

## CHAPTER IV

### SYSTEM DEVELOPMENT

#### 4.1 Research Activities

##### 4.1.1 Research Scenario

In conducting the study required a scenario, the scenario is a sequence of events that will be done in a VM system. This scenario is intended to make research easier in designing the movement of any model used in the VM system. Robot movement is facing a raw table, then take the raw material at the starting point of 0,20,80, the starting point is located on the storage 1. Then the raw material stuck to the end of the robot 1 and then aim and release the raw material at a second point on the conveyor 1, which is at the point 60, 70.80. Then the raw material moves over the conveyor 1 to point 220,70,80 which will be taken by the robot 2. Then the robot 2 will be facing the conveyor 1 and take the raw material. raw material stuck to the end of the robot 2, robot 2 rotates to a point 300, 107, 110 in nc drilling and then release the raw material. Then Nc drilling will process the raw material to be finished product. robot 2 takes the finish product in nc drilling and finished product itself will be attached at the end of the robot 2, robot 2 rotates to a point 220, 70, -70 on the conveyor 2 and release the finished product at that point. Finish product will move on the conveyor 2 to point 60, 70, -70, which will be taken by the robot 1 and put the finished product into storage 2. More detail can be seen in Figure 4.1.

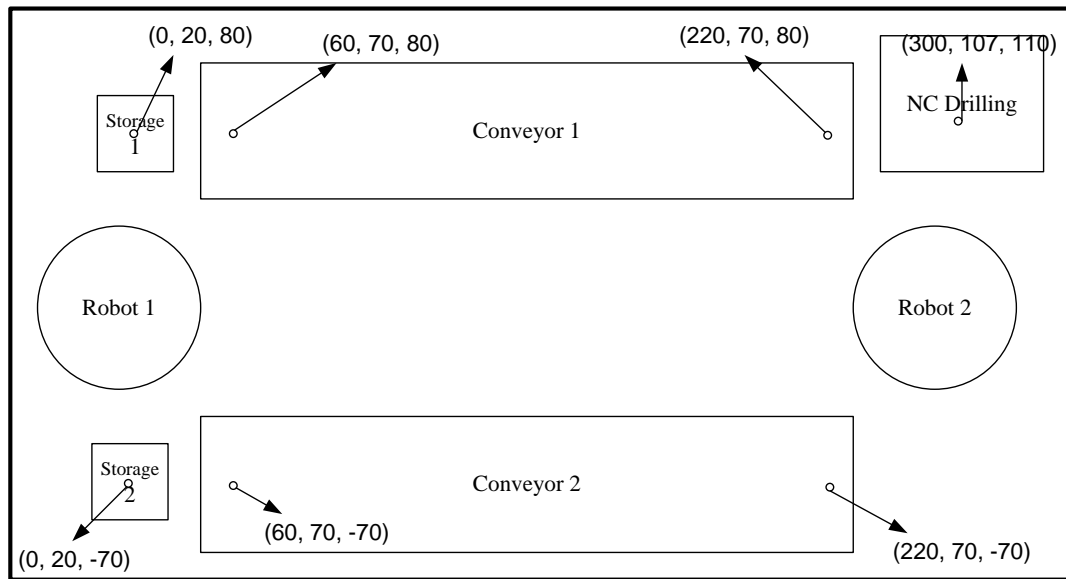


Figure 4.1 Layout Scenario

#### 4.1.2 Determining the Data

In the development of VM takes several objects, including a robot consisting of a waist, shoulder, upper arm, fore arm, arm roll, pitch hand, hand roll, and tools which will then be assembled into a KUKA robot 15. Then using a conveyor, ncdrilling, and box. Waist has a function as the support of all part robot arm, shoulder is the second part of the robot arm that can rotate in a circle or twisting according to the magnitude of the desired degree, upper arm can rotate in a rolling or revolving in accordance with the amount of the desired degree, fore arm can move in a rotational fit with the amount of the desired degree, roll arm can rotate in a circle or twisting in accordance with the amount of the desired angle, pitch hand can rotate in a circle or twisting in accordance with the amount of the desired angle, roll hand can move in a twisting according to the magnitude of the desired angle, the tool is an additional section serves as a pick up and drop objects to be moved. Parts of the robot can be seen in Figure 4.2.

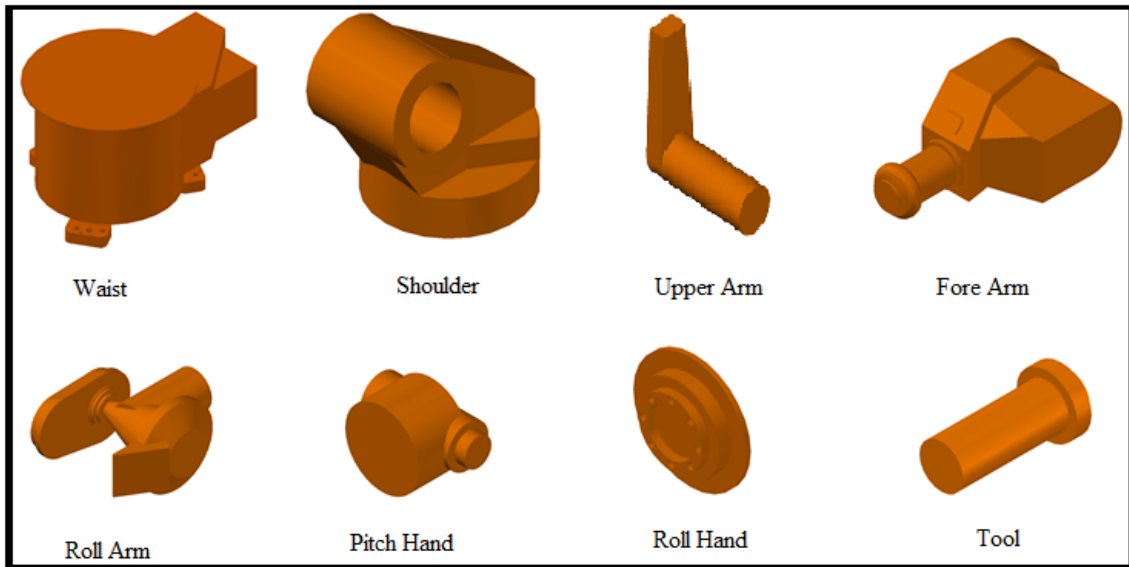


Figure 4.2 Part of Robot Arm

Parts of the robot arm above will be compiled into a KUKA robot arm 15. KUKA Robot Arm 15 is shown in Figure 4.3

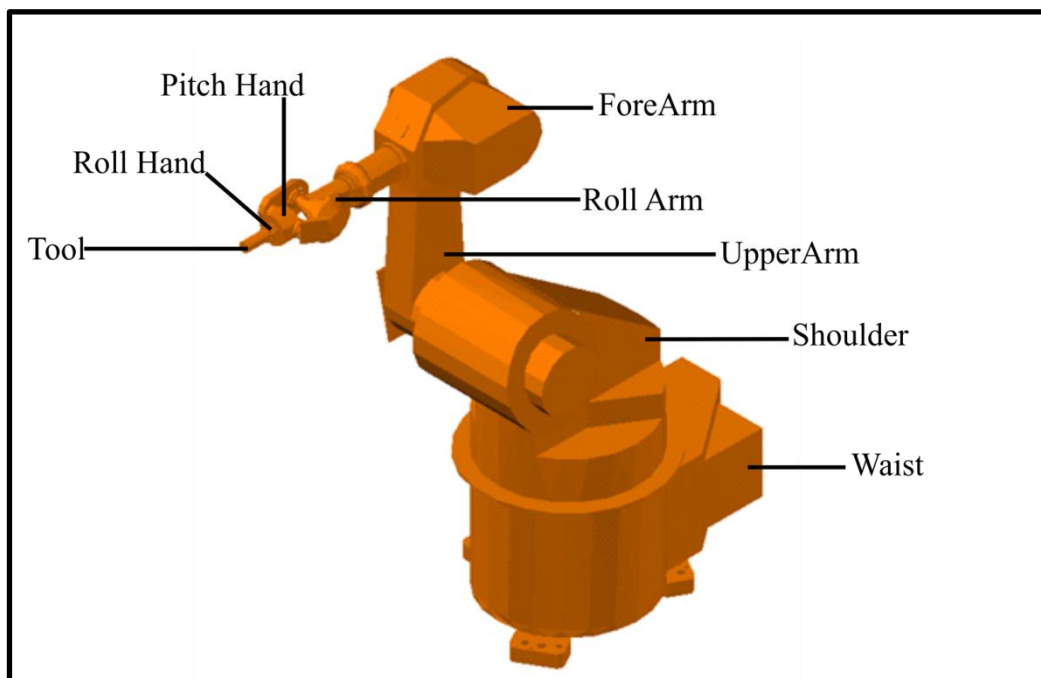


Figure 4.3 Robot Arm KUKA 15

In addition to the robot arm, researchers used a virtual conveyor, ncdrilling virtual and virtual box. Conveyor, ncdrilling and the box can be seen in Figure 4.4; 4.5; and 4.6.

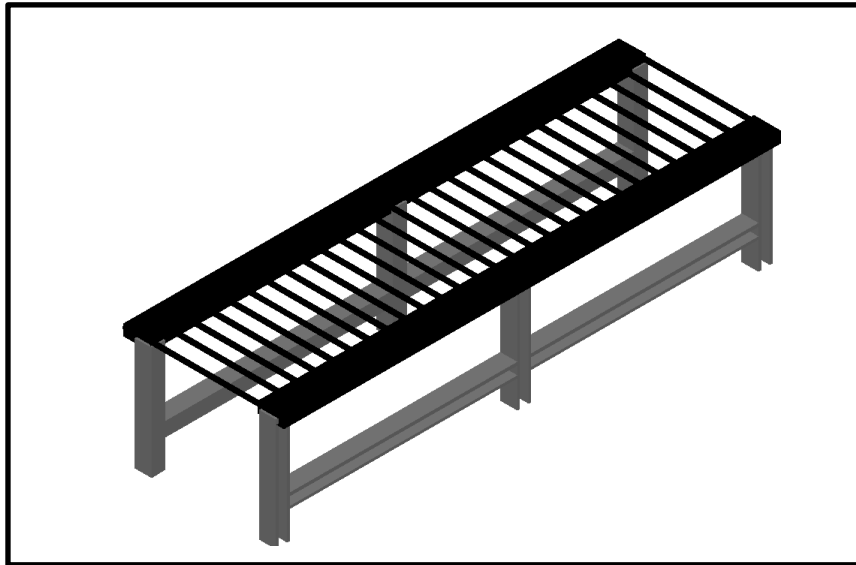


Figure 4.4 Virtual conveyor

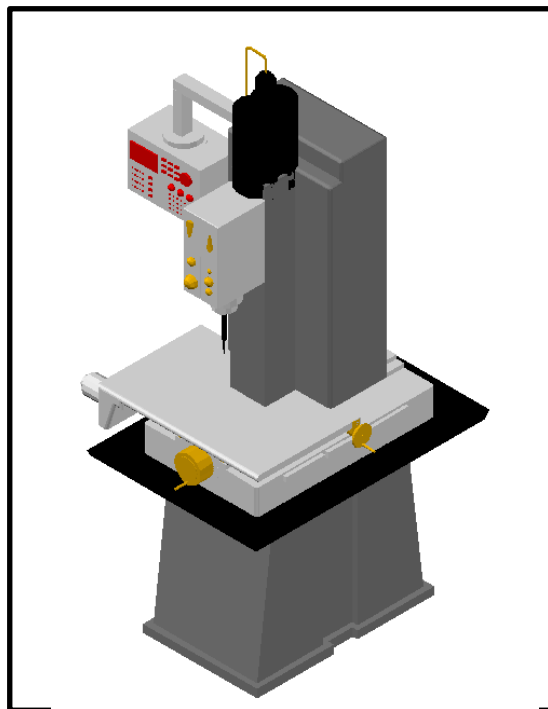


Figure 4.5 Virtual NCDrilling

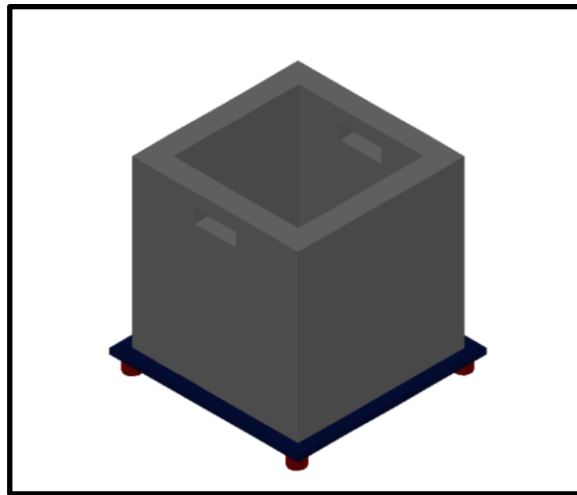


Figure 4.6 Virtual Box

All of the sections above will be prepared in accordance with a layout that has been set previously. Layout of parts above can be seen in Figure 4.7.



Figure 4.7 Layout of VM parts

### 4.1.3 Develop Control system

The purpose of building a control system is to control the movement of the VM model according to the scenario. Control system are used specifically on the robot arm. This research choosing inverse kinematic to move the robot arm. Inverse Kinematic Given an orientation and position for a robotic arm, it want to show That by finding all possible combinations of joint settings, it can place the hand of the robot at this exact point and orientation. This research choosing the method because the inverse kinematic destination point of the robot is known in advance. To apply the method, researcher need a basic matrix for each part of the robot arm to move the robot. Matrixs is obtained from studies previously. This matrix can be seen in Table 4.1.

Table 4.1 Basic Matrix for Each Parts of Robot

<b>Robot 1</b>	<b>Robot 2</b>
waist----- -000.10 0000.00 0000.00 0000.00 0000.00 0000.10 0000.00 0000.00 0000.00 0000.00 -000.10 0000.00 0010.00 0000.00 0010.00 0001.00	waist2----- 0000.10 0000.00 0000.00 0000.00 0000.00 0000.10 0000.00 0000.00 0000.00 0000.00 0000.10 0000.00 0280.00 0000.00 0010.00 0001.00
shoulder----- -000.10 0000.00 0000.00 0000.00 0000.00 0000.10 0000.00 0000.00 0000.00 0000.00 -000.10 0000.00 0010.00 0042.10 0010.00 0001.00	shoulder2----- 0000.10 0000.00 0000.00 0000.00 0000.00 0000.10 0000.00 0000.00 0000.00 0000.00 0000.10 0000.00 0280.00 0042.10 0010.00 0001.00
upperArm----- -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 0079.09 -007.50 0001.00	upperArm2----- 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 0079.09 0027.50 0001.00
foreArm----- -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 0134.09 0001.41 0001.00	foreArm2----- 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 0134.09 0018.59 0001.00

rollArm----- -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 0166.06 0010.00 0001.00	rollArm2----- 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 0166.06 0010.00 0001.00
pitchHand----- -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 0185.66 0004.11 0001.00	pitchHand2----- 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 0185.66 0015.89 0001.00
rollHand----- -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	rollHand2----- 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

After the matrix is obtain, then this matrix can be implemented into the robot by using a programming language.

#### **A. Designing the Movements of Each Robot Parts**

The inverse kinematic robotics problem has been the focus of kinematic analysis for robot manipulators. In order to determine all possible formations to place the end effector of a robot manipulator at a particular point in space, we must compute the movements associated with each joint variable.

Actually, there is a wide range of movement by a robot to pick up and release an object. Movements is well designed in order to become more efficient robot motion. Here the researchers also designed a robot movement to pick up and release an object. Several variations of the robot motion is:

## 1. MovingPose.

MovingPose is the movement of the robot where the robot will be back at the starting position when it would pick up and release an object. To view the programming code can be seen in the picture below.

```
public bool MovingPose(  
{  
define the sequence for bNext 1 and 2 are false  
  
define target 1 and 2 as the initial pose  
  
calculate distance 1 with subtract the target 1 with  
angle of foreArmMesh;  
calculate distance 2 with subtract the target 2 with  
angle of upperArmMesh;  
if distance 1 more than and equal to 0.2 float  
{  
if target1 less than and equal to angle of foreArmMesh;  
calculate angle of foreArmMesh at z point;  
}  
else bNext1 is true;  
if distance2 more than and equal to 0.2float  
{  
if target2 less than and equal to angle of upperArmMesh;  
calculate angle of upperArmMesh at z point;  
}  
else bNext2 is true;  
return bNext1 and bNext2;  
}
```

Figure 4.8 MovingPose Code



## 2. WasitJointRotate

WaistJointRotate is a movement where the waist will turn towards the goal of the robot to pick up an object. To view the programming code can be seen in the picture below.

```
public bool WaistJointRotate(TVMesh targetMesh)
{
    Define target position;
    Define waist joint calculation;
    Define bResult is false
    If {waist calculation more than and 0,1)
    { shoulder mesh rotate Y axis to the left as 0.21}
    else
    define bResult is true;
    return bResult;
}
}
```

Figure 4.9 WaistJointRotate Code

### 3. ReachShoulder1st

ReachShoulder1st is the movement of the robot where the robot move the shoulder first to pick up and release an object. To view the programming code can be seen in the picture below.

```
public bool ReachShoulder1st(TVMesh targetMesh)
{
    Define target (0f, 0f, 0f);

    define bResult1 is false;
    define bResult2 is false;

    input dShoulderJointAngle as
    ShoulderJointAngleCalculation(target);
    if (dShoulderJointAngle more than 0.1f)
    {
        m_upperArmMesh Rotate Z axis to the left as 0.12f, false}
    else bResult1 is true;

    if (bResult1)
    {
        input dElbowJointAngle as
        ElbowJointAngleCalculation(target);
        if (dElbowJointAngle more than 0.1f)
        {
            m_foreArmMesh Rotate Z axis to the left as 0.12f false);
        }
        else bResult2 is true;
    }
    return (bResult1 & bResult2);
}
```

Figure 4.10 ReachShoulder1st Code

#### 4. ReachElbow1st

ReachElbow1st is the movement of the robot where the robot move the elbow joint first to pick up and release an object. To view the programming code can be seen in the picture below.

```

public bool ReachElbow1st(TVMesh targetMesh)
{
    Define target position(0f, 0f, 0f));

    define bResult1 is false;
    define bResult2 is false;

    input dElbowJointAngle as
    ElbowJointAngleCalculation(target);

    if (dElbowJointAngle more than 0.1f)
    {
        m_foreArmMesh Rotate Z axis to the left as 0.12f,
        false);
    }
    else bResult1 is true;

    if (bResult1)
    {
        float dShoulderJointAngle =
        ShoulderJointAngleCalculation(target);

        if (dShoulderJointAngle more than 0.1f)
        {
            m_foreArmMesh Rotate Z axis to the left as 0.12f,
            false);
        }
        else bResult2 is true;
    }

    return (bResult1 & bResult2);
}

```

Figure 4.11 ReachShoulder1st Code

## 5. ReachSimultantToolVertical

ReachsimultantToolVertical is the movement of the robot where the shoulder, elbow and tool moving simultaneously in a vertical to pick up and release an object. To view the programming code can be seen in the picture below.

```

public bool ReachSimultantToolVertical(TVMesh targetMesh)
{
    Define position(0f, 0f, 0f));

    define bResult is false;

    input dShoulderJointAngle as
    ShoulderJointAngleCalculation(target);
    input dElbowJointAngle as
    ElbowJointAngleCalculation(target);

    if ((dShoulderJointAngle more than 0.1f) &
    (dElbowJointAngle more than 0.1f))
    {
        dShoulderJointAngle =
        ShoulderJointAngleCalculation(target);
        if (dShoulderJointAngle more than 0.1f)
        m_foreArmMesh Rotate Z axis to the left as 0.12f,
        false);

        ApproachVerticalDown();

        dElbowJointAngle as ElbowJointAngleCalculation(target);
        if (dElbowJointAngle more than 0.1f)
        m_foreArmMesh Rotate Z axis to the left as 0.12f,
        false);
        ApproachVerticalDown();

    }
    else bResult is true;

    return bResult;
}

```

Figure 4.12 ReachSimultantToolVertical

## B. Flow Diagram

Below is a flow diagram of a robot in a pick up or release the object.

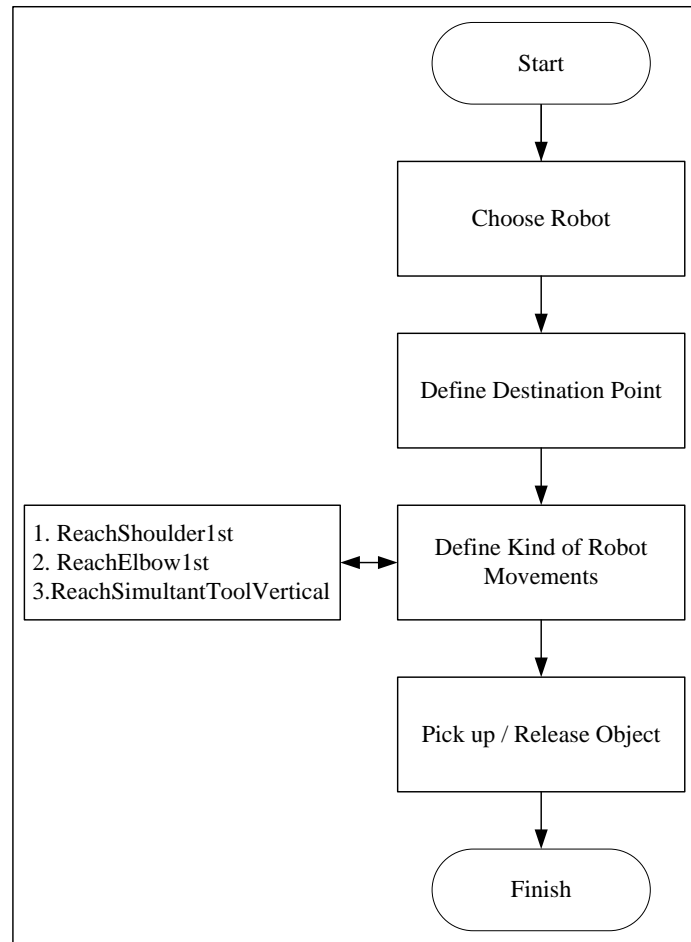


Figure 4.13 Flow Diagram of Robot Assignment

To assign the task to the robot, there are some things to do. The first is selecting the robot, in this study there are two robots that are used. The second is to determine the destination coordinates, this is done so that the end effector of the robot knows where to move. The third is to determine what to use movement, this movement aims to be more efficiently used depending on the location of the object to be retrieved. The fourth is to take or remove the object, the robot used in this study were assigned only to pick up and release objects as a function of material handling.

### **C. SequenceJob**

SequenceJob is a whole sequence of scenarios that have been implemented into the program are made. Starting from Robot 1 took the first point on the raw material storage 1 using ReachShoulder1st movement type and move toward the second point on the conveyor 1 and release the raw material from the type of movement ReachShoulder1st, then the raw material will be moved by the conveyor 1 to the third point. After the raw material up to the third point, the robot 2 will take the raw material from the type of movement ReachSimultantToolVertical then moves toward the fourth point on Ncdrilling, on Ncdrilling raw material will be processed into finished product. After a finished product, robot 2 will take the finished product by using this type of movement ReachSimultantToolVertical then move toward the point of the fifth and finished product release on the conveyor 2. Then the finish product will be driven by the conveyor 2 to the sixth point, after finishing sixth product to the point, robot 1 will take the finished product by using this type of movement and put the finished product ReachElbow1st to seventh points on storage 2 using the type of movement ReachElbow1st.