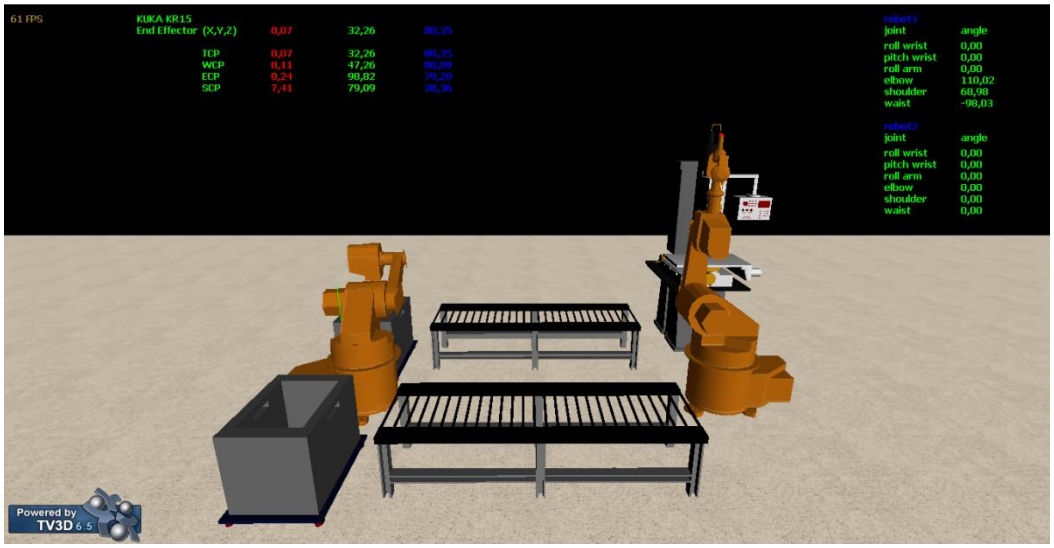
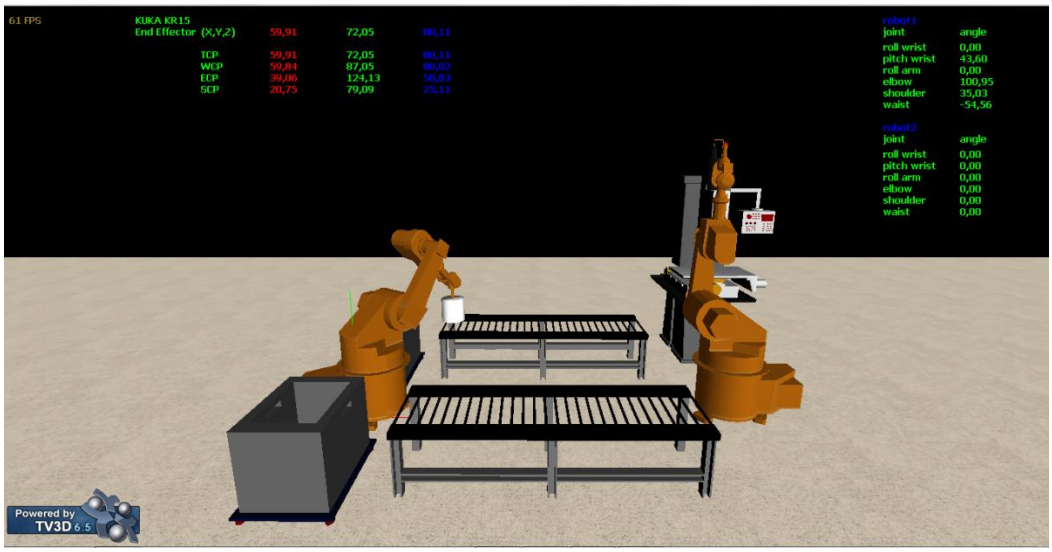
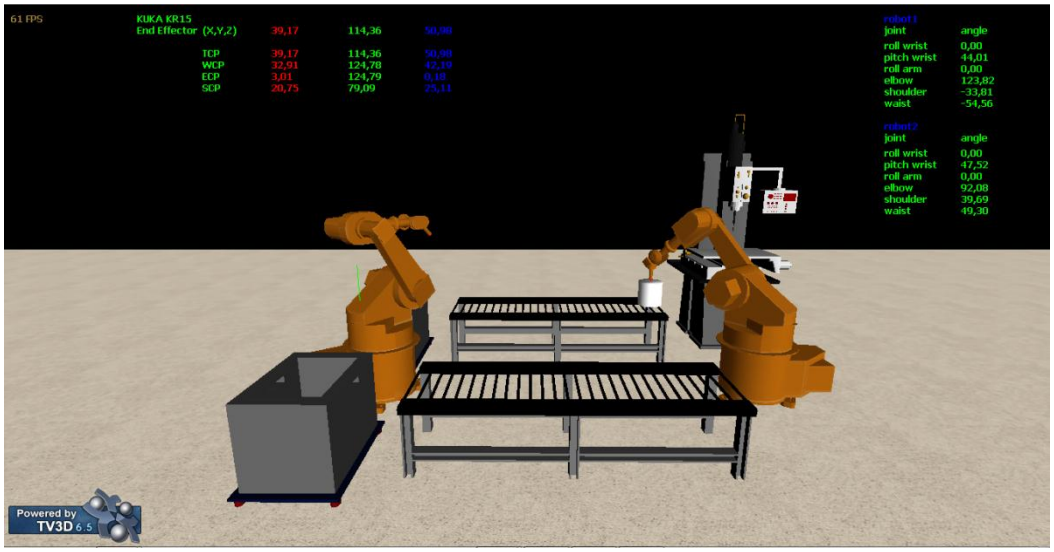
APPENDICES



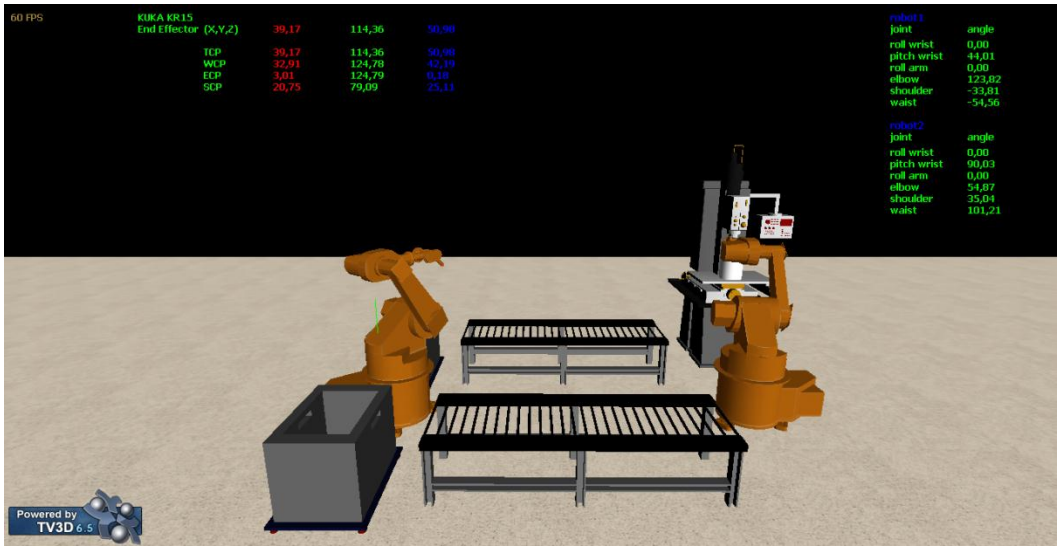
(a)



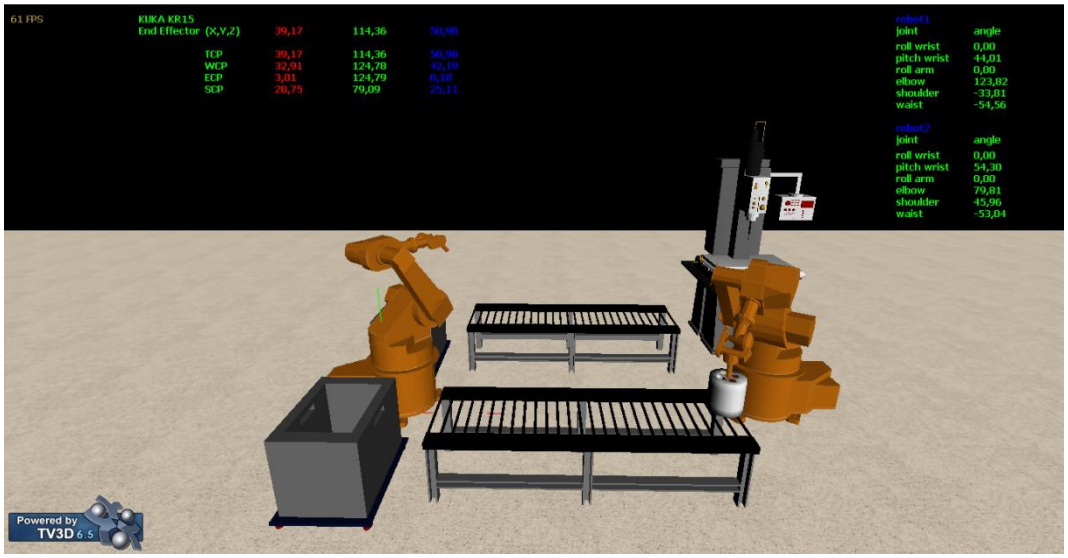
(b)


61 FPS
KUKA KR15
End Effector (X,Y,Z)
39,17
114,36
50,98
TCP
39,17
114,36
50,98
WCP
32,91
124,78
42,19
ECP
3,01
124,79
0,18
SCP
20,75
79,09
25,11
robot1
joint
angle
roll wrist
0,00
pitch wrist
44,01
roll arm
0,00
elbow
123,82
shoulder
-33,81
waist
-54,56
robot2
joint
angle
roll wrist
0,00
pitch wrist
47,52
roll arm
0,00
elbow
92,08
shoulder
39,69
waist
49,30
Powered by
TV3D 6.5

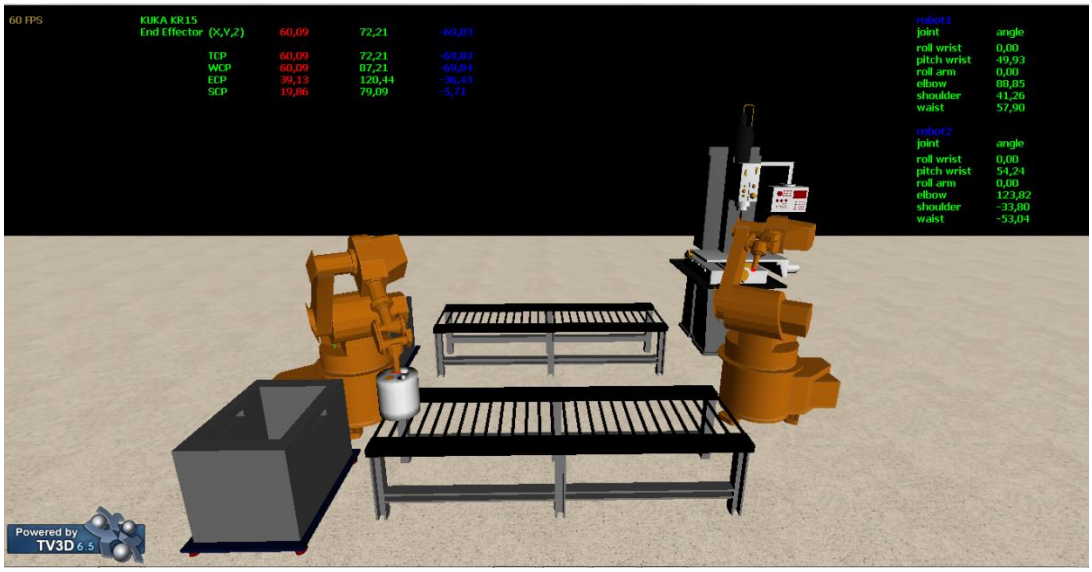
(c)


60 FPS
KUKA KR15
End Effector (X,Y,Z)
39,17
114,36
50,98
TCP
39,17
114,36
50,98
WCP
32,91
124,78
42,19
ECP
3,01
124,79
0,18
SCP
20,75
79,09
25,11
robot1
joint
angle
roll wrist
0,00
pitch wrist
44,01
roll arm
0,00
elbow
123,82
shoulder
-33,81
waist
-54,56
robot2
joint
angle
roll wrist
0,00
pitch wrist
90,03
roll arm
0,00
elbow
54,87
shoulder
35,04
waist
101,21
Powered by
TV3D 6.5

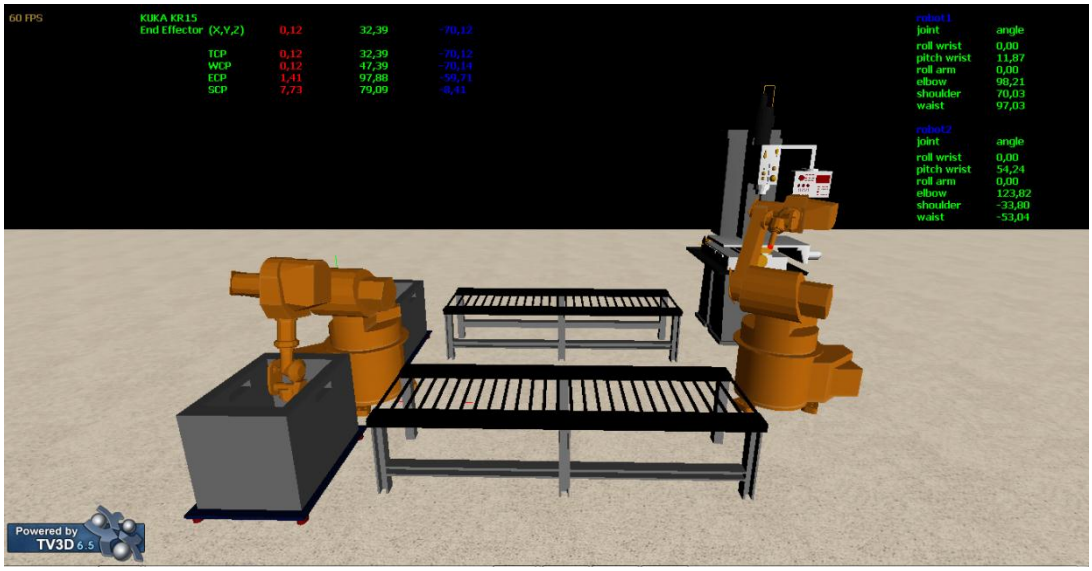
(d)


61 FPS
KUKA KR15
End Effector (X,Y,Z) 39,17 114,36 50,98
TCP 39,17 114,36 50,98
WCP 32,91 124,78 42,19
ECP 3,01 124,79 0,18
SCP 20,75 79,09 25,11
robot1
joint angle
roll wrist 0,00
pitch wrist 44,01
roll arm 0,00
elbow 123,82
shoulder -33,81
waist -54,56
robot2
joint angle
roll wrist 0,00
pitch wrist 54,30
roll arm 0,00
elbow 79,81
shoulder 45,96
waist -53,04
Powered by
TV3D 6.5

(e)


60 FPS
KUKA KR15
End Effector (X,Y,Z) 60,09 72,21 -69,83
TCP 60,09 72,21 -69,83
WCP 60,09 87,21 -60,04
ECP 39,13 120,44 -36,43
SCP 19,86 79,09 -5,71
robot1
joint angle
roll wrist 0,00
pitch wrist 49,93
roll arm 0,00
elbow 88,85
shoulder 41,26
waist 57,90
robot2
joint angle
roll wrist 0,00
pitch wrist 54,24
roll arm 0,00
elbow 123,82
shoulder -33,80
waist -53,04
Powered by
TV3D 6.5

(f)



(g)

a,b,c,d,e,f,g All Process of VM System

Programming Code

```
#region ------ DECLARE OBJECTS FIELD ------
        //mesh
        private TVMesh _roomMesh;
        private TVMesh _floorMesh;
        private TVMesh _cylinderMesh;
        private TVMesh[] _BoxMesh = new TVMesh[2];
        private TVMesh _NcMesh;
        private TVMesh _targetObjectMesh;
        private TVMesh _ObjectMesh;
        private TVMesh[] _conveyorMesh = new TVMesh[2]; //conveyor
        private TVMesh _pointMesh = new TVMesh();

        //home position
        private TV_3DVECTOR _HOMEpos;
        private TV_3DVECTOR _HOMEposDir;

        //end effector position, direction
        private TV_3DVECTOR _endEffCurrentPos;
        private TV_3DVECTOR _endEffCurrentDir;
        private float _endEff2WorldOrigin;

        private TV_3DVECTOR _targetPos;
        private TV_3DVECTOR _targetPosDir;

        //cylinder position
        private TV_3DVECTOR _ptA0 = new TV_3DVECTOR(60, 40, 80);
        private TV_3DVECTOR _ptA1 = new TV_3DVECTOR(60, 70, 80);

        private TV_3DVECTOR _ptB0 = new TV_3DVECTOR(220, 50, 80);
        private TV_3DVECTOR _ptB1 = new TV_3DVECTOR(220, 70, 80);

        private TV_3DVECTOR _ptC0 = new TV_3DVECTOR(0, 20, 70);
```

```
        private TV_3DVECTOR _ptC1 = new TV_3DVECTOR(0, 30, 70);

        private TV_3DVECTOR _ptD0 = new TV_3DVECTOR(300, 100, 110);
        private TV_3DVECTOR _ptD1 = new TV_3DVECTOR(300, 107, 110);

        private TV_3DVECTOR _ptE0 = new TV_3DVECTOR(220, 20, -70);
        private TV_3DVECTOR _ptE1 = new TV_3DVECTOR(220, 70, -70);

        private TV_3DVECTOR _ptF0 = new TV_3DVECTOR(60, 50, -70);
        private TV_3DVECTOR _ptF1 = new TV_3DVECTOR(60, 70, -70);

        private TV_3DVECTOR _ptG0 = new TV_3DVECTOR(0, 20, -70);
        private TV_3DVECTOR _ptG1 = new TV_3DVECTOR(0, 30, -70);

        private TV_3DVECTOR _cylinderPosition = new TV_3DVECTOR();
        private TV_3DVECTOR _cylinderDestination = new TV_3DVECTOR();
        private TV_3DVECTOR _cylinderDirection = new TV_3DVECTOR();
        float _cylinderAngleY = new float();

        private TV_3DVECTOR _ObjectPosition = new TV_3DVECTOR();
        private TV_3DVECTOR _ObjectDestination = new TV_3DVECTOR();
        private TV_3DVECTOR _ObjectDirection = new TV_3DVECTOR();


        private TV_3DVECTOR dVector = new TV_3DVECTOR();
        private TV_3DVECTOR dV2 = new TV_3DVECTOR();

        //joint position, world coordinate
        private TV_3DVECTOR _worldOrigin = new TV_3DVECTOR(0, 0, 0);
        #endregion

#region ------ DECLARE VARIABLES AND CONSTANT FIELD ------
        //directory
        private string _mediaDir = "D:\\RobotProject\\COMMON\\";

        //scale
        private float _scale = 0.1f;

        //simulation loop flag
        private bool _doLoop = true;

        //camera
        private float _cameraPosX;
        private float _cameraPosY;
        private float _cameraPosZ;
        private float _cameraLookAtX;
        private float _cameraLookAtY;
        private float _cameraLookAtZ;

        //table
        private TVMesh[] _ptMesh = new TVMesh[5];
        private float[] _ptAngleXZ = new float[5];

        private int _intRed = new TV_COLOR(1.0f, 0.0f, 0.0f, 1).GetIntColor();
        private int _intGreen = new TV_COLOR(0.0f, 1.0f, 0.0f, 1).GetIntColor();
        private int _intBlue = new TV_COLOR(0.0f, 0.0f, 1.0f, 1).GetIntColor();

        //cylinder dimension
        private float _radius = 10f;
        private float _height = 20f;

        private string _fmt = "0.#0;-0.#0";
```

```
        private float _fTime;            //tick value

        private bool _flagENTER = false;

        #endregion


#region ------- INITIATE FUNCTIONS FIELD -------

        private void InitEngine()
        {
            _engine.AllowMultithreading(true);
            _engine.SetDebugMode(true, true);
            _engine.SetDebugFile(Application.StartupPath + "\\debugfile.txt");
            _engine.Init3DWindowed(this.panel1.Handle);
            _engine.DisplayFPS(true);
            _engine.SetAngleSystem(CONST_TV_ANGLE.TV_ANGLE_DEGREE);
            _engine.SetVSync(true);

_engine.SetWatermarkParameters(CONST_TV_WATERMARKPLACE.TV_WATERMARK_BOTTOMLEFT,
0.001f);
        }

        private void InitScene(string strRenderMode)
        {
            if (strRenderMode.ToUpper() == "POINT")
                _scene.SetRenderMode(CONST_TV_RENDERMODE.TV_POINT);

            if (strRenderMode.ToUpper() == "LINE")
                _scene.SetRenderMode(CONST_TV_RENDERMODE.TV_LINE);

            if (strRenderMode.ToUpper() == "SOLID")
                _scene.SetRenderMode(CONST_TV_RENDERMODE.TV_SOLID);

            _scene.SetShadowParameters(new TV_COLOR(0f, 0f, 0f, 0.5f).GetIntColor(),
false);

        }

        private void InitInputs()
        {
            //input keyboard, mouse
            _input.Initialize(true, false);
        }

        //*******************************
        public void InitMeshes()
        {
            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();

            #region conveyor
            for (int i = 0; i <= 1; i++)
            {
```

```
                _conveyorMesh[i] = new TVMesh();
                _conveyorMesh[i] = _scene.CreateMeshBuilder("conveyor");
                _conveyorMesh[i].LoadXFile(_mediaDir + "kuka_kr15_x\\conveyor1.x");
                _texture.LoadTexture(String.Format(_mediaDir + "Textures\\black.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);

                #region box
                _BoxMesh[i] = _scene.CreateMeshBuilder("graybox");
                _BoxMesh[i].LoadXFile(_mediaDir + "kuka_kr15_x\\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

            }
            #endregion


            #region cylinder
            _cylinderMesh = _scene.CreateMeshBuilder("Cylinder1");
            _cylinderMesh.CreateCylinder(_radius, _height, 30, true);//iPrecision!
jumlah sisi polygon dlm lingkaran, makin banyak makin halus
            _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

            #region NCDrilling
            _NcMesh = _scene.CreateMeshBuilder("ncDrilling");
            _NcMesh.LoadXFile(_mediaDir + "kuka_kr15_x\\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

            #region targetObject
            _targetObjectMesh = _scene.CreateMeshBuilder("targetObject");
            _targetObjectMesh.CreateSphere(1.5f);
            _targetObjectMesh.SetColor(_intRed);

_targetObjectMesh.SetLightingMode(CONST_TV_LIGHTINGMODE.TV_LIGHTING_MANAGED);
            //_targetObjectMesh.SetRotation(180f, 0, 0); //Y axis down
            _targetObjectMesh.SetPosition(300f, 90f + 20f, 110f);
            _targetObjectMesh.SetScale(1f, 1f, 1f);
            _targetObjectMesh.Enable(false);
            #endregion

            #region Object
```

```
            _ObjectMesh = _scene.CreateMeshBuilder("Object");
            _ObjectMesh.LoadXFile(_mediaDir + "kuka_kr15_x\\whitecylinder.x");
           // _ObjectMesh.LoadXFile(_mediaDir + "kuka_kr15_x\\cylinderholes.x");
            _ObjectMesh.SetParent(CONST_TV_NODETYPE.TV_NODETYPE_MESH,
_targetObjectMesh.GetIndex(), 0);
            _ObjectMesh.SetPosition(0, 0, 0);
            _ObjectMesh.SetScale(1f, 1f, 1f);
            _ObjectMesh.SetRotation(90f, 0, 0);
            _ObjectMesh.SetLightingMode(CONST_TV_LIGHTINGMODE.TV_LIGHTING_MANAGED);
            _ObjectMesh.Enable(false);
            #endregion



            #region point mesh
            _ptMesh[1] = _scene.CreateMeshBuilder("ptA");
            _ptMesh[1].CreateSphere(2.5f);
            _ptMesh[1].SetPosition(60f, 70f, 80f);
            _ptMesh[1].SetColor(_intRed);
            _ptMesh[1].SetLightingMode(CONST_TV_LIGHTINGMODE.TV_LIGHTING_MANAGED);
            //_ptMesh[1].Enable(false);
            _ptAngleXZ[1] = _mathLibrary.Direction2Ang(_ptMesh[1].GetPosition().x,
_ptMesh[1].GetPosition().z);

            _ptMesh[2] = _scene.CreateMeshBuilder("ptB");
            _ptMesh[2].CreateSphere(2.5f);
            _ptMesh[2].SetPosition(220f, 70f, 80f);
            _ptMesh[2].SetColor(_intGreen);
            _ptMesh[2].SetLightingMode(CONST_TV_LIGHTINGMODE.TV_LIGHTING_MANAGED);
            //_ptMesh[2].Enable(false);
            _ptAngleXZ[2] = _mathLibrary.Direction2Ang(_ptMesh[2].GetPosition().x,
_ptMesh[2].GetPosition().z);

            _ptMesh[3] = _scene.CreateMeshBuilder("ptC");
            _ptMesh[3].CreateSphere(2.5f);
            _ptMesh[3].SetPosition(220f, 70f, -70f);
            _ptMesh[3].SetColor(_intBlue);
            _ptMesh[3].SetLightingMode(CONST_TV_LIGHTINGMODE.TV_LIGHTING_MANAGED);
            //_ptMesh[3].Enable(false);
            _ptAngleXZ[3] = _mathLibrary.Direction2Ang(_ptMesh[3].GetPosition().x,
_ptMesh[3].GetPosition().z);

            _ptMesh[4] = _scene.CreateMeshBuilder("ptD");
            _ptMesh[4].CreateSphere(2.5f);
            _ptMesh[4].SetPosition(60f, 70f, -70f);
            _ptMesh[4].SetColor(new TV_COLOR(0.2f, 0.5f, 0.1f, 1f).GetIntColor());
            _ptMesh[4].SetLightingMode(CONST_TV_LIGHTINGMODE.TV_LIGHTING_MANAGED);
            //_ptMesh[4].Enable(false);
            _ptAngleXZ[4] = _mathLibrary.Direction2Ang(_ptMesh[4].GetPosition().x,
_ptMesh[4].GetPosition().z);


            _pointMesh = _scene.CreateMeshBuilder("ptE");
            _pointMesh.CreateSphere(2.5f);
            _pointMesh.SetPosition(0f, 20f, 70f);
            _pointMesh.SetColor(new TV_COLOR(0.2f, 0.5f, 0.1f, 1f).GetIntColor());
            _pointMesh.SetLightingMode(CONST_TV_LIGHTINGMODE.TV_LIGHTING_MANAGED);
            #endregion

            //write mesh matrix to a file
            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);

                #endregion
            }

        }
        //*********************************

        private void InitLights()
        {
            _lights.CreateDirectionalLight(new TV_3DVECTOR(-1f, -1f, 1f), 1f, 1f, 1f,
"frontLamp");
        }

        private void InitCamera()
        {
            _cameraPosX = 100;
            _cameraPosY = 100;
            _cameraPosZ = -250;

            _cameraLookAtX = 50;
            _cameraLookAtY = 50;
            _cameraLookAtZ = 250;

            _camera = _scene.GetCamera();
            _camera.SetViewFrustum(60, 1000, 0.1f);
            _camera.SetPosition(_cameraPosX, _cameraPosY, _cameraPosZ);
            _camera.SetLookAt(_cameraLookAtX, _cameraLookAtY, _cameraLookAtZ);
        }

        private void InitViewport()
        {
            _viewport = _engine.CreateViewport(this.Handle, "viewport1");
            _viewport.SetCamera(_camera);
        }

        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);
```

```
        }

        private void InitMaterials()
        {
            int idMat;
            idMat = _materials.CreateLightMaterial(1, 1, 1, 1, 0.015f, 1, "basic
material");
            _materials.SetEmissive(idMat, 0.24f, 0.24f, 0.24f, 1f);
            _materials.SetSpecular(idMat, 1f, 1f, 1f, 1f);
        }

        private void InitRoom()
        {
            _roomMesh = (TVMesh)_scene.CreateMeshBuilder("roommesh");
            _floorMesh = _scene.CreateMeshBuilder("floormesh");


            _floorMesh.AddFloor(_globals.GetTex("RoomTexture"), -5000.0f, -5000 - 0f,
5000.0f, 5000.0f, 0.0f, 100.0f, 100.0f, true);

        }
        #endregion
```