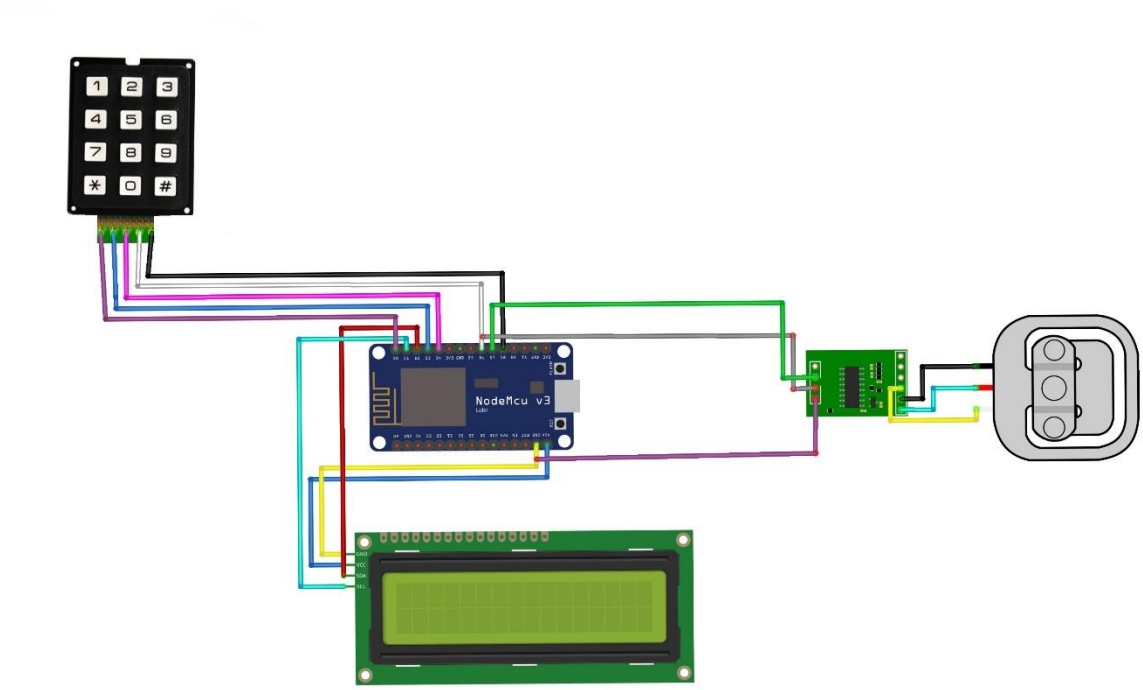
LAMPIRAN

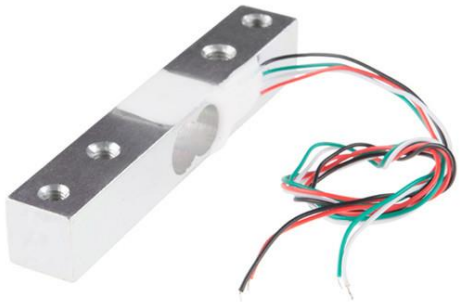
LAMPIRAN 1

Gambar Rangkaian



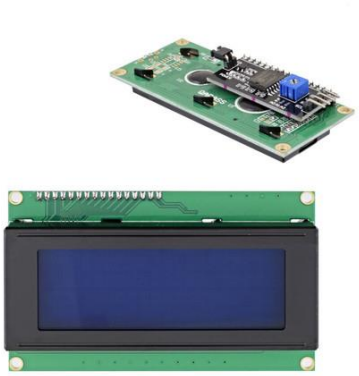
LAMPIRAN 2

Spesifikasi Komponen Alat



Spesifikasi Sensor Loadcell	
Ukuran Sensor	3.16 x 0.51 x 0.51 inch
Max Operation Voltage	15 VDC
Kisaran Berat	5 Kg
Bahan	Alumunium

Spesifikasi Modul HX711	
Operation Voltage	2.7V-5V
Untuk Loadcell	5-200 kg
Dual-Channel	24 Bit



Spesifikasi LCD	
Ukuran	60 Mm 90 Mm
Interface	I2C
Tegangan Suplai	DC 5V
Tampilan	20 Karakter 4 Baris

Spesifikasi Keypad	
Bahan	Plastik+Papan PCB
Ukuran	5x7 Cm
Tampilan	3x4 matrik

Spesifikasi NodeMCU ESP8266	
Mikrokontroler	ESP8266
Input Tegangan	3.3 V ~ 5V
Ukuran *Board*	37 mm x 30 mm
GPIO	13 PIN
Flash Memory	4 MB
Wireles	802.11 b/g/n standar
USB to Serial Converter	CH340G

LAMPIRAN 3

Kode Program Alat

```
//---Library Wifi-------------------
#include <ESP8266WiFi.h>              //sertakan library wifi
//---access point wifi---------------
#define WIFI_SSID       "ANBU"
#define WIFI_PASSWORD   "1234567890"
//#define WIFI_SSID     "ANBU"  //tentukan SSID yang tuju
//#define WIFI_PASSWORD  "1234567890"       //tentukan password SSID yang digunakan

unsigned long previousMillis = 0; //siapkan variable previousMillis dengan tipe data unsigned long, set nilai awal = 0
//const long interval2 = 1000; //siapkan variable interval2 dengan tipe data const long, set nilai awal = 1000

//---------------------------------------
#include <FirebaseESP8266.h>  //sertakan library firebase
#define FIREBASE_HOST "https://avos-880d4-default-rtdb.firebaseio.com"  //tentukan url firebase
#define FIREBASE_AUTH "AIzaSyBolMvjf8F-4V32aXmd9IzgLR46ICvKK8B" //tentukan auth firebase
FirebaseData DataDB; //buat instance baru dari object FirebaseData
//FirebaseData ledData;

//---------------------------------------
#include <NTPClient.h>  //sertakan library NTPClient.h
#include <WiFiUdp.h>  //sertakan library WiFiUdp.h
#include <time.h>   //sertakan library time.h
WiFiUDP ntpUDP; //buat instance ntpUPD dari object WIFIUDP

NTPClient timeClient(ntpUDP, "pool.ntp.org"); //siapkan website ntp server yang dituju
int currentHour, currentMinute, currentSecond, monthDay, currentMonth, currentYear;  //siapkan variabel currentHour, currentMinute, currentSecond dengan
String waktu2, currentDate; //siapkan beberapa variable string

//---------------------------------------
#include <Wire.h> //sertakan lib wire untuk i2c
#include <LiquidCrystal_I2C.h> //sertakan lib LCD i2c
LiquidCrystal_I2C lcd(0x27, 20, 4); //definisikan address, kolom & baris LCD

//---------------------------------------
#include <Keypad_I2C.h> //sertakan lib keypad lcd
#define I2CADDR 0x26      //A0 sambung ke ground

const byte ROWS = 4; //jumlah baris keypad
const byte COLS = 3; //jumlah kolom keypad
char keys[ROWS][COLS] = { //matrix array urutan pin keypad
  {'1', '2', '3'},
  {'4', '5', '6'},
  {'7', '8', '9'},
  {'*', '0', '#'}
};
//definisikan urutan pin keypad disambung ke pin modul i2c :
//susun secara lurus: C2 pin paling kiri,  R2 pin paling kanan

#define C2 7  //definisikan Coloum 2 sebagai pin 7
#define R1 6  //definisikan Row 1 sebagai pin 6
#define C1 5  //definisikan Coloum 1 sebagai pin 5
#define R4 4  //definisikan Row 4 sebagai pin 4 //pin 3 i2c tidak digunakan
#define C3 2  //definisikan Coloum 3 sebagai pin 2
#define R3 1  //definisikan Row 3 sebagai pin 1
#define R2 0  //definisikan Row 2 sebagai pin 0

// Digitran keypad, bit numbers of PCF8574 i/o port
byte rowPins[ROWS] = {R1, R2, R3, R4};  //buat susunan array Row
byte colPins[COLS] = {C1, C2, C3};      //buat susunana array Coloum

TwoWire *jwire = &Wire;   //test passing pointer to keypad lib
Keypad_I2C kpd( makeKeymap(keys), rowPins, colPins, ROWS, COLS, I2CADDR, PCF8574, jwire );
//Keypad_I2C kpd( makeKeymap(keys), rowPins, colPins, ROWS, COLS, I2CADDR );

String pin_id, amount, weight, price; //siapkan beberapa variable string
String nama, berat, harga; //siapkan beberapa variable string
float berat2, harga2; //siapkan beberapa variable float
int i = 0; //siapkan beberapa variable integer
bool login = 0; //siapkan beberapa variable boolean

//---------------------------------------
///#include <Arduino.h>
#include "HX711.h"  //sertakan lib modul hx711
const int LOADCELL_DOUT_PIN = D6; //definisikan pin DOUT
const int LOADCELL_SCK_PIN = D7; //definisikan pin SCK
HX711 scale; //buat instance dari object HX711

//---------------------------------------
#include <ArduinoJson.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClient.h>
WiFiClient wifiClient;
 const char* host      = "http://api.timezonedb.com/v2/get-time-zone?key=NEBL8NB6CW2Z&format=json&fields=formatted&by=zone&zone=Asia/Jakarta";

// Variables to accept data
String payload;
String nowday;
String nowmonth;
String nowyear;
String nowhour;
String nowmin;
String nowsec;

//---------------------------------------
void setup() { //fungsi setup
  Serial.begin(115200); //set baudrate serial
  delay(200); //beri delay sebentar
  Serial.println("Timbangan digital versi 1.2");
  pinMode(LED_BUILTIN, OUTPUT); //set led internal sebagai output

  //0.init lcd
  lcd.init();       //init lcd
  lcd.backlight();  //init backlight
  menu_awal();      //panggil fungsi menu_awal()

  //1. Start wifi
  konekwifi();      //panggil fungsi konekwifi()

  //2. Firebase begin
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH); //aktifkan firebase
  Firebase.reconnectWiFi(true);  //aktifkan koneksi ulang firebase

  //3. NTP begin
//  timeClient.begin(); //aktifkan firebase
//  timeClient.setTimeOffset(3600 * 7); //seting timezone = +7

  //3. Timezonedb begin
```

```
  //3. Timezonedb begin
  Serial.print( "Timezonedb begin" );
  tzdb();
  parse_response_json();
  print_waktu();
  delay(200); //beri delay sebentar

  //4. Keypad i2c begin
  jwire->begin( ); //siapkan keypad i2c
  //  kpd.begin( makeKeymap(keys) );
  kpd.begin( ); //aktifkan keypad i2c
  Serial.print( "start with pinState = " );
  Serial.println( kpd.pinState_set( ), HEX ); //normalnya terdeteksi 7F

  //5. Loadcell begin
  Serial.println("Initializing the scale");
  scale.begin(LOADCELL_DOUT_PIN, LOADCELL_SCK_PIN); //aktifkan modul hx711
  delay(1000); //beri delay
  setup_loadcell(); //panggil fungsi setup_loadcell()

  //tes set nilai awal
  //pin_id = "123456";
  //amount = "20000";
  //weight = "2";
}

void loop() { //fungsi loop
main: //label main untuk looping menu awal lcd
  char key = kpd.getKey(); //baca keypad & simpan pada variable key
  delay(200); //beri delay sebentar
  if (key) { //periksa isi key
    Serial.println(key); //print ke serial

    if (key == '*') { //jika key=*
      lcd.setCursor(0, 3); //set kursor
      lcd.print("Input Pin: "); //tampilan pesan di lcd
x1:
      key = kpd.getKey(); //baca lagi keypad
      delay(200); //beri delay sebentar

      if (key && key != '#') { //periksa isi key
        lcd.print(key); //tampilkan ke lcd
        pin_id = pin_id + key; //tampung berapa pun digitnya ke variable pin_id
      }

      else if (key == '#') { //jika isi key=# panggil fungsi berikutnya
        lcd.print(key); //print key # ke lcd
        get_firebase(); //panggil fungsi untuk baca data firebase

        if (nama == "null") { //jika respon firebase=null
          lcd.setCursor(0, 3); //set kursor
          lcd.print("Pin keliru          "); //beri pesan ke lcd
          Serial.println("user tidak ditemukan"); //beri pesan ke serial
          pin_id = ""; //bersihkan lagi variabel pin_id
          delay(3000); //kasih delay agak lama
          menu_awal(); //panggil tampilan menu awal
          goto main; //kembali ke label main
        }
        else if (nama != "null") { //jika respon firebase bukan null
          login = true; //flag login seting jadi true
          Serial.println("user ditemukan"); //beri pesan ke serial
          menu_lcd(); //panggil fungsi menu_lcd()
          goto main2; //pindah ke label main2
        }
      }
      goto x1; //looping terus ke label x1 selama masih input pin
    }
  }

  timer_millis(); //panggil fungsi millis jika diperlukan
  goto main;  //looping terus ke label main

main2: //label main2 untuk looping di menu baca loadcell
  key = kpd.getKey(); //baca keypad lagi
  delay(200); //beri delay sebentar

  if (key == '#') { //jika key # maka kirim data ke firebase
    lcd.setCursor(14, 3); //set kursor
    lcd.print("Kirim"); //beri pesan di lcd
    amount = String(berat2); //pindahkan nilai yang terbaca
    weight = String(harga2); //pindahkan nilai yang terbaca
    //baca_ntp(); //baca waktu ke ntp
    tzdb();
    parse_response_json();
    print_waktu();

    set_firebase(); //kirim data ke firebase
    delay(2000); //beri delay agak lama
    menu_lcd(); //panggil tampilan menu lcd
  }

  else if (key == '0') { //jika key=0 maka logout
    login = false; //set flag login jadi false
    pin_id = ""; //kosongkan lagi variable pin_id
    menu_awal(); //panggil menu awal
    goto main; //kembali ke label main
  }

  timer_millis(); //panggil millis untuk baca loadcell rutin
  goto main2; //looping ke label main2 selama masih login
}

void timer_millis() { //fungsi millis
  //if (Firebase.ready() && (millis() - previousMillis >= 1000)) { //baca RTC setiap 1000ms atau 1 detik
  if (millis() - previousMillis >= 1000) { //jalankan millis setiap 1000ms atau 1 detik
    previousMillis = millis(); //masukan nilai currentMillis ke variable previousMillis
    digitalWrite(LED_BUILTIN, !digitalRead(LED_BUILTIN)); //buat toggle nyala lampu led
    if (login) {  //jika flag login = true
      cek_loadcell(); //baca loadcell
      //berat2 = random(1, 10); //nilai random hanya untuk tes; perlu nonaktifkan
      harga2 = berat2/1000 * price.toInt(); //hitung harga total = berat x price
      lcd.setCursor(7, 2); //set kursor
      lcd.print("          "); //bersihkan dulu layar dari tampilan sebelumnya
      lcd.setCursor(7, 2);  //set kursor
      lcd.print(berat2); //tampilkan berat ke lcd
      lcd.setCursor(7, 3);  //set kursor
      lcd.print("          ");  //bersihkan dulu layar dari tampilan sebelumnya
      lcd.setCursor(7, 3);  //set kursor
      lcd.print(harga2); //tampilkan harga ke lcd
    }
  }
}

void menu_awal() { //fungsi menu awal
  lcd.clear(); //bersihkan semua tamilan lcd
  lcd.setCursor(0, 0); //set kursor
  lcd.print(" TIMBANGAN DIGITAL "); //tampilan ke lcd
  lcd.setCursor(0, 1); //set kursor
  lcd.print("         BSN  "); //tampilan ke lcd
  lcd.setCursor(0, 2); //set kursor
  lcd.print("  AL IHYA ULUMADDIN    "); //tampilan ke lcd
  lcd.setCursor(0, 3); //set kursor
  lcd.print("(*)Login           "); //tampilan ke lcd
}

void menu_lcd() { //fungsi menu lcd
  lcd.clear(); //bersihkan semua tampilan layar
```

```
  lcd.setCursor(0, 0); //set kursor
  lcd.print(" TIMBANGAN DIGITAL "); //tampilan ke lcd
  lcd.setCursor(0, 1);  //set kursor
  lcd.print("Nama : "); //tampilan ke lcd
  lcd.print(nama); //tampilan ke lcd
  lcd.setCursor(0, 2); //set kursor
  lcd.print("Berat:"); //tampilan ke lcd
  lcd.setCursor(0, 3);  //set kursor
  lcd.print("Harga:"); //tampilan ke lcd
  lcd.setCursor(0, 3); //set kursor
  //lcd.print("(0)Logout   Kirim(#)");
}

void get_firebase() { //fungsi baca database firebase
  Serial.println("Get Firebase"); //tampilkan ke serial
  if (Firebase.ready()) { //jika firebase ready
    Firebase.getString(DataDB, ("users/" + pin_id + "/profile/name")); //baca field name
    //nama = DataDB.stringData(); //ini masih ada tanda kutip " "
    nama = DataDB.to<const char *>(); //ini bisa menghilangkan tanda kutif
    Serial.print("Nama: "); //tampilkan ke serial
    Serial.println(nama);  //tampilkan ke serial
    delay(100); //beri delay sebentar

    Firebase.getString(DataDB, ("users/price")); //baca field price
    price = DataDB.to<const char *>(); //ini bisa menghilangkan tanda kutif
    Serial.print("Harga: ");  //tampilkan ke serial
    Serial.println(price); //tampilkan ke serial
  }
  else Serial.println("Firebase not ready");  //tampilkan ke serial
}

void set_firebase() { //fungsi set database firebase
  Serial.println("Set Firebase");  //tampilkan ke serial
  Serial.println(amount);  //tampilkan ke serial
  Serial.println(weight);  //tampilkan ke serial
  Firebase.setString(DataDB, "users/" + pin_id + "/history/" + currentDate + " " + waktu2 + "/amount", amount); //set field amount
  Firebase.setString(DataDB, "users/" + pin_id + "/history/" + currentDate + " " + waktu2 + "/weight", weight); //ste field weight
}

void baca_ntp() { //fungsi baca ntp
  timeClient.update(); //panggil fungsi timeClient.update() pada lib ntp
  delay(200);
  unsigned long epochTime = timeClient.getEpochTime(); //panggil fungsi timeClient.getEpochTime() dan simpan hasilnya pada variable epochTime dengan ti
  //String formattedTime = timeClient.getFormattedTime(); //panggil fungsi timeClient.getFormattedTime() dan simpan hasilnya pada variable formattedTim
  waktu2 = timeClient.getFormattedTime(); //panggil & simpan ke variable waktu2
  currentHour = timeClient.getHours();     //panggil fungsi timeClient.getHours() dan simpan hasilnya pada variable currentHour
  currentMinute = timeClient.getMinutes();  //panggil fungsi timeClient.getMinutes() dan simpan hasilnya pada variable currentMinute
  currentSecond = timeClient.getSeconds();  //panggil fungsi timeClient.Seconds() dan simpan hasilnya pada variable currentSecond


  struct tm *ptm = gmtime ((time_t *)&epochTime); //buat a time structure
  monthDay = ptm->tm_mday;
  currentMonth = ptm->tm_mon + 1;
  currentYear = ptm->tm_year + 1900;
  //Print complete date:
  currentDate = String(currentYear) + "-" + String(currentMonth) + "-" + String(monthDay);
  //  Serial.print("Current date: ");
  //  Serial.println(currentDate);
}

void setup_loadcell() {
  if (scale.is_ready()) {
    Serial.println("1. Before setting up the scale:");
    Serial.print("a. read: \t\t");
    Serial.println(scale.read());      // print a raw reading from the ADC
    delay(100);

    Serial.print("b. read average: \t");
    Serial.println(scale.read_average(20));   // print the average of 20 readings from the ADC
    delay(100);

    Serial.print("c. get value: \t\t");
    Serial.println(scale.get_value(5));   // print the average of 5 readings from the ADC minus the tare weight (not set yet)
    delay(100);

    Serial.print("d. get units: \t\t");
    Serial.println(scale.get_units(5), 1);  // print the average of 5 readings from the ADC minus tare weight (not set) divided
    // by the SCALE parameter (not set yet)
    delay(100);
  }
  else Serial.println("HX711 not ready1");

  if (scale.is_ready()) {
    //    scale.set_scale(-478.507);
    scale.set_scale(91783 / 198); //hp sony 91783/198 ram > isi ini error
    //scale.set_scale(-471.497);                  // this value is obtained by calibrating the scale with known weights; see the README for details
    delay(100);
    scale.tare();               // reset the scale to 0
    delay(100);
  }
  else Serial.println("HX711 not ready2");

  if (scale.is_ready()) {
    Serial.println("2. After setting up the scale:");

    Serial.print("a. read: \t\t");
    Serial.println(scale.read());                 // print a raw reading from the ADC
    delay(100);

    Serial.print("b. read average: \t");
    Serial.println(scale.read_average(20));       // print the average of 20 readings from the ADC
    delay(100);

    Serial.print("c. get value: \t\t");
    Serial.println(scale.get_value(5));   // print the average of 5 readings from the ADC minus the tare weight, set with tare()
    delay(100);

    Serial.print("d. get units: \t\t");
    Serial.println(scale.get_units(5), 1);        // print the average of 5 readings from the ADC minus tare weight, divided
    // by the SCALE parameter set with set_scale
    delay(100);
  }

  else Serial.println("HX711 not ready3");
}

void cek_loadcell() {
  Serial.print("one reading:\t");
  Serial.print(scale.get_units(), 1);
  berat2 = 0;
  berat2 = scale.get_units();
  if (berat2<1) berat2 = 0;
  Serial.print("\t| average:\t");

  Serial.println(berat2);
  scale.power_down();             // put the ADC in sleep mode
  delay(100);
  scale.power_up();
}
```

```
void konekwifi() {
  WiFi.disconnect();
  delay(200);
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  //Connecting...
  while (WiFi.status() != WL_CONNECTED) {
   Serial.print("...."); //wait
   delay(1000);
  }
  Serial.println();
  Serial.println("Connected");
}


void tzdb() {
  int httpCode = 0;                                   // Variable to hold received data
  HTTPClient http;                                    // Declare an object of class HTTPClient

  Serial.println("Connecting to TimezoneDB...");

  http.begin(wifiClient, host);                                 // Connect to site
  httpCode = http.GET();                              // Check if data is coming in

  while (httpCode == 0) {                             // if no data is in
    delay(1000);                                      // wait a sec
    http.begin(wifiClient, host);                               // and try again
    httpCode = http.GET();
  }

  payload = http.getString();                         // Save response as string
  Serial.println(payload);                            // Show response

  http.end();                                               // Close connection to timezonedb
  //WiFi.mode(WIFI_OFF);                              // Close connection to WiFi
}

void parse_response_json() {

  /*sample json response:
    {
    "status": "OK",
    "message": "",
    "formatted": "2022-01-12 15:18:07"
    }
  */
  //DynamicJsonBuffer jsonBuffer; //json5

  DynamicJsonDocument doc(1024); //json6
  String input = payload;
  //JsonObject& root = jsonBuffer.parseObject(input); //json5
  deserializeJson(doc, input); //json6

  //String tanggal = root[String("formatted")];
  String tanggal = doc[String("formatted")];
  Serial.println(tanggal);
  //2022-01-12 16:06:28
  nowday    = tanggal.substring(0,4);
  nowmonth  = tanggal.substring(5,7);
  nowyear   = tanggal.substring(8,10);

  nowhour   = tanggal.substring(11,13);
  nowmin    = tanggal.substring(14,16);
  nowsec    = tanggal.substring(17,19);

  waktu2 = nowyear +'-'+ nowmonth +'-'+ nowday;
  currentDate = nowhour +':'+ nowmin +':'+ nowsec;
  }

  void print_waktu(){
  // Print integers without leading zeros - use in calculations
  //  Serial.print("Today's date is:   ");
  //  Serial.print(y);
  //  Serial.print("-");
  //  Serial.print(mo);
  //  Serial.print("-");
  //  Serial.println(d);
  //
  //  Serial.print("Current local time is:   ");
  //  Serial.print(h);
  //  Serial.print(":");
  //  Serial.print(mi);
  //  Serial.print(":");
  //  Serial.println(s);
  //  Serial.println();

  //Print characters with leading zeros - use for lcd etc.
  Serial.print("Tanggal : ");
  Serial.print(nowyear);
  Serial.print("-");
  Serial.print(nowmonth);
  Serial.print("-");
  Serial.println(nowday);


  Serial.print("Waktu   : ");
  Serial.print(nowhour);
  Serial.print(":");
  Serial.print(nowmin);
  Serial.print(":");
  Serial.println(nowsec);
  Serial.println();
}
```

Link kode program aplikasi

https://drive.google.com/file/d/1D6MEH3D7VHQJpvdKSQXf9rCPGawTL0ni/view?usp=share_link

LAMPIRAN 4

Observasi Di Bank Sampah Al Ihya Ulumaddin


PERSEPPA
5


BR 1-7 (CRUMBLE)