DAFTAR GAMBAR

DAFTAR TABEL

Lampiran Datasheet Sensor Arus

LEM

Current Transducers HTC 1000..3000-S

$I_{PN\,DC} = \pm 1000..3000\ A$

$V_{OUT} = \pm 10\ V$

For the electronic measurement of currents: AC,DC, pulsed, mixed, with a galvanic isolation between the primary circuit (high power) and the secondary circuit (electronic circuit).

Electrical data

Primary continuous direct current (nominal) $I_{PN\,DC}$ (A)	Primary current measuring range I_{PM} (A)	Type
1000	± 1100	**HTC 1000-S**
2000	± 2200	**HTC 2000-S**
3000	± 3300	**HTC 3000-S**

V_C	Supply voltage (± 3 %)	± 15	V
I_C	Current consumption	< ± 20	mA
R_{IS}	Insulation resistance @ 500 VDC	> 500	MΩ
V_{out}	Output voltage (Analog) @ $\pm I_{PN\,DC}$, R_L=2kΩ, T_A=25°C	± 10	V
R_{OUT}	Output internal resistance	< 100	Ω
V_d	Rms voltage for AC isolation test, 50 Hz,1min	2.5	kV
R_L	Load resistance	≥ 2	kΩ

Accuracy-Dynamic performance data

X	Accuracy @ $I_{PN\,DC}$, T_A = 25°C	< ± 1	%of $I_{PN\,DC}$
ε_L	Linearity error (0 .. ± $I_{PN\,DC}$)	< ± 1	%of $I_{PN\,DC}$
V_{OE}	Eletrical offset voltage @ T_A = 25°C	< ± 30	mV
V_{OH}	Hysteresis offset voltage @ I_P = 0, after an excursion of 1 x $I_{PN\,DC}$	< ± 50	mV
TCV_{OE}	Temperature coefficient of V_{OE}	< ± 1.0	mV/K
TCV_{OUT}	Temperature coefficient of V_{OUT}	≤ ± 0.1	%/K
t_r	Response time to 90% of $I_{PN\,DC}$ step @ di/dt = 100A/µs	≤ 10	µs
BW	Frequency bandwidth (- 3dB)	DC .. 10	kHz

General data

T_A	Ambient operating temperature	- 40 .. + 85	°C
T_S	Ambient storage temperature	- 40 .. + 85	°C
m	Mass	450	g
	Standards[1]	EN 50155	

Note :
[1] Specification according to IEC 61000-4-3 are not guaranteed between 260 and 290 MHz (value higher by 5% than the specification).

Features

- Hall effect measuring principle
- Galvanic insulation between primary and secondary circuit
- Insulated plastic case recognized according to UL 94-V0

Advantages

- Easy installation
- Compact
- High immunity to external interference
- Low power consumption

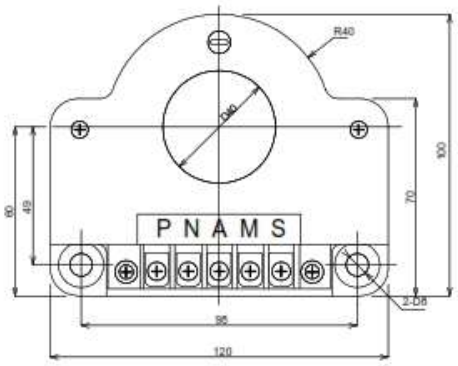
Application

- Traction

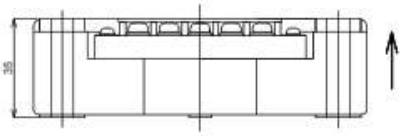
HALAMAN INI SENGAJA DIKOSONGKAN

Dimensions HTC 1000..3000-S (in mm. 1 mm = 0.0394 inch)

Front view



Bottom view



Pins Assignment

P : +V
N : -V
A : Output
M : 0V
S : 0V

Safety

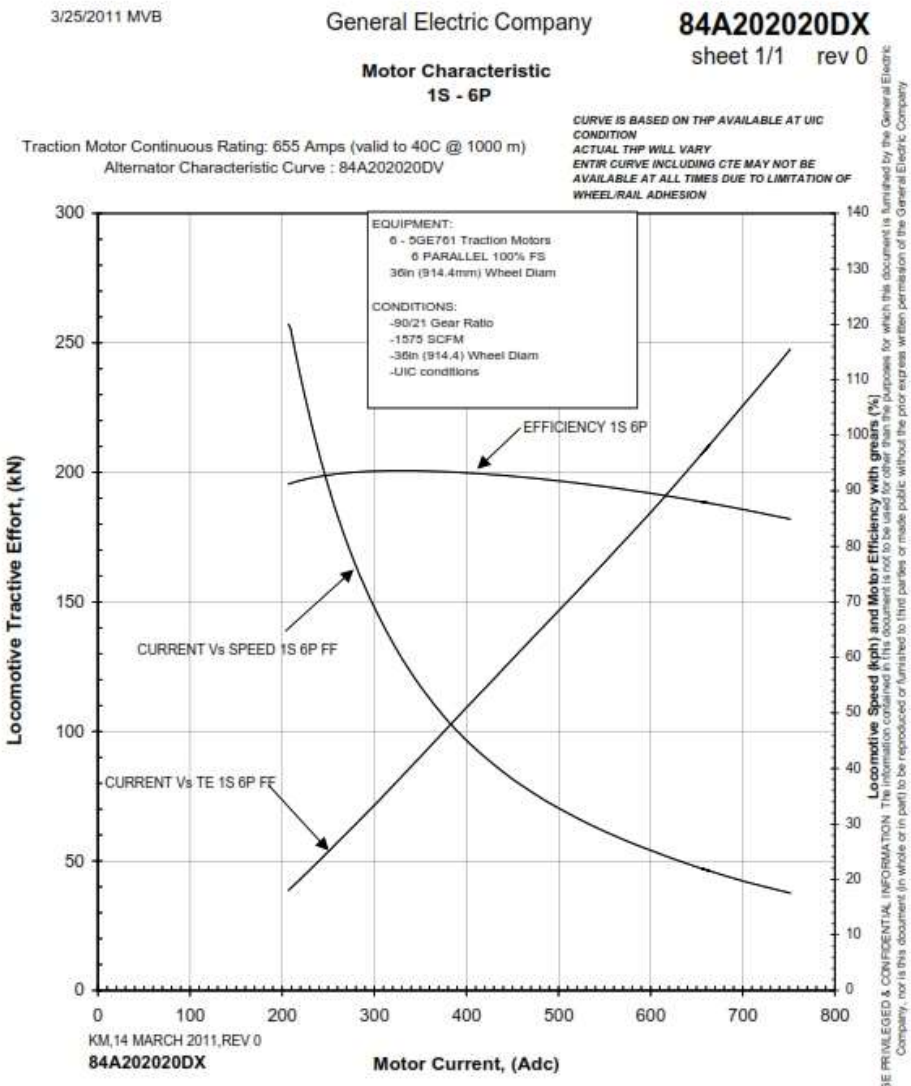
This transducer must be used in electric/electronic equipment with respect to applicable standards and safety requirements in accordance with the following manufacturer's operating instructions.

Caution, risk of electrical shock

When operating the transducer, certain parts of the module can carry hazardous voltage (eg. primary busbar, power supply). Ignoring this warning can lead to injury and/or cause serious damage.
This transducer is a built-in device, whose conducting parts must be inaccessible after installation.
A protective housing or additional shield could be used.
Main supply must be able to be disconnected.

HALAMAN INI SENGAJA DIKOSONGKAN

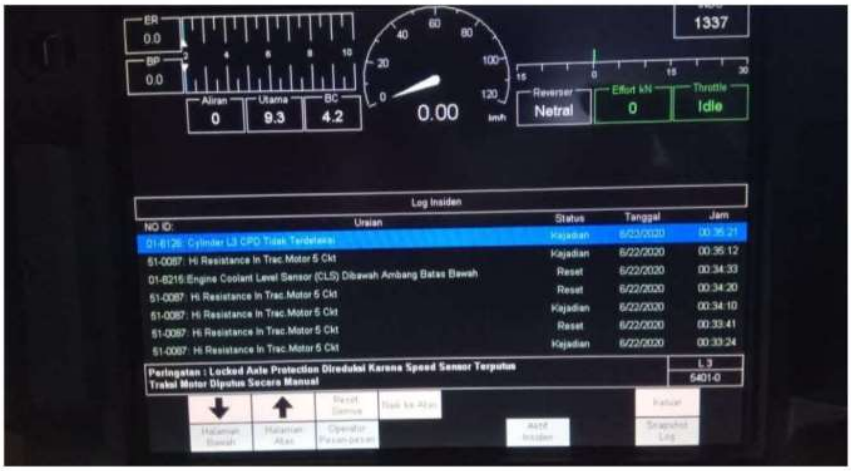
Lampiran Karakteristik Traksi Motor



-Tolerance on curve is +/- 5%
-Entire curve may not be achievable at all times due to limitation of wheel/rail adhesion.
-Actual THP will vary.

HALAMAN INI SENGAJA DIKOSONGKAN

Lampiran Kodisi Hi Resistant Lokomotif CC 206



imagination at work

HALAMAN INI SENGAJA DIKOSONGKAN

Lampiran Program

```
#define BLYNK_PRINT Serial
TFT_eSPI tft = TFT_eSPI();
RTC_DS1307 rtc;
File myFile;
DateTime now;
char auth[] = "U0QULD4nL8Dc53kbP4_0-V2xZOdweMJc";
char ssid[] = "bonek";
char pass[] = "a1b2c3d4";
const int chipSelect = SS;
char daysOfTheWeek[7][12] = {"Minggu", "Senin", "Selasa", "Rabu", "Kamis",
"Jumat", "Sabtu"};
int Sensor1, Sensor2, Sensor3;
const int pin_Sens1 = 39;
const int pin_Sens2 = 34;
const int pin_Sens3 = 35;
#define RED2RED 0
#define GREEN2GREEN 1
#define BLUE2BLUE 2
#define BLUE2RED 3
#define GREEN2RED 4
#define RED2GREEN 5
#define TFT_BLACK       0x0000      /*   0,   0,   0 */
#define TFT_NAVY        0x000F      /*   0,   0, 128 */
#define TFT_DARKGREEN   0x03E0      /*   0, 128,   0 */
#define TFT_DARKCYAN    0x03EF      /*   0, 128, 128 */
#define TFT_MAROON      0x7800      /* 128,   0,   0 */
#define TFT_PURPLE      0x780F      /* 128,   0, 128 */
#define TFT_OLIVE       0x7BE0      /* 128, 128,   0 */
#define TFT_LIGHTGREY   0xD69A      /* 211, 211, 211 */
#define TFT_GREY        0x7BEF      /* 128, 128, 128 */
#define TFT_BLUE        0x001F      /*   0,   0, 255 */
#define TFT_GREEN       0x07E0      /*   0, 255,   0 */
#define TFT_CYAN        0x07FF      /*   0, 255, 255 */
#define TFT_RED         0xF800      /* 255,   0,   0 */
#define TFT_MAGENTA     0xF81F      /* 255,   0, 255 */
#define TFT_YELLOW      0xFFE0      /* 255, 255,   0 */
#define TFT_WHITE       0xFFFF      /* 255, 255, 255 */
#define TFT_ORANGE      0xFDA0      /* 255, 180,   0 */
#define TFT_GREENYELLOW 0xB7E0      /* 180, 255,   0 */
#define TFT_PINK        0xFE19      /* 255, 192, 203 */
#define TFT_BROWN       0x9A60      /* 150,  75,   0 */
#define TFT_GOLD        0xFEA0      /* 255, 215,   0 */
#define TFT_SILVER      0xC618      /* 192, 192, 192 */
#define TFT_SKYBLUE     0x867D      /* 135, 206, 235 */
```

HALAMAN INI SENGAJA DIKOSONGKAN

```
#define TFT_VIOLET      0x915C      /* 180,  46, 226 */
#include <Wire.h>
#include <SPI.h>
#include <SD.h>
#include <TFT_eSPI.h>
#include "RTClib.h"
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#include "inisialisasi.h"
#include "EmonLib.h"
EnergyMonitor emon1, emon2, emon3;
void setup() {
 Serial.begin(115200);
 delay(10);
 tft.init();
 tft.setRotation(1);
 Tampil_pembuka();
 while (!SD.begin(SS)) {
  Tampil_cekSD1();
 }
 Tampil_cekSD2();
 //  delay(1000);
 //  tft.fillScreen(TFT_BLACK);
 //  tesSD();
 delay(2000);
 while (! rtc.begin()) {
  Tampil_RTC1();
 }
 Tampil_RTC2();
 delay(2000);
 while (! rtc.isrunning()) {
  Tampil_RTC3();
  //    setRTC();
 }
 Tampil_wifi1();
 WiFi.begin(ssid, pass);
 int wifi_ctr = 0;
 while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  tft.setCursor(0, 140, 0);
  tft.setTextColor(TFT_PINK);
  tft.setTextFont(4);
  tft.print(".");
 }
 Tampil_wifi2();
```

HALAMAN INI SENGAJA DIKOSONGKAN

```
  Blynk.begin(auth, ssid, pass);
  kalibrasi_sensor();
}
void loop() {
  bacaRTC();
  bacaSensor();
  tft.fillScreen(TFT_BLACK);
  Tampil_waktu();
  Tampil_sensor();
  cek_alert();
  Blynk.run();
  Blynk.virtualWrite(V0, Sensor1);
  Blynk.virtualWrite(V1, Sensor2);
  Blynk.virtualWrite(V2, Sensor3);
  Blynk.virtualWrite(V3, Sensor1);
  Blynk.virtualWrite(V4, Sensor2);
  Blynk.virtualWrite(V5, Sensor3);
  delay(20);
}
void kalibrasi_sensor() {
  emon1.current(pin_Sens1, 1000.1);
  emon2.current(pin_Sens2, 1000.1);
  emon3.current(pin_Sens3, 1000.1);
}
void bacaSensor() {
  Sensor1 = emon1.calcIrms(1480);
  Sensor2 = emon2.calcIrms(1480);
  Sensor3 = emon3.calcIrms(1480);
}
void setRTC() {
  rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
}
void bacaRTC() {
  now = rtc.now();
}
void tulisSD1() {
  myFile = SD.open("/dataLOG_Monitoring.txt", FILE_APPEND);
  if (myFile) {
    tft.setCursor(5, 210, 0);
    tft.setTextColor(TFT_RED);
    tft.setTextFont(3);
    tft.println("Sedang menulis ke test.txt...");
    myFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    myFile.print(" -> ");
    myFile.print(now.day(), DEC);
    myFile.print("/");
```

HALAMAN INI SENGAJA DIKOSONGKAN

```
    myFile.print(now.month(), DEC);
    myFile.print("/");
    myFile.print(now.year(), DEC);
    myFile.print("  <>  ");
    myFile.print(now.hour(), DEC);
    myFile.print(":");
    myFile.print(now.minute(), DEC);
    myFile.print(":");
    myFile.print(now.second(), DEC);
    myFile.print("  <>  ");
    myFile.println("-> Traksi Motor 1 Detected <-");
    myFile.close();
    tft.setCursor(5, 230, 0);
    tft.setTextColor(TFT_RED);
    tft.setTextFont(3);
    tft.println("Selesai menulis ke test.txt...");
  } else {
    tft.setCursor(5, 210, 0);
    tft.setTextColor(TFT_RED);
    tft.setTextFont(3);
    tft.println("tidak dapat membuka test.txt...");
  }
}
void tulisSD2() {
  myFile = SD.open("/dataLOG_Monitoring.txt", FILE_APPEND);
  if (myFile) {
    tft.setCursor(5, 210, 0);
    tft.setTextColor(TFT_RED);
    tft.setTextFont(3);
    tft.println("Sedang menulis ke test.txt...");
    myFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    myFile.print(" -> ");
    myFile.print(now.day(), DEC);
    myFile.print("/");
    myFile.print(now.month(), DEC);
    myFile.print("/");
    myFile.print(now.year(), DEC);
    myFile.print("  <>  ");
    myFile.print(now.hour(), DEC);
    myFile.print(":");
    myFile.print(now.minute(), DEC);
    myFile.print(":");
    myFile.print(now.second(), DEC);
    myFile.print("  <>  ");
    myFile.println("-> Traksi Motor 2 Detected <-");
    myFile.close();
```

HALAMAN INI SENGAJA DIKOSONGKAN

```
    tft.setCursor(5, 230, 0);
    tft.setTextColor(TFT_RED);
    tft.setTextFont(3);
    tft.println("Selesai menulis ke test.txt...");
   } else {
    tft.setCursor(5, 210, 0);
    tft.setTextColor(TFT_RED);
    tft.setTextFont(3);
    tft.println("tidak dapat membuka test.txt...");
   }
}
void tulisSD3() {
  myFile = SD.open("/dataLOG_Monitoring.txt", FILE_APPEND);
  if (myFile) {
    tft.setCursor(5, 210, 0);
    tft.setTextColor(TFT_RED);
    tft.setTextFont(3);
    tft.println("Sedang menulis ke test.txt...");
    myFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    myFile.print(" -> ");
    myFile.print(now.day(), DEC);
    myFile.print("/");
    myFile.print(now.month(), DEC);
    myFile.print("/");
    myFile.print(now.year(), DEC);
    myFile.print("  <>  ");
    myFile.print(now.hour(), DEC);
    myFile.print(":");
    myFile.print(now.minute(), DEC);
    myFile.print(":");
    myFile.print(now.second(), DEC);
    myFile.print("  <>  ");
    myFile.println("-> Traksi Motor 3 Detected <-");
    myFile.close();
    tft.setCursor(5, 230, 0);
    tft.setTextColor(TFT_RED);
    tft.setTextFont(3);
    tft.println("Selesai menulis ke test.txt...");
   } else {
    tft.setCursor(5, 210, 0);
    tft.setTextColor(TFT_RED);
    tft.setTextFont(3);
    tft.println("tidak dapat membuka test.txt...");
   }
}
```

HALAMAN INI SENGAJA DIKOSONGKAN

```
void bacaSD() {
  myFile = SD.open("/dataLOG_Monitoring.txt");
  if (myFile) {
    tft.setCursor(60, 0, 0);
    tft.println("test.txt:");
    while (myFile.available()) {
      tft.setCursor(80, 0, 0);
      tft.println(myFile.read());
    }
    myFile.close();
  } else {
    tft.setCursor(120, 0, 0);
    tft.println("tidak dapat membuka test.txt...");
  }
}
void tesSD() {
  switch (SD.cardType()) {
    case CARD_NONE:
      tft.setCursor(10, 10, 0);
      tft.setTextColor(TFT_RED);
      tft.setTextFont(4);
      tft.print("Card type: ");
      tft.println("NONE");
      break;
    case CARD_MMC:
      tft.setCursor(10, 10, 0);
      tft.setTextColor(TFT_RED);
      tft.setTextFont(4);
      tft.print("Card type: ");
      tft.println("MMC");
      break;
    case CARD_SD:
      tft.setCursor(10, 10, 0);
      tft.setTextColor(TFT_RED);
      tft.setTextFont(4);
      tft.print("Card type: ");
      tft.println("SD");
      break;
    case CARD_SDHC:
      tft.setCursor(10, 10, 0);
      tft.setTextColor(TFT_RED);
      tft.setTextFont(4);
      tft.print("Card type: ");
      tft.println("SDHC");
      break;
    default:
```

HALAMAN INI SENGAJA DIKOSONGKAN

```
    tft.setCursor(10, 10, 0);
    tft.setTextColor(TFT_RED);
    tft.setTextFont(4);
    tft.print("Card type: ");
    tft.println("Unknown");
  }
  tft.setCursor(10, 40, 0);
  tft.setTextColor(TFT_RED);
  tft.setTextFont(4);
  tft.print("Card size:  ");
  tft.println((float)SD.cardSize() / 1000);
  tft.setCursor(10, 70, 0);
  tft.setTextColor(TFT_RED);
  tft.setTextFont(4);
  tft.print("Total bytes: ");
  tft.println(SD.totalBytes());
  tft.setCursor(10, 100, 0);
  tft.setTextColor(TFT_RED);
  tft.setTextFont(4);
  tft.print("Used bytes: ");
  tft.println(SD.usedBytes());
  //  File dir =  SD.open("/");
  //  printDirectory(dir, 0);
}
void printDirectory(File dir, int numTabs) {
  while (true) {
    File entry =  dir.openNextFile();
    if (! entry) {
      // no more files
      break;
    }
    for (uint8_t i = 0; i < numTabs; i++) {
      Serial.print('\t');
    }
    Serial.print(entry.name());
    if (entry.isDirectory()) {
      Serial.println("/");
      printDirectory(entry, numTabs + 1);
    } else {
      // files have sizes, directories do not
      Serial.print("\t\t");
      Serial.print(entry.size(), DEC);
      time_t lw = entry.getLastWrite();
      struct tm * tmstruct = localtime(&lw);
```

HALAMAN INI SENGAJA DIKOSONGKAN

```
      Serial.printf("\tLAST WRITE: %d-%02d-%02d %02d:%02d:%02d\n",
(tmstruct->tm_year) + 1900, (tmstruct->tm_mon) + 1, tmstruct->tm_mday,
tmstruct->tm_hour, tmstruct->tm_min, tmstruct->tm_sec);
    }
    entry.close();
  }
}
void Tampil_pembuka() {
  tft.fillScreen(TFT_BLACK);
  tft.setCursor(0, 0, 0);
  tft.setTextColor(TFT_PINK);
  tft.setTextFont(4);
  tft.println("inisialisasi SDcard. . .");
}
void Tampil_cekSD1() {
  tft.fillScreen(TFT_BLACK);
  tft.setCursor(0, 30, 0);
  tft.setTextColor(TFT_RED);
  tft.setTextFont(4);
  tft.println("SDcard Tidak Terdeteksi");
}
void Tampil_cekSD2() {
  tft.fillScreen(TFT_BLACK);
  tft.setCursor(0, 30, 0);
  tft.setTextColor(TFT_RED);
  tft.setTextFont(4);
  tft.println("SDcard Terdeteksi dengan baik");
}
void Tampil_RTC1() {
  tft.fillScreen(TFT_BLACK);
  tft.setCursor(0, 60, 0);
  tft.setTextColor(TFT_BLUE);
  tft.setTextFont(4);
  tft.println("RTC tidak ada");
}
void Tampil_RTC2() {
  tft.fillScreen(TFT_BLACK);
  tft.setCursor(0, 60, 0);
  tft.setTextColor(TFT_BLUE);
  tft.setTextFont(4);
  tft.println("RTC Terdeteksi dengan baik");
}
void Tampil_RTC3() {
  tft.fillScreen(TFT_BLACK);
  tft.setCursor(0, 90, 0);
  tft.setTextColor(TFT_RED);
```

HALAMAN INI SENGAJA DIKOSONGKAN

```
  tft.setTextFont(4);
  tft.println("RTC perlu Setting ulang");
}
void Tampil_wifi1() {
  tft.fillScreen(TFT_BLACK);
  tft.setCursor(0, 120, 0);
  tft.setTextColor(TFT_RED);
  tft.setTextFont(4);
  tft.println("Wifi sedang proses connect ... ");
}
void Tampil_wifi2() {
  tft.fillScreen(TFT_BLACK);
  tft.setCursor(0, 120, 0);
  tft.setTextColor(TFT_RED);
  tft.setTextFont(4);
  tft.println("Wifi Connected");
}
void Tampil_sensor() {
  tft.setCursor(76, 160, 0);
  tft.setTextColor(TFT_BLUE);
  tft.setTextFont(4);
  tft.print("Amp");
  tft.setCursor(216, 160, 0);
  tft.setTextColor(TFT_BLUE);
  tft.setTextFont(4);
  tft.print("Amp");
  tft.setCursor(355, 160, 0);
  tft.setTextColor(TFT_BLUE);
  tft.setTextFont(4);
  tft.print("Amp");
  int xpos1 = 40, ypos = 65, radius = 60;
  int xpos2 = 180;
  int xpos3 = 320;
  ringMeter(Sensor1, 0, 2000, xpos1, ypos, radius, " Watts", RED2RED);
  ringMeter(Sensor2, 0, 2000, xpos2, ypos, radius, " mA", RED2RED);
  ringMeter(Sensor3, 0, 2000, xpos3, ypos, radius, " Volt", RED2RED);
}
void Tampil_waktu() {
  tft.setCursor(140, 5, 0);
  tft.setTextColor(TFT_RED);
  tft.setTextFont(4);
  tft.print(daysOfTheWeek[now.dayOfTheWeek()]);
  tft.print(" -> ");
  tft.print(now.day(), DEC);
  tft.print("/");
  tft.print(now.month(), DEC);
```

HALAMAN INI SENGAJA DIKOSONGKAN

```
  tft.print("/");
  tft.println(now.year(), DEC);
  tft.setCursor(190, 30, 0);
  tft.setTextColor(TFT_RED);
  tft.setTextFont(4);
  tft.print(now.hour(), DEC);
  tft.print(":");
  tft.print(now.minute(), DEC);
  tft.print(":");
  tft.println(now.second(), DEC);
}
void cek_alert() {
  if (Sensor1 < 50) {
    tft.setCursor(175, 190, 0);
    tft.setTextColor(TFT_RED);
    tft.setTextFont(4);
    tft.println("WARNING");
    tft.setCursor(80, 220, 0);
    tft.setTextColor(TFT_RED);
    tft.setTextFont(4);
    tft.println(" -> Traksi Motor 1 Detected <-");
    Blynk.notify(" -> Traksi Motor 1 Detected <-");
    tulisSD1();
    delay(3000);
  }
  if (Sensor2 < 50) {
    tft.setCursor(175, 190, 0);
    tft.setTextColor(TFT_RED);
    tft.setTextFont(4);
    tft.println("WARNING");
    tft.setCursor(80, 250, 0);
    tft.setTextColor(TFT_RED);
    tft.setTextFont(4);
    tft.println(" -> Traksi Motor 2 Detected <-");
    Blynk.notify(" -> Traksi Motor 2 Detected <-");
    tulisSD2();
    delay(3000);
  }
  if (Sensor3 < 50) {
    tft.setCursor(175, 190, 0);
    tft.setTextColor(TFT_RED);
    tft.setTextFont(4);
    tft.println("WARNING");
    tft.setCursor(80, 280, 0);
    tft.setTextColor(TFT_RED);
    tft.setTextFont(4);
```

HALAMAN INI SENGAJA DIKOSONGKAN

```
    tft.println(" -> Traksi Motor 3 Detected <-");
    Blynk.notify(" -> Traksi Motor 3 Detected <-");
    tulisSD3();
    delay(3000);
   }
}
int ringMeter(int value, int vmin, int vmax, int x, int y, int r, const char *units,
byte scheme)
{
  x += r; y += r;
  int w = r / 3;
  int angle = 150;
  int v = map(value, vmin, vmax, -angle, angle);
  byte seg = 3;
  byte inc = 6;
  int colour = TFT_BLUE;
  for (int i = -angle + inc / 2; i < angle - inc / 2; i += inc) {
    float sx = cos((i - 90) * 0.0174532925);
    float sy = sin((i - 90) * 0.0174532925);
    uint16_t x0 = sx * (r - w) + x;
    uint16_t y0 = sy * (r - w) + y;
    uint16_t x1 = sx * r + x;
    uint16_t y1 = sy * r + y;
    float sx2 = cos((i + seg - 90) * 0.0174532925);
    float sy2 = sin((i + seg - 90) * 0.0174532925);
    int x2 = sx2 * (r - w) + x;
    int y2 = sy2 * (r - w) + y;
    int x3 = sx2 * r + x;
    int y3 = sy2 * r + y;
    if (i < v) {
      switch (scheme) {
        case 0: colour = TFT_RED; break;
        case 1: colour = TFT_GREEN; break;
        case 2: colour = TFT_BLUE; break;
        default: colour = TFT_BLUE; break;
      }
      tft.fillTriangle(x0, y0, x1, y1, x2, y2, colour);
      tft.fillTriangle(x1, y1, x2, y2, x3, y3, colour);
    }
    else // Fill in blank segments
    {
      tft.fillTriangle(x0, y0, x1, y1, x2, y2, TFT_GREY);
      tft.fillTriangle(x1, y1, x2, y2, x3, y3, TFT_GREY);
    }
  }
  char buf[10];
```

HALAMAN INI SENGAJA DIKOSONGKAN

```
  byte len = 3; if (value > 999) len = 5;
  dtostrf(value, len, 0, buf);
  buf[len] = ' '; buf[len + 1] = 0;
  tft.setTextSize(1);
  tft.setTextColor(TFT_WHITE, TFT_BLACK);
  tft.setTextColor(colour, TFT_BLACK);
  tft.setTextDatum(MC_DATUM);
  if (r > 84) {
    tft.setTextPadding(55 * 3);
    tft.drawString(buf, x, y, 8);
  }
  else {
    tft.setTextPadding(3 * 14); // Allow for 3 digits each 14 pixels wide
    tft.drawString(buf, x, y, 4); // Value in middle
  }
  return x + r;
}
```