1. Kode Program Pada Client

```
unit uCLIENT;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
Forms,
  Dialogs, StdCtrls, ScktComp, ExtCtrls, XPMan, ComCtrls,
ThinTrackBar;

type
  TfMain = class(TForm)
    ClientSocket1: TClientSocket;
    Label1: TLabel;
    Label2: TLabel;
    txtIpServer: TEdit;
    txtPort: TEdit;
    btnConnect: TButton;
    txtMessage: TEdit;
    btnSend: TButton;
    Label3: TLabel;
    cbDirection: TComboBox;
    XPManifest1: TXPManifest;
    ColorDialog1: TColorDialog;
    barR: TThinTrackBar;
    barG: TThinTrackBar;
    barB: TThinTrackBar;
    lblR: TLabel;
    lblG: TLabel;
    lblB: TLabel;
    ColorView: TShape;
    barDelay: TThinTrackBar;
    Label4: TLabel;
    procedure btnConnectClick(Sender: TObject);
    // merupakan button konek dan disconek
    procedure txtPortKeyPress(Sender: TObject; var Key: Char);
    //edit untuk menginputkan port dengan variabel char
    procedure ClientSocket1Error(Sender: TObject; Socket:
    TCustomWinSocket;
      ErrorEvent: TErrorEvent; var ErrorCode: Integer);
    //merupakan pesan kesalahan jika gagal koneksi dengan server atau
    server mati
    procedure ClientSocket1Disconnect(Sender: TObject;
      Socket: TCustomWinSocket);
    //merupakan perintah diskonek dengan server
    procedure ClientSocket1Connect(Sender: TObject;
      Socket: TCustomWinSocket);
    //merupakan perintaha konek dengan server
    procedure btnSendClick(Sender: TObject);
    //merupakan button send
    procedure ClientSocket1Connecting(Sender: TObject;
      Socket: TCustomWinSocket);
    procedure cbDirectionSelect(Sender: TObject);
    procedure cbDirectionChange(Sender: TObject);
    procedure barRChange(Sender: TObject);
    procedure barGChange(Sender: TObject);
    procedure barBChange(Sender: TObject);
    //merupakan perintah untuk mengatur warna RGB
    procedure FormShow(Sender: TObject);
```

```
    procedure barDelayChange(Sender: TObject);
    //merupakan perintah untuk mengatur dalay

  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  fMain: TfMain;
  direction : Integer = -1;
  colR, colG, colB : byte;
  iDelay : byte = 5;
  //merupakan variabel yang digunakan.
implementation

{$R *.dfm}

procedure TfMain.btnConnectClick(Sender: TObject);
begin
  //merupakan button konek keserver
  if (txtIpServer.Text='') or (txtPort.Text='') then
  //digunakan untuk menyimpan ip server dan port ketika di inputkan
  begin
    ShowMessage('IP Server & Port tidak boleh kosong');
    exit;
  end;
  //merupakan peringatan tidak boleh mengosongkan ip atau port
  if btnConnect.Caption='Connect' then
  begin
    // Masukin IP dan Port Server
    ClientSocket1.Host:=txtIpServer.Text;
    ClientSocket1.Port:=StrToInt(txtPort.Text);
    ClientSocket1.Active:= true; // Connect
  End
  else
    ClientSocket1.Active := false; // Disconnect
end;

procedure TfMain.txtPortKeyPress(Sender: TObject; var Key: Char);
begin
  if not (Key in ['0'..'9',#8,#13])  then  Key:=#0;
//digunakan untuk menentukan inputan hanya angka saja
end;



procedure TfMain.ClientSocket1Error(Sender: TObject;
  Socket: TCustomWinSocket; ErrorEvent: TErrorEvent;
  var ErrorCode: Integer);
begin
  ErrorCode := 0; // Ini penting untuk mendisablekan pesan error
  ShowMessage('Failed Connecting to Server...!!!');
  fMain.Caption := 'Idle';
end;
//perintah ini dijalankan jika tidak terkoneksi dengan server.
procedure TfMain.ClientSocket1Disconnect(Sender: TObject;
  Socket: TCustomWinSocket);
Begin
```

```
  // Bagian ini dijalankan pada saat Client Diskonek dari server
  ShowMessage('Disconnected from Server: ' + txtIpServer.Text);
  btnConnect.Caption := 'Connect';
  fMain.Caption := 'Idle';
end;

procedure TfMain.ClientSocket1Connect(Sender: TObject;
  Socket: TCustomWinSocket);
begin
  // Bagian ini di jalankan pada saat Client berhasil terkoneksi
dengan Server
  ShowMessage('Connected to Server: ' + txtIpServer.Text);
  btnConnect.Caption := 'Disconnect';
  fMain.Caption := 'Connected to: ' + txtIpServer.Text;
end;

procedure TfMain.btnSendClick(Sender: TObject);
var
  kata, kataTemp : String;
  lengthKata, i, j: Integer;
//merupakan variabel kata,i,j
begin

  // set direction 0/1 untuk langsung memilih arah
  direction := 0; // 0 = Kanan ke Kiri, 1 = Kiri ke Kanan

  // Check connection to arduino
  if not ClientSocket1.Active then
  begin
    ShowMessage('Not Connected to Server');
    btnConnect.SetFocus;
    exit;
  end;

  // Check direction tidak dipilih
  if direction = -1 then
  begin
    ShowMessage('Direction Not Valid');
    cbDirection.SetFocus;
    exit;
  end;

  // Check direction LEFT TO RIGHT, Inv Kata
  if direction = 1 then begin
    // Kata Dibalik, misal [ Ady -> ydA ]
    kataTemp := txtMessage.Text;
    lengthKata := length(kataTemp);
    i := 1;
    for j:=lengthKata downto 1 do begin
      insert(kataTemp[j], kata, i);
      inc(i);
    end;
  else
    kata := txtMessage.Text;
    //kata di isi dengan kata sesuai dengan yang diinputkan tidak
   perlu di invers terlebih dahulu.

  // Format pengiriman data dari delphi ke arduino untuk 1x
  pengiriman data
  // 0x02 - Text - 0x03 - Direction(0:RL,1:LR) - 0x04 Warna(R[0:255],
  G[0:255], B[0:255], D[0:25]) - 0x05
```

```
  // 0x02 - Text - 0x03 - [0:1] - 0x04 - (R[0:255],G[0:255],
  B[0:255], D[0:25]) - 0x05
  ClientSocket1.Socket.SendText(#02);
  //merupakan header atau aturan untuk mengirim kata
  ClientSocket1.Socket.SendText(trim(kata));
  //perintah untuk mengirim kata sesuai dengan yang di inputkan.
  ClientSocket1.Socket.SendText(#03);
  //merupakan header atau aturan untuk mengirim arah
  ClientSocket1.Socket.SendText(IntToStr(direction));
  //perintah untuk mengirim arah
  ClientSocket1.Socket.SendText(#04);
  //merupakan  header atau aturan untuk mengirim warna dan delay

  ClientSocket1.Socket.SendText('R'+IntToStr(colR)+'G'+IntToStr(colG
  )+'B'+IntToStr(colB)+'D'+IntToStr(iDelay));
  //merupakan perintah untuk mengirim warna dan delay
  ClientSocket1.Socket.SendText(#05);
  //merupakan akhir atau penutup dalam satu kali proses pengiriman
  data.
end;

{procedure TfMain.ClientSocket1Read(Sender: TObject;
  Socket: TCustomWinSocket);

procedure TfMain.ClientSocket1Connecting(Sender: TObject;
  Socket: TCustomWinSocket);
begin
  fMain.Caption := 'Connecting to: ' + txtIpServer.Text;
end;
//merupakan parameter perintah konek.
procedure TfMain.cbDirectionSelect(Sender: TObject);
begin
  direction := cbDirection.ItemIndex;
end;
//merupakan parameter dari penentu arah tampil
procedure TfMain.cbDirectionChange(Sender: TObject);
begin
  direction := cbDirection.ItemIndex;
end;

procedure TfMain.barRChange(Sender: TObject);
begin
  colR := barR.Position;
  ColorView.Brush.Color := RGB(colR,colG,colB);
end;
//merupakan parameter untuk mengatur warna
procedure TfMain.barGChange(Sender: TObject);
begin
  colG := barG.Position;
  ColorView.Brush.Color := RGB(colR,colG,colB);
end;
//merupakan parameter untuk mengatur warna
procedure TfMain.barBChange(Sender: TObject);
begin
  colB := barB.Position;
  ColorView.Brush.Color := RGB(colR,colG,colB);
end;
//merupakan parameter untuk mengatur warna
procedure TfMain.FormShow(Sender: TObject);
begin
  ColorView.Brush.Color := RGB(colR,colG,colB);
```

```
end;
//merupakan parameter untuk mengatur warna
procedure TfMain.barDelayChange(Sender: TObject);
begin
  iDelay := barDelay.Position;
end;
//merupakan parameter untuk mengatur delay
end.
```

2. Kode Program Pada Server Arduino

```
#include <SPI.h>
#include <Ethernet.h>
#include <Adafruit_GFX.h>
#include <RGBmatrixPanel.h>

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress ip(192, 168, 1, 253);
IPAddress gateway(192, 168, 1, 254);
IPAddress subnet(255, 255, 255, 0);

EthernetServer server(23); // 4,10,11,12,13,A0,A1

#define CLK 8 //11
#define LAT A5 //10
#define OE  9
#define A   A2
#define B   A3
#define C   A4

RGBmatrixPanel matrix(A, B, C, CLK, LAT, OE, false);

char str[] = "Hello World";
int dir = 0, colorR = 80, colorG = 80, colorB = 80, iDelay = 5;
int cursorX = matrix.width(), cursorXmin = length(str) * -12;

void setup() {
    Ethernet.begin(mac, ip, gateway, subnet);
    server.begin();
    Serial.begin(9600);
    Serial.print("Server address: ");
Serial.println(Ethernet.localIP());
    Serial.println("Listening...");

    matrix.begin();
    matrix.setTextWrap(false);
    matrix.setTextSize(2);
}

void loop() {
    checkNetwork();
    runDisplay(dir, colorR, colorG, colorB, iDelay);
}

void checkNetwork(){
//||=================================================================
========================||
//|| 0x02 - Text - 0x03 - (0=RL,1=LR) - 0x04 - (R[0:255], G[0:255],
B[0:255], D[0:25]) - 0x05 ||
```

```
//||=========================================================================
=========================||
//|| Ex: 0x02-"Text Value"-0x03-"0"-0x04-"R1G2B3D4"-0x05

    boolean readText = false, readDirection = false, readColor =
false;
    boolean readR = false, readG = false, readB = false, readD =
false;
    boolean applyConfiguration = false;
    String sColorR = "", sColorG = "", sColorB = "", sDelay = "";
    byte index = 0;
    EthernetClient client = server.available();

    while(client) {
        while(client.available() > 0) {
            byte thisChar = client.read(); // Serial.write(thisChar);

            if(thisChar == 0x02){ readText = true; }
            if(thisChar == 0x03){ readText = false; readDirection =
true; }
            if(thisChar == 0x04){ readDirection = false; readColor =
true; }
            if(thisChar == 0x05){ readColor = false;
applyConfiguration = true; break; }

            if(readText && (thisChar != 0x02)){ str[index++] =
thisChar; }
            if(readDirection && (thisChar != 0x03)){ dir = thisChar -
'0'; }
            if(readColor && (thisChar != 0x04)){
                if(thisChar == 'R'){ readR = true; }
                if(thisChar == 'G'){ readR = false; readG = true; }
                if(thisChar == 'B'){ readG = false; readB = true; }
                if(thisChar == 'D'){ readB = false; readD = true; }

                if(readR && (thisChar != 'R')){
                    if(isDigit(thisChar)){ sColorR += (char)thisChar;
}
                    colorR = sColorR.toInt();
                }
                if(readG && (thisChar != 'G')){
                    if(isDigit(thisChar)){ sColorG += (char)thisChar;
}
                    colorG = sColorG.toInt();
                }
                if(readB && (thisChar != 'B')){
                    if(isDigit(thisChar)){ sColorB += (char)thisChar;
}
                    colorB = sColorB.toInt();
                }
                if(readD && (thisChar != 'D')){
                    if(isDigit(thisChar)){ sDelay += (char)thisChar;
}
                    iDelay = sDelay.toInt();
                }
            }
        }

        if(applyConfiguration){
            str[index] = '\0';
            cursorX = matrix.width();
```

```
            cursorXmin = length(str) * -12;
            Serial.print("From Delphi: ");
            Serial.print(str); Serial.print("[" + String(length(str))
+ "]"); Serial.print(",");
            Serial.print(dir); Serial.print(",");
            Serial.print(colorR,HEX); Serial.print(",");
            Serial.print(colorG,HEX); Serial.print(",");
            Serial.print(colorB,HEX); Serial.print(",");
            Serial.print(iDelay,HEX); Serial.println();
            break;
        }
    }
}

void runDisplay(int direction, int R, int G, int B, int wait){
    matrix.fillScreen(matrix.Color888(0, 0, 0));
    matrix.setTextColor(matrix.Color888(R, G, B, true));
    matrix.setCursor(cursorX, 1);
    matrix.print(str);

    if(direction == 0){ // RIGTH TO LEFT
        if((--cursorX) < cursorXmin) cursorX = matrix.width();
    } else { // LEFT TO RIGHT
        if((++cursorX) > matrix.width()) cursorX = length(str) * -12;
    }
    delay(wait);
}

size_t length(const char *s){
    return (strlen(s) + 1);
}
```