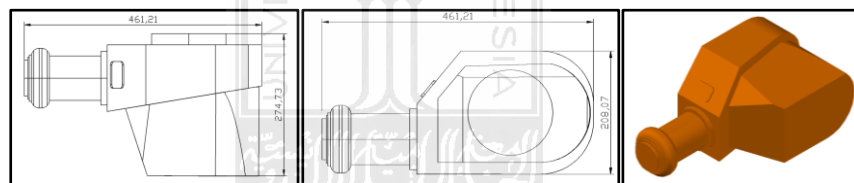


Figure 4.4 Forearm (a) top view wireframe, (b) front view wireframe, (c) realistic view

e. Roll Arm

Roll arm is the fifth constituent part after the Waist, Shoulder, Upper arm and Forearm. Roll arm can move and rotate with twisting according to the desired degrees. In addition to direct contact with the Forearm, Roll arm also directly related to the Pitch hand. Figure 4.5 describes the Roll arm



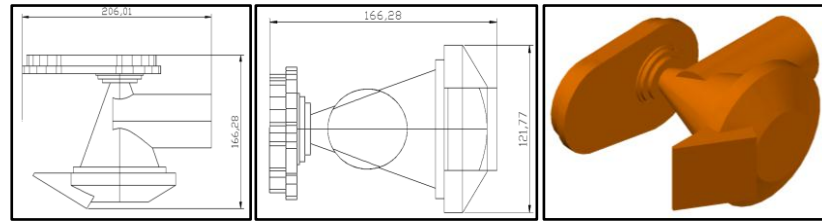


Figure 4.5 Roll Arm (a) top view wireframe, (b) left view wireframe, (c) realistic view.

#### f. Pitch Hand

Pitch hand is the sixth constituent part after Waist, Shoulder, Upper arm, Forearm and Roll arm. Pitch hand can move and rotate with revolving according to the desired degrees. In addition to direct contact with Roll arm, Pitch hand also directly related to roll hand. Figure 4.6 describes the pitch hand.

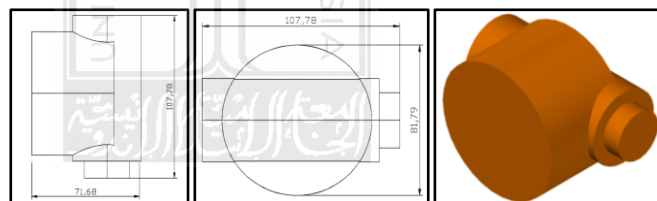


Figure 4.6 Pitch Hand (a) top view wireframe, (b) left view wireframe, (c) realistic view.

#### g. Roll Hand

Roll hand is the seventh constituent part after Waist, Shoulder, Upper arm, Forearm, Roll arm and Pitch hand. Roll hand can move and rotate with twisting. In addition to direct contact with Roll arm, Pitch Hand also directly related to an additional part, that is tool. Figure 4.7 describes the roll hand.

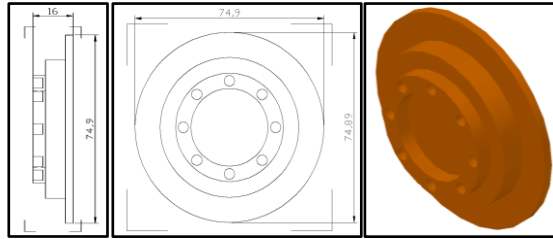


Figure 4.7 Roll Hand (a) top view wireframe, (b) left view wireframe, (c) realistic view.

#### h. Tool

Tool is an additional part in the preparation of the robot arm. This tool is paired after Roll hand. It can be changed as needed. In this case the tool that is used will move the object to be processed by attaching the tool's corner with the surface of the object. This tool can move and rotate with twisting according to the desired degrees. Figure 4.8 describes the tool.

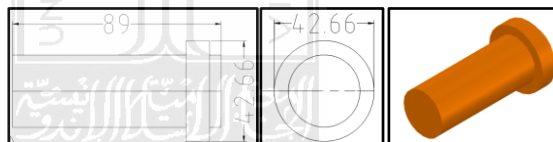


Figure 4.8 Tool (a) top view wireframe, (b) left view wireframe, (c) realistic view.

Robot arm that consists of seven main parts and an additional part as described above then it will be assembled into robot arm KUKA 15. Figure 4.9 describes the Robot arm KUKA 15.

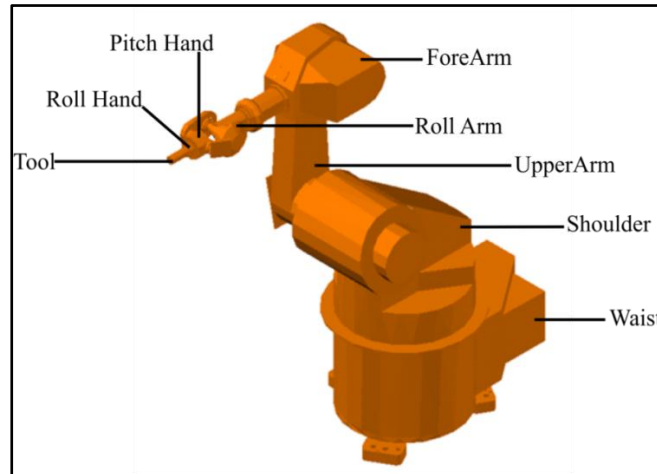


Figure 4.9 Robot arm KUKA 15

Above robot arm still can be moved over yet perfectly if all of parts of robot are not connected to each other. From robot arm above, the joint for each part are as follows:

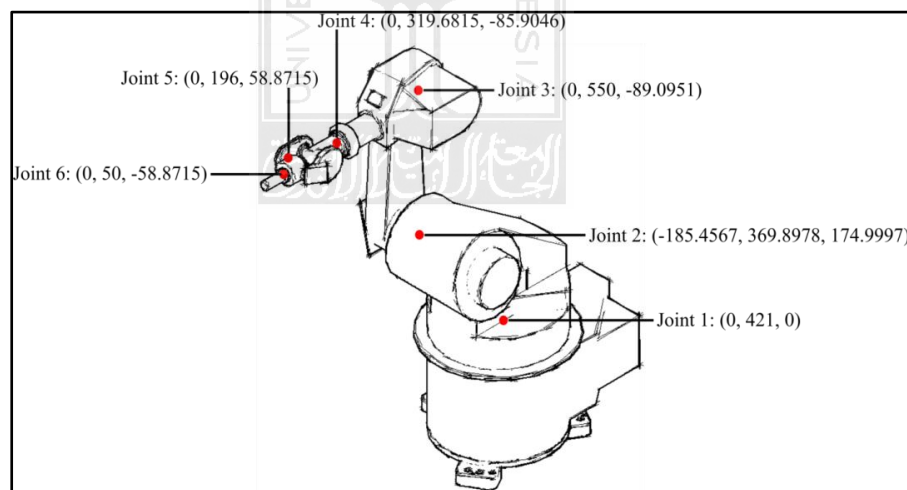


Figure 4.10 Joints of Robot arm KUKA 15

As shown in Figure 4.10 above, robot arm that was built has six joints or six degrees of freedom. Each joint can be moved in accordance with their respective functions. Joint 1 at (0, 421, 0) is also called the waist joint, this joint uniting the waist and shoulder of the robot arm and it can move in a twisting. Joint 2 at (-185.4567,

369.8978, 174.9997) is also called the shoulder joint, this joint uniting the shoulder and upper arm of the robot arm and it can be move in a revolving way. Joint 3 at (0, 550, -89.0951) is also called the elbow joint, this joint uniting the upper arm and forearm of the robot arm and it can be move in a rotational. Joint 4 at (0, 319.6815, -85.9046) is also called the roll arm joint. This joint uniting the forearm and roll arm and it can be move in a twisting way. Joint 5 at (0, 196, 58.8715) is also called pitch hand joint. This joint uniting the roll arm and pitch hand of the robot arm and it can be move in a revolving way. The last joint is joint 6 at (0, 50, -58.8715) is also called roll hand joint, this joint uniting the pitch hand and roll hand and it can be move in a twisting way.

Robot arm that has been built has reach distance as far as 140.12 cm in radius, Figure 4.11 explains how to determine the radius.

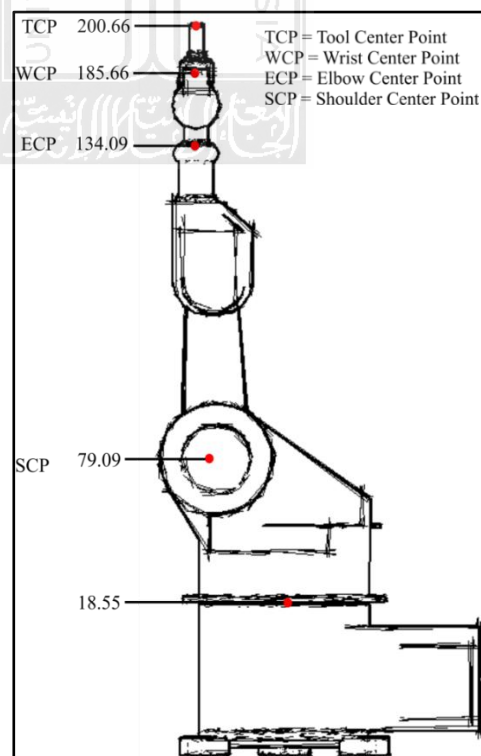


Figure 4.11 Robot arm in left view

Distance reach of robot arm =  $18.55 + (ECP - SCP) + (WCP - ECP) + (TCP - WCP)$

Distance reach of robot arm =  $18.55 + (134.09 - 79.09) + (185.66 - 134.09) +$   
 $(200.66 - 185.66)$

Distance reach of robot arm =  $18.55 + 55 + 51.57 + 15$

Distance reach of robot arm = 140.12

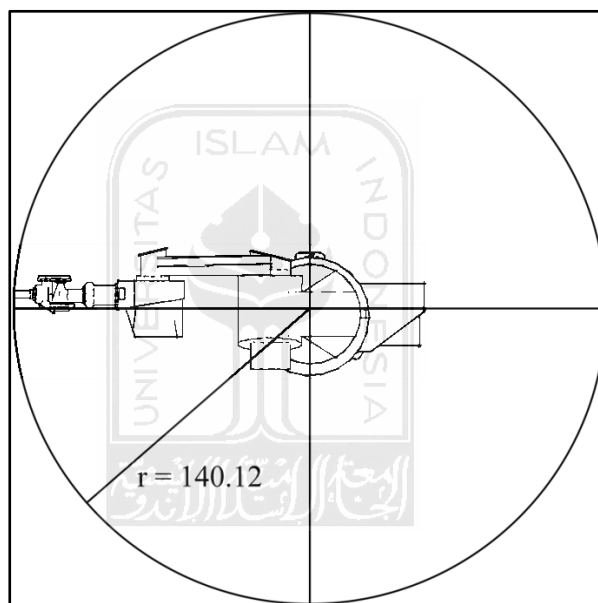


Figure 4.12 Robot arm's reach distance

#### 4.1.2 Conveyor

In this system, conveyor used as material handling. Such as move the object from one side of the conveyor to the other side of conveyor. The shape of conveyor used is I letter, because this shape is more flexible and more savings the placement than another shape of conveyor, such as L or U letter. Figure 4.13 describes the conveyor model.

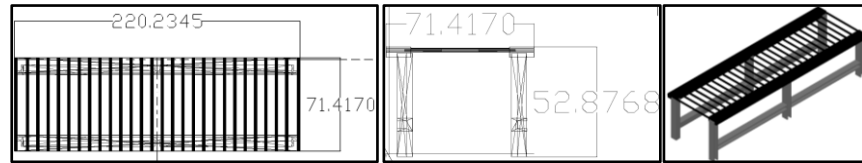


Figure 4.13 Conveyor (a) top view wireframe, (b) left view wireframe, (c) realistic view

#### 4.1.3 NC Drilling

NC Drilling in this system is to process the object. NC Drilling in this research will punch work piece with 4 holes, then become a finish product. Figure 4.14 describes the NC drilling model.

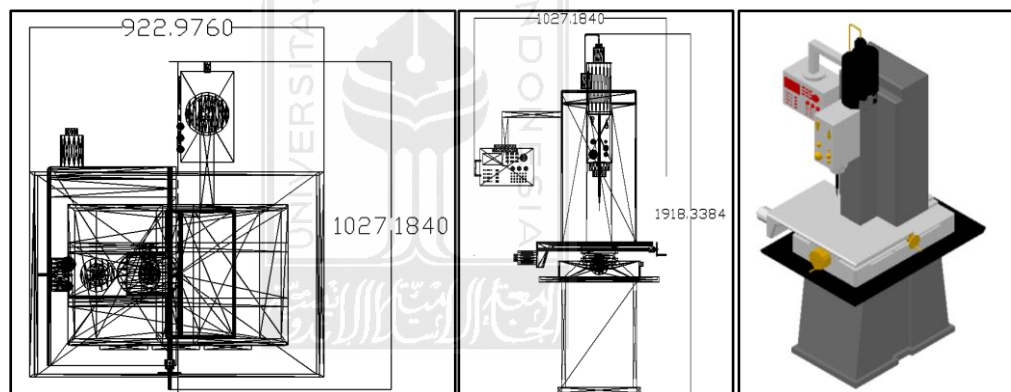


Figure 4.14 NC Drilling (a) top view wireframe, (b) left view wireframe, (c) realistic view.

#### 4.1.4 Storage

In this research, storage that is used is place to put work piece and finish product. Figure 4.15 describes the storage model.

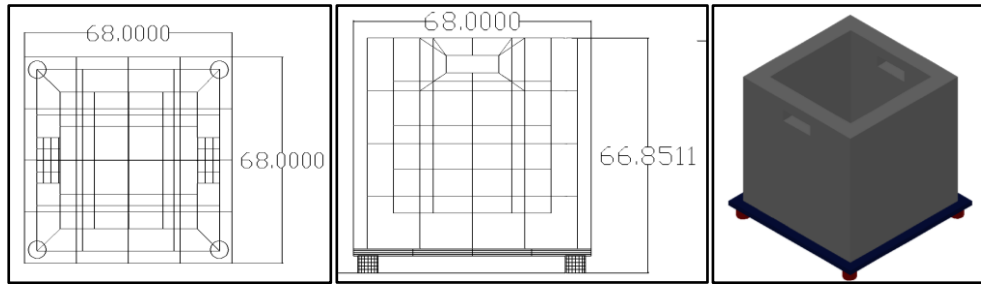


Figure 4.15 Storage (a) top view wireframe, (b) left view wireframe, (c) realistic view.

#### 4.1.5 Work piece

Work piece that is used in this system is a solid cylinder with size and pictures as follows:



Figure 4.16 Work piece (a) top view wireframe, (b) left view wireframe, (c) realistic view.

#### 4.1.6 Finish product

Finish product here is the result of work piece that has been processed by punching it with 4 holes. Figure 4.17 describes the finish product model.

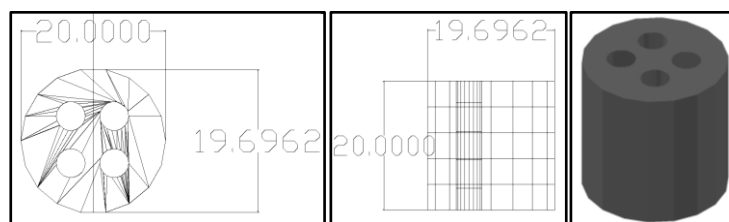


Figure 4.17 Finish product (a) top view wireframe, (b) left view wireframe, (c) realistic view.

## 4.2 Determining the Data

This sub-chapter discusses about the data to be extracted in order to control object movement.

### 4.2.1 Data Object Position

Models that have been made will be inserted into computer software, later on the arrangement of those models will form a layout of VM that is mini virtual factory.

(Unit = cm)

#### a. Robot arm

This robot arm consists of seven main parts and an additional part. It will be assembled into robot arm KUKA 15. In this research needs two pieces of robot arm with the same type. This robot arm will be placed in  $(x, y, z) = (10, 0, 10)$  for first robot arm, and  $(x, y, z) = (280, 0, 10)$  for second robot arm.

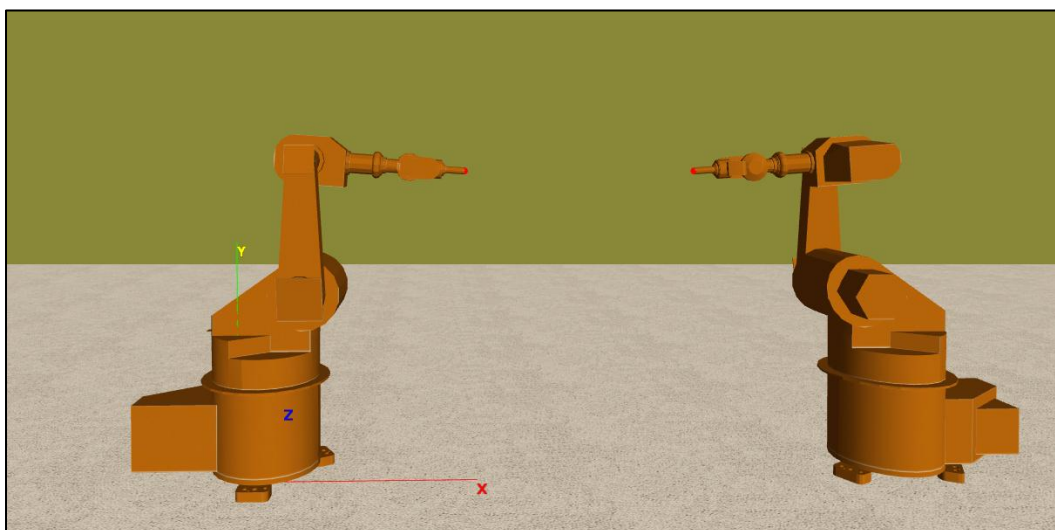


Figure 4.18 Robot arm position



b. Conveyor

In this research needs two pieces of conveyor with the same type. This conveyor placed at  $(x, y, z) = (140, 8, 50)$  for first conveyor, and  $(140, 8, -100)$  for the second conveyor.

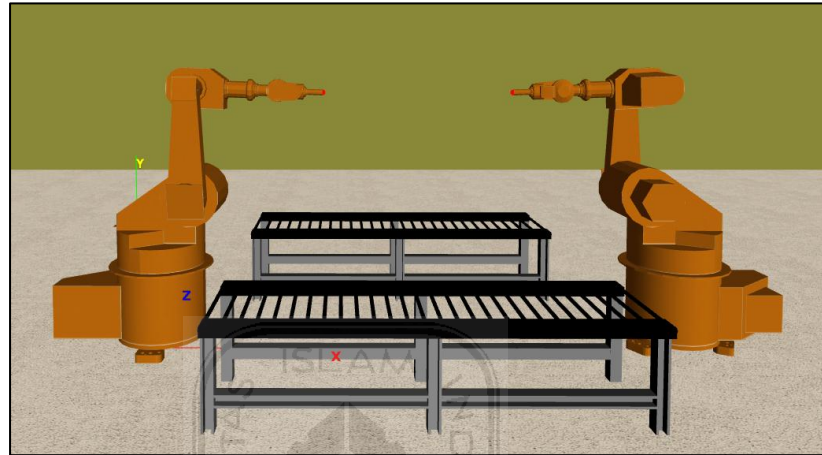


Figure 4.19 Conveyor position

c. NC Drilling

In this research need an NC drilling putted beside the second robot arm. NC Drilling is placed at  $(x, y, z) = (300, 0, 140)$ .



Figure 4.20 NC drilling position

d. Storage

In this system needs two storages with the same type, but it is used for different goods. This storage is placed at  $(x, y, z) = (0, 0, -45)$  for the first storage, and  $(0, 0, 110)$  for the second storage.

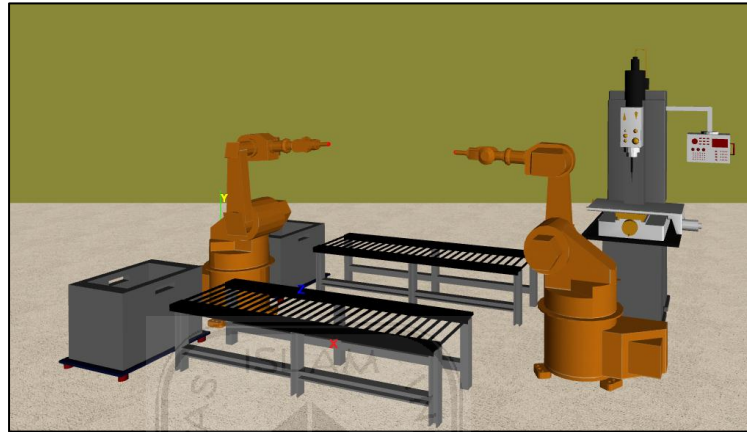


Figure 4.21 Storages position

e. Work piece

In this system work piece is putted beside the first robot arm. Work piece is placed at  $(x, y, z) = (0, 20, 80)$ .

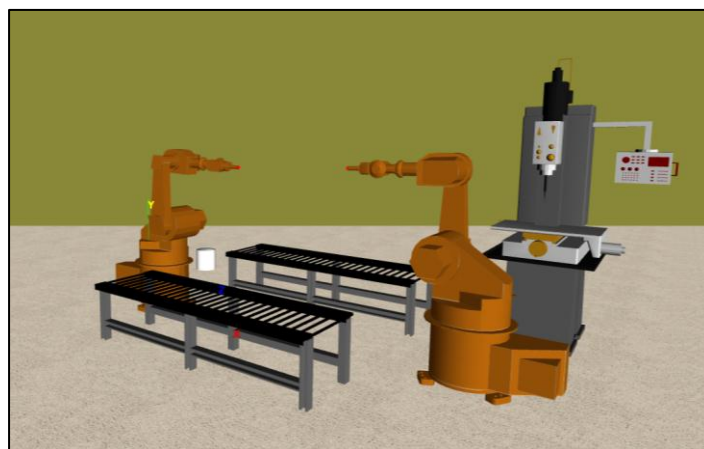


Figure 4.22 Work piece position

f. Finish product

In this system finish product putted in NC drilling. Finish product is placed at  $(x, y, z) = (0, 20, 0)$ .

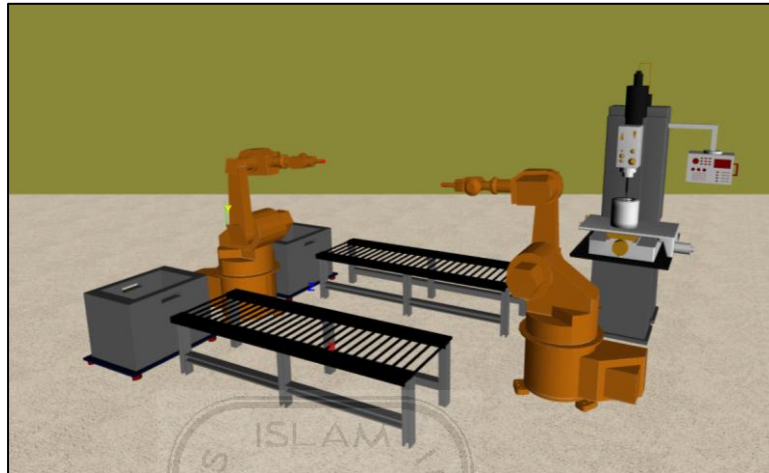


Figure 4.23 Finish product position

#### 4.2.2 Data Object Movement

In this process the object that being moving are work piece and finish product. Work piece move from  $(x, y, z) = (0, 20, 80)$  this is the position where the work piece is still inside the storage. Figure 4.24 describes the work piece movement.

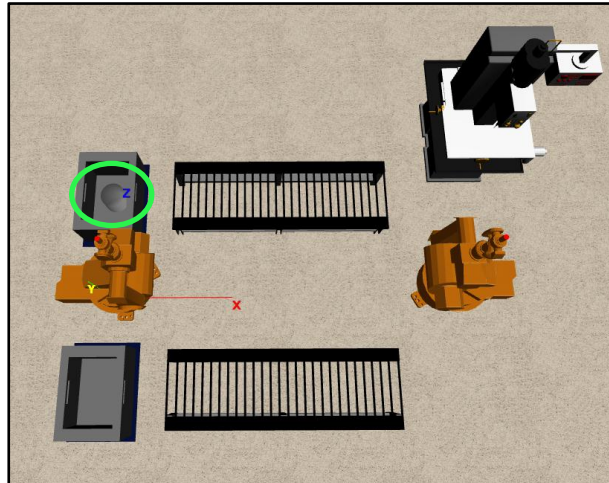


Figure 4.24 Initial positions for work piece inside the storage

Then the work piece go to (60, 40, 80) position where the work piece is above the first conveyor after being moved by the first robot arm. Figure 4.25 describes the second movement of the work piece.

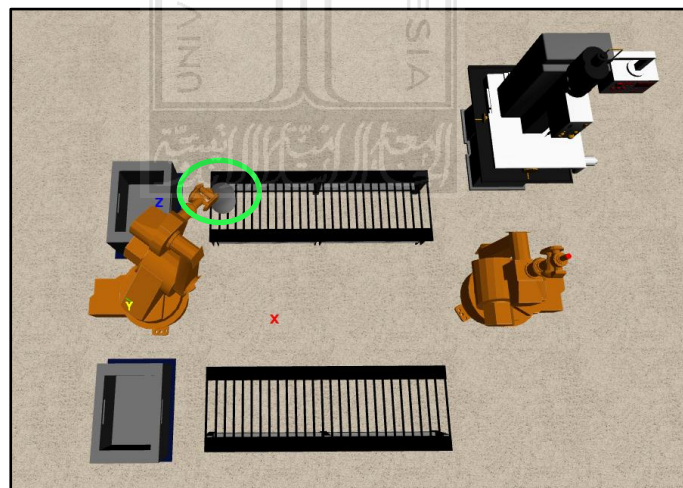


Figure 4.25 Work pieces above the first conveyor

Then the work piece go to (220, 50, 80) position where the work piece is above the first conveyor which will be moved by the second robot arm. Figure 4.26 describes the third movement of the work piece.

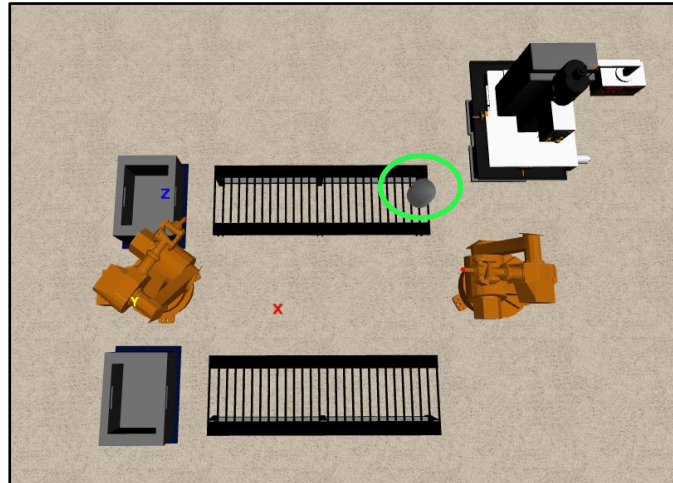


Figure 4.26 Work pieces above the other edge of the first conveyor

And the work piece go to  $(300, 100, 110)$  this is the last destination point for the work piece; this point is above NC Drilling machine.

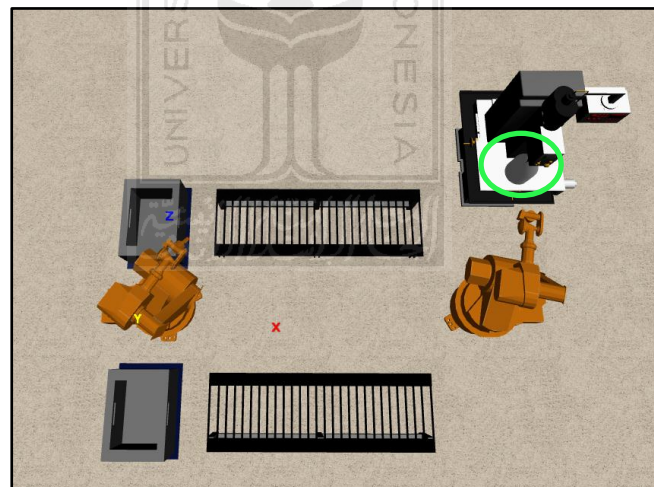


Figure 4.27 Work pieces in NC drilling

After that, finish product will move from  $(x, y, z) = (300, 100, 110)$  this is the starting position for finish product, where it is above the NC Drilling machine.

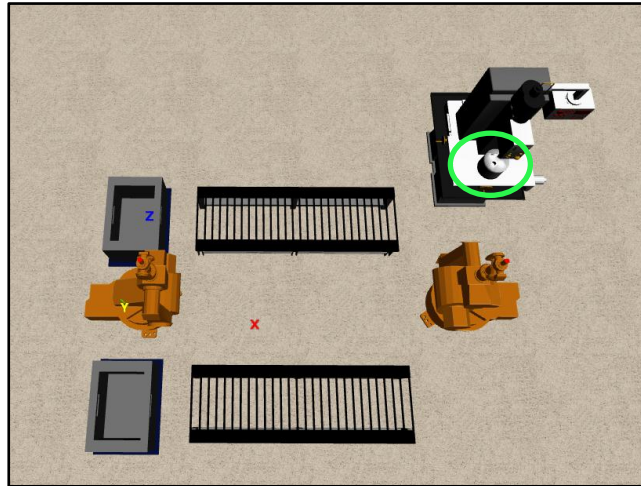


Figure 4.28 First position of finish product in NC drilling

Then the finish product go to  $(220, 20, -70)$  position where the finish product is above the second conveyor after being moved by the second robot arm, Figure 4.29 describes the second movement of the finish product.

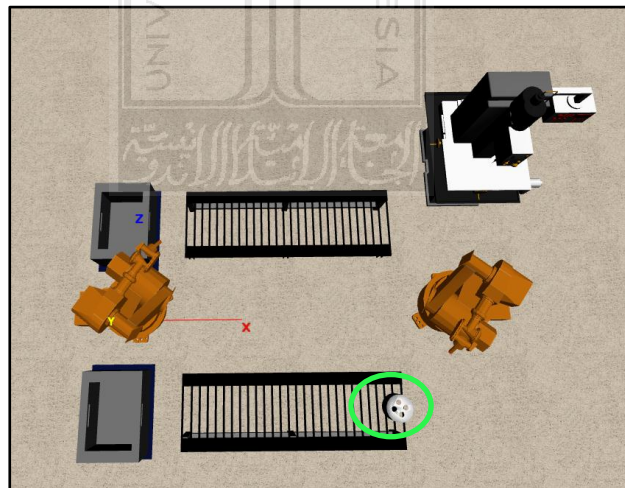


Figure 4.29 Finish products above the second conveyor

Then the finish product go to  $(60, 50, -70)$  position where finish product is above the second conveyor which will be moved to the next position by the first robot arm.

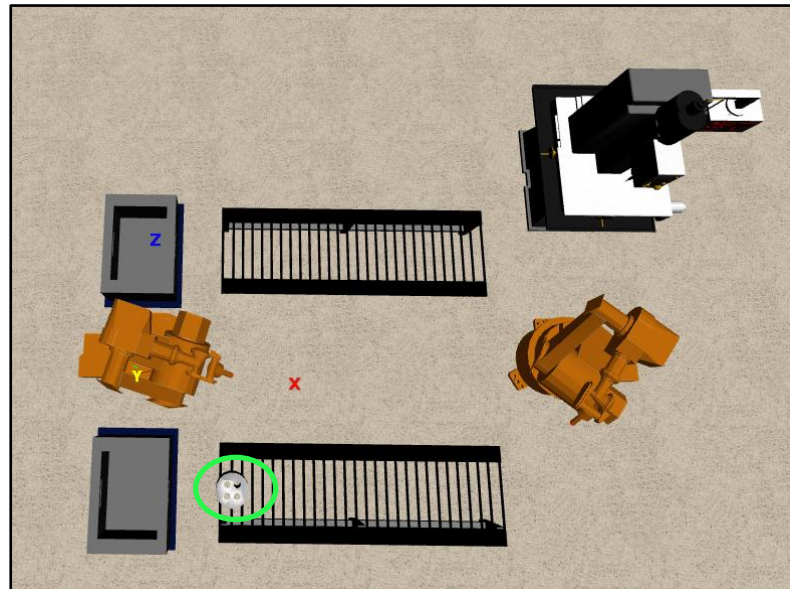


Figure 4.30 Finish products above the other edge of the second conveyor

And then the finish product go to  $(0, 20, -70)$  this is the last point for finish product, this point inside the second storage that will become place for finish product.

Figure 4.31 describes the laying of finish product.

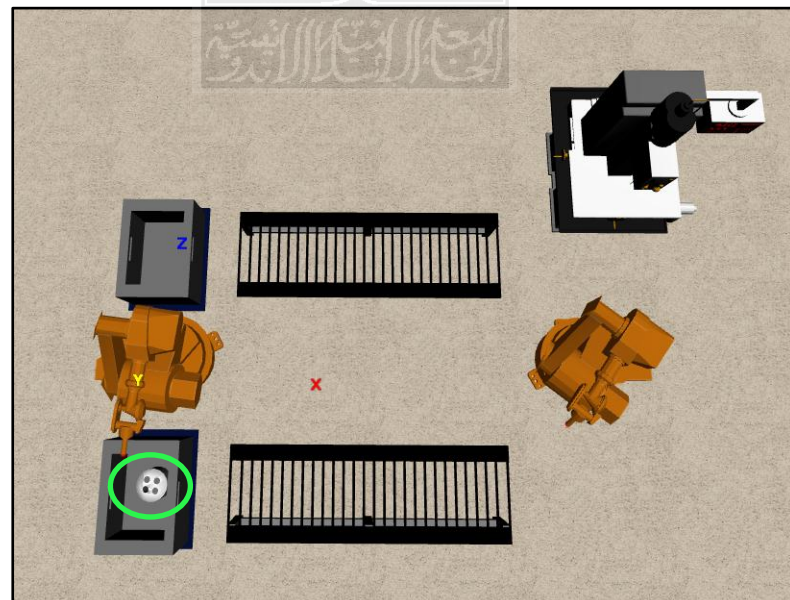


Figure 4.31 Finish products inside the second storage

### 4.2.3 Data Joints

Data joint obtained from the result of a merger between each part of robot arm. This data used to connect between one parts to another part of robot arm, so that when the robot arm is moved these parts remain at a predetermined position. The data joint are:

1. Joint 1 = (0, 421, 0)
2. Joint 2 = (-185.4567, 369.8978, 174.9997)
3. Joint 3 = (0, 550, -89.0951)
4. Joint 4 = (0, 319.6815, -85.9046)
5. Joint 5 = (0, 196, 58.8715)
6. Joint 6 = (0, 50, -58.8715)

### 4.3 System Development

Development of VM System in this research is using TrueVision3D as the engine and C# as programming language. Broadly, this research consisted of inputting process of .X file into the VE, and then performs data extraction. Development of this system can be seen in Figure 4.32.

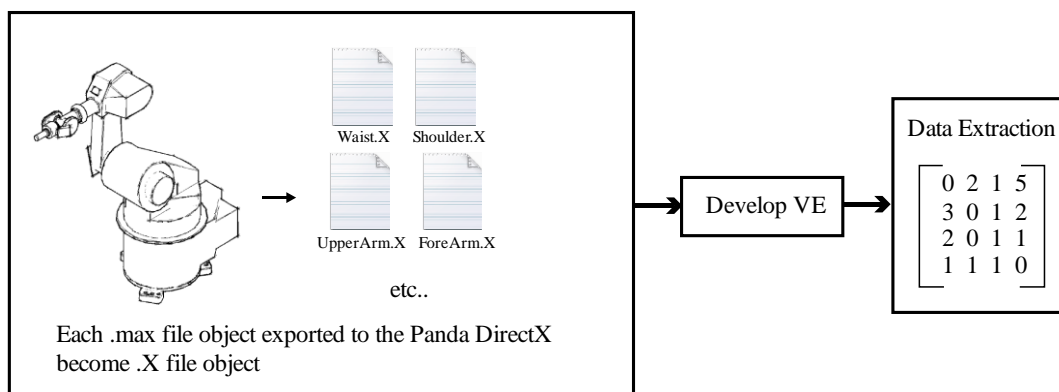


Figure 4.32 System Development



To build a VM needs a virtual object that have several stage of development, which is the design and transfer stages. As explained in the previous sub-chapter, the virtual object is constructed using 3Ds Max 2008 and generate a *.Max* file. Then this *.Max* file should be transferred to the VE that is built using TrueVision3D , however *.Max* file cannot be directly transferred, this file must be exported using Panda DirectX Converter first which will generate an *.X* file. This export process done because it can save memory so that the resulting program can be run smoothly and lightly.

Development of VM aims to create an environment as similar as possible to the original, so that users can interact and feel the same with the real one. In development of VE needs a virtual object which will be arranged within the virtual room that will form a virtual layout of the VE.

Virtual objects are required from the transferring files process from *.Max* into *.X* (as described in the paragraph above). Then this file will be called using *LoadXFile* function. Figure 4.33 is the *pseudocode* of loading the virtual object.

```

#region NCDrilling
  Define mesh for NCDrilling
  Calling NCDrilling file using LoadXFile
  Set position NCDrilling
  Set scale NCDrilling
  Set rotation NCDrilling
  Adjust the lighting on NCDrilling
#endregion

```

Figure 4.33 *Pseudocode* load object

All the objects needed to be transferred to the VE using the same function.

Objects that have been transferred will be structured like Figure 4.45.

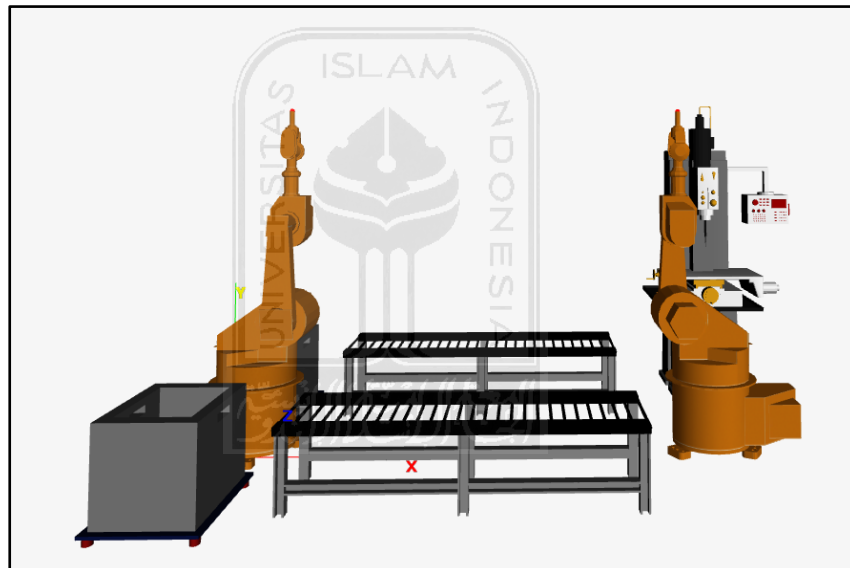


Figure 4.34 Virtual object in VE

Next step is to create a virtual room becomes more attractive and eye catching, with giving walls and floors into the virtual room. Figure 4.35 describes the function to create virtual room.

```

private TVMesh _roomMesh;
private TVMesh _floorMesh;
_roomMesh = new TVMesh();
InitRoom();
private void InitTextures()
{
    Calling texture if the wall using LoadTexture
}
private void InitRoom()
{
    Define mesh for room and floor
    Set floor size
    -5000.0f, -5000 - 0f, 5000.0f, 5000.0f, 0.0f, 100.0f, 100.0f,
    true);
}

```

Figure 4.35 Pseudocode to create a room

Once those functions are entered into the program, then the virtual room will be like Figure 4.36.

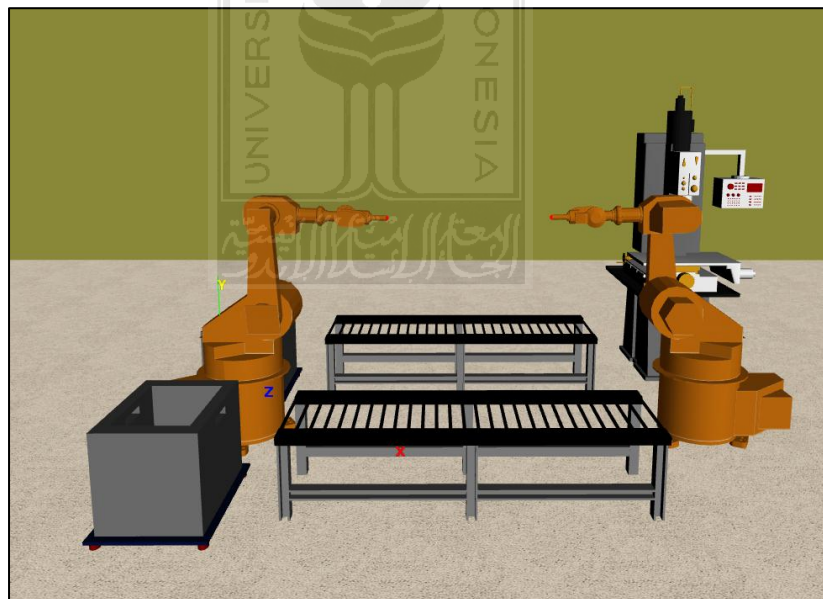


Figure 4.36 Virtual Room

Virtual room is constructed consists of two robot arms, two conveyors, two storages, an NC Drilling, a work piece and a finish product. All of them still cannot be

called as the perfect VE, because it cannot be able to interact with the users. Therefore, to move the objects needs data extraction.

#### 4.4 Data Extraction

Data extraction was carried out to assist in moving objects that is robot arm. Actually, to move these objects not used data extraction but uses the result from data extraction that will be used in moving objects. The result of data extraction is transformation matrix.

The resulting matrix becomes the transformation mesh matrix, including rotation, translation and scaling. This matrix can be used together with an external physic engine. In this case the type of transformation matrix used is perspective projection. Means that the object has a smaller projection when far from the center of projection and has a larger projection when it is closer. Or in another words, when a human looks into the screen, then the object will look smaller than the actual object. The function used to obtain the matrix is a *GetMatrix* function. The arrangement functions as follows:

```
//write mesh matrix to a file
if (_bWrite2File)
{#region LINK MATRIX
    Define the name of the file used to save the matrix
    Name each matrix and call them using GetMatrix function
#endregion}
```

Figure 4.37 Pseudocode to get the matrix

Once these functions are entered into the program, then the desired matrix will be obtained. These matrixes will be used to control the object movement of a virtual robot arm that has been built. The following is transformation matrix resulting from the data extraction that has been done



LINK MATRIX	foreArm-----
Robot 1	-000.10 0000.00 0000.00 0000.00
waist-----	0000.00 0000.10 0000.00 0000.00
-000.10 0000.00 0000.00 0000.00	0000.00 0000.00 -000.10 0000.00
0000.00 0000.10 0000.00 0000.00	0028.55 0134.09 0001.41 0001.00
0000.00 0000.00 -000.10 0000.00	rollArm-----
0010.00 0000.00 0010.00 0001.00	-000.10 0000.00 0000.00 0000.00
shoulder-----	0000.00 0000.10 0000.00 0000.00
-000.10 0000.00 0000.00 0000.00	0000.00 0000.00 -000.10 0000.00
0000.00 0000.10 0000.00 0000.00	0028.55 0166.06 0010.00 0001.00
0000.00 0000.00 -000.10 0000.00	pitchHand-----
0010.00 0042.10 0010.00 0001.00	-000.10 0000.00 0000.00 0000.00
upperArm-----	0000.00 0000.10 0000.00 0000.00
-000.10 0000.00 0000.00 0000.00	0000.00 0000.00 -000.10 0000.00
0000.00 0000.10 0000.00 0000.00	0028.55 0185.66 0004.11 0001.00
0000.00 0000.00 -000.10 0000.00	rollHand-----
0028.55 0079.09 -007.50 0001.00	-000.10 0000.00 0000.00 0000.00
	0000.00 0000.10 0000.00 0000.00
	0000.00 0000.00 -000.10 0000.00
	0028.55 0190.66 0010.00 0001.00

Figure 4.38 Matrix of first robot arm

LINK MATRIX	foreArm2-----
Robot 2	0000.10 0000.00 0000.00 0000.00
waist2-----	0000.00 0000.10 0000.00 0000.00
0000.10 0000.00 0000.00 0000.00	0000.00 0000.00 0000.10 0000.00
0000.00 0000.10 0000.00 0000.00	0261.45 0134.09 0018.59 0001.00
0000.00 0000.00 0000.10 0000.00	rollArm2-----
0280.00 0000.00 0010.00 0001.00	0000.10 0000.00 0000.00 0000.00
shoulder2-----	0000.00 0000.10 0000.00 0000.00
0000.10 0000.00 0000.00 0000.00	0000.00 0000.00 0000.10 0000.00
0000.00 0000.10 0000.00 0000.00	0261.45 0166.06 0010.00 0001.00
0000.00 0000.00 0000.10 0000.00	pitchHand2-----
0280.00 0042.10 0010.00 0001.00	0000.10 0000.00 0000.00 0000.00
upperArm2-----	0000.00 0000.10 0000.00 0000.00
0000.10 0000.00 0000.00 0000.00	0000.00 0000.00 0000.10 0000.00
0000.00 0000.10 0000.00 0000.00	0261.45 0185.66 0015.89 0001.00
0000.00 0000.00 0000.10 0000.00	rollHand2-----
0261.45 0079.09 0027.50 0001.00	0000.10 0000.00 0000.00 0000.00
	0000.00 0000.10 0000.00 0000.00
	0000.00 0000.00 0000.10 0000.00
	0261.45 0190.66 0010.00 0001.00

Figure 4.39 Matrix of second robot arm

## CHAPTER V

### DISCUSSION

This chapter will discuss the result of the research that has been done on the development VM System including object modeling and data extraction.

#### 5.1 Testing a System Built

This section discuss the testing system has been built by comparing the angle of the manual calculation with the result of the program. It is done in order to prove that the model used in the program in accordance with the model that was designed at the beginning or in other words the test is conducted to validate the model. To perform this test need data about the size of robot arm that are the length of the shoulder joint to the elbow joint and the elbow joint to the wrist joint, those length are:

Table 5.1 Example data joint's length of robot arm

Robot arm 1	
Shoulder – elbow joint (Lc)	55
Elbow – wrist joint (La)	51.57

To test the system need an example of robot movement. Figure 5.1 is an example of the movement of the robot arm model.



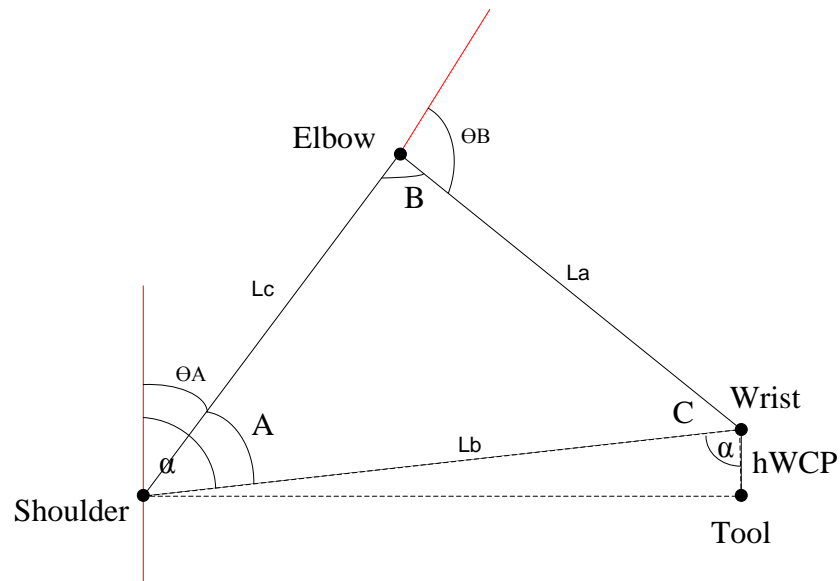


Figure 5.1 Example of the robot arm model

In this discussion the value of  $L_a$ ,  $L_b$  and  $L_c$  are known then find the value of  $\alpha$  using:

$$\cos \alpha = h_{WCP} / L_b$$

$h_{WCP}$  obtained from the difference in height between the SCP and WCP from the surface. In the robot arm model, the high of SCP and WCP are 79.09 and 87.05, and for  $L_b$  are 67.94, so the value of  $\alpha$  is:

$$\cos \alpha = (87.05 - 79.09) / 67.94$$

$$\alpha = 83.27$$

After that, should find the value of  $A$  using the cosines law as follows:

$$\cos A = (L_b^2 + L_c^2 - L_a^2) / 2L_b.L_c$$

$$A = 48.19$$

Further, find the  $\Theta_A$  by subscribing  $\alpha$  with  $A$ , so that the robot arm rotation angle obtained in this mode is 35 degrees for the shoulder.

To calculate the angle at the elbow, the value of B must be sought using the cosines law.

$$\text{Cos B} = (\text{La}^2 + \text{Lc}^2 - \text{Lb}^2) / 2\text{La.Lc}$$

$$\text{B} = 79.14$$

After that, find the  $\Theta\text{B}$  by subscribing the straight angle that is 180 degrees with B.

$$\Theta\text{B} = 180 - 79.1$$

$$\Theta\text{B} = 100.9 \text{ degrees}$$

Below is a comparison of  $\Theta\text{A}$  and  $\Theta\text{B}$  obtained manually and from the result of the program that was built.

Table 5.2 Value of  $\Theta\text{A}$  and  $\Theta\text{B}$

	Manual calculation	Result from the program
Shoulder ( $\Theta\text{A}$ )	35.0	35.0
Elbow ( $\Theta\text{B}$ )	100.9	100.9

In this discussion also compares the position of TCP in robot arm, whether the position of TCP from the robot arm built are same or not with the position of TCP on the program made. For TCP from the robot arm built the position is 200.6 and for TCP on the program that has been made the position are 200.6 also.

## CHAPTER VI

### CONCLUSION AND RECOMMENDATION

Conclusions and recommendations below are obtained from the result of research and in accordance with the problem formulations of research that has been initially determine. The conclusions and recommendations are as follows:

#### 6.1 Conclusions

The result of the research can be concluding as follows:

1. Development VM System including object modeling by using AutoCAD 2008 and 3D Max 2008 as its design software to create three dimensional models, and using TrueVision3D as the engine with C# from Microsoft Visual Studio as its programming language is running well in accordance with the scenario that was design at the beginning.
2. Data extraction for VM System by using TrueVision3D and C# as the programming language obtained a clear result, which are transformation matrices that will be used to control object movement.

## 6.2 Recommendations

For further research should be added a few things that can enhance and develop the system performance, that are:

1. The system built should use the simple script in controlling object movement, in order to facilitate the user in controlling the objects.
2. The system built should discuss both the inventory of raw material and finished product.
3. The system built should use the appropriate scheduling with real system
4. The system built should discuss the production cost needed.



## REFERENCES

- Amos H C Ng, Adolfsson. Josef, Sundberg. Martin, De Vin. leo J, (2008). Virtual manufacturing for press line monitoring and diagnostic. *International Journal of Machine Tools and Manufacture*. 48 no 5, 565-575.
- Arayici. Y, Aouad.G, (2004). Divercity: distributed virtual workspace for enhancing communications and collaboration within the construction industry. *In: A. Dikbas, R. Scherer (Eds.), eWork and eBusiness in Architecture, Engineering and Construction*, Taylor & Francis Group, London, UK.
- Asfahl. C. Ray, (1992). *Robots and manufacturing automation*. John Wiley & Sons, Inc. New York.
- Banerjee. P & Zetu. D, (2001). *Virtual manufacturing*. John Wiley & Sons, Inc. New York.
- Bouchlaghem. D, Shang. H, Whyte. J, Ganah. A, (2004). Visualisation in architecture, engineering and construction (AEC). *Automation in Construction*. 14 no 3, 287–295.
- Chen. Ke-Zhang, Feng. Xing-Yang, Wang. Feng, Feng. Xin-An, (2007). A virtual manufacturing system for components made of a multiphase perfect material. *Computer-Aided Design*. 39 no 2, 112-124.
- Chen. Lienjing, Bender. P, Renton. P, El-Wardany. T, (2002). Integrated Virtual Manufacturing Systems for Process Optimisation and Monitoring. *CIRP Annals - Manufacturing Technology*, 51 no 1, 409-412.
- Ellis. R.E, (1994). What are virtual environment?. *IEEE Comput. Graph. Appl*. 14, 17-22.
- Fumarola. Michele, Poelman. Ronald, (2011). Generating virtual environments of real world facilities: Discussing four different approaches. *Automation in Construction*. 20 no 3, 263-269.

- Groover. M.P, Weiss. M, Nagel. R.N, Odrey. N.G, (1986). *Industrial robotics*. McGraw-Hill, Inc.
- Hariyadi. Efendi. 2009. OOP Object oriented programming part-1. March 9 2009. Bongkar Online. [www.bongkar.co.id/informasi-teknologi/hard-coding/817-oop-object-oriented-programming-part-1.html](http://www.bongkar.co.id/informasi-teknologi/hard-coding/817-oop-object-oriented-programming-part-1.html). February 1 2012.
- Heathfield. Heather, Armstrong. Jim, Kirkham. Nigel. (1991). Object-oriented design and programming in medical decision support. *Computer Methods and Programs in Biomedicine*, 36, no 2, 239-251.
- Ingemansson. A, Oscarsson. J, (2006). Simulation and production improvement techniques. *In: Proceedings if the 39th CIRP International Seminar on Manufacturing Systems*, Bled, Slovenia.
- Iwata. K, M. Onosato, K. Teramoto, S. Osaki, (1997). Virtual manufacturing systems as advanced information infrastructure for integrated manufacturing resources and activities. *CIRP*. 46 no 1, 335-338.
- Kam. C, Fischer. M, (2004). Capitalizing on early project decision-making opportunities to improve facility design, construction, and life-cycle performance—POP, PM4D, and decision dashboard approaches. *Automation in Construction*. 13 no 1, 53–65.
- Kim. Yong-Sik, Yang. Jeongsam, Han. Soonhung. (2006). A multichannel visualization module for virtual manufacturing. *Computers in Industry*, 57, 653-662.
- Klar. R, (2002). Extended discrete event simulation including restriction criteria for material flow analysis. M.Sc. thesis, University of Skovde.
- Koren. Yorem, (1983). *Computer control of manufacturing systems*. McGraw-Hill, Inc. United States of America.
- Lee. W. B, Cheung. C. F, Li. J. G, (2001). Applications of virtual manufacturing in materials processing. *Journal of Materials Processing Technology*. 113 no 1-3, 416-423.

- Lee. K. I, S.D. Noh, (1997). Virtual manufacturing system --- a test-bed of engineering activities. *CIRP*. 46, 347-350.
- MathImage. 2011. Transformation Matrix. July 2011.  
[http://mathforum.org/mathimages/index.php/Transformation\\_Matrix#3d\\_Transformation\\_Applet](http://mathforum.org/mathimages/index.php/Transformation_Matrix#3d_Transformation_Applet). February 10 2012.
- Matthew W. Rohrer, (2000). Seeing is believing: The Importance of Visualization in Manufacturing Simulation, *Winter Simulation Conference*.
- Mclean CR, Bloom HM, Hopp TH, (1982). The virtual manufacturing cell. *In: Proceedings of the 4th IFAC/IFIP conference on information control problems in manufacturing technology*. Gaithersburg, MD; 1982, 1-9.
- Mujber. T. S, Szecsi, Hashmi. M. S. J, (2004). Virtual reality applications in manufacturing process simulation. *Journal of Materials Processing Technology*. 155-156, 1834-1838.
- Nagy. Fracis. N & Siegler. Andras, (1987). *Engineering foundations of robotics*. Prentice-Hall International (UK) Ltd. United States of America.
- Offodile. O.F, L. Layek, Malek. Abdel. (2002). The virtual manufacturing paradigm: The impact of IT/IS outsourcing on manufacturing strategy. *Int. J. Production Economics*, 75, 147-159.
- Park. Sang C, (2005). A methodology for creating a virtual model for a flexible manufacturing system. *Computer in Industry*. 56 no 7, 734-746.
- Pham. D.T, Gault. R. S, (1998). A comparison of rapid prototyping technologies. *International Journal of Machine Tools and Manufacture*. 38 no 10-11, 1257-1287.
- R.S. Tiruvannamalai, (2002). *Virtual Reality in Design and Manufacturing Applications*.
- Rosenblum. LJ, Bryson. S, Feiner. S.K, (1995). Virtual reality unbound. *IEEE Comput. Graph. Appl*. 15, 19-21.

- S. Jayaram, H. Connacher, K. Lyons, (1997). Virtual assembly using virtual reality techniques, *Comp. Aided Des.* 29, 557–584.
- Shiratuddin. M.F., W. Thabet, (2007). Information rich virtual design review environment. *Proceedings of the 4th W78 Conference, Maribor 2007 & 5th ITCEDU Workshop & 14th EG-ICE Workshop.*
- Slomp. Jannes, Chowdary. B. V, Suresh. N. C, (2005). Design of virtual manufacturing cells: a mathematical programming approach. *Robotics and Computer-Integrated Manufacturing*, 21 no 3, 273-288.
- Solding. P, Eriksson. P, De Vin LJ, (2003). Discrete event simulation in operational production planning--an outline. *In: Proceedings of the IMC-20, Conference, Cork, Ireland.* 181-187.
- Sundberg. M, Ng AHC, Adolfsson. J, De Vin LJ, (2006). Simulation supported service and maintenance in manufacturing. *In: Proceedings of the IMC23, Jordanstown, UK.* 559-566.
- Sutherland. I.E, (1965). The ultimate display. *Proceedings of the 1965 IFIP Congress.* 2, 506-508.
- Sutherland.I.E, (1968). A head mounted three dimensional display. *Proceedings of the Fall Joint Computer Conference.* 1, 757-764.
- TheServerside. 2008. Object-oriented programming (OOP). March 2008. <http://searchsoa.techtarget.com/definition/object-oriented-programming>. February 2 2012.
- TrueVision3D. xxxx. TV3D Engine Overview. <http://www.truevision3d.com/page-14-3d-development-with-tv3d>. February 10 2012.
- Waterman. N.A, Dickens. P, (1994). Rapid product development in the USA. *Europe and Japan, World Class Design To Manufacture.* 1 no 3, 27–36.
- Wikipedia. 2007. TrueVision3D. September 2007. <http://en.wikipedia.org/wiki/Truevision3D>. February 10 2012.



Wikipedia. 2009. Autodesk 3ds Max. December 2009.  
[http://en.wikipedia.org/wiki/3ds\\_Max](http://en.wikipedia.org/wiki/3ds_Max). February 2 2012.

Wikipedia. 2011. AutoCAD. November 2011. <http://en.wikipedia.org/wiki/AutoCAD>.  
February 2 2012.

Woksepp. S, Olofsson. T, (2006). Using virtual reality in a large-scale industry project. *The Journal of Information Technology in Construction*. 11, 627–640.

Zhang. Yu, Liu. Hongwu, (2009). Application of Rapid Prototyping Technology in Die Making of Diesel Engine. *Tsinghua Science & Technology*. 14 S1, 127-131.



## ATTACHEMENTS

### 1. Function to define Robot arm 1 and 2

```
robot1.SetName("robot1");
robot1.SetPosition(10, 0, 10);
robot1.SetRotateY(180f);
robot1.SetScale(0.1f);
robot1.InitializeRobot();

robot2.SetName("robot2");
robot2.SetPosition(280, 0, 10f);
robot2.SetScale(0.1f);
robot2.InitializeRobot();
```

### 2. Function to define Conveyor

```
#region conveyor
_conveyorMesh[i] = new TVMesh();
_conveyorMesh[i] = _scene.CreateMeshBuilder("conveyor");
_conveyorMesh[i].LoadXFile(_mediaDir + "conveyor\\conveyor1.x");
_texture.LoadTexture(String.Format(_mediaDir + "Textures\\silver.bmp",
Application.StartupPath));
_conveyorMesh[i].SetLightingMode(CONST_TV_LIGHTINGMODE.TV_LIGHTING_MANAGED
);
_conveyorMesh[i].SetPosition(140.0f, 8.0f, -150.0f * (i + 1) + 200.0f);
_conveyorMesh[i].SetScale(0.8f, 0.8f, 0.8f);
_conveyorMesh[i].SetRotation(-90, 0, 0);
#endregion
```

### 3. Function to define Storage

```
#region box
_BoxMesh[i] = _scene.CreateMeshBuilder("graybox");
_BoxMesh[i].LoadXFile(_mediaDir + "box\\graybox.x");
_BoxMesh[i].SetPosition(0, 0, 155.0f*(i+1)-200.0f);
_BoxMesh[i].SetScale(0.8f, 1.1f, 0.8f);
_BoxMesh[i].SetRotation(-90, 0, 0);
_BoxMesh[i].SetLightingMode(CONST_TV_LIGHTINGMODE.TV_LIGHTING_MANAGED);
#endregion
```

#### 4. Function to define Work piece

```
#region cylinder
_cylinderMesh = _scene.CreateMeshBuilder("Cylinder1");
_cylinderMesh.CreateCylinder(_radius, _height, 30, true);
_cylinderMesh.SetTexture(_globals.GetTex("RawTexture"));
_cylinderMesh.SetMaterial(_globals.GetMat("basic Material"));
_cylinderMesh.SetPosition(_ptC0.x, _ptC0.y + _height / 2, _ptC0.z);
_cylinderMesh.SetLightingMode(CONST_TV_LIGHTINGMODE.TV_LIGHTING_MANAGED);
_cylinderAngleY =
_mathLibrary.Direction2Ang(_cylinderMesh.GetPosition().x,
_cylinderMesh.GetPosition().z);
_cylinderMesh.Enable(true);
#endregion
```

#### 5. Function to define NC Drilling

```
#region NCDrilling
_NcMesh = _scene.CreateMeshBuilder("ncDrilling");
_NcMesh.LoadXFile(_mediaDir + "nc\\ncdrilling.x");
_NcMesh.SetPosition(300, 0, 140);
_NcMesh.SetScale(0.11f, 0.11f, 0.11f);
_NcMesh.SetRotation(-90, 180, 90);
_NcMesh.SetLightingMode(CONST_TV_LIGHTINGMODE.TV_LIGHTING_MANAGED);
#endregion
```

#### 6. Function to define Finished product

```
#region Object
_ObjectMesh = _scene.CreateMeshBuilder("Object");
_ObjectMesh.LoadXFile(_mediaDir + "object\\whitecylinder.x");
_ObjectMesh.SetParent(CONST_TV_NODETYPE.TV_NODETYPE_MESH,
_targetObjectMesh.GetIndex(), 0);
_ObjectMesh.SetPosition(0, 20, 0);
_ObjectMesh.SetScale(1f, 1f, 1f);
_ObjectMesh.SetRotation(0, 0, 0);
_ObjectMesh.SetMeshCenter(0, 20, 0);
_ObjectMesh.SetLightingMode(CONST_TV_LIGHTINGMODE.TV_LIGHTING_MANAGED);
_ObjectMesh.Enable(true);
#endregion
```

## 7. Function to get the matrix

```
if (_bWrite2File)
{
#region LINK MATRIX
Write2File("*****", false);
Write2File("LINK MATRIX", false);
Write2File("*****", false);
Write2File("world coordinate", true);
Write2File("waist-----",
robot1.m_waistMesh.GetMatrix());
Write2File("shoulder-----",
robot1.m_shoulderMesh.GetMatrix());
Write2File("upperArm-----",
robot1.m_upperArmMesh.GetMatrix());
Write2File("foreArm-----",
robot1.m_foreArmMesh.GetMatrix());
Write2File("rollArm-----",
robot1.m_rollArmMesh.GetMatrix());
Write2File("pitchHand-----",
robot1.m_pitchHandMesh.GetMatrix());
Write2File("rollHand-----",
robot1.m_rollHandMesh.GetMatrix());
Write2File("*****", false);
Write2File(" ", false);

Write2File("*****", false);
Write2File("LINK MATRIX", false);
Write2File("*****", false);
Write2File("world coordinate2", true);
Write2File("waist2-----",
robot2.m_waistMesh.GetMatrix());
Write2File("shoulder2-----",
robot2.m_shoulderMesh.GetMatrix());
Write2File("upperArm2-----",
robot2.m_upperArmMesh.GetMatrix());
Write2File("foreArm2-----",
robot2.m_foreArmMesh.GetMatrix());
Write2File("rollArm2-----",
robot2.m_rollArmMesh.GetMatrix());
Write2File("pitchHand2-----",
robot2.m_pitchHandMesh.GetMatrix());
Write2File("rollHand2-----",
robot2.m_rollHandMesh.GetMatrix());
Write2File("*****", false);
Write2File(" ", false);
#endregion
}
```

## 8. Function to load texture

```
private void InitTextures()
{
    _texture.LoadTexture(_mediaDir + "Textures\\3.bmp", "RoomTexture", -1, -1,
        CONST_TV_COLORKEY.TV_COLORKEY_NO, true);
    _texture.LoadTexture(_mediaDir + "Textures\\warna.bmp", "ArmTexture", -1,
        -1, CONST_TV_COLORKEY.TV_COLORKEY_NO, true);
}
```

## 9. Function to load part of robot arm

```
#region create meshes
#region waist
m_waistMesh = m_scene.CreateMeshBuilder(m_robotName+"_waist");
m_waistMesh.LoadXFile(m_mediaDir + "kuka_kr15_x\\DH\\waistDH.x");
m_waistMesh.SetTexture(m_globals.GetTex("robotArmTexture"));
m_waistMesh.SetMaterial(m_globals.GetMat("robot material"));
m_waistMesh.SetLightingMode(CONST_TV_LIGHTINGMODE.TV_LIGHTING_MANAGED);
m_waistMesh.SetPosition(m_worldOrigin.x + m_dX, m_worldOrigin.y + m_dY,
    m_worldOrigin.z + m_dZ);
m_waistMesh.SetRotation(m_rotateX, m_rotateY, m_rotateZ);
m_waistMesh.SetScale(m_scale, m_scale, m_scale);
#endregion

#region shoulder
m_shoulderMesh = m_scene.CreateMeshBuilder(m_robotName + "_shoulder");
m_shoulderMesh.LoadXFile(m_mediaDir + "kuka_kr15_x\\DH\\shoulderDH.x");
m_shoulderMesh.SetTexture(m_globals.GetTex("robotArmTexture"));
m_shoulderMesh.SetMaterial(m_globals.GetMat("robot material"));
m_shoulderMesh.SetLightingMode(CONST_TV_LIGHTINGMODE.TV_LIGHTING_MANAGED);
m_shoulderMesh.SetParent(CONST_TV_NODETYPE.TV_NODETYPE_MESH,
    m_waistMesh.GetEntityIndex(), 0);
m_shoulderMesh.SetPosition(m_waistJointLocalPos.x, m_waistJointLocalPos.y,
    m_waistJointLocalPos.z);
m_shoulderMesh.SetScale(m_scale, m_scale, m_scale);
#endregion

#region upper arm
m_upperArmMesh = m_scene.CreateMeshBuilder(m_robotName + "_upperarm");
m_upperArmMesh.LoadXFile(m_mediaDir + "kuka_kr15_x\\DH\\upperarmDH.x");
m_upperArmMesh.SetTexture(m_globals.GetTex("robotArmTexture"));
m_upperArmMesh.SetMaterial(m_globals.GetMat("robot material"));
m_upperArmMesh.SetLightingMode(CONST_TV_LIGHTINGMODE.TV_LIGHTING_MANAGED);
m_upperArmMesh.SetParent(CONST_TV_NODETYPE.TV_NODETYPE_MESH,
    m_shoulderMesh.GetEntityIndex(), 0);
m_upperArmMesh.SetPosition(m_shoulderJointLocalPos.x,
    m_shoulderJointLocalPos.y, m_shoulderJointLocalPos.z);
m_upperArmMesh.SetScale(m_scale, m_scale, m_scale);
#endregion
```

```

#region forearm
m_foreArmMesh = m_scene.CreateMeshBuilder(m_robotName + "_forearm");
m_foreArmMesh.LoadXFile(m_mediaDir + "kuka_kr15_x\\DH\\foreArmDH.x");
m_foreArmMesh.SetTexture(m_globals.GetTex("robotArmTexture"));
m_foreArmMesh.SetMaterial(m_globals.GetMat("robot material"));
m_foreArmMesh.SetLightingMode(CONST_TV_LIGHTINGMODE.TV_LIGHTING_MANAGED);
m_foreArmMesh.SetParent(CONST_TV_NODETYPE.TV_NODETYPE_MESH,
m_upperArmMesh.GetEntityIndex(), 0);
m_foreArmMesh.SetPosition(m_elbowJointLocalPos.x, m_elbowJointLocalPos.y,
m_elbowJointLocalPos.z);
m_foreArmMesh.SetScale(m_scale, m_scale, m_scale);
#endregion

#region rollarm
m_rollArmMesh = m_scene.CreateMeshBuilder(m_robotName + "_rollarm");
m_rollArmMesh.LoadXFile(m_mediaDir + "kuka_kr15_x\\DH\\rollarmDH.x");
m_rollArmMesh.SetTexture(m_globals.GetTex("robotArmTexture"));
m_rollArmMesh.SetMaterial(m_globals.GetMat("robot material"));
m_rollArmMesh.SetLightingMode(CONST_TV_LIGHTINGMODE.TV_LIGHTING_MANAGED);
m_rollArmMesh.SetParent(CONST_TV_NODETYPE.TV_NODETYPE_MESH,
m_foreArmMesh.GetEntityIndex(), 0);
m_rollArmMesh.SetPosition(m_rollArmJointLocalPos.x,
m_rollArmJointLocalPos.y, m_rollArmJointLocalPos.z);
m_rollArmMesh.SetScale(m_scale, m_scale, m_scale);
#endregion

#region pitch hand
m_pitchHandMesh = m_scene.CreateMeshBuilder(m_robotName + "_pitchhand");
m_pitchHandMesh.LoadXFile(m_mediaDir + "kuka_kr15_x\\DH\\pitchhandDH.x");
m_pitchHandMesh.SetTexture(m_globals.GetTex("robotArmTexture"));
m_pitchHandMesh.SetMaterial(m_globals.GetMat("robot material"));
m_pitchHandMesh.SetLightingMode(CONST_TV_LIGHTINGMODE.TV_LIGHTING_MANAGED);
;
m_pitchHandMesh.SetParent(CONST_TV_NODETYPE.TV_NODETYPE_MESH,
m_rollArmMesh.GetEntityIndex(), 0);
m_pitchHandMesh.SetPosition(m_pitchHandJointLocalPos.x,
m_pitchHandJointLocalPos.y, m_pitchHandJointLocalPos.z);
m_pitchHandMesh.SetScale(m_scale, m_scale, m_scale);
#endregion

#region roll hand
m_rollHandMesh = m_scene.CreateMeshBuilder(m_robotName + "_rollhand");
m_rollHandMesh.LoadXFile(m_mediaDir + "kuka_kr15_x\\DH\\rollhandDH.x");
m_rollHandMesh.SetTexture(m_globals.GetTex("robotArmTexture"));
m_rollHandMesh.SetMaterial(m_globals.GetMat("robot material"));
m_rollHandMesh.SetLightingMode(CONST_TV_LIGHTINGMODE.TV_LIGHTING_MANAGED);
m_rollHandMesh.SetParent(CONST_TV_NODETYPE.TV_NODETYPE_MESH,
m_pitchHandMesh.GetEntityIndex(), 0);
m_rollHandMesh.SetPosition(m_rollHandJointLocalPos.x,
m_rollHandJointLocalPos.y, m_rollHandJointLocalPos.z);
m_rollHandMesh.SetScale(m_scale, m_scale, m_scale);
#endregion

```