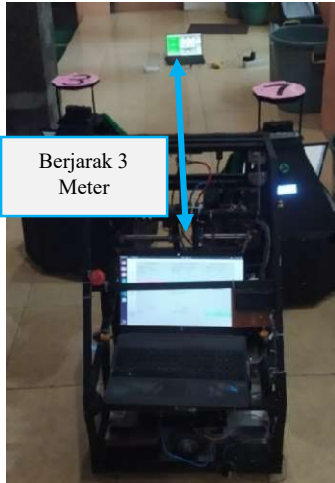
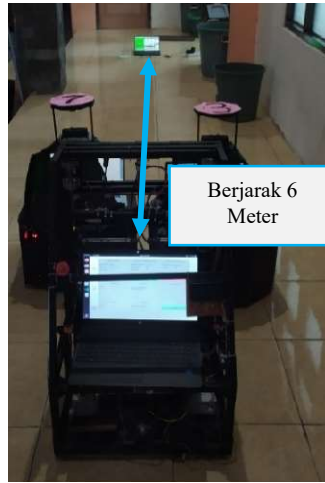


LAMPIRAN A

Dokumentasi Pengujian *Jarak Access Point* dengan robot



Gambar A.1 Pengujian 3 Meter



Gambar A.2 Pengujian 6 Meter



Gambar A.3 Pengujian 9 Meter



Gambar A.4 Pengujian 12 Meter



Gambar A.5 Pengujian 15 Meter

LAMPIRAN B

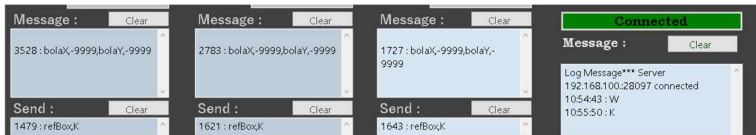
Dokumentasi Pengujian Referee Box

B.1. Konfigurasi tim sebagai tim cyan.

1. Kick Off Cyan



Gambar B. 1 Kick Off Cyan Tampilan pada Referee Box



Gambar B. 2 Kick Off Cyan Tampilan pada Base Station

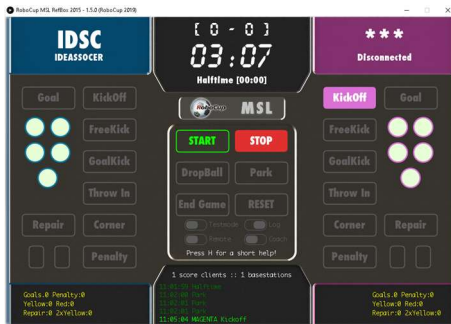


Gambar B. 3 Kick Off Cyan Tampilan pada Interface Robot

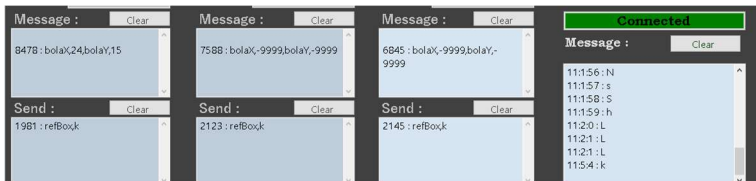


Gambar B. 4 Kick Off Cyan Tampilan pada LCD Robot

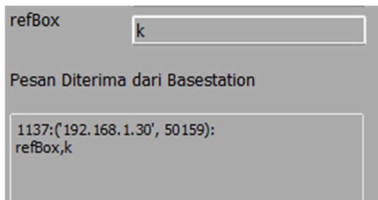
2. Kick Off Magenta



Gambar B. 5 Kick Off Magenta Tampilan pada Referee Box



Gambar B. 6 Kick Off Magenta Tampilan pada Base Station

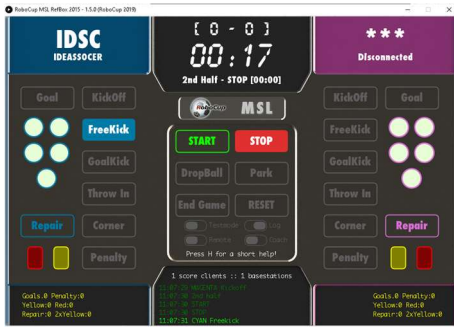


Gambar B. 7 Kick Off Magenta Tampilan pada Interface Robot



Gambar B. 8 Kick Off Magenta Tampilan pada LCD Robot

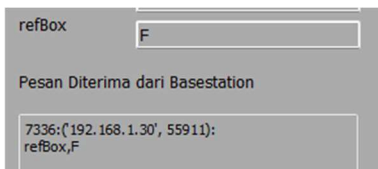
3. Free Kick Cyan



Gambar B. 9 Free Kick Cyan Tampilan pada Referee Box



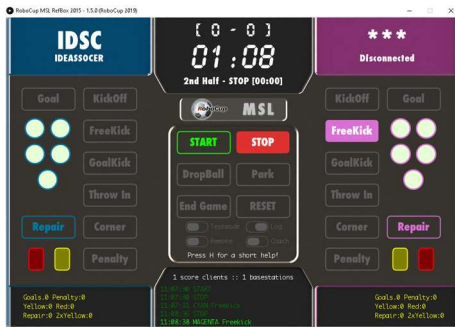
Gambar B. 10 Free Kick Cyan Tampilan pada Base Station



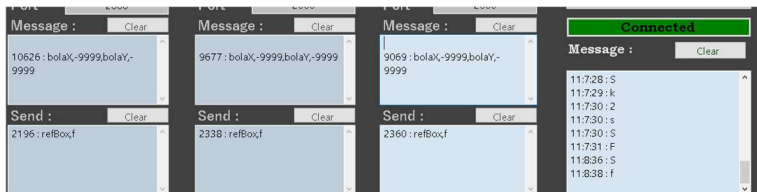
Gambar B. 11 Free Kick Cyan Tampilan pada Interface Robot

Gambar B. 12 Free Kick Cyan Tampilan pada LCD Robot

4. Free Kick Magenta



Gambar B. 13 Free Kick Magenta Tampilan pada Referee Box



Gambar B. 14 Free Kick Magenta Tampilan pada Base Station



Gambar B. 15 Free Kick Magenta Tampilan pada Interface Robot



Gambar B. 16 Free Kick Magenta Tampilan pada LCD Robot

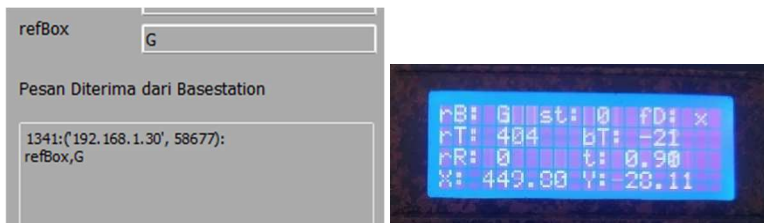
5. Goal Kick Cyan



Gambar B. 17 Goal Kick Cyan Tampilan pada Referee Box



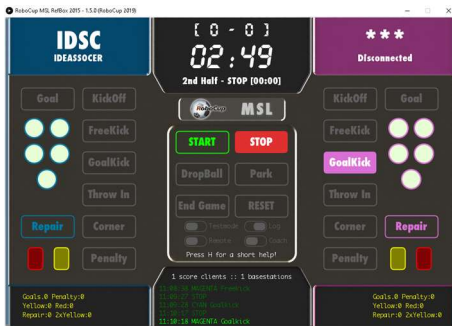
Gambar B. 18 Goal Kick Cyan Tampilan pada Base Station



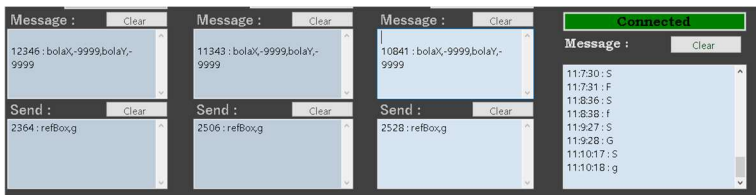
Gambar B. 19 Goal Kick Cyan Tampilan pada Interface Robot

Gambar B. 20 Goal Kick Cyan Magenta Tampilan pada LCD Robot

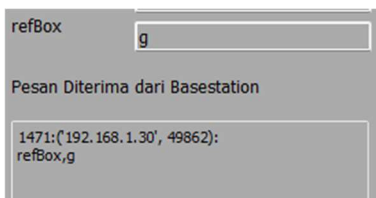
6. Goal Kick Magenta



Gambar B. 21 Goal Kick Magenta Tampilan pada Referee Box



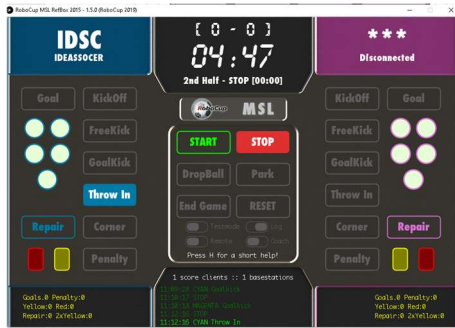
Gambar B. 22 Goal Kick Magenta Tampilan pada Base Station



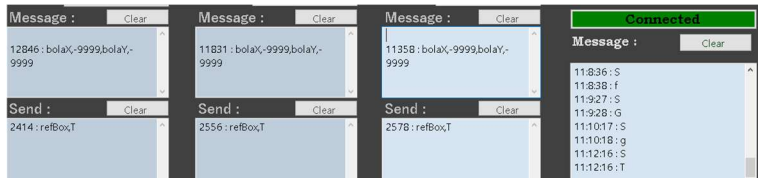
Gambar B. 23 Goal Kick Magenta Tampilan pada Interface Robot

Gambar B. 24 Goal Kick Magenta Tampilan pada LCD Robot

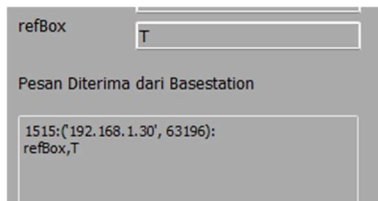
7. Throw In Cyan



Gambar B. 25 Throw In Cyan Tampilan pada Referee Box



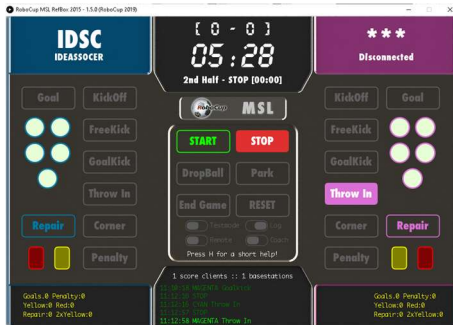
Gambar B. 26 Throw In Cyan Tampilan pada Base Station



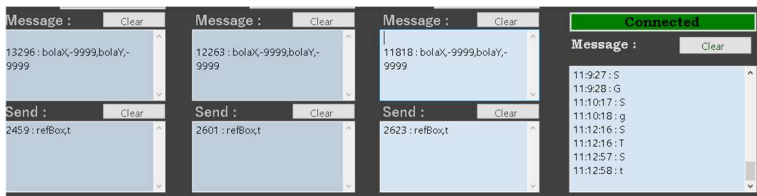
Gambar B. 27 Throw In Cyan Magenta Tampilan pada Interface Robot

Gambar B. 28 Throw In Cyan Tampilan pada LCD Robot

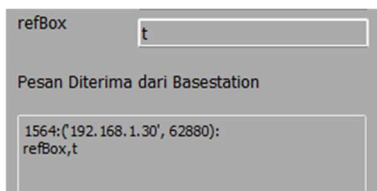
8. Throw In Magenta



Gambar B. 29 Throw In Magenta Tampilan pada Referee Box



Gambar B. 30 Throw In Magenta Tampilan pada Base Station

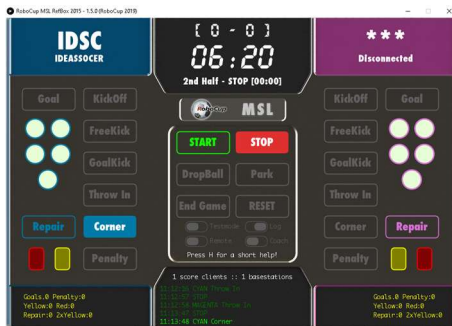


Gambar B. 31 Throw In Magenta Tampilan pada Interface Robot

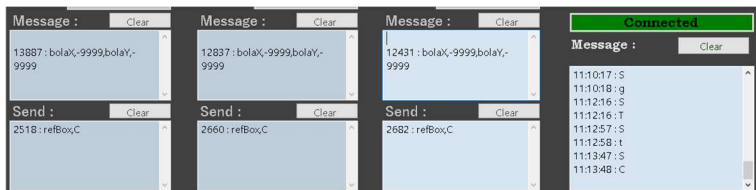


Gambar B. 32 Throw In Magenta Tampilan pada LCD Robot

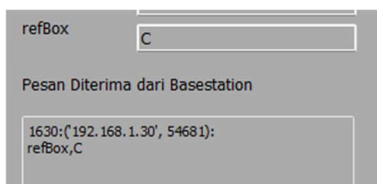
9. Corner Kick Cyan



Gambar B. 33 Corner Kick Cyan Tampilan pada Referee Box



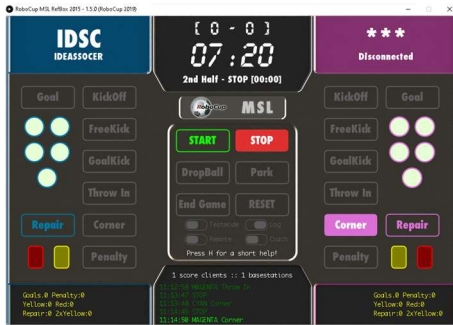
Gambar B. 34 Corner Kick Cyan Tampilan pada Base Station



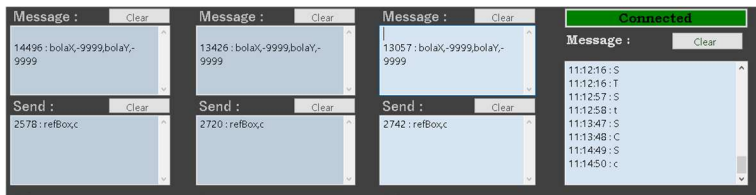
Gambar B. 35 Corner Kick Cyan Tampilan pada Interface Robot

Gambar B. 36 Corner Kick Cyan Tampilan pada LCD Robot

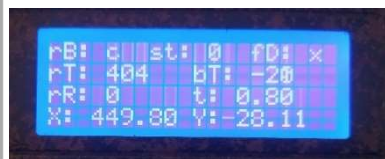
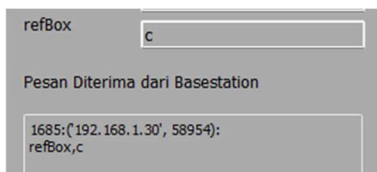
10. Corner Kick Magenta



Gambar B. 37 Corner Kick Magenta Tampilan pada Referee Box



Gambar B. 38 Corner Kick Magenta Tampilan pada Base Station



Gambar B. 39 Corner Kick Magenta Tampilan pada Interface Robot

Gambar B. 40 Corner Kick Magenta Tampilan pada LCD Robot

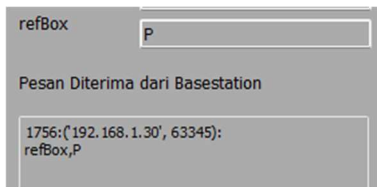
11. Penalty Cyan



Gambar B. 41 Penalty Cyan Tampilan pada Referee Box



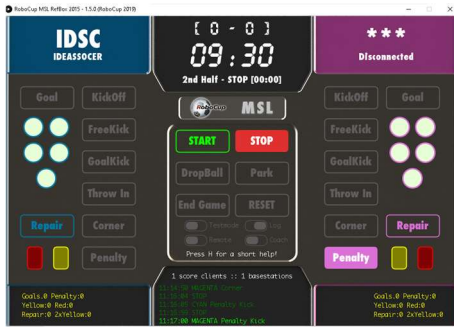
Gambar B. 42 Penalty Cyan Tampilan pada Base Station



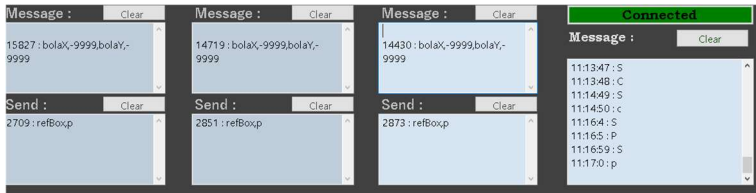
Gambar B. 43 Penalty Cyan Tampilan pada Interface Robot

Gambar B. 44 Penalty Cyan Tampilan pada LCD Robot

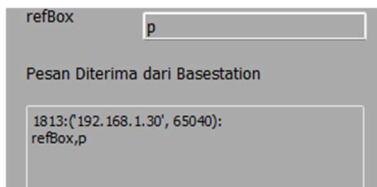
12. Penalty Magenta



Gambar B. 45 Penalty Magenta Tampilan pada Referee Box



Gambar B. 46 Penalty Magenta Tampilan pada Base Station

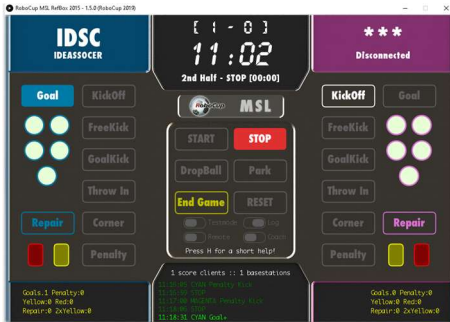


Gambar B. 47 Penalty Magenta Tampilan pada Interface Robot

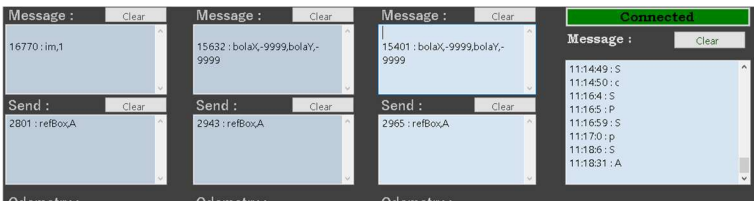


Gambar B. 48 Penalty Magenta Tampilan pada LCD Robot

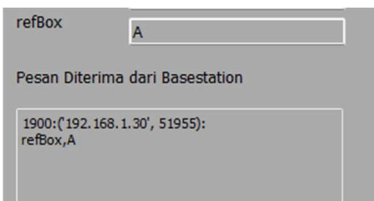
13. Goal Cyan



Gambar B. 49 Goal Cyan Tampilan pada Referee Box



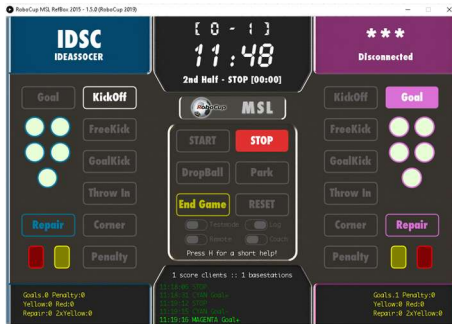
Gambar B. 50 Goal Cyan Tampilan pada Base Station



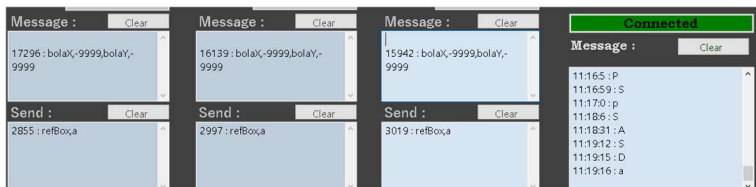
Gambar B. 51 Goal Cyan Tampilan pada Interface Robot

Gambar B. 52 Goal Cyan Tampilan pada LCD Robot

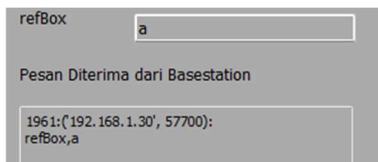
14. Goal Magenta



Gambar B. 53 Goal Magenta Tampilan pada Referee Box



Gambar B. 54 Goal Magenta Tampilan pada Base Station



Gambar B. 55 Goal Magenta Tampilan pada Interface Robot

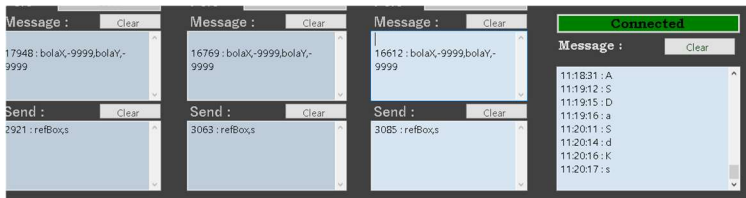


Gambar B. 56 Goal Magenta Tampilan pada LCD Robot

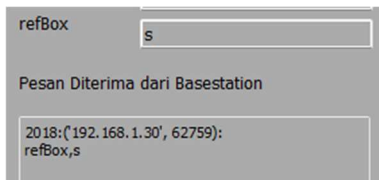
15. Start



Gambar B. 57 Start Tampilan pada Referee Box



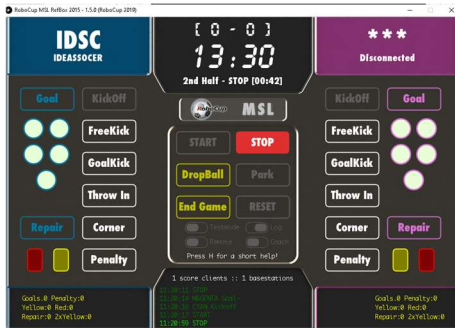
Gambar B. 58 Start Tampilan pada Base Station



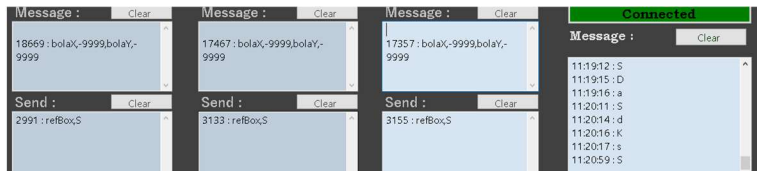
Gambar B. 59 Start Tampilan pada Interface Robot

Gambar B. 60 Start Tampilan pada LCD Robot

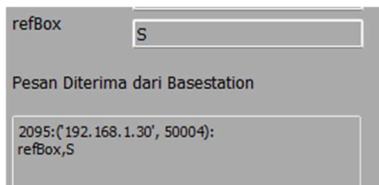
16. Stop



Gambar B. 61 Stop Tampilan pada Referee Box



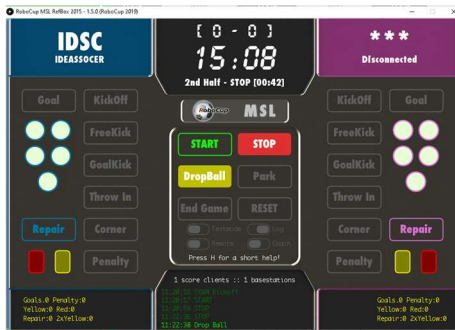
Gambar B. 62 Stop Tampilan pada Base Station



Gambar B. 63 Stop Tampilan pada Interface Robot

Gambar B. 64 Stop Tampilan pada LCD Robot

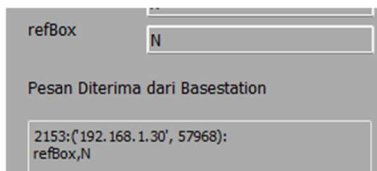
17. Drop Ball



Gambar B. 65 Drop Ball Tampilan pada Referee Box



Gambar B. 66 Drop Ball Tampilan pada Base Station



Gambar B. 67 Drop Ball Tampilan pada Interface Robot

Gambar B. 68 Drop Ball Tampilan pada LCD Robot

18. Park



Gambar B. 69 Park Tampilan pada Referee Box



Gambar B. 70 Park Tampilan pada Base Station



Gambar B. 71 Park Tampilan pada Interface Robot

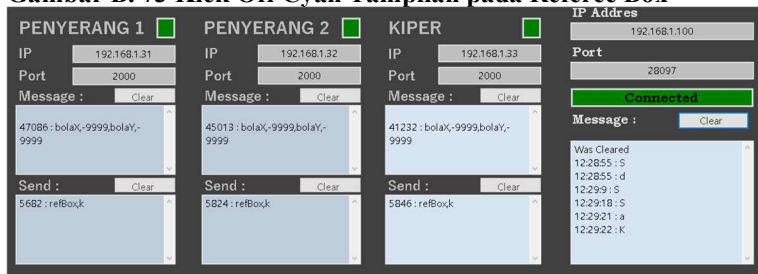
Gambar B. 72 Park Tampilan pada LCD Robot

B.2. Konfigurasi tim sebagai tim Magenta.

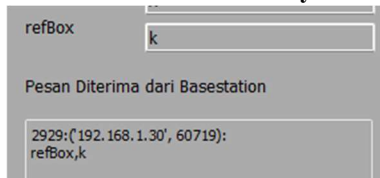
1. Kick Off Cyan



Gambar B. 73 Kick Off Cyan Tampilan pada Referee Box



Gambar B. 74 Kick Off Cyan Tampilan pada Base Station



Gambar B. 75 Kick Off Cyan Tampilan pada Interface Robot

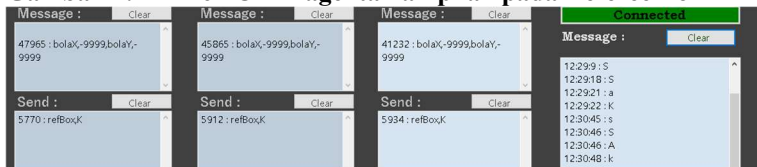


Gambar B. 76 Kick Off Cyan Tampilan pada LCD Robot

2. Kick Off Magenta



Gambar B. 77 Kick Off Magenta Tampilan pada Referee Box



Gambar B. 78 Kick Off Magenta Tampilan pada Base Station

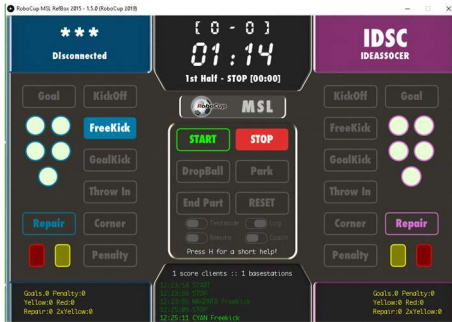


Gambar B. 79 Kick Off Magenta Tampilan pada Interface Robot



Gambar B. 80 Kick Off Magenta Tampilan pada LCD Robot

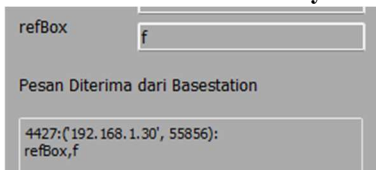
3. Free Kick Cyan



Gambar B. 81 Free Kick Cyan Tampilan pada Referee Box



Gambar B. 82 Free Kick Cyan Tampilan pada Base Station



Gambar B. 83 Free Kick Cyan Tampilan pada Interface Robot

Gambar B. 84 Free Kick Cyan Tampilan pada LCD Robot

4. Free Kick Magenta



Gambar B. 85 Free Kick Magenta Tampilan pada Referee Box



Gambar B. 86 Free Kick Magenta Tampilan pada Base Station



Gambar B. 87 Free Kick Magenta Tampilan pada Interface Robot

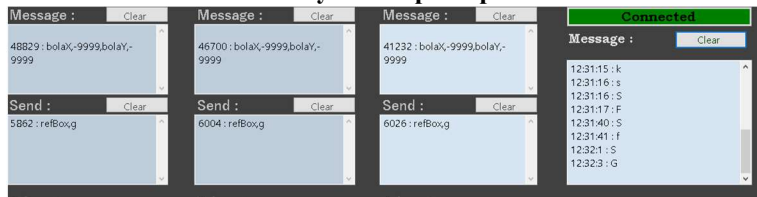


Gambar B. 88 Free Kick Magenta Tampilan pada LCD Robot

5. Goal Kick Cyan



Gambar B. 89 Goal Kick Cyan Tampilan pada Referee Box



Gambar B. 90 Goal Kick Cyan Tampilan pada Base Station



Gambar B. 91 Goal Kick Cyan Tampilan pada Interface Robot



Gambar B. 92 Goal Kick Cyan Tampilan pada LCD Robot

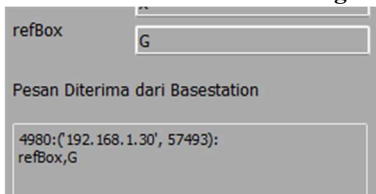
6. Goal Kick Magenta



Gambar B. 93 Goal Kick Magenta Tampilan pada Referee Box



Gambar B. 94 Goal Kick Magenta Tampilan pada Base Station

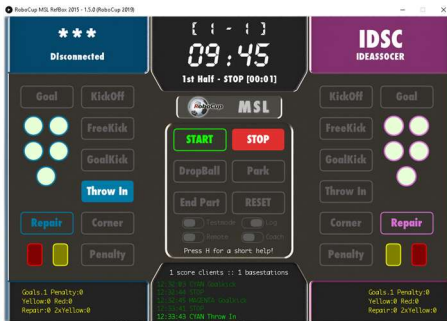


Gambar B. 95 Goal Kick Magenta Tampilan pada Interface Robot

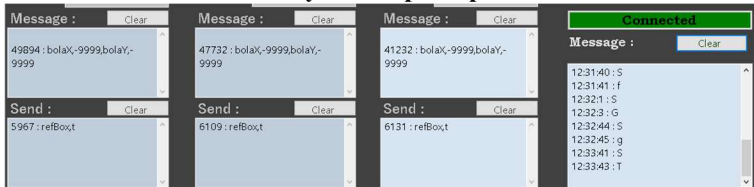


Gambar B. 96 Goal Kick Magenta Tampilan pada LCD Robot

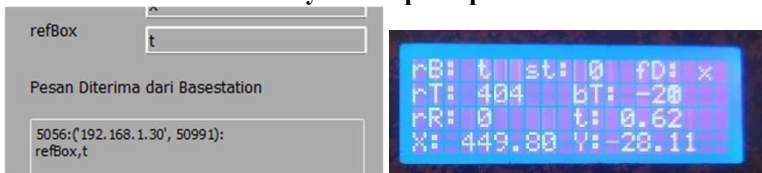
7. Throw In Cyan



Gambar B. 97 Throw In Cyan Tampilan pada Referee Box



Gambar B. 98 Throw In Cyan Tampilan pada Base Station



Gambar B. 99 Throw In Cyan Magenta Tampilan pada Interface Robot

Gambar B. 100 Throw In Cyan Tampilan pada LCD Robot

8. Throw In Magenta



Gambar B. 101 Throw In Magenta Tampilan pada Referee Box



Gambar B. 102 Throw In Magenta Tampilan pada Base Station



Gambar B. 103 Throw In Magenta Tampilan pada Interface Robot

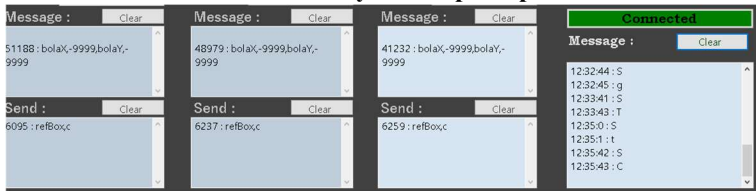


Gambar B. 104 Throw In Magenta Tampilan pada LCD Robot

9. Corner Kick Cyan



Gambar B. 105 Corner Kick Cyan Tampilan pada Referee Box



Gambar B. 106 Corner Kick Cyan Tampilan pada Base Station



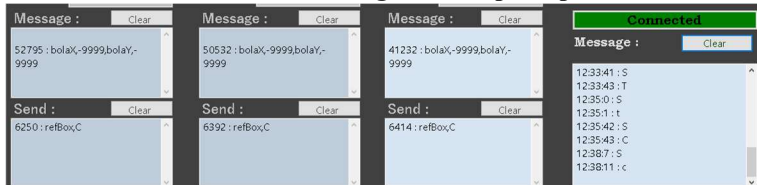
Gambar B. 107 Corner Kick Cyan Tampilan pada Interface Robot

Gambar B. 108 Corner Kick Cyan Tampilan pada LCD Robot

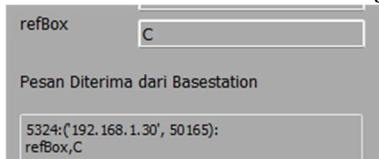
10. Corner Kick Magenta



Gambar B. 109 Corner Kick Magenta Tampilan pada Referee Box



Gambar B. 110 Corner Kick Magenta Tampilan pada Base Station



Gambar B. 111 Corner Kick Magenta Tampilan pada Interface Robot



Gambar B. 112 Corner Kick Magenta Tampilan pada LCD Robot

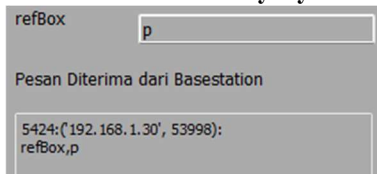
11. Penalty Cyan



Gambar B. 113 Penalty Cyan Tampilan pada Referee Box



Gambar B. 114 Penalty Cyan Tampilan pada Base Station



Gambar B. 115 Penalty Cyan Tampilan pada Interface Robot

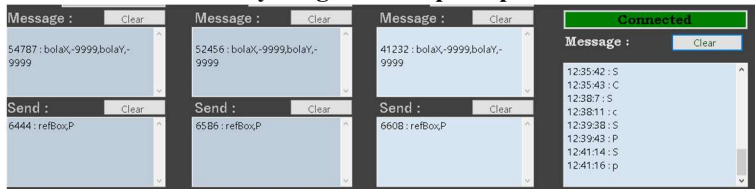


Gambar B. 116 Penalty Cyan Tampilan pada LCD Robot

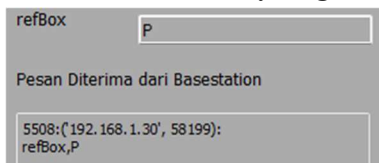
12. Penalty Magenta



Gambar B. 117 Penalty Magenta Tampilan pada Referee Box



Gambar B. 118 Penalty Magenta Tampilan pada Base Station



Gambar B. 119 Penalty Magenta Tampilan pada Interface Robot



Gambar B. 120 Penalty Magenta Tampilan pada LCD Robot

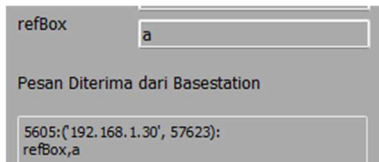
13. Goal Cyan



Gambar B. 121 Goal Cyan Tampilan pada Referee Box



Gambar B. 122 Goal Cyan Tampilan pada Base Station



Gambar B. 123 Goal Cyan Tampilan pada Interface Robot

Gambar B. 124 Goal Cyan Tampilan pada LCD Robot

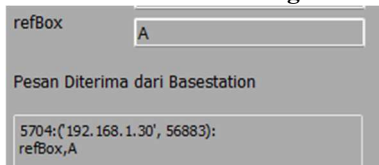
14. Goal Magenta



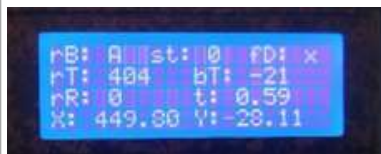
Gambar B. 125 Goal Magenta Tampilan pada Referee Box



Gambar B. 126 Goal Magenta Tampilan pada Base Station

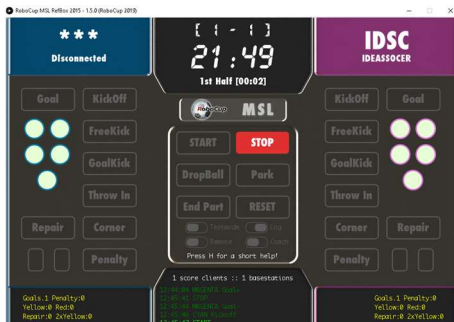


Gambar B. 127 Goal Magenta Tampilan pada Interface Robot

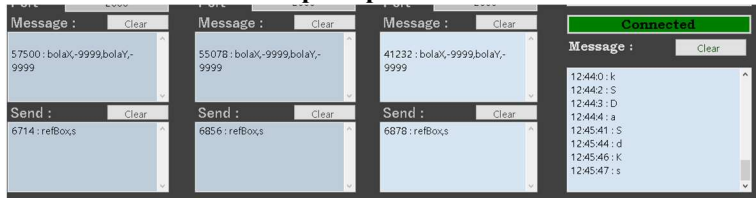


Gambar B. 128 Goal Magenta Tampilan pada LCD Robot

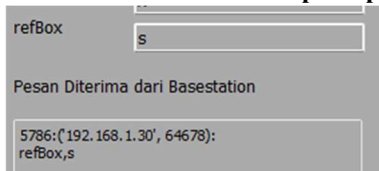
15. Start



Gambar B. 129 Start Tampilan pada Referee Box



Gambar B. 130 Start Tampilan pada Base Station



Gambar B. 131 Start Tampilan pada Interface Robot

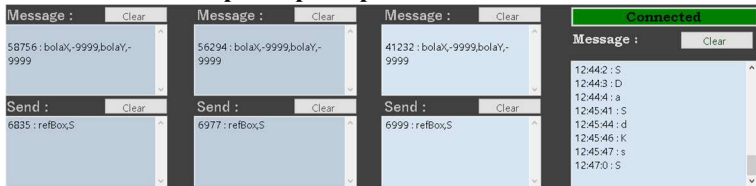


Gambar B. 132 Start Tampilan pada LCD Robot

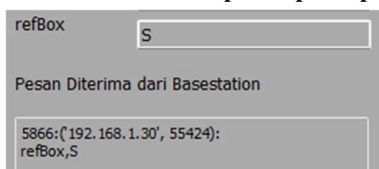
16. Stop



Gambar B. 133 Stop Tampilan pada Referee Box



Gambar B. 134 Stop Tampilan pada Base Station



Gambar B. 135 Stop Tampilan pada Interface Robot



Gambar B. 136 Stop Tampilan pada LCD Robot

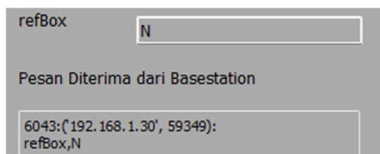
17. Drop Ball



Gambar B. 137 Drop Ball Tampilan pada Referee Box



Gambar B. 138 Drop Ball Tampilan pada Base Station



Gambar B. 139 Drop Ball Tampilan pada Interface Robot

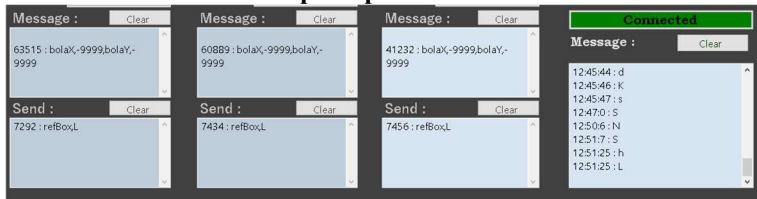


Gambar B. 140 Drop Ball Tampilan pada LCD Robot

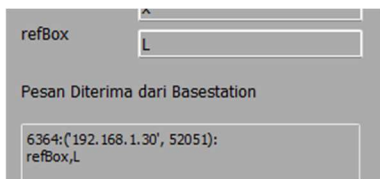
18. Park



Gambar B. 141 Park Tampilan pada Referee Box



Gambar B. 142 Park Tampilan pada Base Station



Gambar B. 143 Park Tampilan pada Interface Robot



Gambar B. 144 Park Tampilan pada LCD Robot

LAMPIRAN C

Listing Program Base Station

```
using SuperSimpleTcp;
using System.Net;
using System.Net.Sockets;
using System.Text;
using Newtonsoft.Json;
using System.Windows.Forms;
using System;
using OpenCvSharp;
using System.Threading;
using OpenCvSharp.Flann;

namespace Bastation_Ideassoccer
{
    public partial class Form1 : Form
    {
        private static int lebarLapangan = 450;
        private static int panjangLapangan = 600;
        private bool doneLoadedForm = false, errorSwitch = true;
        Thread udpThread;
        Thread camP1Thread, camP2Thread, camKThread;
        private bool isRunningCamP1 = false, isRunningCamP2 = false,
isRunningCamK = false;
        private UdpClient udpClientCamP1, udpClientCamP2,
udpClientCamK;
        private Bitmap img =
(Bitmap)Bitmap.FromFile(@"C:\Users\ASUS\Desktop\Tugas
Akhir\Visual Basic\Pict\Robot.png");
        static SimpleTcpClient ClientReffBox;
        Boolean teamCyan = true;
        long count_IPListClient, count_send_P1, count_send_P2,
count_send_K, count_msg_P1, count_msg_P2, count_msg_K;
        int mode;
        bool auto_ip = false;
```

```

string datarefbox = "x";
public Form1()
{
    InitializeComponent();
    SendToRobot.Start();
    resetBola.Start();
    sendRefBox.Start();
    pageAllCamera = new All_Camera();
    pageCamP1 = new FormCam();
    pageCamP2 = new FormCam();
    pageCamK = new FormCam();
}
private void Form1_Load(object sender, EventArgs e)
{
    this.Controls.Add(pageCamP1);
    pageCamP1.Visible = false;
    this.pageCamP1.PortChanged += new
System.EventHandler(this.btnPortCamP1_click);

    this.Controls.Add(pageCamP2);
    pageCamP2.Visible = false;
    this.pageCamP2.PortChanged += new
System.EventHandler(this.btnPortCamP2_click);

    this.Controls.Add(pageCamK);
    pageCamK.Visible = false;
    this.pageCamK.PortChanged += new
System.EventHandler(this.btnPortCamK_click);

    this.Controls.Add(pageAllCamera);
    pageAllCamera.Visible = false;

    udpThread = new Thread(serverThread);
    udpThread.IsBackground = true;
    udpThread.Start();
}

```



```

ReadJson();
ReadJsonRobot("Penyerang1");
ReadJsonRobot("Penyerang2");
ReadJsonRobot("Kiper");
//mode = int.Parse(ReadText("mode"));
writeIPServer();
writeOnRobot();
// WriteJson();
grapich_bola1();
grapich_robot1();

grapich_bola2();
grapich_robot2();

grapich_bolaK();
grapich_robotK();
doneLoadedForm = true;
errorTime.Stop();
}
private void btnConnectReff_Click(object sender, EventArgs e)
{
    try
    {
        if (btnConnectReff.BackColor == Color.Red)
        {
            ClientReffBox = new SimpleTcpClient(txtIP_Reff.Text,
Convert.ToInt16(txtPort_Reff.Text));
            ClientReffBox.Events.Connected += Client_Connected;
            ClientReffBox.Events.Disconnected +=
Client_Disconnected;
            ClientReffBox.Events.DataReceived +=
Client_DataReceived;
            ClientReffBox.Events.DataSent += Client_DataSent;
            ClientReffBox.Keepalive.EnableTcpKeepAlives = true;
            ClientReffBox.Settings.MutuallyAuthenticate = false;

```

```

        ClientReffBox.Settings.AcceptInvalidCertificates = true;
        ClientReffBox.Settings.ConnectTimeoutMs = 5000;
        ClientReffBox.ConnectWithRetries(2000);
    }
    else
    {
        //ClientReffBox.Disconnect();
        ClientReffBox.DisconnectAsync();
    }
}
catch (TimeoutException error){
    DateTime now = DateTime.Now;
    txtErrorMsg.Text = now+ " : " + error.Message;
}
}
private void Client_DataReceived(object sender,
DataReceivedEventArgs e)
{
    this.Invoke((MethodInvoker)delegate
    {
        String data = Encoding.UTF8.GetString(e.Data.Array, 0,
e.Data.Count);
        DateTime now = DateTime.Now;
        string jam = now.Hour + ":" + now.Minute + ":" + now.Second;
        txtMsg_Reff.Text += jam + " : "+data +
Environment.NewLine;
        // perlu diganti dari data team ke data yang diperlukan robot

        // variable data merupakan pesan yang diterima dari RefBox
        if (teamCyan == true){ // jika konfigurasi tim yaitu cyan
            // data yang dikirim merupakan data murni RefBox
            string datakirim = "refBox," + data;
            datarefbox = data;
            sendAll(datakirim);
        }
        else if (teamCyan == false){ // jika konfigurasi tim yaitu cyan

```

```

String databaru;
if (data == "K") databaru = "k";
else if (data == "k") databaru = "K";
else if (data == "A") databaru = "a";
else if (data == "a") databaru = "A";
else if (data == "F") databaru = "f";
else if (data == "f") databaru = "F";
else if (data == "G") databaru = "g";
else if (data == "g") databaru = "G";
else if (data == "T") databaru = "t";
else if (data == "t") databaru = "T";
else if (data == "C") databaru = "c";
else if (data == "c") databaru = "C";
else if (data == "P") databaru = "p";
else if (data == "p") databaru = "P";
else if (data == "B") databaru = "b";
else if (data == "b") databaru = "B";
else databaru = data;
datarefbox = databaru;
string datakirim = "refBox," + databaru + ",";
sendAll(datakirim);
    }
});
}
public void serverThread()
{
    UdpClient      udpClient      =      new
UdpClient(Convert.ToInt16(txtPortServer.Text));
    while (true)
    {
        try
        {
            IPEndPoint      RemoteIpEndPoint      =      new
IPEndPoint(IPAddress.Any, Convert.ToInt16(txtPortServer.Text));
            Byte[]      receiveBytes      =      udpClient.Receive(ref
RemoteIpEndPoint);

```

```

        string          returnData          =
Encoding.ASCII.GetString(receiveBytes);

        this.Invoke(new MethodInvoker(delegate ()
        {
            String ipclient = RemoteIpEndPoint.Address.ToString();
            if (ipclient == txtIP_P1.Text && Penyerang1.on == true)
            {
                Penyerang1 = parsing_data(returnData, Penyerang1);
                Penyerang1.newData = true;
                if (Penyerang1.freeDataNew == true &&
Penyerang2.on == true && Penyerang1.on == true)
                {
                    sendP2("freeData," + Penyerang1.freeData + ",");
                    Penyerang1.freeDataNew = false;
                }

                txtMsg_P1.Text = Environment.NewLine +
count_msg_P1 + " : " + returnData.ToString();
                count_msg_P1 += 1;
                grapich_bola1();
                grapich_robot1();
            }
            else if (ipclient == txtIP_P2.Text && Penyerang2.on ==
true)
            {
                Penyerang2 = parsing_data(returnData, Penyerang2);
                Penyerang2.newData = true;
                if (Penyerang2.freeDataNew == true &&
Penyerang1.on == true && Penyerang2.on == true)
                {
                    sendP1("freeData," + Penyerang2.freeData + ",");
                    Penyerang2.freeDataNew = false;
                }
            }
        }
    )
}

```

```

        txtMsg_P2.Text = Environment.NewLine +
count_msg_P2 + " : " + returnData.ToString();
        count_msg_P2 += 1;
        grapich_bola2();
        grapich_robot2();
    }
    else if (ipclient == txtIP_K.Text && Kiper.on == true)
    {
        Kiper = parsing_data(returnData, Kiper);
        Kiper.newData = true;
        txtMsg_K.Text = Environment.NewLine +
count_msg_K + " : " + returnData.ToString();
        count_msg_K += 1;
        grapich_bolaK();
        grapich_robotK();
    }
    else
    {
        if(auto_ip == true) {
            UdpClient udpClient = new UdpClient();
            Byte[] senddata =
Encoding.ASCII.GetBytes("who");
            if (case_auto_ip == 0)
            {
                udpClient.Connect(ipclient,
Convert.ToInt16(txtPort_P1.Text));
                udpClient.Send(senddata, senddata.Length);
                udpClient.Close();
                case_auto_ip = 1;
            }
            else if (case_auto_ip == 1)
            {
                udpClient.Connect(ipclient,
Convert.ToInt16(txtPort_P2.Text));
                udpClient.Send(senddata, senddata.Length);
                udpClient.Close();
            }
        }
    }
}

```

```

        case_auto_ip = 2;
    }
    else if (case_auto_ip == 2)
    {
        udpClient.Connect(ipclient,
Convert.ToInt16(txtPort_K.Text));
        udpClient.Send(senddata, senddata.Length);
        udpClient.Close();
        case_auto_ip = 0;
    }

    if (returnData == "im,1") txtIP_P1.Text = ipclient;
    else if (returnData == "im,2") txtIP_P2.Text =
ipclient;
    else if (returnData == "im,K") txtIP_K.Text =
ipclient;
    }
    IPListClient.Text = count_IPListClient + " : ";
    IPListClient.Text += ipclient;
    IPListClient.Text += ":";
    IPListClient.Text += returnData;
    count_IPListClient += 1;
    }
    }));
}
catch (Exception e)
{
    DateTime now = DateTime.Now;
    txtErrorMsg.Text = now + " : " + e.Message;
}
}
}
private DataRobot parsing_data(string data_recv, DataRobot
dataRobot)
{
    String[] data = data_recv.Split(',');

```

```

int length = data.Length;
for (int i = 0; i < length; i += 2)
{
    try
    {
        if (data[i] == "odomX") dataRobot.odomX =
Int32.Parse(data[i + 1]);
        else if (data[i] == "odomY") dataRobot.odomY =
Int32.Parse(data[i + 1]);
        else if (data[i] == "bolaX")
        {
            dataRobot.bolaX = Int32.Parse(data[i + 1]);
            if (dataRobot.bolaX != -9999) dataRobot.newDataBola =
true;
        }
        else if (data[i] == "bolaY") dataRobot.bolaY =
Int32.Parse(data[i + 1]);
        else if (data[i] == "state") dataRobot.state =
Int32.Parse(data[i + 1]);
        else if (data[i] == "theta") dataRobot.theta =
Int32.Parse(data[i + 1]);
        else if (data[i] == "refBox") dataRobot.refBox = data[i + 1];
        else if (data[i] == "freeData")
        {
            dataRobot.freeData = data[i + 1];
            dataRobot.freeDataNew = true;
        }
        else if (data[i] == "port")
        {
            if(dataRobot.name == "1") txtPort_P1.Text = data[i + 1];
            else if (dataRobot.name == "2") txtPort_P2.Text = data[i
+ 1];

            else if (dataRobot.name == "K") txtPort_K.Text = data[i +
1];
        }
    }
}

```

```

        catch (Exception e)
        {
            DateTime now = DateTime.Now;
            txtErrorMsg.Text = now + " : " + e.Message;
        }
    }
    return dataRobot;
}
private void sendP1(String data)
{
    try
    {
        if (Penyerang1.on) {
            UdpClient udpClient = new UdpClient();
            udpClient.Connect(txtIP_P1.Text,
Convert.ToInt16(txtPort_P1.Text));
            Byte[] senddata = Encoding.ASCII.GetBytes(data);
            udpClient.Send(senddata, senddata.Length);
            txtSend_P1.Text = count_send_P1 + " : " + data +
Environment.NewLine;
            count_send_P1 += 1;
        }
    }
    catch (Exception e)
    {
        DateTime now = DateTime.Now;
        txtErrorMsg.Text = now + " : " + e.Message;
    }
}
private void sendP2(String data)
{
    try
    {
        if (Penyerang2.on)
        {

```



```

        UdpClient udpClient = new UdpClient();
        udpClient.Connect(txtIP_P2.Text,
Convert.ToInt16(txtPort_P2.Text));
        Byte[] senddata = Encoding.ASCII.GetBytes(data);
        udpClient.Send(senddata, senddata.Length);

        txtSend_P2.Text = count_send_P2 + " : " + data +
Environment.NewLine;
        count_send_P2 += 1;
    }
}
catch (Exception e)
{
    DateTime now = DateTime.Now;
    txtErrorMsg.Text = now + " : " + e.Message;
}
}
private void sendK(String data)
{
    try
    {
        if (Kiper.on)
        {
            UdpClient udpClient = new UdpClient();
            udpClient.Connect(txtIP_K.Text,
Convert.ToInt16(txtPort_K.Text));
            Byte[] senddata = Encoding.ASCII.GetBytes(data);
            udpClient.Send(senddata, senddata.Length);

            txtSend_K.Text = count_send_K + " : " + data +
Environment.NewLine;
            count_send_K += 1;
        }
    }
    catch (Exception e)
    {

```

```

        DateTime now = DateTime.Now;
        txtErrorMsg.Text = now + " : " + e.Message;
    }
}

private void sendAll(String data)
{
    sendP1(data);
    sendP2(data);
    sendK(data);
}

private void SendToRobot_Tick(object sender, EventArgs e)
{
    DateTime now = DateTime.Now;
    labelTime.Text = now.ToString();
    if (Penyerang1.bolaX != -9999 && Penyerang1.on &&
Penyerang1.newDataBola)
    {
        bola.X = Penyerang1.bolaX;
        bola.Y = Penyerang1.bolaY;
        resetBola.Stop();
        resetBola.Start();
        Penyerang1.newDataBola = false;
    }
    else if (Penyerang2.bolaX != -9999 && Penyerang2.on &&
Penyerang2.newDataBola)
    {
        bola.X = Penyerang2.bolaX;
        bola.Y = Penyerang2.bolaY;
        resetBola.Stop();
        resetBola.Start();
        Penyerang2.newDataBola = false;
    }
    else if (Kiper.bolaX != -9999 && Kiper.on &&
Kiper.newDataBola)
    {
        bola.X = Kiper.bolaX;

```

```

        bola.Y = Kiper.bolaY;
        resetBola.Stop();
        resetBola.Start();
        Kiper.newDataBola = false;
    }
else
{
    bola.X = -9999;
    bola.Y = -9999;
}

if (Penyerang1.on == true)
{
    bool befsend = false;
    string ToRobot = "";
    if (Penyerang2.on == true && Penyerang2.newData == true)
    {
        Penyerang2.newData= false;
        ToRobot = "player,1";
        ToRobot += ",odomX,";
        ToRobot += Penyerang2.odomX.ToString();
        ToRobot += ",odomY,";
        ToRobot += Penyerang2.odomY.ToString();
        ToRobot += ",state,";
        ToRobot += Penyerang2.state.ToString();
        befsend = true;
    }
    else if (Penyerang2.on == true && Penyerang2.newData ==
false) {
        Penyerang2.newData = false;
        ToRobot = "player,1";
        befsend = true;
    }
    else
    {
        ToRobot = "player,0";
    }
}

```

```

        befsend = true;
    }
    if (Penyerang1.bolaX == -9999 && bola.X != -9999)
    {
        if (befsend == true)
        {
            ToRobot += ",";
            befsend = false;
        }
        ToRobot += "bolaX,";
        ToRobot += bola.X.ToString();
        ToRobot += ",bolaY,";
        ToRobot += bola.Y.ToString();
    }
    if(ToRobot!="") sendP1(ToRobot);
}
if (Penyerang2.on == true)
{
    bool befsend = false;
    string ToRobot = "";
    if (Penyerang1.on == true && Penyerang1.newData == true)
    {
        Penyerang1.newData = false;
        ToRobot = "player,1";
        ToRobot += ",odomX,";
        ToRobot += Penyerang1.odomX.ToString();
        ToRobot += ",odomY,";
        ToRobot += Penyerang1.odomY.ToString();
        ToRobot += ",state,";
        ToRobot += Penyerang1.state.ToString();
        befsend = true;
    }
    else if (Penyerang1.on == true)
    {
        Penyerang1.newData = false;
        ToRobot = "player,1";
    }
}

```

```

        befsend = true;
    }
    else if (Penyerang1.on == false)
    {
        Penyerang1.newData = false;
        ToRobot = "player,0";
        befsend = true;
    }
    if (Penyerang2.bolaX == -9999 && bola.X != -9999)
    {
        if (befsend == true)
        {
            ToRobot += ",";
            befsend = false;
        }
        ToRobot += "bolaX,";
        ToRobot += bola.X.ToString();
        ToRobot += ",bolaY,";
        ToRobot += bola.Y.ToString();
    }
    if (ToRobot != "") sendP2(ToRobot);
}
if (Kiper.on == true)
{
    string ToRobot = "";
    if (Kiper.bolaX == -9999 && bola.X != -9999)
    {
        ToRobot += "bolaX,";
        ToRobot += bola.X.ToString();
        ToRobot += ",bolaY,";
        ToRobot += bola.Y.ToString();
    }
    if (ToRobot != "") sendK(ToRobot);
}
txt_odom_P1.Text = "OdomX : " +
Penyerang1.odomX.ToString() + "\r\nOdomY : " +

```

```

Penyerang1.odomY.ToString() + "\r\nTetha      : " +
Penyerang1.theta.ToString() + "\r\nbolaX      : " +
Penyerang1.bolaX.ToString() + "\r\nbolaY      : " +
Penyerang1.bolaY.ToString();
    txt_odom_P2.Text = "OdomX      : " +
Penyerang2.odomX.ToString() + "\r\nOdomY      : " +
Penyerang2.odomY.ToString() + "\r\nTetha      : " +
Penyerang2.theta.ToString() + "\r\nbolaX      : " +
Penyerang2.bolaX.ToString() + "\r\nbolaY      : " +
Penyerang2.bolaY.ToString();
    txt_odom_K.Text = "OdomX : " + Kiper.odomX.ToString() +
"\r\nOdomY : " + Kiper.odomY.ToString() + "\r\nTetha      : " +
Kiper.theta.ToString() + "\r\nbolaX      : " + Kiper.bolaX.ToString() +
"\r\nbolaY      : " + Kiper.bolaY.ToString();
    }
}

```

LAMPIRAN D

Listing Program Interface Robot

```
import datetime,time,socket,serial, math,json,os,sys
import data.Millis as Millis
from data.config import *
import data.dataRobot as dataRobot
import data.dataTerima as dataTerima

from PyQt5 import QtCore, QtWidgets, uic, QtGui
from PyQt5.QtCore import QTimer
import serial.tools.list_ports

import cv2
from data.objectDetect import objectDetect
from data.makeCircle import frameCircleModule
from data.masking import centerMask
import data.utils as utils
import pickle
class KomunikasiThread(QtCore.QThread):
    any_signal = QtCore.pyqtSignal(int)
    error_signal = QtCore.pyqtSignal(Exception)
    def __init__(self):
        super(KomunikasiThread, self).__init__()
    def parse_dataTerima(self,data):
        parsed_data = data.split(',')
        i = 0
        while i < len(parsed_data):
            if (parsed_data[i] == "odomX"): dataTerima.odomX =
int(parsed_data[i+1])
            elif (parsed_data[i] == "odomY"): dataTerima.odomY =
int(parsed_data[i+1])
            elif (parsed_data[i] == "theta"): dataTerima.theta =
int(parsed_data[i+1])
            elif (parsed_data[i] == "bolaX"): dataTerima.bolaX =
int(parsed_data[i+1])
            elif (parsed_data[i] == "bolaY"): dataTerima.bolaY =
int(parsed_data[i+1])
```

```

        elif (parsed_data[i] == "state"): dataTerima.state =
int(parsed_data[i+1])
        elif (parsed_data[i] == "player"): dataTerima.player =
parsed_data[i+1]
        elif (parsed_data[i] == "refBox"): dataTerima.refBox =
parsed_data[i+1]
        elif (parsed_data[i] == "freeData"): dataTerima.freeData =
parsed_data[i+1]

        elif (parsed_data[i] == "KDBolaT"):dataRobot.KDBolaT =
int(parsed_data[i+1])
        elif (parsed_data[i] == "KDBolaR"): dataRobot.KDBolaR =
int(parsed_data[i+1])
        elif (parsed_data[i] == "KDGawangT"): dataRobot.KDGawangT
= int(parsed_data[i+1])
        elif (parsed_data[i] == "KDGawangR"): dataRobot.KDGawangR
= int(parsed_data[i+1])

        elif (parsed_data[i] == "sendCam"):
            dataTerima.sendCam = int(parsed_data[i+1])
            print("sendCam",dataTerima.sendCam)
        elif (parsed_data[i] == "portSend"):dataTerima.portSend =
int(parsed_data[i+1])
        i = i+2
def run(self):
    print("\nstarting thread komunikasi")
    global UDP
    global sock_recv, sock_send,data_terima_basestation,
count_data_terima_basestation
    while True:
        if UDP['useUDP'] == True:
            try:
                data, addr = sock_recv.recvfrom(1024)
                strbaru = data.decode('utf-8')
                if(strbaru == "who"):
                    sock_send.sendto(("im,"+namaRobot).encode('utf-8'),
(UDP['IP_Destination'], UDP['PORT_Destination']))
                    data_terima_basestation = str(addr) + '\n' + strbaru
                    self.parse_dataTerima(strbaru)

```



```

        if(dataTerima.freeData == ""):
            dataTerima.freeData = 'x'
        if(dataTerima.refBox == ""):
            dataTerima.refBox = 'x'
        self.any_signal.emit(1)
        count_data_terima_basestation +=1
    except Exception as e:
        self.error_signal.emit(e)
        print("Error:", e)
        print("gagal terima udp")
        time.sleep(0.1)
    else:
        time.sleep(1)
    print("\nending thread komunikasi")
def stop(self):
    print('Stopping thread...')
    self.terminate()
class SerialThread(QtCore.QThread):
    any_signal = QtCore.pyqtSignal(int)
    error_signal = QtCore.pyqtSignal(Exception)
    def __init__(self):
        super(SerialThread, self).__init__()
    def send_basestation(self,data):
        global UDP
        # print("ke basestation : ", data)
        global sock_send, count_data_kirim_basestation
        if UDP['useUDP'] == True:
            try:
                sock_send.sendto(data.encode('utf-8'), (UDP['IP_Destination'],
UDP['PORT_Destination']))
                count_data_kirim_basestation +=1
            except Exception as e:
                self.error_signal.emit(e)
                print("Error:", e)
                print("gagal mengirim udp")
    def send_ser(self,data):
        global ser, count_data_kirim_serial
        global Serial
        if Serial['useSerial'] == True:

```

```

try:
    data = data + '\n'
    ser.write(data.encode('utf-8'))
    count_data_kirim_serial+=1
except Exception as e:
    self.error_signal.emit(e)
    print("Error:", e)
    print("gagal mengirim serial")
def parse_dataRobot(self,data):
    parsed_data = data.split(',')
    i = 0
    while i < len(parsed_data):
        if (parsed_data[i] == "odomX"): dataRobot.odomX =
int(parsed_data[i+1])
        elif (parsed_data[i] == "odomY"): dataRobot.odomY =
int(parsed_data[i+1])
        elif (parsed_data[i] == "state"): dataRobot.state =
int(parsed_data[i+1])
        elif (parsed_data[i] == "theta"): dataRobot.theta =
int(parsed_data[i+1])
        elif (parsed_data[i] == "freeData"): dataRobot.freeData =
parsed_data[i+1]
        i = i+2
def run(self):
    print("\nstarting thread serial")
    global Serial, UDP
    waktu = 0
    waktuSendSerAll = 0
    waktuSer = 0
    waktuSendUDPPort = 0
    global data_kirim_basestation, data_terima_serial,
data_kirim_serial
    global ser, count_data_terima_serial
    newDataSerial = False
    while (1):
        if Serial['useSerial'] == True:
            try:
                if ser.in_waiting > 0:
                    data = ser.readline()

```

```

data_terima_serial = data.decode('utf-8')
# print(data_terima_serial)
try:
    self.parse_dataRobot(data_terima_serial)
except Exception as e:
    print("Error:", e)
    print("gagal parsing")
newDataSerial = True
count_data_terima_serial += 1
# print("TERIMA SERIAL : ",data_terima_serial)
except Exception as e:
    self.error_signal.emit(e)
    print("Error:", e)
    print("gagal menerima serial")
try:
    if(Millis.millis())>waktuSer):
        waktuSer = Millis.millis()+Serial["interval"] #waktu
        kirim serial
        data_kirim_serial = self.getStringkirimSerial()
        self.send_ser(data_kirim_serial)
        if(Millis.millis())>waktuSendSerAll):
            waktuSendSerAll = Millis.millis()+2000 #waktu kirim
            serial semua data
            data_kirim_serial =
            "state,"+str(dataTerima.state)+",player,"+str(dataTerima.player)+",freeD
            ata,"+str(dataTerima.freeData)+",refBox,"+str(dataTerima.refBox)
            data_kirim_serial += ","
            self.send_ser(data_kirim_serial)
        except Exception as e:
            self.error_signal.emit(e)
            print("Error:", e)
            print("gagal mengirim serial")
            time.sleep(0.1)
    if(UDP['useUDP']== True):
        if(Millis.millis())>waktuSendUDPPort):
            waktuSendUDPPort = Millis.millis()+2000 #waktu kirim
            serial semua data
            self.send_basestation("port,"+str(UDP["PORT_Local"]))
            self.send_basestation("im,"+namaRobot)

```

```

        if(Millis.millis(>waktu):
            waktu = Millis.millis()+UDP["interval"]
            data_kirim_basestation
            "bolaX,"+str(dataRobot.bolaX)+" ,bolaY,"+str(dataRobot.bolaY)
            if(newDataSerial):
                data_kirim_basestation +=", "
                data_kirim_basestation +=data_terima_serial
                self.send_basestation(data_kirim_basestation)
                self.any_signal.emit(1)
                time.sleep(0.01)
            print("\nending thread serial")
def stop(self):
    print('Stopping thread...')
    self.terminate()
def getStringKirimSerial(self):
    dataKirim = "bolaT,"+str(dataRobot.bolaKirimT)
    dataKirim += ",refBox,"+str(dataTerima.refBox)
    # if(dataTerima.refBox != dataTerima.last_refBox):
    #     dataKirim += ",refBox,"+str(dataTerima.refBox)
    #     dataTerima.last_refBox = dataTerima.refBox
    if(dataRobot.bolaKirimR != dataRobot.last_bolaKirimR):
        dataKirim += ",bolaR,"+str(dataRobot.bolaKirimR)
        dataRobot.last_bolaKirimR = dataRobot.bolaKirimR
    if(dataRobot.robotT!=dataRobot.last_robotT):
        dataKirim += ",robotT,"+str(dataRobot.robotT)
        dataRobot.last_robotT = dataRobot.robotT
    if(dataRobot.robotR!=dataRobot.last_robotR):
        dataKirim += ",robotR,"+str(int(dataRobot.robotR))
        dataRobot.last_robotR = dataRobot.robotR
    if(dataTerima.state!=dataTerima.last_state):
        dataKirim += ",state,"+str(dataTerima.state)
        dataTerima.last_state = dataTerima.state
    if(dataTerima.player!=dataTerima.last_player):
        dataKirim += ",player,"+str(dataTerima.player)
        dataTerima.last_player = dataTerima.player
    if(dataTerima.freeData!=dataTerima.last_freeData):
        dataKirim += ",freeData,"+str(dataTerima.freeData)
        dataTerima.last_freeData = dataTerima.freeData
    if(dataRobot.KDBolaT!=dataRobot.last_KDBolaT):

```

```

        dataKirim += ",KDBT,"+str(dataRobot.KDBolaT)
        dataRobot.last_KDBolaT = dataRobot.KDBolaT
    if(dataRobot.KDBolaR!=dataRobot.last_KDBolaR):
        dataKirim += ",KDBR,"+str(dataRobot.KDBolaR)
        dataRobot.last_KDBolaR = dataRobot.KDBolaR
    if(dataRobot.KDGawangT!=dataRobot.last_KDGawangT):
        dataKirim += ",KDGT,"+str(dataRobot.KDGawangT)
        dataRobot.last_KDGawangT = dataRobot.KDGawangT
    if(dataRobot.KDGawangR!=dataRobot.last_KDGawangR):
        dataKirim += ",KDGR,"+str(dataRobot.KDGawangR)
        dataRobot.last_KDGawangR = dataRobot.KDGawangR
    if(dataRobot.MusuhT!=dataRobot.last_MusuhT):
        dataKirim += ",musuhT,"+str(dataRobot.MusuhT)
        dataRobot.last_MusuhT = dataRobot.MusuhT
    if(dataRobot.MusuhR!=dataRobot.last_MusuhR):
        dataKirim += ",musuhR,"+str(dataRobot.MusuhR)
        dataRobot.last_MusuhR = dataRobot.MusuhR
    dataKirim += ","
    return dataKirim
app = QtWidgets.QApplication(sys.argv)
main_ui = gui()
main_ui.show()
sys.exit(app.exec_())

```

LAMPIRAN E

Listing Program Arduino

```
#include "variable_nasional.h"
#define maks_odom 255
#define maks_pwm_kinematik 130

valPID PID_X = { 7, 0.04, 0.04, maks_odom, -maks_odom };
valPID PID_Y = { 7, 0.04, 0.04, maks_odom, -maks_odom };
valPID PID_Z = { 6, 0.009, 0.02, maks_odom, -maks_odom };

valPID PID_CAMbot = {2, 0, 0.02, maks_odom, -maks_odom };
valPID PID_CAMz = {4.5, 0, 0.02, 200, -200};
valPID PID_CAMzG = {4, 0.009, 0.02, 50, -50};
valPID PID_CAMz_foll = {6, 0, 0.02, maks_odom, -maks_odom};
valPID PID_CAMx = { 22, 0, 0.02, maks_odom, -maks_odom };

void setup() {
  initialVar();
  serial_setup();
  motor_setup();
  drible_setup();
  compas_setup();
  tombol_setup();
  kick_setup();
}

unsigned long delayTendang;

void loop() {
  parsing_pc();
  get_last_bola();
  compas();
  readTombol();
  serialSlave();
  odometry();
  peta();
  onDisplay();
  dribleSwitch();
```

```

mainLogic();
 kirim_pc();
 //foll_ball(0, 3);
 }
 void serial_setup() {
  Serial.begin(9600);
  while (!Serial);
  Serial1.begin(9600);
  while (!Serial1);
  Serial2.begin(9600);
  while (!Serial2);
  Serial3.begin(9600);
  while (!Serial3);

  Serial1.print("y");
  Serial2.print("y");
  Serial3.print("y");
 }
 void parsing_pc() {
  if (Serial.available()) {
   String data_recv = Serial.readStringUntil('\n');
   // Serial.println(data_recv);
   int lengthMsg;
   String* dataMsg = split(data_recv, ',', lengthMsg);
   for (int i = 0; i < lengthMsg; i += 2) {
    if (dataMsg[i] == "odomX") udp_terima.odomX = dataMsg[i +
1].toInt();
    else if (dataMsg[i] == "odomY") udp_terima.odomY = dataMsg[i +
1].toInt();
    else if (dataMsg[i] == "bolaT") udp_terima.lastBolaT = dataMsg[i +
1].toInt();
    else if (dataMsg[i] == "bolaR") udp_terima.lastBolaR = dataMsg[i +
1].toInt();
    else if (dataMsg[i] == "robotT")udp_terima.robotT = dataMsg[i +
1].toInt();
    else if (dataMsg[i] == "robotR") udp_terima.robotR = dataMsg[i +
1].toInt();
    else if (dataMsg[i] == "state") udp_terima.state = dataMsg[i +
1].toInt();

```

```

    else if (dataMsg[i] == "player")udp_terima.player = cchar(dataMsg[i
+ 1]);
    else if (dataMsg[i] == "freeData") udp_terima.freeData =
cchar(dataMsg[i + 1]);
    else if (dataMsg[i] == "refBox") udp_terima.refBox =
cchar(dataMsg[i + 1]);
    else if (dataMsg[i] == "KDBT") udp_terima.KDBolaT = dataMsg[i +
1].toInt();
    else if (dataMsg[i] == "KDBR") udp_terima.KDBolaR = dataMsg[i +
1].toInt();
    else if (dataMsg[i] == "KDG T") udp_terima.KDGawangT =
dataMsg[i + 1].toInt();
    else if (dataMsg[i] == "KDGR") udp_terima.KDGawangR =
dataMsg[i + 1].toInt();
    }
    delete[] dataMsg;
}
}
}

```

```

unsigned long waktuKirimSerial;
void kirim_pc() {
    if (millis() > waktuKirimSerial) {
        waktuKirimSerial = millis() + 100;
        Serial.println("odomX," + String((int)X.lap) + ",odomY," +
String((int)Y.lap) + ",theta," + String((int)newYaw) + ",state," +
String(udp_kirim.state)+",freeData," + String(udp_kirim.freeData));
    }
}
}

```


BIODATA



Nama : Yusuf Firmansyah
Tempat, Tanggal Lahir : Cilacap, 28 Maret 2002
Alamat : Jalan Anggur Kel. Karangandri, Kec. Kesugihan, Kab. Cilacap, Prov. Jateng
Email : yufiyusuffirmansyah@gmail.com
No. Telepon : 081930946245
Motto : Napas tuh manual, jangan otomatis. Kaya gini nih, tarik hempppp, keluarin huffff, gitu terus jangan berhenti.
Hobi : Ketiduran

Riwayat Pendidikan:

- | | | |
|----|------------------------------------------------------------------|-----------------|
| 1. | SDN Tegalkamulyan 1 Cilacap | Tahun 2008-2014 |
| 2. | SMP Negeri 4 Cilacap | Tahun 2014-2017 |
| 3. | SMK Negeri 2 Cilacap
Teknik Pembangkit Tenaga Listrik | Tahun 2017-2020 |
| 4. | Politeknik Negeri Cilacap
Program Studi D3 Teknik Elektronika | Tahun 2020-2023 |

Penulis telah mengikuti seminar hasil Tugas Akhir pada tanggal 28 Juli 2023 sebagai salah satu persyaratan untuk memperoleh gelar Ahli Madya(A.Md).