REPUBLIK INDONESIA
KEMENTERIAN HUKUM DAN HAK ASASI MANUSIA

SURAT PENCATATAN
CIPTAAN

Dalam rangka pelindungan ciptaan di bidang ilmu pengetahuan, seni dan sastra berdasarkan Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta, dengan ini menerangkan:

Nomor dan tanggal permohonan	:	EC00201821905, 23 Juli 2018
Pencipta		
Nama	:	**Slamet Winardi, Dadang Supriyatno, , dkk**
Alamat	:	Jl. Jemur Ngawinan No. 46-D, Surabaya, Jawa Timur, -
Kewarganegaraan	:	Indonesia
Pemegang Hak Cipta		
Nama	:	**Slamet Winardi**
Alamat	:	Jl. Jemur Ngawinan No. 46-D , Surabaya, Jawa Timur, -
Kewarganegaraan	:	Indonesia
Jenis Ciptaan	:	**Buku**
Judul Ciptaan	:	**Manual User Dives Digital Identification Vehicle System**
Tanggal dan tempat diumumkan untuk pertama kali di wilayah Indonesia atau di luar wilayah Indonesia	:	20 Juni 2018, di Surabaya
Jangka waktu pelindungan	:	Berlaku selama hidup Pencipta dan terus berlangsung selama 70 (tujuh puluh) tahun setelah Pencipta meninggal dunia, terhitung mulai tanggal 1 Januari tahun berikutnya.
Nomor pencatatan	:	000112537

adalah benar berdasarkan keterangan yang diberikan oleh Pemohon.
Surat Pencatatan Hak Cipta atau produk Hak terkait ini sesuai dengan Pasal 72 Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.

a.n. MENTERI HUKUM DAN HAK ASASI MANUSIA
DIREKTUR JENDERAL KEKAYAAN INTELEKTUAL

Dr. Freddy Harris, S.H., LL.M., ACCS.
NIP. 196611181994031001

MANUAL USER
DIVES DIGITAL IDENTIFICATION VEHICLE SISTEM

Disusun Oleh :

Slamet Winardi,S.T.,M.T.
Dr. Ir. Dadang Supriyatno, MT
Benediktus Anindito, S.Kom, M.MT
Natalia Damastuti, ST, MT
Bayu Saputra

Diterbitkan oleh :

Narotama University Press

NAROTAMA University Press

Manual User Dives Digital Identification Vehicle Sistem
/disusun oleh Slamet Winardi, dkk.
40 hal; viii ; editor, Seger S.S

Copyright © 2017 oleh Narotama University Press

Hak Cipta Dilindungi Undang-Undang
© All Rights Reserved

Dilarang mengutip dan memperbanyak sebagian atau seluruhnya dalam bentuk apapun tanpa ijin tertulis dari Penerbit

© Cetakan Pertama Juni 2018

Ukuran Buku : A5 (14,8 x 21 cm)

Penyusun : Slamet Winardi,S.T.,M.T.
Dr. Ir. Dadang Supriyatno, MT
Benediktus Anindito, S.Kom, M.MT
Natalia Damastuti, ST, MT
Bayu Saputra

Editor : Seger S.S
Layout/Setting : G. Purwantoro
Design Cover : Gatut Purwantoro

ISBN : 978-602-6557-32-2

© HAK CIPTA DILINDUNGI UNDANG-UNDANG
Isi diluar tanggungjawab Penerbit

Diterbitkan oleh **Narotama *University Press***
Jl. Arief Rachman Hakim No.51 Surabaya 60117
Telp: 031-5946404, 5995578 Fax: 031-5931213
Website: www.narotama.ac.id
Email: narotamapress@narotama.ac.id

KATA PENGANTAR

Terimakasih kami panjatkan kepada Tuhan Yang Maha Esa atas segala rahmat dan nikmatnya serta kesehatan yang selalu diberikan kepada kami. Atas berkatnya kami dapat menuliskan sebuah buku manual untuk mengoperasi-kan sebuah software hasil Penelitian Unggulan Perguruan Tinggi yang kami peroleh pada tahun 2017.

Sebuah software identifikasi kendaraan bermotor secara digital yang kami berinama DIVeS (*Digital Identification Vehicle System*) untuk melengkapi proses pengoperasian software tersebut. Terimakasih kami sampaikan juga kepada Dirjen Ristek Dikti yang telah memberikan kesempatan kepada tim penulis untuk melakasanakan penelitian yang menghasilkan beberapa luaran.

Terimakasih juga kepada Rektor dan LPPM Universitas Narotama yang telah memberikan kesempatan dan dorongan untuk melakukan penelitian tersebut hingga menghasilkan sebuah buku manual ini. Semoga buku manual ini dapat bermanfaat bagi pembaca serta memberikan sebuah ide dan pengetahuan yang baru tentang digitalisasi identitas kendaraan bermotor.

Manual buku ini intinya terdiri dari 3 bagian yaitu panduan software untuk client kendaraan bermotor sebagai identitas digital kendaraan bermotor, yang kedua sebagai scanner untuk menangkap sinyal wifi dari kendaraan bermotor yang memancarkan identitas kendaraan bermotor tersebut, dan yang ketiga adalah sebuah server yang berfungsi untuk

menyimpan database, menampilkan dan memproses system tersebut.

Mungkin masih banyak kekurangan dari buku manual tersebut yang kami buat, saran dan kritik yang membangun kami butuhkan untuk menyempurnakan buku manual ini. Terakhir saya ucapkan terimakasih kepada semua pihak sehingga penulis dapat menyelesaikannya.

Surabaya, Mei 2018

Tim Penulis

DAFTAR ISI

KATA PENGANTAR iii

DAFTAR ISI v

DAFTAR GAMBAR vi

1. Install Arduino IDE 3

2. Install Board Wemos di Arduino IDE 3

INSTALASI CHIP DIVES 6

1. Instalasi Board Wemos sebagai *Client* 10

2. Instalasi Board Wemos D1 Mini sebagai *Scanner Client* 14

3. Melakukan *Testing* pada kedua Alat Identifikasi Kendaraan 16

DAFTAR GAMBAR

Gambar 1 Download Page Arduino IDE..... 1
Gambar 2 Tampilan awal Arduino IDE 3
Gambar 3 Koneksi Ke Board Arduino 5
Gambar 4 Koneksi ke Comm Port di Komputer 5
Gambar 5 Program Blink 6
Gambar 6 Install Board Wemos ke Arduino IDE 7
Gambar 7 Path json Wemos D1 mini 7
Gambar 8 JSON Board manager Arduino IDE 8
Gambar 9 Install board wemos dan modul ESP8266 9
Gambar 10 Mengenal Board Wemos D1 Mini 9
Gambar 11 Seting Board Wemos D1 Mini 10
Gambar 12 Seting Port yang Digunakan Arduino 11
Gambar 13 Code Client Identifikasi Kendaraan 12
Gambar 14 Daftar Pemilik Kendaraan 13
Gambar 15 Seting Wemos D1 Mini 14
Gambar 16 Seting Port Arduino yang Digunakan..... 15
Gambar 17 Kode Scanner Identifikasi 16
Gambar 18 Blok Diagram Sistem Identifikasi Kendaraan Bermotor..... 18
Gambar 19 Tampilan Data Daftar Kendaraan Bermotor Yang Teridentifikasi 19
Gambar 20 Detail Data Kendaraan Bermotor 20

SANKSI PELANGGARAN PASAL 113

Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta:

(1) Setiap Orang yang dengan tanpa hak melakukan pelanggaran hak ekonomi sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf i untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 1 (satu) tahun dan/atau pidana denda paling banyak Rp 100.000.000 (seratus juta rupiah).

(2) Setiap Orang yang dengan tanpa hak dan/atau tanpa izin Pencipta atau pemegang Hak Cipta melakukan pelanggaran hak ekonomi Pencipta sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf c, huruf d, huruf f, dan/atau huruf h untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 3 (tiga) tahun dan/atau pidana denda paling banyak Rp 500.000.000,00 (lima ratus juta rupiah).

(3) Setiap Orang yang dengan tanpa hak dan/atau tanpa izin Pencipta atau pemegang Hak Cipta melakukan pelanggaran hak ekonomi Pencipta sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf a, huruf b, huruf e, dan/atau huruf g untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 4 (empat) tahun dan/atau pidana denda paling banyak Rp 1.000.000.000,00 (satu miliar rupiah).

(4) Setiap Orang yang memenuhi unsur sebagaimana dimaksud pada ayat (3) yang dilakukan dalam bentuk pembajakan, dipidana dengan pidana penjara paling lama 10 (sepuluh) tahun dan/atau pidana denda paling banyak Rp 4.000.000.000,00 (empat miliar rupiah).

Universitas Narotama Surabaya

INSTALASI SISTEM

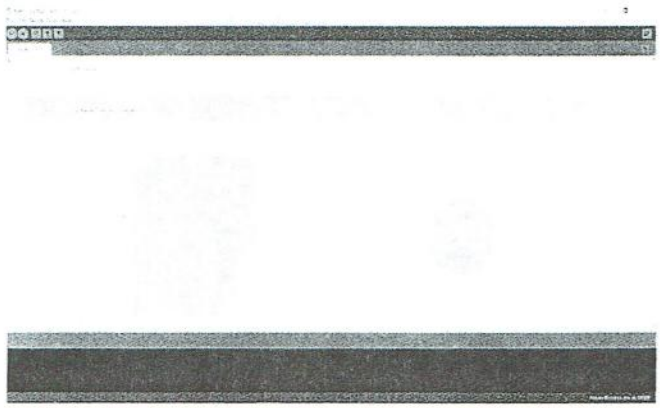
1. **Install Arduino IDE**

   Buka browser anda, masukan https://www.arduino.cc/en/Main/Software di kolom url browser anda. Akan muncul seperti gambar berikut:

Gambar 1. Download Page Arduino IDE

Sesuaikan sistem operasi yang sedang di gunakan, diasumsikan bahwa menggunakan sistem operasi Windows, jadi silahkan anda klik Windows Installer. Setelah diklik, Arduino IDE proses di download. Setelah terdownload maka install program Arduino IDE tersebut.

Setelah selesai install, sekarang masuk ke program Arduino IDE yang sudah terinstal. Selain versi Windows installer dapat dipilih versi zip. Untuk versi zip, setelah melakukan download program lakukan ekstrak file tersebut sehingga menjadi kumpulan file Arduino yang tersimpan dalam sebuah folder Windows. Tampilannya akan seperti berikut:

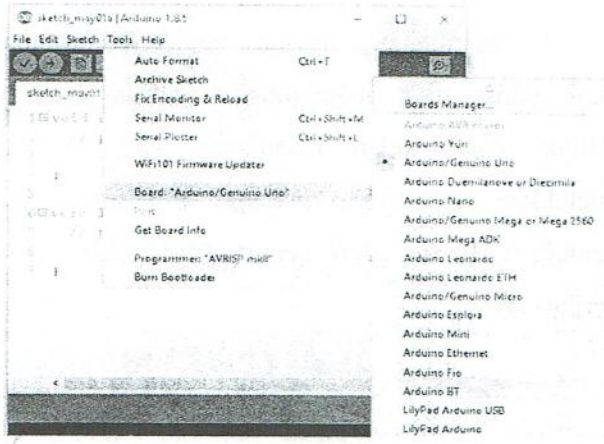


Gambar 2. Tampilan awal Arduino IDE

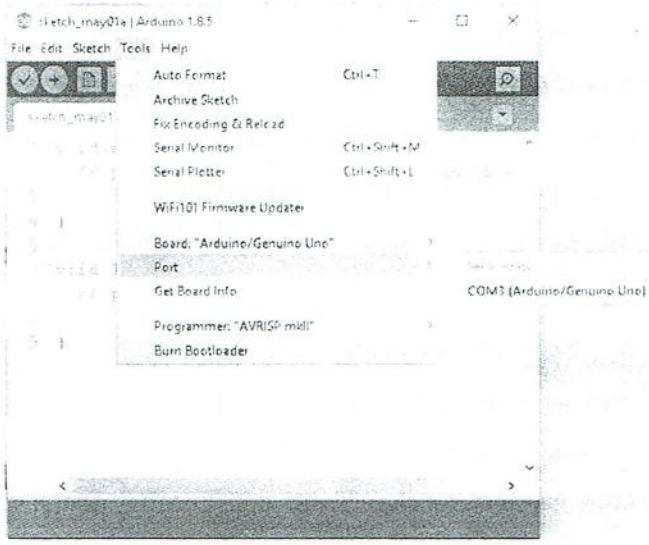
Setelah terekstrak dan file Arduino dijalankan maka secara default akan muncul dua buah fungsi (void) yaitu void setup yang befungsi mendeklarasikan fungsi-fungsi Arduino dan void setup yang berfungsi untuk membuat program utama.

Sebelum membuat program pastikan mikrokontroler terhubung dengan computer yang dipastikan dengan melihat ***tools*** pada IDE Arduino untuk memastikan Board Arduino yang digunakan sama dengan seting di softwarenya, missal menggunakan mikrokontroler Arduino Uno pastikan boardnya juga dipilih sesuai dengan hardwarenya.

Selanjutnya memastikan port yang dinamakan Arduino telah tercentang atau terpilih oleh software Arduino, jika belum ada lakukan update driver serial. Untuk gambar koneksi board dan koneksi serial seperti ditunjukkan dibawah ini.

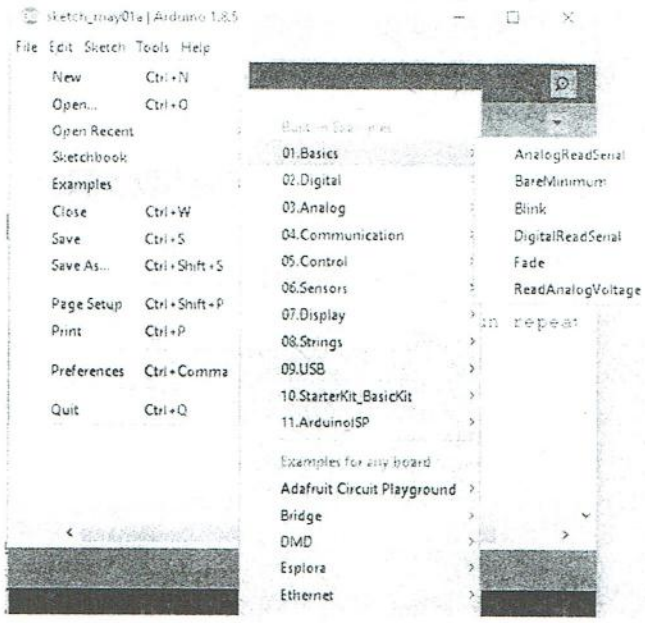


Gambar 3. Koneksi Ke Board Arduino



Gambar 4. Koneksi ke Comm Port di Komputer

Untuk menguji apakah boardnya berfungsi atau tidak dapat diuji menggunakan program dasar yaitu "Blink" yang telah disediakan oleh Arduino dan diletakkan pada folder ***example*** dan sub folder Basic kemudian cari file Blink, seperti ditunjukkan pada gambar berikut ini.



Gambar 5. Program Blink

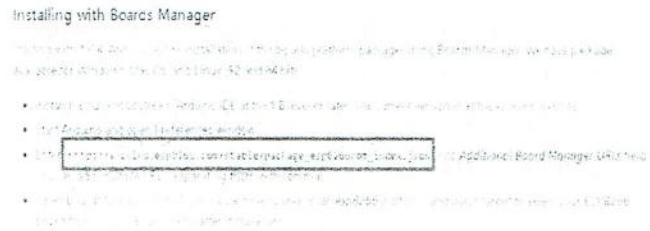
2. Install Board Wemos di Arduino IDE

Setelah Arduino IDE ter-install, selanjutnya harus menginstall Board Wemos agar bisa melakukan pengembangan di Board Wemos melalui program Arduino IDE. Pertama buka browser lalu ketikan link ini yang ada di kolom url browser berikut ini. https://github.com/esp8266/Arduino disitu dapat ditemukan tutorial bagaimana install board wemos di board manager Arduino

IDE. Berikut ditunjukan bagaimana instalasi board wemos di board manager Arduino IDE:

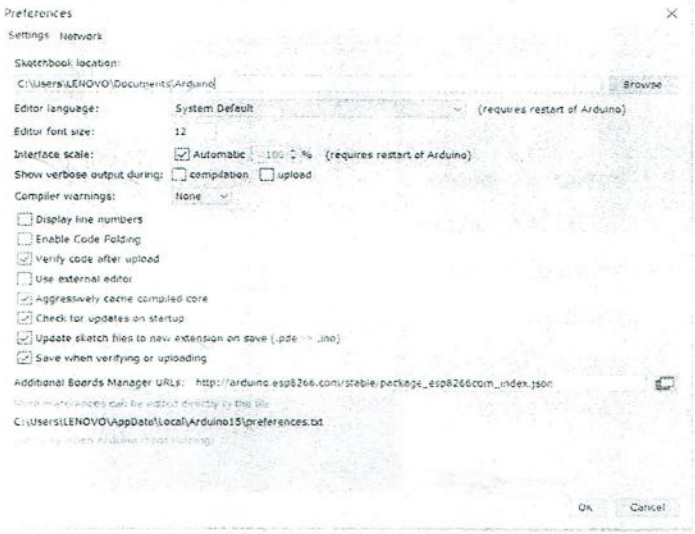
Gambar 6. Install Board Wemos ke Arduino IDE

Jika belum menemukan path json untuk board Wemos ini, dapat dicari di github. Alamat URL untuk keperluan di atas dapat dibrowsing di Google Search Engine dengan kata kunci "*github json esp8266*" kemudian cari title yang mengandung alamat link path json wemos seperti ditunjukkan pada gambar di bawah ini.



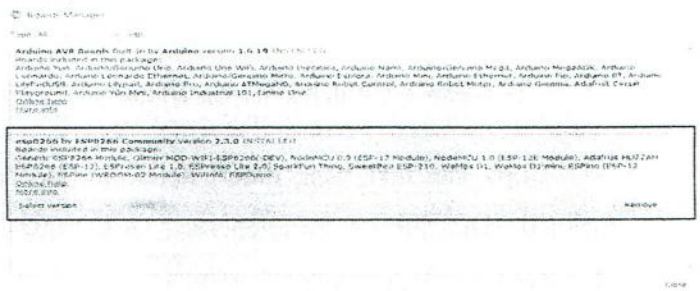
Gambar 7. Path json Wemos D1 mini

Copy kan text yang diberi kotak warna merah ke Arduino IDE. File > Preferences > Additional Boards Manager Urls. Seperti gambar berikut:



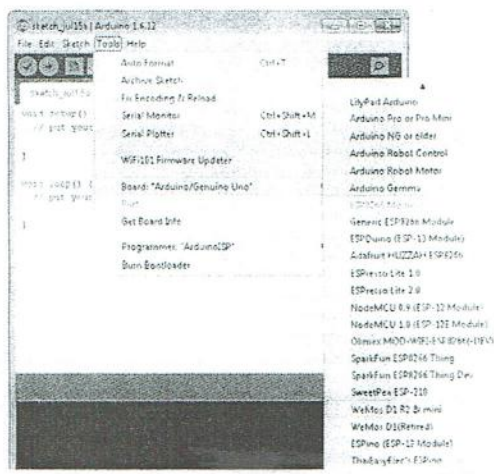
Gambar 8. JSON Board manager Arduino IDE

Setelah itu update board manager dengan seting di Tools > Board "Arduino/Genuino Ino" > Board Manager. Selanjutnya tulis di kolom search **ESP8266** seperti gambar berikut.

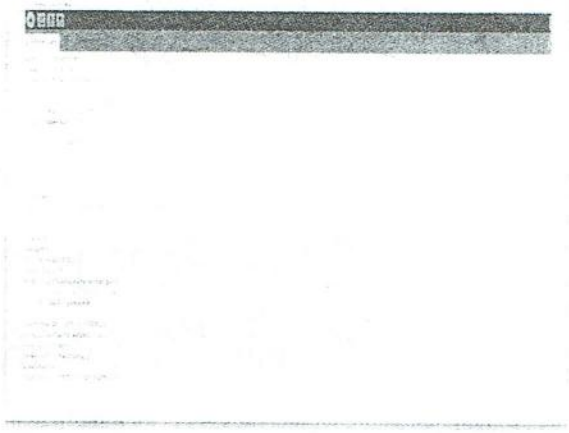


Gambar 9. Install board wemos dan modul ESP8266

Setelah semua proses di atas selesai, maka board ESP8266 akan muncul pada pilihan menu *Tools* -> *Board* seperti ini:



Gambar 10. Mengenal Board Wemos D1 Mini



Gambar 13. Code Client Identifikasi Kendaraan

d. Ganti nilai *ssid* dengan nopol yang sudah terdaftar di *WebService* identifikasi kendaraan http://identifikasi.inodroid-narotama.id melalui admin. Seperti gambar berikut.

Gambar 14. Daftar Pemilik Kendaraan

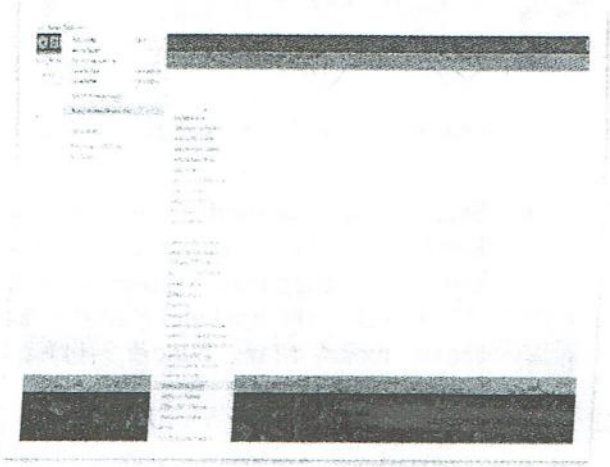
Setelah terdaftar maka di Arduino IDE klik tombol *Upload.* Dari sini telah berhasil melakukan instalasi Board Wemos D1 Mini sebagai *Client.*

INSTALASI CHIP DIVES

1. **Instalasi Board Wemos sebagai *Client***

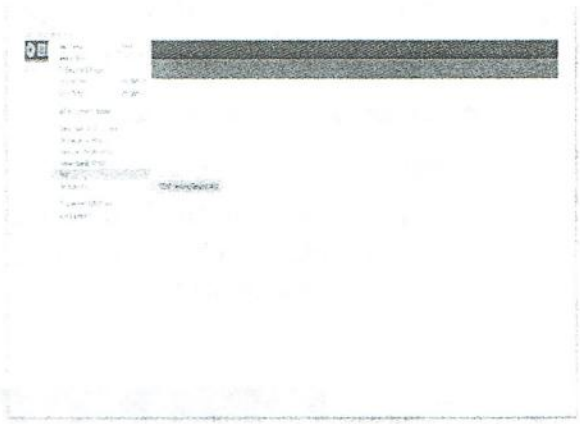
   Dalam instalasi *Client* di Board Wemos dibutuhkan Editor Arduino IDE yang sudah dijelaskan di atas langkah-langkah instalasi-nya, dan juga Board Wemos D1 Mini. Berikut langkah-langkah instalasi *Client* di Board Wemos D1 Mini.

   a. Buka Arduino IDE lalu ubah Settingan Board yang semula default ke "Wemos D1 Mini", seperti gambar berikut.



Gambar 11. Seting Board Wemos D1 Mini

b. Setelah Arduino IDE dibuka dan Board sudah diubah sesuai dengan kebutuhan, selanjutnya hubungkan USB dari Board Wemos D1 Mini ke USB Port PC/Laptop. Setelah itu hubungkan ke Arduino IDE dengan mengubah Port, seperti gambar berikut.

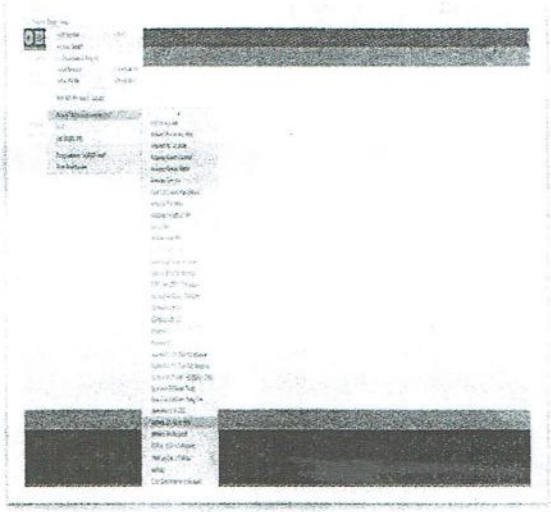


Gambar 12. Seting Port yang Digunakan Arduino

c. Selanjutnya buka sumber kode dari Identifikasi Kendaraan *Client* dengan cara "File->Open" atau bisa menggunakan kombi-nasi keyboard "CTRL+O". lalu arahkan ke sumber kode, jika sudah maka akan muncul gambar seperti berikut.

2. **Instalasi Board Wemos D1 Mini sebagai *Scanner Client***

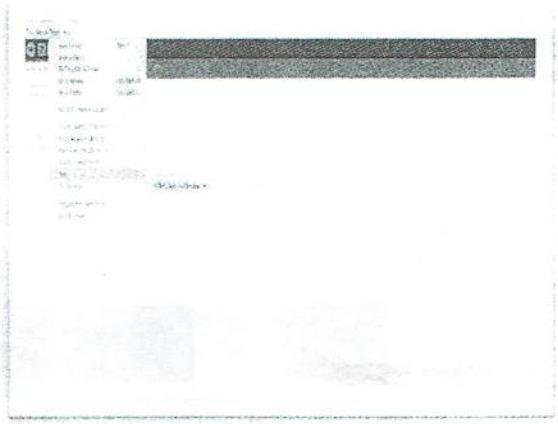
Setelah melakukan instalasi Board Wemos D1 Mini sebagai *Client* selanjutnya lakukan instalasi Board Wemos D1 Mini sebagai *Scanner Client* yang berfungsi sebagai alat perekam nopol yang sudah tertanam pada Board Wemos D1 Mini. Berikut langkah-langkah instalasi Board Wemos D1 Mini sebagai *Scanner Client*.



Gambar 15. Seting Wemos D1 Mini

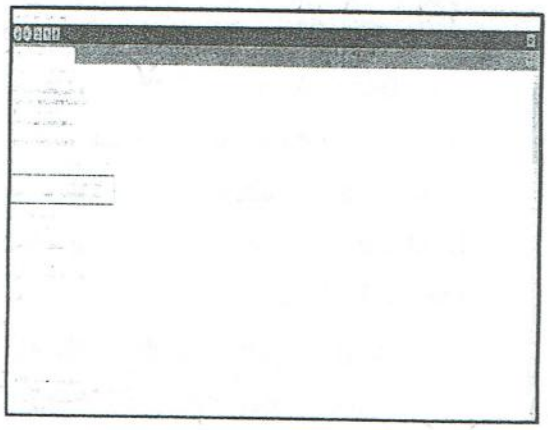
a. Buka Arduino IDE anda lalu ubah Settingan Board yang semula default ke "Wemos D1 Mini", seperti Gambar 15 berikut.

b. Setelah Arduino IDE dibuka dan Board sudah diubah sesuai dengan kebutuhan, selanjutnya hubungkan USB dari Board Wemos D1 Mini ke USB Port PC/Laptop. Setelah itu hubungkan ke Arduino IDE dengan mengubah Port, seperti gambar berikut.



Gambar 16. Seting Port Arduino yang Digunakan

c. Selanjutnya buka sumber kode dari Identifikasi Kendaraan *Scanner Client* dengan cara "File->Open" atau bisa menggunakan kombinasi keyboard "CTRL+O". lalu arahkan ke sumber kode, jika sudah maka akan muncul gambar seperti berikut.



Gambar 17. Kode Scanner Identifikasi

d. Setelah berhasil membuka sumber kode Identifikasi Kendaraan *Scanner List* perhatikan kotak hitam pada gambar di atas. Ada tulisan *ssid & pass*. Ubah nilai *ssid & pass* sesuai dengan koneksi WiFi yang saat ini anda gunakan. Setelah diubah, silahkan

anda klik tombol *Upload* dan tunggu beberapa saat maka alat *Scanner Client* sudah terinstall di Board Wemos D1 Mini.

3. **Melakukan *Testing* pada kedua Alat Identifikasi Kendaraan**

Setelah melakukan kedua tahap instalasi pada Board Wemos D1 Mini sebagai client dan scanner identifikasi kendaraan bermotor, berikutnya dilakukan *Testing* dari proses identifikasi kendaraan langkah-langkahnya sebagai berikut.

a. Pasang alat client pada kendaraan bermotor (sepeda motor dana tau mobil) kemudian beri tegangan modul tersebut pada saat kendaraan bermotor dinyalakan. Modul client tersebut akan memancarkan sinyal wifi berupa SSID plat nomor kendaraan bermotor sebagai identitas digital kendaraan tersebut

b. Pasang modul scanner pada lokasi tertentu yang digunakan untuk mendeteksi

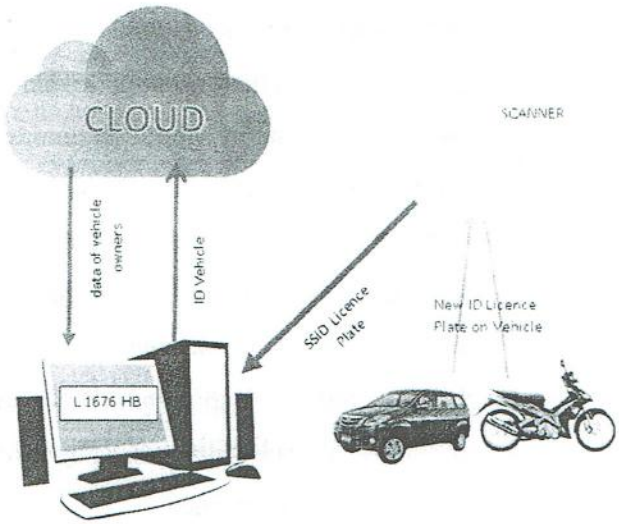
kendaraan bermotor tersebut, misalnya di pintu masuk parker atau di lokasi parker kendaraan bermotor. Modul scanner tersebut akan mendeteksi sinyal wifi yang dipancarkan oleh client yang terpasang di kendaraan bermotor, berupa SSID kendaraan bermotor masing-masing.

c. Buka page http://identifikasi.inodroid-narotama. id pada computer yang terhubung dengan jaringan internet.

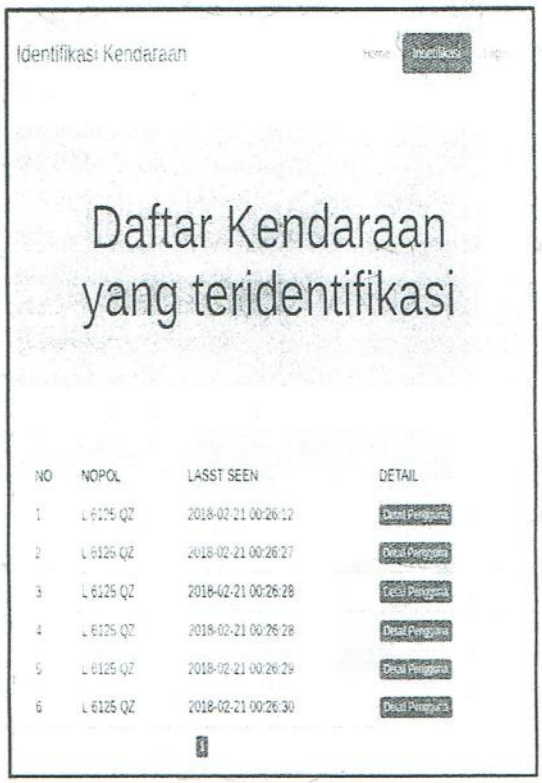
d. Kendaraan bermotor yang berada di area scanner akan otomatis ke deteksi dan dimasukkan dalam sebuah database komputer

e. komputer akan merekam plat nomor kendaraan bermotor tersebut dan menampilkan di halaman web tersebut.

f. Cara mendeteksi sinyal wifi client oleh scanner dan ditampilkan dalam layar komputer ditunjukkan dalam diagram di bawah ini.

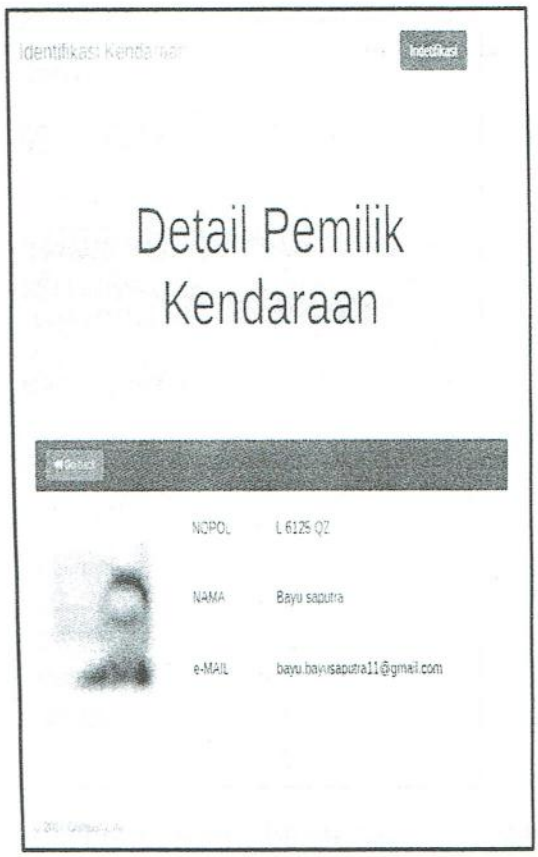


Gambar 18. Blok Diagram Sistem Identifikasi Kendaraan Bermotor



Gambar 19. Tampilan data daftar kendaraan bermotor yang teridentifikasi

Untuk melihat detail data kendaraan dapat diklik tombol detail pengguna, maka sistem akan menampilkan data secara detail yang ditunjukan dalam gambar berikut ini.



Gambar 20. Detail Data Kendaraan Bermotor

LAMPIRAN PROGRAM CLIENT

```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>

//silahkan anda isi ssid sesuai dengan nopol anda dan
kosongkan passwordnya
const char *ssid = "L 6125 QZ";
const char *password = "null";

ESP8266WebServer server(80);

void handleRoot() {
server.send(200, "text/html", "<h1>You are
connected</h1>");
}

void setup() {
	delay(1000);
	Serial.begin(115200);
	Serial.println();
	Serial.print("Configuring access point...");
	WiFi.softAP(ssid, password);

	IPAddress myIP = WiFi.softAPIP();
	Serial.print("AP IP address: ");
	Serial.println(myIP);
	server.on("/", handleRoot);
	server.begin();
	Serial.println("HTTP server started");
}
```

```
void loop() {
        server.handleClient();}
```

LAMPIRAN PROGRAM SCANNER

```
#include <Arduino.h>
#include <ESP8266WiFi.h>
#include <ESP8266WiFiMulti.h>
#include <ESP8266HTTPClient.h>
#include <LiquidCrystal.h>
#include <string.h>
#define MAX_STRING_LEN 20

ESP8266WiFiMulti WiFiMulti;
LiquidCrystal lcd(D0, D5, D6, D7, D3, D4);
String val_nopol;

String baris1, baris2, baris3, baris4;

int LoopTime_Soll = 9;
int LoopTime_Angepasst = LoopTime_Soll;
int LoopTime_Bisher = LoopTime_Soll;
unsigned long LoopTime_Start = 0;

void setup() {

  Serial.begin(115200);

  lcd.begin(20, 4);
  lcd.print("Setup LCD Success");
  Serial.println("Setup LCD Success");
  delay(5000);
  lcd.clear();

  WiFi.mode(WIFI_STA);
  WiFi.disconnect();
  lcd.print("Scanner is Works!");
```

```
  Serial.println("Scanner is Works!");
  delay(5000);
  lcd.clear();

  WiFiMulti.addAP("LAB.PAUD", "12345678");
  while((WiFiMulti.run() != WL_CONNECTED)){
    lcd.print("HTTP is Works!");
    Serial.println("HTTP is Works!");
    delay(5000);
    lcd.clear();
  }

}

void loop() {
 val_nopol = "";
 Serial.println("scan start");

 int n = WiFi.scanNetworks();
 Serial.println("scan done");
 if (n == 0)
   Serial.println("no networks found");
 else
 {
   Serial.print(n);
   Serial.println(" networks found");
   for (int i = 0; i < n; ++i)
   {
     Serial.print(i + 1);
     Serial.print(": ");
     Serial.print(WiFi.SSID(i));
     Serial.print(" (");
     Serial.print(WiFi.RSSI(i));
```

```
    Serial.print(")");
    Serial.println((WiFi.encryptionType(i)              ==
ENC_TYPE_NONE)?" ":"*");
    sendRequest(WiFi.SSID(i));
    delay(10);
   }
  }
  Serial.println("");

  LoopTime_Bisher = millis() - LoopTime_Start;
  if(LoopTime_Bisher < LoopTime_Soll){
   delay(LoopTime_Soll - LoopTime_Bisher);
  }
  LoopTime_Angepasst = millis() - LoopTime_Start;
  LoopTime_Start = millis();
}

void sendRequest(String nopol){
  int setcursor = 0;
  HTTPClient http;


http.begin("http://128.199.133.21/identifikasi/public/ap
i/communicate");
  http.addHeader("Content-Type", "application/x-www-
form-urlencoded");

  int httpCode = http.POST("nopol=" + nopol);
  String payload = http.getString();

  Serial.print("Code: ");
  Serial.println(httpCode);
  Serial.println("Payload: " + payload);
```

```
  if(payload.toInt() == 200){
    displayCursor(nopol + " is found");
    delay(1000);
  }

  Serial.println("This nopol: " + nopol);

  http.end();
}

void displayCursor(String nopol){
  baris4 = baris3;
  baris3 = baris2;
  baris2 = baris1;
  baris1 = nopol;

  lcd.setCursor(0,0);
  lcd.print(baris1);
  lcd.setCursor(0,1);
  lcd.print(baris2);
  lcd.setCursor(0,2);
  lcd.print(baris3);
  lcd.setCursor(0,3);
  lcd.print(baris4);}
```

LAMPIRAN PROGRAM WEB SERVICE

```
<?php

namespace App\Http\Controllers;
```

```
use
Illuminate\Foundation\Bus\DispatchesJ
obs;
use Illuminate\Routing\Controller as
BaseController;
use
Illuminate\Foundation\Validation\Vali
datesRequests;
use
Illuminate\Foundation\Auth\Access\Aut
horizesRequests;

class Controller extends
BaseController
{
    use AuthorizesRequests,
DispatchesJobs, ValidatesRequests;
}

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Kendaraan;
use App\History;

class CommunicateController extends
Controller {

    public function
checkNopol(Request $request) {
        $nopol = $request->nopol;
```

```
        $resultNopol =
Kendaraan::where("nopol", $nopol)-
>get();

        if ($resultNopol->count() >
0) {
            $dataKendaraan =
$resultNopol->first();
            $dataHistory = [
                "id_kendaraan" =>
$dataKendaraan->id,
                "nopol" =>
$dataKendaraan->nopol,
                "tgl_update" =>
date("Y-m-d H:i:s")
            ];
            $resultHistory =
 History::create($dataHistory);
            if ($resultHistory) {
                return 200;
            } else {
                return 502;
            }
        } else {
            return 500;
        }
    }

}


<?php

namespace App\Http\Controllers;
```

```
use Illuminate\Http\Request;
use App\Pengguna;

class PenggunaController extends
Controller {

    public function index() {
        $status = $this-
>retrovitResponse("success",
"success", 200, false);
        return response()-
>json(array("status" => $status,
"data" => Pengguna::all()), 200);
    }

    public function show(Pengguna
$pengguna) {
        $status = $this-
>retrovitResponse("success",
"success", 200, false);
        return response()-
>json(array("status" => $status,
"data" => $pengguna));
    }

    public function store(Request
$request) {
        $pengguna =
Pengguna::create($request->all());
        $status = $this-
>retrovitResponse("success",
"success", 201, false);
```

```
        return response()-
>json(array("status" => $status,
"data" => $pengguna), 201);
    }

    public function update(Request
$request, Pengguna $pengguna) {
        $pengguna->update($request-
>all());
        $status = $this-
>retrovitResponse("success",
"success", 200, false);

        return response()-
>json(array("status" => $status,
"data" => $pengguna), 200);
    }

    public function delete(Pengguna
$pengguna) {
        $pengguna->delete();
        $status = $this-
>retrovitResponse("success",
"success", 200, false);

        return response()-
>json(array("status" => $status),
204);
    }

    public function
retrovitResponse($type, $message,
$code, $error) {
        $data = array(
```

```
            "type" => $type,
            "message" => $message,
            "code" => $code,
            "error" => $error
        );
        return $data;
    }

}

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Kendaraan;

class KendaraanController extends
Controller {

    public function index() {
        $status = $this-
>retrovitResponse("success",
"success", 200, false);
        return response()-
>json(array("status" => $status,
"data" => Kendaraan::all()), 200);
    }

    public function show(Kendaraan
$kendaraan) {
        $status = $this-
>retrovitResponse("success",
"success", 200, false);
```

```
        return response()-
>json(array("status" => $status,
"data" =>
Kendaraan::where("id_pengguna",
$kendaraan->id_pengguna)->first()));
    }

    public function store(Request
$request) {
        $kendaraan =
Kendaraan::create($request->all());
        $status = $this-
>retrovitResponse("success",
"success", 201, false);

        return response()-
>json(array("status" => $status,
"data" => $kendaraan), 201);
    }

    public function update(Request
$request, Kendaraan $kendaraan) {
        $kendaraan->update($request-
>all());
        $status = $this-
>retrovitResponse("success",
"success", 200, false);

        return response()-
>json(array("status" => $status,
"data" => $kendaraan), 200);
    }
```

```
    public function delete(Kendaraan
$kendaraan) {
        $kendaraan->delete();
        $status = $this-
>retrovitResponse("success",
"success", 200, false);

        return response()-
>json(array("status" => $status),
204);
    }

    public function
retrovitResponse($type, $message,
$code, $error) {
        $data = array(
            "type" => $type,
            "message" => $message,
            "code" => $code,
            "error" => $error
        );
        return $data;
    }

}

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\History;
use Illuminate\Support\Facades\DB;
```

```
class HistoryController extends
Controller {

    public function index() {
        $data =
DB::table('histories')
            ->join('kendaraans',
'kendaraans.id', '=',
'histories.id_kendaraan')
            ->join('penggunas',
'penggunas.id', '=',
'kendaraans.id_pengguna')
            ->select('histories.*',
'kendaraans.id_pengguna',
'kendaraans.merk_kendaraan',
'penggunas.nama')
            -
>orderBy('histories.tgl_update',
'desc')
            ->get();

        $status = $this-
>retrovitResponse("success",
"success", 200, false);
        return response()-
>json(array("status" => $status,
"data" => $data), 200);
    }

    public function lastUpdate() {
        $data =
DB::table('histories')
```

```
            ->join('kendaraans',
'kendaraans.id', '=',
'histories.id_kendaraan')
            ->join('penggunas',
'penggunas.id', '=',
'kendaraans.id_pengguna')
            ->select('histories.*',
'kendaraans.id_pengguna',
'penggunas.nama')
            -
>orderBy('histories.tgl_update',
'desc')
            ->first();
        $status = $this-
>retrovitResponse("success",
"success", 200, false);
        return response()-
>json(array("status" => $status,
"data" => $data), 200);
    }

    public function show(History
$history) {
        $status = $this-
>retrovitResponse("success",
"success", 200, false);
        return response()-
>json(array("status" => $status,
"data" =>
History::where("id_kendaraan",
$history->id_kendaraan)->get()));
    }
```

```
    public function store(Request
$request) {
        $history =
History::create($request->all());
        $status = $this-
>retrovitResponse("success",
"success", 201, false);

        return response()-
>json(array("status" => $status,
"data" => $history), 201);
    }

    public function update(Request
$request, History $history) {
        $history->update($request-
>all());
        $status = $this-
>retrovitResponse("success",
"success", 200, false);

        return response()-
>json(array("status" => $status,
"data" => $history), 200);
    }

    public function delete(History
$history) {
        $history->delete();
        $status = $this-
>retrovitResponse("success",
"success", 200, false);
```

```
        return response()-
>json(array("status" => $status),
204);
    }

    public function
retrovitResponse($type, $message,
$code, $error) {
        $data = array(
            "type" => $type,
            "message" => $message,
            "code" => $code,
            "error" => $error
        );
        return $data;
    }

}
```

DAFTAR PUSTAKA

Limantara, A. D., Krisnawati,L.D., Winardi, S., Mudjanarko, S.W., Solusi Pengawasan Kebijakan Mengatasi Kemacetan Jalan dan Parkir Kota Berbasis Internet Cerdas, *Prosiding Seminar Nasional Teknologi dan Rekayasa Informasi*, Fakultas Ilmu Komputer Universitas Brawijaya, (2017), ISBN: 978-602-50782-0-0, pages 1-6.

Mudjanarko, S. W. & Winardi, S. & Limantara, A. D. Pemanfaatan Internet of Things Sebagai Solusi Manajemen Transportasi Kendaraan Sepeda Motor. *Prosiding Seminar Nasional Aplikasi Teknologi & Prasarana Wilayah (ATPW)*, 5 Agustus, (2017) .Surabaya.

Mudjanarko, S. W., Harnen, 2013, Behaviour Model of Motor Cycle User in Selecting Parking Location (Case study in Surabaya City of Indonesia), *Journal of Basic and Applied Scientific Research, J. Basic. Appl. Sci. Res.*, 3(7)842-846, (2013), ISSN 2090-4304.

Mudjanarko, S.W., Winardi, S., Prasetijo, J., Harmanto, D., Chip Number Vehicle Applications as Part of Internet of Thing (IoT), *Sustainable Transportation Infrastructures Smart Driving Research Center (SDRC)*, Penerbit UTHM, ISBN 978-967-2110-70-5, page 84

Saputra, B., Winardi, S., Mudjanarko, S. W., & Inayati, I., Identitas Digital Kendaraan Bermotor Dengan Chip Wemos Untuk Solusi Program Plat Nomor Genap Ganjil, *Prosiding Seminar Nasional Ilmu Terapan (SNITER) Universitas Widya Kartika,* (2017), Surabaya, ISSN: 2597-7067 B03-1 – B03-8

Winardi, S., Kamisutara, M., Purworusmiardi, T., Sukoco, A., Mudjanarko, S. W., Internet of Things (IoT) as Green City Economic Development Smart Transportation Sistem, MATEC Web of Conferences, (2017), EDP Sciences , Volume 138, pages 07015

Universitas Narotama Surabaya