



REPUBLIK INDONESIA
KEMENTERIAN HUKUM DAN HAK ASASI MANUSIA

SURAT PENCATATAN CIPTAAN

Dalam rangka perlindungan 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 : EC00201821907, 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 : **Identifikasi Kendaraan Bermotor Dengan Chip ESP8266**

Tanggal dan tempat diumumkan untuk pertama kali di wilayah Indonesia atau di luar wilayah Indonesia : 20 Juni 2018, di Surabaya

Jangka waktu perlindungan : 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 : 000112536

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

IDENTIFIKASI KENDARAAN BERMOTOR dengan CHIP ESP8266

Slamet Winardi | Dadang Supriyatno | Benediktus Anindito
Sri Wiwoho Mujanarko | Natalia Damastuti | Bayu Saputra

MPR Kamera

WAL RIGOR

Relay / Digital I/O

Embedded

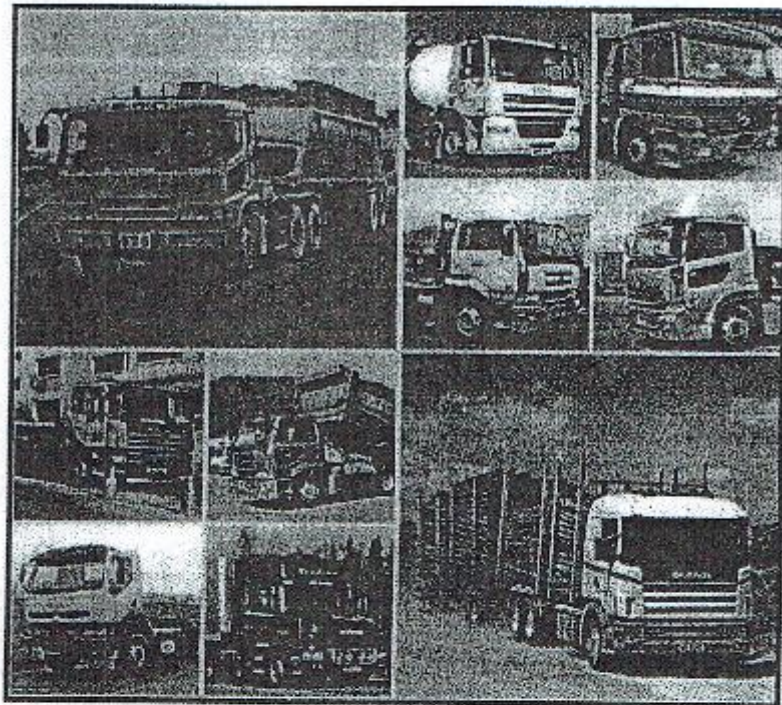
Lat: 28.44
Log: 117.20



Narotama University Press

Slamet Winardi | Dadang Supriyatno | Benediktus Anindito
Sri Wiwoho Mujanarko | Natalia Damastuti | Bayu Saputra

IDENTIFIKASI KENDARAAN BERMOTOR DENGAN CHIP ESP8266



Diterbitkan oleh
Narotama University Press



NAROTAMA University Press

Identifikasi Kendaraan Bermotor dengan Chip ESP8266
/disusun oleh Slamet Winardi, dkk. 92 hal; [viii]; editor, Seger

Copyright © 2018 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: B5 (17,6 x 25 cm)

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

Editor : Seger
Layout/Setting : Gatut Purwantoro
Design Cover : Gatut Purwantoro

ISBN 978-602-6557-33-9

© 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

Atas berkat rahmat Tuhan Yang Maha Esa yang selalu melimpahi kesehatan kepada penulis, sehingga penulis mampu menyelesaikan buku teks ini sebagai hasil dari luaran penelitian yang penulis beserta tim mengajukan penelitian pada tahun 2017 hingga 2019, namun pada tahun 2018 ini penulis dapat merampungkannya.

Buku tentang identifikasi kendaraan bermotor secara digital dengan sinyal wifi ini kami persembahkan kepada almamater kami yang telah memberikan fasilitas untuk melakukan penelitian dan merestui kami semua, semoga buku ini dapat bermanfaat untuk menambah ilmu bagi sivitas akademika dan masyarakat umum.

Terimakasih kami ucapkan kepada Ristek Dikti yang telah memberikan dana penelitian kami hingga dapat menyelesaikan buku teks ini. Terimakasih kepada Rektor Universitas Narotama atas berkenannya memberikan tempat untuk melakukan penelitian kami ini. Terimakasih juga kepada LPPM yang telah mempermudah pelaksanaan penelitian ini sehingga semua kegiatan dapat dilakukan dengan baik. Taka da gading yang tak retak, buku ini masih banyak kekurangan dan memerlukan banyak penyempurnaan, kami tim penulis menerima saran dan kritik membangun untuk menyempurnakan buku kami.

Surabaya, Mei 2018

Tim Penyusun

DAFTAR ISI

KATA PENGANTAR	iii
DAFTAR ISI	v
1.1. Pengertian Kendaraan Bermotor	1
1.1.1. Truk.....	2
1.1.2. Bus	3
1.1.3. Mobil Pribadi.....	4
1.1.4. Sepeda Motor.....	5
1.2. Identitas Kendaraan Bermotor	5
1.2.1. Surat Tanda Nomor Kendaraan	6
1.2.1.1. Identitas Pemilik dan Nomor Registrasi.....	6
1.2.1.2. Data Detail Kendaraan Bermotor dan Masa Berlaku	7
1.2.2. Tanda Nomor Kendaraan Bermotor.....	7
1.2.3. Buku Pemilik Kendaraan Bermotor	8
1.3. Nomor Rangka dan Nomor Mesin	9
2.1. Apa itu Wifi?	11
2.1.1. WiFi Half Duplex	12
2.1.2. Channel Bandwidth.....	12
2.2. Konsep Bekerjanya Wifi	13
2.2.3. Hotspot Wifi.....	15
2.3. Wifi Standar IEEE	16
2.4. Akses Protokol Wifi.....	17
2.5. Quality Of Service (QOS) Wifi	18
2.5.1. WiFi Multimedia Extensions (WME).....	18
2.5.2. WiFi Scheduled Multimedia (WSM).....	19
2.6. Keamanan Wifi.....	20
2.7. Layanan Jaringan Wifi	20

2.8.	Radio Mudalasi Wifi.....	23
3.1	Fitur.....	25
3.2.	Parameter-parameter.....	26
3.3.	Diskripsi Pin.....	27
3.4.	Flashing ESP8266.....	29
3.4.1.	Material yang dibutuhkan.....	30
3.4.2.	Instalasi Software Yang Dibutuhkan.....	30
3.4.3.	Bagaimana Menghubungkan ESP8266 dengan Arduino.....	31
3.4.4.	Rangkaian Pembagi Tegangan.....	32
3.4.5.	Hubungkan Komputer dengan Arduino.....	32
3.4.6.	Menguji Koneksi ESP8266 dengan Arduino.....	33
3.4.7.	Upload Skecth Kosong Arduino.....	34
3.4.8.	Menghapus Flash Memori.....	34
3.4.9.	Flash Firmware ESP8266.....	35
3.4.10.	Menghubungkan ke wifi ESP8266.....	35
3.5.	Command AT Dasar.....	36
3.5.1.	Perintah AT.....	36
3.5.2.	Restart Modul.....	37
3.5.3.	Melihat Info Versi AT Command.....	37
3.5.4.	Enter Deep-sleep Mode.....	37
3.5.5.	AT Command Echo.....	38
3.5.6.	Reset Factory.....	38
3.5.7.	Konfigurasi UART.....	39
3.5.8.	Konfigurasi UART yang sedang berjalan.....	39
3.5.9.	Konfigurasi UART Default.....	40
3.5.10.	Mode Sleep.....	41
3.5.11.	Set Nilai Maksimum Daya untuk TX RF.....	42
3.5.12.	Set Daya RF TX sama dengan VDD33.....	42
4.1.	Langkah-langkah Instalasi Wemos (ESP8266).....	45
4.2.	Fungsi Pin pada Wemos D1 & R2 dan Mini.....	48

5.1.	WiFiScan	51
5.2.	Access Point Dengan Wemos	53
5.3.	Wifi Web Server dengan Wemos.....	56
5.4.	Wifi Client	59
6.1.	Client Identifikasi Kendaraan Bermotor	64
6.2.	Scanner Identifikasi Kendaraan Bermotor	66
6.3.	Web Service Identifikasi Kendaraan Bermotor	70
	Direktorat Jendral Hubungan Darat Kementrian Perhubungan Republik Indonesia, "Lalu Lintas dan Angkutan Jalan". http://hubdat.dephub.go.id/uu/288-uu-nomor-22-tahun-2009-tentang-lalu-lintas-dan-angkutan-jalan , diakses Mei 2018.....	81
	Direktorat Jendral Hubungan Darat Kementrian Perhubungan Republik Indonesia, "Pemeriksaan Kendaraan Bermotor di Jalan", http://hubdat.dephub.go.id/peraturan-pemerintah/81-pp-no-42-tahun-1993-ttg-pemeriksaan-kendaraan-bermotor-di-jalan , diakses Mei 2018	81
	https://www.temukanpengertian.com/2013/01/pengertian-client.html , "Pengertian Client "	82
	http://sewuweb.com/cara-kerja-dan-fungsi-access-point/ , "Cara Kerja dan Fungsi Access Point"	82
	https://idcloudhost.com/pengertian-web-server-dan-fungsinya/ , "Pengertian Web Server dan Fungsinya"	82
	DAFTAR PUSTAKA	83

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

BAB I

IDENTITAS KENDARAAN BERMOTOR

1.1. Pengertian Kendaraan Bermotor

Kendaraan bermotor adalah kendaraan yang digerakkan oleh peralatan teknik untuk pergerakannya, dan digunakan untuk transportasi darat. Umumnya kendaraan bermotor menggunakan mesin pembakaran dalam, namun mesin listrik dan mesin lainnya juga dapat digunakan. Kendaraan bermotor memiliki roda, dan biasanya berjalan diatas jalanan. Klasifikasi kendaraan bermotor ini bervariasi tergantung masing-masing negara. ISO 3833:1977 adalah standar untuk tipe dan definisi kendaraan darat.

Berdasarkan UU No. 14 tahun 1992 yang dimaksud dengan peralatan teknik dapat berupa motor atau peralatan lainnya yang berfungsi untuk mengubah suatu sumber daya energi tertentu menjadi tenaga gerak kendaraan bermotor yang bersangkutan. Pengertian kata berada dalam ketentuan ini adalah terpasang pada tempat sesuai dengan fungsinya. Termasuk dalam pengertian kendaraan bermotor

adalah kereta gandengan atau kereta tempelan yang dirangkaikan dengan kendaraan bermotor sebagai penariknya. (<http://wahyu-smekers.blogspot.co.id/>, 2017)

1. TRUK
2. BUS
3. MOBIL PRIBADI
4. SEPEDA MOTOR

Sampai tahun 2010, ada lebih dari 1 miliar kendaraan bermotor di seluruh dunia, tidak termasuk kendaraan *off-road* dan kendaraan berat. Kepemilikan kendaraan per kapita global adalah 148 kendaraan beroperasi tiap 1000 orang. Amerika Serikat adalah negara yang memiliki jumlah kendaraan bermotor terbanyak di dunia, dengan 239,8 juta kendaraan tahun 2010.

Kepemilikan kendaraan per kapita di Amerika Serikat juga tertinggi di dunia, yaitu 769 kendaraan per 1000 penduduk. Republik Rakyat Tiongkok mempunyai jumlah kendaraan terbanyak kedua di dunia, dengan jumlah 78 juta unit dan sejak 2009 juga menjadi pasar kendaraan terbesar di dunia. Pada tahun 2011, 80 juta mobil dan kendaraan komersial diproduksi di seluruh dunia, 18,4 juta unit diantaranya diproduksi di Cina. ([https://id.wikipedia.org/wiki, "Kendaraan Bermotor"](https://id.wikipedia.org/wiki/Kendaraan_Bermotor))

1.1.1. Truk

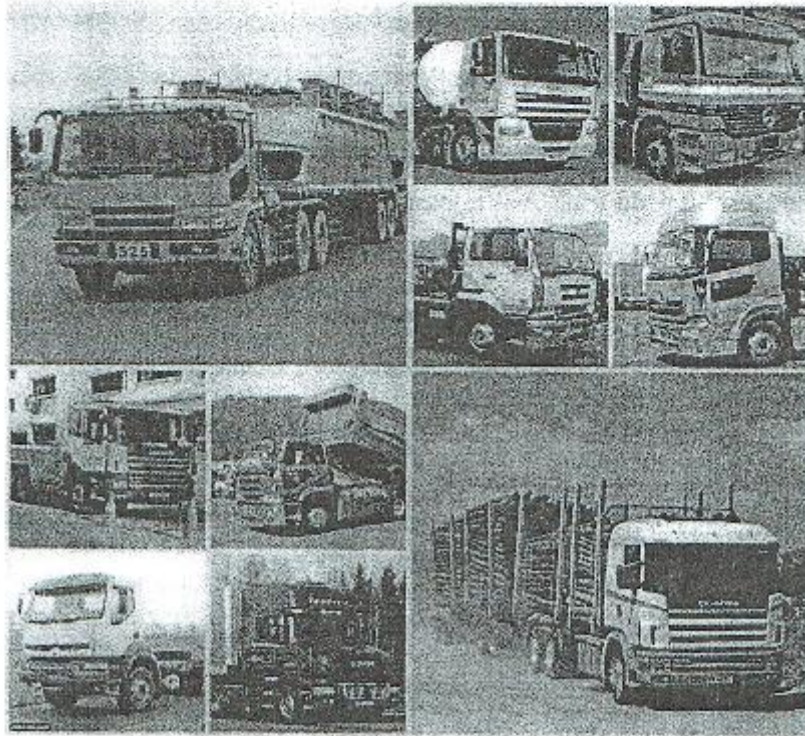
Truk merupakan sebuah kendaraan bermotor yang digunakan khusus untuk mengangkut barang. Jenis kendaraan bermotor ini biasa disebut juga sebagai mobil barang. Mobil Truk ini biasa digunakan dalam bidang konstruksi dan perdagangan. Contoh truk kecil yang

sering kita temui di jalan-jalan adalah mobil pick-up. Semakin besar ukuran truk semakin memiliki banyak sumbu roda.

Sumbu roda ini berfungsi sebagai tempat roda, sehingga makin banyak sumbu roda truk, makin banyak ban yang digunakan dan beban yang diterimapun makin besar. Sumbu mobil truk paling banyak adalah 6 sumbu. Tabel berikut adalah daftar hubungan daya angkut dengan konfigurasi sumbu truk :

Tabel 1.1. Konfigurasi dan Jenis Truk

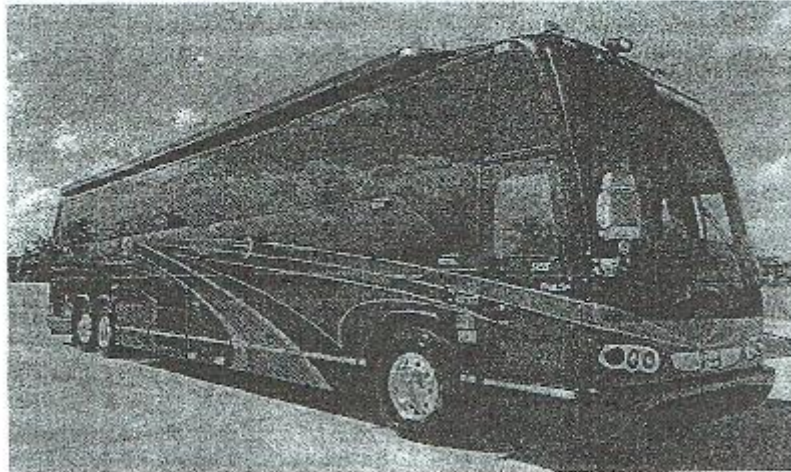
Konfigurasi Sumbu	Jumlah Sumbu	Jenis	JBI Kelas II	JBI Kelas III	Jumlah Ban
1 - 1	2	Truk Engkel Tunggal	12 ton	12 ton	4
1 - 2	2	Truk Engkel Ganda	16 ton	14 ton	6
1.1 - 2	3	Truk Trinton	18 ton	16 ton	8
1 - 2.2	3	Truk Tronton	22 ton	20 ton	10
1.1 - 2.2	4	Truk Trinton	30 ton	26 ton	12
1 - 2 - 2.2	4	Truk Trailer Engkel	34 ton	28 ton	14
1 - 2 - 2.2.2 1 - 2.2 - 2.2	5	Truk Trailer Engkel Truk Trailer Tronton	40 ton	32 ton	18
1 - 2.2 - 2.2.2	6	Truk Trailer Tronton	43 ton	40 ton	22



Gambar 1.1. Jenis Kendaraan Truk

1.1.2. Bus

Bus merupakan kendaraan besar beroda, digunakan untuk membawa penumpang dalam jumlah yang banyak. Ukuran dan berat kendaraan bus ini sangat besar dibandingkan dengan mobil penumpang umum. Kendaraan bus di Indonesia bisa dibagi ke dalam beragam kelompok, yakni berdasar pada ukuran, type, kelas dan jarak. Berdasar pada ukuran, kendaraan bus dibedakan jadi tiga type, yakni bus ukuran besar, bus ukuran tengah serta bus ukuran kecil. Sedang type bus berdasar pada kelas dibedakan jadi bus kelas ekonomi, bus kelas usaha, bus kelas executive serta bus kelas *super executive*.



Gambar 1.2. Kendaraan Bus

1.1.3. Mobil Pribadi



Gambar 1.3. Deretan Jenis Mobil Pribadi

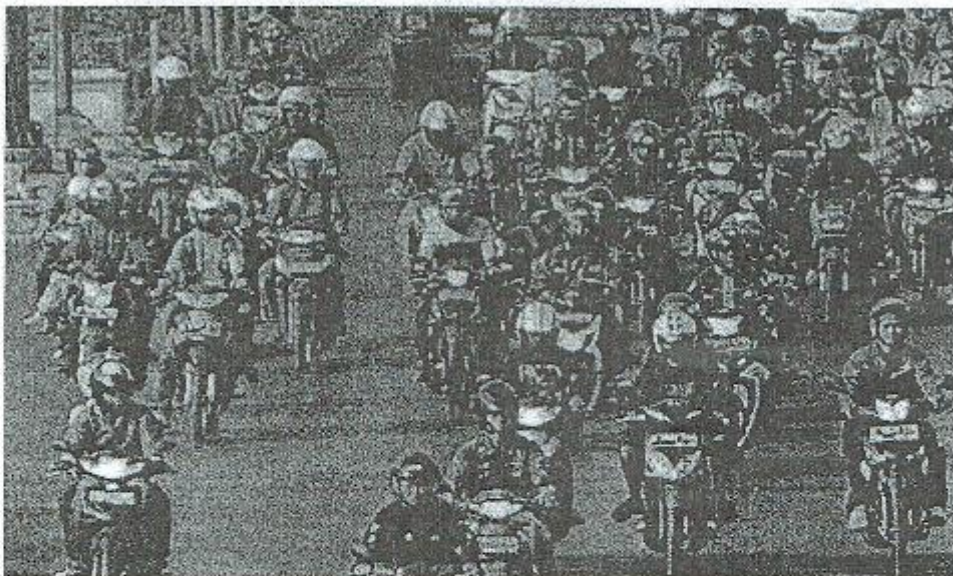
Dewasa ini cukup banyak jenis dan ragam kendaraan bermotor berjenis mobil, mulai dari ukuran yang kecil dsampai dengan yang ukuran besar. Model mobil pribadi yang diproduksi semakin beragam sesuai dengan tuntutan kebutuhan konsumen, fungsi dan kegunaan bahkan gaya hidup, dari tipe sedan sampai dengan minibus.

Data yang dirilis oleh Korps Lalu Lintas Kepolisian Negara Republik Indonesia mencatat, jumlah mobil pribadi yang masih beroperasi di seluruh Indonesia pada 2013 mencapai 11 484 514 unit (pada website BPS).

1.1.4. Sepeda Motor

Sepeda motor merupakan kendaraan beroda dua yang digerakkan oleh sebuah mesin 2 Tak atau 4 Tak. Pengguna sepeda motor di Indonesia sangat banyak karena harganya yang relatif murah, terjangkau untuk beberapa kalangan dan penggunaan bahan bakarnya irit serta biaya operasionalnya juga sangat rendah.

Menurut BPS (Badan Pusat Statistik) pada tahun 2013 populasi sepeda motor di Indonesia mencapai 84 732 652 unit naik 11% dari tahun sebelumnya.



Gambar 1.4. Kendaraan Bermotor Roda Dua

1.2. Identitas Kendaraan Bermotor

Menurut Undang-undang Nomor 22 Tahun 2009 tentang Lalu Lintas dan Angkutan Jalan, secara tegas menyatakan bahwa setiap kendaraan bermotor wajib memasang nomor polisi di kendaraannya masing-masing. Polisi pun dengan tegas menyatakan bahwa syarat kendaraan dapat melaju di jalan raya adalah dengan memasang nomor polisi atau pelat nomor sebagai identitas kendaraan tersebut.



Gambar 1.5. Plat Nomor Kendaraan Bermotor

Ini dasar hukum yang mewajibkan setiap kendaraan memiliki identitas berupa pelat nomor (TNKB) maupun Surat Tanda Nomor Kendaraan (STNK). Untuk Proses Jual beli kendaraan bermotor juga dilengkapi berupa Buku Pemilik Kendaraan Bermotor (BPKB).

1.2.1. Surat Tanda Nomor Kendaraan

Menurut Undang-undang Nomor 22 tahun 2009 pasal 64 tentang registrasi dan identifikasi kendaraan bermotor bahwa pada ayat 1 berbunyi "Setiap Kendaraan Bermotor wajib diregistrasikan", registrasi kendaraan bermotor yang dimaksud bertujuan tertib administrasi, untuk pengendalian dan pengawasan kendaraan bermotor

yang dioperasikan di Indonesia, mempermudah penyidikan pelanggaran dan/atau kejahatan, untuk perencanaan, operasional manajemen, dan rekayasa lalu lintas dan angkutan jalan.



Gambar 1.6. Surat Tanda Nomor Kendaraan Bermotor

1.2.1.1. Identitas Pemilik dan Nomor Registrasi

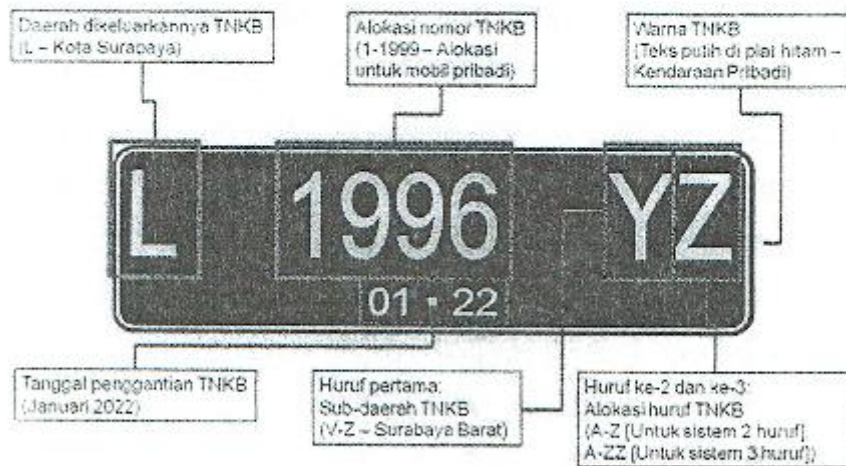
Sesuai dengan Undang-undang 22 tahun 2009 pasal 68 ayat 2, di dalam STNK terdapat Identitas Pemilik dan nomor registrasi Kendaraan bermotor. Nomor registrasi kendaraan bermotor mencakup kode wilayah yang telah ditentukan oleh Kepolisian Negara Republik Indonesia dan nomor registrasi yang ditentukan oleh pihak berwenang atau sesuai dengan permintaan sendiri dengan membayar sejumlah tarif yang berlaku pada PP No. 60 Tahun 2016. Sedangkan Identitas Pemilik yang tercantum dalam STNK ini terdiri dari Nama Pemilik dan Alamat Pemilik Kendaraan Bermotor.

1.2.1.2. Data Detail Kendaraan Bermotor dan Masa Berlaku

Pasal 68 ayat 2,3, dan 4 telah mencakup data detail kendaraan bermotor yaitu : Merk Kendaraan Bermotor, type, Jenis, Model, Tahun Pembuatan, Isi silinder, Nomor rangka/NIK/VIN, Nomor mesin, Warna kendaraan bermotor, Bahan bakar, Warna TNKB, Tahun Registrasi, Nomor BPKB, dan kode lokasi. Masa berlaku Surat Tanda Kendaraan Bermotor sesuai pasal 70 adalah selama 5 tahun dan harus dimintakan pengesahan setiap tahun di kantor samsat sebagai bukti bahwa pajak kendaraan bermotor telah terbayarkan ke kas negara.

1.2.2. Tanda Nomor Kendaraan Bermotor

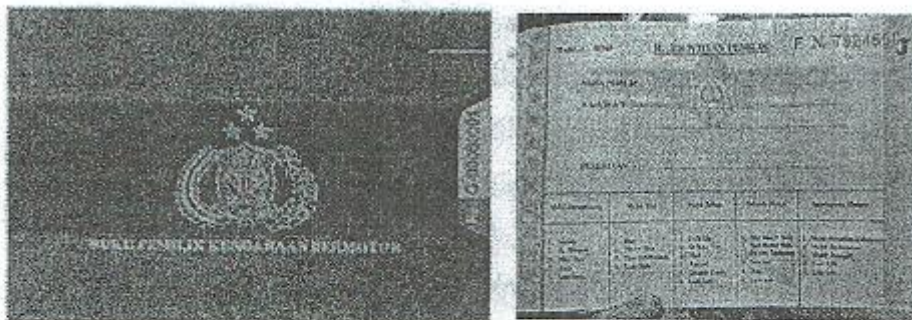
Setiap kendaraan bermotor yang beroperasi di jalan raya harus dilengkapi Tanda Nomor Kendaraan Bermotor (TNKB) sesuai dengan Undang-undang 22 Tahun 2009 pasal 68. TNKB yang dimaksud harus memuat kode wilayah, nomor registrasi dan masa berlaku kendaraan bermotor tersebut. TNKB ini harus memenuhi syarat tertentu yang ada dalam undang-undang tersebut yang mencakup bentuk, ukuran, bahan, warna, dan cara pemasangannya. Pada ayat 5 TNKB ini dapat diterbitkan khusus dan/atau rahasia, dan bila pemilik menghendaki nomor khusus dapat memesan nomor khusus tersebut dengan membayar sejumlah tarif sesuai dengan PP No. 60 tahun 2016. TNKB ini berlaku selama 5 tahun dan akan diganti yang baru bila telah habis masa berlakunya.



Gambar 1.7. Tanda Nomor Kendaraan Bermotor

1.2.3. Buku Pemilik Kendaraan Bermotor

Sebagai bukti kepemilikan kendaraan bermotor, seorang pemilik harus memiliki dan memegang tanda bukti yang berupa BPKB. Sesuai dengan pasal 70 BPKB ini disimpan oleh pemilik selama tidak dipindah tangankan ke pemilik yang lain.

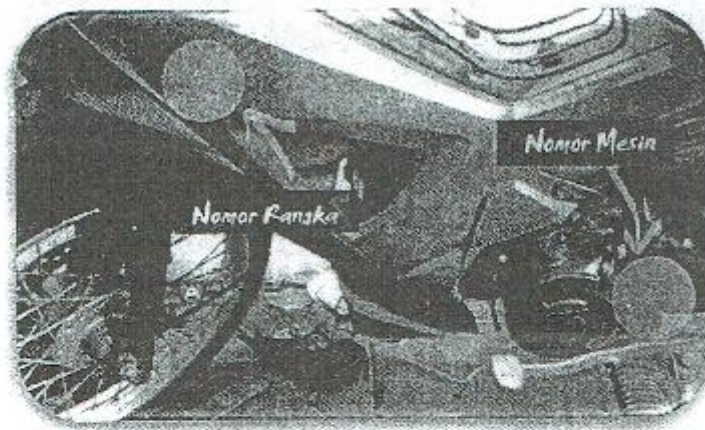


Gambar 1.8. Buku Pemilik Kendaraan Bermotor

Menurut Peraturan Kapolri No. 5 tahun 2015 pelayanan pengesahan STNK tahunan dan pembayaran pajak kendaraan bermotor dan SWDKLLJ (Jasa Raharja) tanpa harus membawa BPKB. Setelah 5 tahun barulah untuk mengganti STNK dan TNKB diperlukan BPKB untuk dibawa sebagai bukti kepemilikan kendaraan bermotor.

1.3. Nomor Rangka dan Nomor Mesin

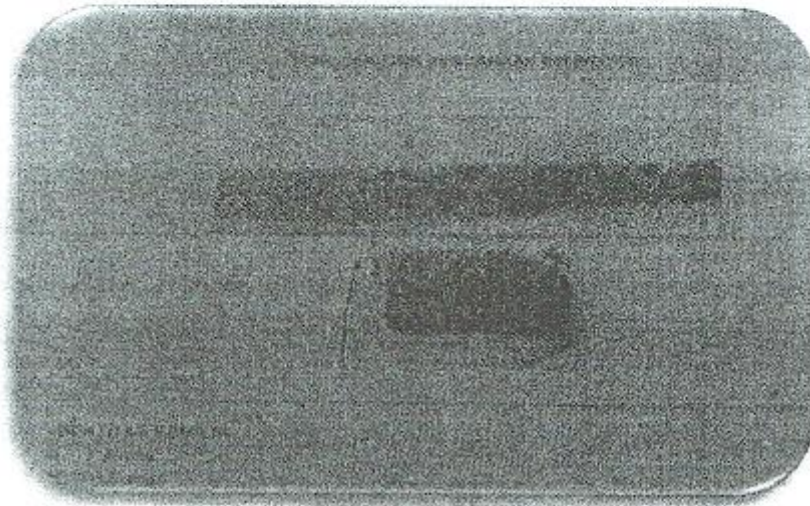
Untuk menjamin dan melindungi keaslian, keabsahan, kebenaran dan keamanan tanda bukti pendaftaran kendaraan bermotor adalah dengan melakukan pemeriksaan fisik terhadap kendaraan bermotor yang didaftarkan untuk dicocokkan dengan tanda bukti pendaftaran yang sedang diproses berupa STNK dan BPKB. Cek fisik kendaraan bermotor merupakan langkah pembuktian antara fisik kendaraan bermotor dengan surat-suratnya.



Gambar 1.9. Nomor Rangka dan Nomor Mesin pada Sepeda Motor

Cek fisik secara rutin dilakukan oleh pihak Sistem Administrasi Manunggal di bawah Satu Atap (SAMSAT) selama 5 tahun sekali setelah masa pajak kendaraan bermotor habis. Cek fisik

yang dilakukan untuk melihat nomor rangka dan nomor mesin yang ada pada setiap kendaraan bermotor.



Gambar 1.10. Bukti Gesek Nomor Rangka dan Nomor Mesin

BAB II

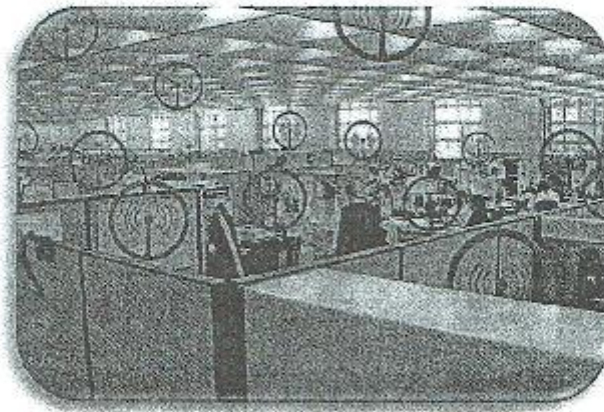
WIFI (*WIRELESS FIDELITY*)

2.1. Apa itu Wifi?

WiFi adalah singkatan dari *Wireless Fidelity*. WiFi ini didasarkan pada standar IEEE 802.11 dan terutama teknologi jaringan area lokal (LAN) yang dirancang untuk menyediakan jangkauan broadband dalam bangunan. Sistem WiFi saat ini mendukung kecepatan data *physical-layer* sebesar 54 Mbps dan biasanya mengcover area dalam ruangan sejauh ± 50 meter.

WiFi telah menjadi standar untuk konektivitas broadband di rumah-rumah, kantor, dan lokasi hotspot publik. Sistem dapat memberikan jangkauan sekitar ± 50 meter dari access point.

WiFi menawarkan kecepatan data yang jauh lebih tinggi daripada sistem 3G, beroperasi di atas bandwidth 20 MHz yang lebih besar, namun sistem WiFi tidak dirancang untuk mendukung mobilitas dengan kecepatan tinggi.



Gambar 2.1. Sinyal-sinyal Wifi

Salah satu keuntungan yang signifikan dari WiFi melalui WiMAX dan 3G adalah ketersediaan terminal perangkat yang luas. Sebagian besar laptop yang dimiliki mempunyai interface WiFi built-in. Interface Wifi sekarang juga sedang dibangun ke dalam berbagai perangkat, diantaranya PDA, telepon nirkabel, telepon seluler, kamera, dan media player.

2.1.1. WiFi Half Duplex

Semua jaringan WiFi menggunakan sistem berbasis TDD, dimana access point dan *mobile station* semua bersaing untuk menggunakan saluran yang sama. Karena operasi media bersama, maka semua jaringan WiFi bersifat half duplex. Ada vendor peralatan yang memasarkan WiFi dengan konfigurasi mesh, tapi implementasi itu menggabungkan teknologi yang tidak didefinisikan dalam standar.

2.1.2. Channel Bandwidth

Di banyak belahan dunia, frekuensi yang pakai oleh Wi-Fi dengan pengguna tidak memerlukan untuk mendapatkan izin dari pengatur lokal. 802.11a menggunakan frekuensi yang lebih tinggi (5 MHz), oleh sebab itu daya jangkauannya lebih sempit, yang lainnya menggunakan versi Wi-Fi yang paling luas dalam pasaran sekarang ini (berdasarkan dalam IEEE 802.11b/g) beroperasi pada 2.400 GHz sampai 2.483,50 GHz. Dengan begitu mengijinkan operasi dalam 11 channel (masing-masing 5 MHz).

Tabel 2.1. Spesifikasi Wifi

Spesifikasi	Kecepatan	Frekuensi Band	Cocok dengan
802.11b	11 Mb/s	~2.4 GHz	b
802.11a	54 Mb/s	~5 GHz	a
802.11g	54 Mb/s	~2.4 GHz	b, g
802.11n	100 Mb/s	~2.4 GHz	b, g, n

2.2. Konsep Bekerjanya Wifi

Seperti handphone, untuk mengirimkan informasi dari satu perangkat ke perangkat yang lain, jaringan Wi-Fi menggunakan gelombang radio. Komputer harus mempunyai perangkat nirkabel yang akan menerjemahkan data yang dikirim menjadi sinyal radio.

Sinyal yang akan dikirim, melalui antena, untuk decoder dikenal sebagai router. Setelah diterjemahkan, data akan dikirim ke Internet melalui koneksi kabel Ethernet. Sebagai jaringan nirkabel akan bekerja sebagai lalu lintas dua arah, data yang diterima dari

Internet juga akan melewati router yang akan dikodekan menjadi sinyal radio yang akan diterima oleh Wireless Adapter pada komputer.

2.2.1. Sinyal Radio

Sinyal Radio adalah kuncinya yang memungkinkan jaringan WiFi. Sinyal radio ini ditransmisikan dari antena WiFi diterima oleh receiver WiFi, seperti komputer dan handphone yang dilengkapi dengan kartu WiFi. Kapanpun komputer menerima salah satu dari sinyal dalam jangkauan jaringan WiFi, yang biasanya 50 - 100 meter dari antena, WiFi card membaca sinyal dan menciptakan koneksi internet antara pengguna jaringan tanpa menggunakan kabel.



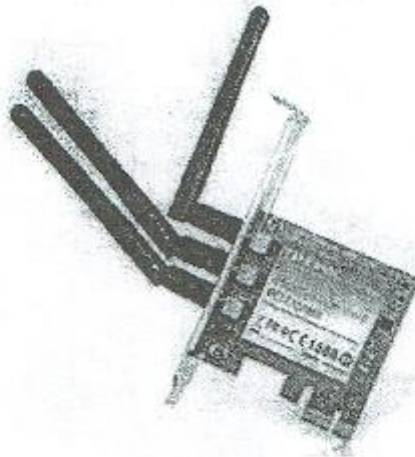
Gambar 2.2. Sinyal Radio dan Access Point

Jalur akses, terdiri dari antena dan router, merupakan sumber utama yang mentransmisikan dan menerima gelombang radio. Antena bekerja menguatkan sinyal dan mentransmisikan gelombang radio yang lebih panjang dengan Radius 50-100 meter, yang digunakan di tempat umum sedangkan router yang lebih lemah namun efektif lebih cocok untuk rumah dengan transmisi radio 30-50 meter.

2.2.2. Wifi Card

Bisa dibayangkan sebuah WiFi card sebagai kabel yang tak kelihatan oleh mata yang menghubungkan komputer ke antena untuk koneksi langsung ke internet. WiFi card bisa bersifat eksternal atau internal. Jika WiFi card tidak terpasang di komputer, maka dapat membeli perangkat antena USB untuk dipasang di komputer sebagai wifi eksternal.

Port USB, atau memiliki kartu ekspansi yang dilengkapi antena terpasang langsung ke komputer. (Seperti yang ditunjukkan pada gambar di bawah). Untuk laptop, sekarang sudah dilengkapi dengan wifi card tinggal menghubungkan dengan access point yang tertangkap sinyalnya.



Gambar 2.3. Wifi Card

2.2.3. Hotspot Wifi

Hotspot WiFi dibuat dengan memasang titik akses ke koneksi internet. Titik akses mentransmisikan sinyal nirkabel dalam jarak

dekat. Biasanya menjangkau sekitar 50 meter. Saat perangkat yang mempunyai koneksi WiFi seperti PC atau *handphone* menemukan hotspot, maka perangkat dapat terhubung ke jaringan tanpa kabel. Sebagian besar hotspot berada di tempat yang mudah diakses oleh masyarakat umum seperti bandara, kedai kopi, hotel, toko buku, dan lingkungan kampus.

802.11b adalah spesifikasi yang paling umum untuk hotspot di seluruh dunia. Standar 802.11g terbalik, kompatibel dengan 802.11b tapi 802.11a menggunakan rentang frekuensi yang berbeda dan memerlukan perangkat keras yang terpisah seperti adaptor a, a / g, atau a / b / g. Jaringan WiFi publik terbesar adalah disediakan oleh penyedia layanan internet (ISP); Mereka membebankan biaya kepada pengguna yang ingin mengakses internet.



Gambar 2.4. Hotspot Wifi

Hotspot semakin berkembang di seluruh dunia. Di Indonesia semakin menjamur lokasi hotspot wifi, di pusat perbelanjaan, di bandara, terminal, stasiun, kantor, hotel, kampus dan lain-lain. Di Kampus Universitas Narotama Surabaya juga dibangun tempat-tempat hotspot yang dapat diakses oleh semua sivitas akademika. Setiap komputer notebook dengan integrasi wireless, adaptor wireless yang terpasang pada Motherboard oleh produsen, atau adaptor wireless seperti kartu PCMCIA dapat mengakses jaringan wireless.

Beberapa Hotspot memerlukan password WEP untuk terhubung, yang dianggap sebagai personal dan keamanan. Sedangkan untuk koneksi terbuka, siapapun dengan WiFi card bisa memiliki akses ke hotspot tersebut. Jadi untuk memiliki akses internet di bawah WEP, pengguna harus memasukkan kode password WEP.

2.3. Wifi Standar IEEE

Standar 802.11 didefinisikan melalui beberapa spesifikasi WLAN. Ini mendefinisikan sebuah interface *over-the-air* antara wireless client dan base station atau antara dua client secara wireless.

Ada beberapa spesifikasi dalam keluarga 802.11:

- **802.11:** Hal ini berkaitan dengan *wireless LAN* dan menyediakan transmisi 1- atau 2 Mbps di *bandwidth* 2,4 GHz menggunakan *frekuensi-hopping spread spectrum (FHSS)* atau *Direct-sequence spread spectrum (DSSS)*.
- **802.11a:** Ini adalah ekstensi untuk 802.11 yang berkaitan dengan wireless LAN dan berjalan dengan kecepatan 54 Mbps di band 5 GHz. 802.11a menggunakan *Orthogonal Frequency*

Division Multiplexing (OFDM) encoding scheme dibandingkan dengan FHSS atau DSSS.

- **802.11b:** WiFi tingkat tinggi 802.11 adalah perpanjangan 802.11 yang berkaitan dengan wireless LAN dan menghasilkan koneksi dengan kecepatan transmisi 11 Mbps (dengan mundur ke 5,5, 2, dan 1 Mbps tergantung kekuatan sinyal) di bandwidth 2,4 GHz. Spesifikasi 802.11b hanya menggunakan DSSS. Perhatikan bahwa 802.11b sebenarnya sebuah amandemen terhadap standar 802.11 asli yang ditambahkan pada tahun 1999 untuk mengizinkan wireless. Fungsionalitas untuk analog dengan koneksi Ethernet.
- **802.11g:** Ini berkaitan dengan wireless LAN dan menyediakan bandwidth 2,4 GHz dengan kecepatan 20+ Mbps atau lebih.

Tabel 2.2. Perbandingan Teknis Sistem Wifi

Fitur	WiFi (802.11b)	WiFi (802.11a/g)
Aplikasi Utama	Wireless LAN	Wireless LAN
Band Frekuensi	2.4 GHz ISM	2.4 GHz ISM (g) 5 GHz U-NII (a)
Bandwidth kanal	25 MHz	20 MHz
Half/Full Duplex	Half	Half
Teknologi Radio	Direct Sequence Spread Spectrum	OFDM (64 kanal)
Efisiensi Bandwidth	≤ 0.44 bps/Hz	≤ 2.7 bps/Hz
Modulasi	QPSK	BPSK, QPSK, 16-, 64-QAM
FEC	Tidak ada	Convolutional Code
Enskripsi	Optional- RC4m (AES in 802.11i)	Optional- RC4(AES in 802.11i)
Mobility	Dalam Pengembangan	Dalam Pengembangan
Mesh	Vendor Proprietary	Vendor Proprietary
Access Protocol	CSMA/CA	CSMA/CA

2.4. Akses Protokol Wifi

Wireless LAN IEEE 802.11 menggunakan media akses kontrol protokol yang disebut *Carrier Sense Multiple Access with Collision Avoidance* (CSMA/CA). Sedangkan namanya serupa dengan *Ethernet's Carrier Sense Multiple Access dengan Collision Detection* (CSMA / CD), tetapi konsep operasinya sama sekali berbeda.

Sistem WiFi yang digunakan untuk konfigurasi media adalah *half* dupleks, dimana semua stasiun mentransmisikan dan menerima saluran radio yang sama. Masalah yang mendasar dari sebuah sistem radio adalah stasiun tidak dapat menerima data saat sedang mengirim, dan karenanya tidak mungkin mendeteksi tabrakan data. Karena itu, pengembang spesifikasi 802.11 menciptakan mekanisme untuk menghindari tabrakan data yang disebut *Distributed Control Function* (DCF).

Menurut DCF, stasiun WiFi hanya akan mentransmisikan data bila salurannya jelas. Semua Transmisi data diketahui, jika sebuah terminal tidak menerima data yang dikirim, kemungkinan terjadi tabrakan data dan akan retries setelah interval waktu tunggu yang acak. Insiden tabrakan data akan meningkat seiring dengan meningkatnya lalu lintas data atau dalam situasi di mana terminal bergerak tidak terjadi komunikasi.

2.5. Quality Of Service (QoS) Wifi

Rencana untuk menggabungkan kemampuan *Quality Of Service* (QoS) dalam teknologi WiFi dengan standar IEEE 802.11e.

Standar 802.11e akan mencakup dua mode operasi, yang bisa digunakan untuk memperbaiki layanan suara:

- *WiFi Multimedia Extensions* (WME): Wajib
- *WiFi Scheduled Multimedia* (WSM): Opsional

2.5.1. WiFi Multimedia Extensions (WME)

WiFi Multimedia Extensions menggunakan protokol yang disebut Enhanced Multimedia Distributed Control Access (EDCA), yang merupakan perluasan dari versi terdistribusi dari Fungsi Kontrol Terdistribusi (Distributed Control Function/DCF) yang didefinisikan dalam MAC 802.11 asli.

Bagian yang disempurnakan bahwa EDCA akan menentukan delapan tingkat prioritas akses ke saluran wireless yang dibagikan, seperti DCF asli, akses EDCA adalah protokol berbasis *contention* yang menggunakan seperangkat *wait interval* dan waktu *back-off* yang dirancang untuk menghindari tabrakan data. Namun, dengan DCF semua terminal menggunakan nilai yang sama dan karenanya memiliki prioritas yang sama untuk mentransmisikan data pada saluran.

Dengan EDCA, masing-masing prioritas akses berbeda diberi rentang interval waktu tunggu dan *counter back-off* yang berbeda. Transmisi dengan prioritas akses yang lebih tinggi diberikan interval yang lebih pendek. Standar ini juga mencakup mode *packet-bursting* yang memungkinkan jalur akses atau terminal bergerak untuk memesan saluran dan mengirim paket 3 sampai 5 paket secara berurutan.

2.5.2. WiFi Scheduled Multimedia (WSM)

Consistent delay services yang benar dapat diberikan dengan *WiFi Scheduled Multimedia (WSM)* yang bersifat opsional. WSM beroperasi seperti yang digunakan *Point Control Function (PCF)* yang didefinisikan dengan MAC 802.11 asli. Di WSM, access point secara berkala mem-*broadcast* pesan kontrol yang memaksa semua terminal melakukan perawatan saluran yang sibuk dan tidak dapat mentransmisikan data.

Selama periode ini, akses poling pendapat masing-masing terminal yang didefinisikan untuk layanan *time sensitive*. Untuk menggunakan opsi WSM, perangkat perlu mengirimkan profil lalu lintas data yang menjelaskan bandwidth, Latency, dan persyaratan jitter. Jika access point tidak memiliki sumber daya yang memadai memenuhi profil lalu lintas data, itu akan kembali mengeluarkan sinyal busy.

2.6. Keamanan Wifi

Keamanan telah menjadi salah satu kekurangan utama WiFi, meski sistem enkripsinya lebih baik. Sekarang telah tersedia Enkripsi di WiFi yang sifatnya opsional, ada tiga teknik yang berbeda untuk proses enkripsi. Teknik ini adalah sebagai berikut:

- *Wired Equivalent Privacy (WEP)*
Enkripsi 40-atau 104-bit berbasis RC4 dengan *static key*.
- *WiFi Protected Access (WPA)*
Enkripsi ini merupakan standar baru dari Aliansi WiFi yang menggunakan kunci WEP 40 atau 104-bit, tapi merubah kunci

dalam setiap paket. Untuk mengubah fungsi kunci disebut *Temporal Key Integrity Protocol* (TKIP).

- *IEEE 802.11i/WPA2*

IEEE telah merampungkan standar 802.11i, yang didasarkan pada enkripsi yang jauh lebih kuat, teknik ini disebut *Advanced Encryption Standard*. Aliansi WiFi menunjuk produk yang sesuai dengan standar 802.11i sebagai WPA2. Namun, penerapan 802.11i memerlukan upgrade perangkat keras.

2.7. Layanan Jaringan Wifi

WiFi merupakan istilah dari "*Wireless Fidelity*", memiliki pengertian yaitu sekumpulan standar yang dipakai untuk Jaringan Lokal Nirkabel (*Wireless Local Area Networks* – WLAN) yang didasarkan pada spesifikasi IEEE 802.11. Standar terbaru dari spesifikasi 802.11a atau b, seperti 802.11g, saat ini sedang dalam penyusunan, spesifikasi terbaru tersebut menawarkan banyak peningkatan mulai dari luas jangkauan yang lebih jauh hingga kecepatan transfernya.

Pada mulanya Wi-Fi digunakan untuk perangkat *wireless* dan Jaringan Lokal (LAN), namun saat ini lebih banyak digunakan untuk mengakses internet. Hal ini memungkinkan seseorang dengan komputer yang dilengkapi *wireless card* atau *notepad* untuk terhubung dengan internet dengan menggunakan *access point* (atau dikenal dengan *hotspot*) terdekat. Spesifikasi Wi-Fi dirancang berdasarkan spesifikasi IEEE 802.11. Sekarang ini ada empat variasi dari 802.11, yaitu: 802.11a, 802.11b, 802.11g, and 802.11n.

Spesifikasi b adalah produk pertama Wi-Fi. Variasi g dan n merupakan salah satu produk yang memiliki penjualan terbanyak pada 2005. Di belahan dunia, frekuensi yang digunakan oleh Wi-Fi, 802.11a menggunakan frekuensi yang lebih tinggi dan oleh karena itu daya jangkauannya lebih sempit. Versi Wi-Fi yang paling banyak dipasarkan sekarang ini (berdasarkan dalam IEEE 802.11b/g) beroperasi pada 2,4 GHz. Serta mengizinkan operasi dalam 11 channel (masing-masing 5 MHz), berpusat di frekuensi berikut:

- Channel 1 – 2,412 GHz;
- Channel 2 – 2,417 GHz;
- Channel 3 – 2,422 GHz;
- Channel 4 – 2,427 GHz;
- Channel 5 – 2,432 GHz;
- Channel 6 – 2,437 GHz;
- Channel 7 – 2,442 GHz;
- Channel 8 – 2,447 GHz;
- Channel 9 – 2,452 GHz;
- Channel 10 – 2,457 GHz;
- Channel 11 – 2,462 GHz

Secara teknis operasional, Wi-Fi merupakan salah satu varian teknologi komunikasi dan informasi yang bekerja pada jaringan dan perangkat WLANs (*wireless local area network*). Dengan kata lain, Wi-Fi adalah nama dagang (*certification*) yang diberikan pabrikan kepada perangkat telekomunikasi (Internet) yang bekerja di jaringan WLANs dan sudah memenuhi kualitas interoperability yang dipersyaratkan.

Wi-Fi (*Wireless Fidelity*) adalah koneksi tanpa kabel seperti handphone dengan mempergunakan teknologi radio sehingga pemakainya dapat mentransfer data dengan cepat dan aman. Wi-Fi tidak hanya dapat digunakan untuk mengakses internet, Wi-Fi juga dapat digunakan untuk membuat jaringan tanpa kabel dimanapun. Karena itu banyak orang mengasosiasikan Wi-Fi dengan kebebasan karena teknologi Wi-Fi memberikan kebebasan kepada pemakainya untuk mengakses internet atau mentransfer data dari ruang meeting, kamar hotel, kampus, dan café-café yang bertanda “**Wi-Fi Hot Spot**”.

Juga menjadi salah satu kelebihan dari Wi-Fi adalah kecepatannya yang beberapa kali lebih cepat dari modem kabel yang tercepat. Jadi pemakai Wi-Fi tidak lagi harus berada di dalam ruang kantor untuk bekerja. Wi-Fi dapat di akses dengan komputer, laptop, atau Smartphone yang telah dikonfigurasi dengan Wi-Fi. Dua contoh utama ini adalah wireless ISP dan jaringan mesh WiFi di seluruh kota.

- **Wireless ISP**

Salah satu bisnis yang tumbuh dari WiFi adalah *Wireless* ISP (WISP). Ini merupakan ide untuk menjual layanan akses Internet dengan menggunakan teknologi wireless LAN dan koneksi Internet bersama di tempat umum yang ditunjuk sebagai hot spot.

Dari sudut pandang teknis, akses ke layanan terbatas berdasarkan transmisi berbagai teknologi WLAN. Anda harus berada di hot spot (yaitu dalam 100 m dari Jalur akses) untuk menggunakannya. Dari sudut pandang bisnis, pengguna berlangganan layanan operator tertentu dengan biaya bulanan

atau mengakses layanan berdasarkan permintaan berdasarkan biaya per jam. Sementara biaya bulanan biasanya paling hemat biaya, ada beberapa pengaturan akses antar operator, jadi harus berada di hot spot yang dioperasikan oleh operator untuk mengakses layanan.

- Mesh Network WiFi

Untuk mengatasi keterbatasan jangkauan, vendor seperti Mesh Networks dan Tropos Networks memiliki kemampuan jaringan mesh yang dikembangkan menggunakan teknologi radio WiFi. Gagasan jaringan jaring radio adalah pesan dapat disampaikan melalui sejumlah jalur akses ke pusat kontrol jaringan. Jaringan ini biasanya dapat mendukung mobilitas saat koneksi dialihkan dari jalur akses ke mobile stasion.

2.8. Radio Mudalasi Wifi

Sistem WiFi menggunakan dua teknik transmisi radio primer, yaitu :

- 802.11b (\leq 11 Mbps): *Link* radio 802.11b menggunakan teknik sekumpulan spektrum urutan langsung yang disebut *complementary coded keying* (CCK). Bit stream diproses dengan pengkodean khusus dan kemudian dimodulasi menggunakan *Quadrature Phase Shift Keying* (QPSK)
- 802.11a dan g (\leq 54 Mbps): Sistem 802.11a dan g menggunakan *Orthogonal Frequency Division Multiplexing* 64 kanal (OFDM). Dalam sistem modulasi OFDM, pita radio yang tersedia dibagi menjadi beberapa sub-saluran dan

beberapa bit dikirim masing-masing. Pemancar mengkodekan aliran bit pada 64 subcarrier menggunakan *Binary Phase Shift Keying* (BPSK), *Quadrature Phase Shift Keying* (QPSK), atau satu dari dua tingkat *Quadrature Amplitude Modulation* (16, atau 64-QAM). Beberapa informasi yang ditransmisikan bersifat berlebihan, sehingga penerima tidak harus menerima semua sub-carrier untuk merekonstruksi informasi.

BAB III

ESP8266

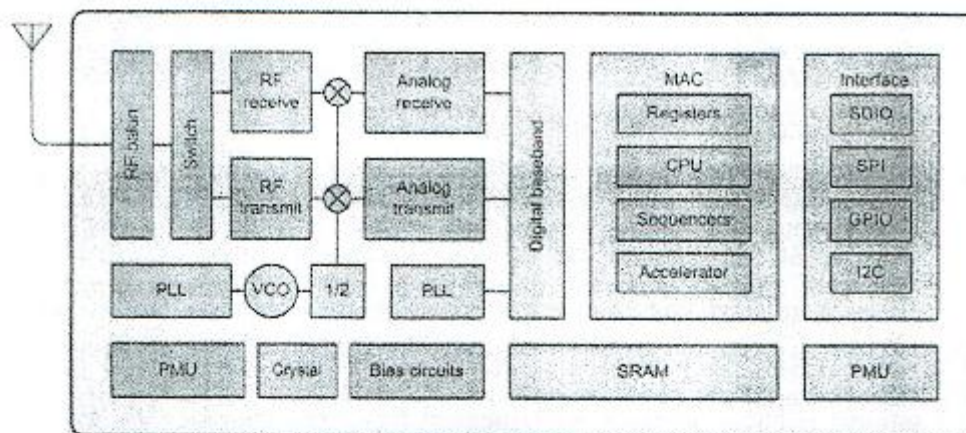
ESP8266 merupakan modul WiFi yang murah dan mudah untuk ditambahkan fungsionalitas WiFi ke minimum sistem mikrokontroler yang ada melalui koneksi serial UART. Modul ini dapat diprogram ulang sebagai perangkat terhubung WiFi secara mandiri hanya cukup menambahkan daya.

ESP8266 adalah solusi jaringan Wi-Fi lengkap dan mandiri yang dapat mengembangkan aplikasi perangkat lunak, atau melalui prosesor aplikasi lain yang dapat meningkatkan kemampuan jaringan Wi-Fi. ESP8266 merupakan perangkat terpasang dan sebagai satu-satunya aplikasi prosesor wifi, memori flash bisa langsung digunakan sebagai memori eksternal.

Memori *cache built-in* akan membantu meningkatkan kinerja sistem dan mengurangi kebutuhan memori. Ketika akses Internet lewat wireless ESP8266 ini diasumsikan bertugas sebagai adaptor Wi-Fi, dan dapat ditambahkan ke desain berbasis mikrokontroler, dengan koneksi sederhana hanya dengan antarmuka SPI/SDIO. Pengolahan dan kapasitas penyimpanan pada ESP8266 sangat besar (4 MB), dapat

diintegrasikan melalui GPIO. Chip ESP8266 yang terintegrasi, termasuk balok sakelar antenna, konverter daya, sehingga rangkaian eksternal minimal, dan modul front-end, secara keseluruhan solusi yang dirancang untuk meminimalkan ruang yang menempati PCB.

Sistem yang dilengkapi dengan fitur unggulan terkemuka ESP8266 adalah: VoIP hemat energi dengan cepat beralih antara pola *sleep/ wake*, dengan sinyal radio adaptif berdaya rendah, fungsi pemrosesan sinyal front-end, pemecahan masalah dan karakteristik sistem radio yang meluas menghubungkan seluler/Bluetooth/ Interferensi DDR/ LVDS / LCD.



Gambar 3.1. Blok Diagram ESP8266

3.1 Fitur

- ✓ 802.11 b/g/n
- ✓ Integrated low power 32-bit MCU
- ✓ Integrated 10-bit ADC
- ✓ Integrated TCP/IP protocol stack

- ✓ Integrated TR switch, balun, LNA, power amplifier and matching network
- ✓ Integrated PLL, regulators, and power management units
- ✓ Supports antenna diversity
- ✓ Wi-Fi 2.4 GHz, support WPA/WPA2
- ✓ Support STA/AP/STA+AP operation modes
- ✓ Support Smart Link Function for both Android and iOS devices
- ✓ Support Smart Link Function for both Android and iOS devices
- ✓ SDIO 2.0, (H) SPI, UART, I2C, I2S, IRDA, PWM, GPIO
- ✓ STBC, 1x1 MIMO, 2x1 MIMO
- ✓ A-MPDU & A-MSDU aggregation and 0.4s guard interval
- ✓ Deep sleep power < 5uA
- ✓ Wake up and transmit packets in < 2ms
- ✓ Standby power consumption of < 1.0mW (DTIM3)
- ✓ +20dBm output power in 802.11b mode
- ✓ Operating temperature range -40C ~ 125C

3.2. Parameter-parameter

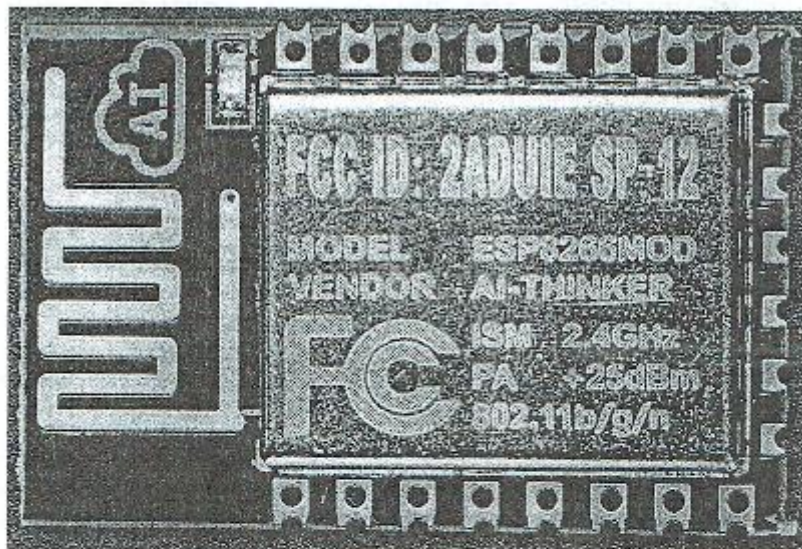
Tabel 3.1. Parameter ESP8266

Kategori	Item	Nilai
Parameter WiFi	Protokol WiFi	802.11 b/g/n
	Range Frekuensi	2,4GHz - 2,5 GHz
Parameter Hardware	Peripheral Bus	UART/IISPI/I2C/I2S/IR Remote Control GPIO/PWM
	Tegangan operasi	3.0 ~ 3.6 V
	Arus	Rata-rata 80mA
	Cakupan Temperatur	-40° ~ 125 °

	Operasi	
	Cakupan Ambient	Temperatur Normal Temperatur
	Ukuran Kemasan	16mm*24mm*3mm
	External Interface	N/A
Parameter Software	Mode WiFi	Station/softAP/SoftAP station +
	Keamanan	WPA/WPA2
	Enkripsi	WEP/TKIP/AES
	Upgrade Firmware	UART Download/OTA(via network)/via host
	Pengembangan Software	Supports Cloud Server Development/SDK
	Network Protokol	IPV4, TCP/UDP/HTTP/FTP
	Konfigurasi User	Set Instruksi AT, Cloud Server, Android/iOS App

3.3. Diskripsi Pin

Terdapat 22 pin yang dimiliki oleh ESP8266, yang definisi pin dijelaskan pada Tabel 3.2 di bawah ini.



Gambar 3.2. Chip ESP8266

Tabel 3.2. Diskripsi Pin Esp8266

No	Nama Pin	Fungsi
1	RST	Modul Reset
2	ADC	Conversi A/D, Teg. Input 0-1V nilai 0-1024
3	EN	Pin Enable Chip, Aktif High
4	IO16	GPIO16, mode sleep/wake up
5	IO14	GPIO14, HSPI_CLK
6	IO12	GPIO12, HSPI_MISO
7	IO13	GPIO13, HSPI_MOSI, UART0_CTS
8	VCC	3,3V power supply
9	CS0	Ch Ip Selection
10	MISO	Slave Output Main Input
11	IO9	GPIO9
12	IO10	GPIO10
13	MOSI	Main Output Slave Input
14	SCLK	Clock
15	GND	Ground
16	IO15	GPIO15, MTDO, HSPICS, UART0_RTS
17	IO2	GPIO2, UART1_TXD
18	IO0	GPIO0
19	IO4	GPIO4
20	IO5	GPIO5
21	RXD	UART0_RXD, GPIO3
22	TXD	UART0_TXD, GPIO1

Tabel 3.3. Pin Mode

Mode	GPIO15	GPIO0	GPIO2
UART	Low	Low	High
Flash Boot	Low	High	High

Tabel 3.4. Sensitivitas Penerimaan Sinyal

Parameters	Min	Typical	Max	Unit
Input frequency	2412		2484	MHz
Input impedance		50		Ω
Input reflection			-10	dB
Output power of PA for 72.2Mbps	15.5	16.5	17.5	dBm
Output power of PA for 11b mode	19.5	20.5	21.5	dBm
Sensitivity				
DSSS, 1Mbps		-98		dBm
CCK, 11Mbps		-91		dBm
6Mbps (1/2 BPSK)		-93		dBm
54Mbps (3/4 64-QAM)		-75		dBm
HT20, MCS7 (65Mbps, 72.2Mbps)		-72		dBm
Adjacent Channel Rejection				
OFDM, 6Mbps		37		dB
OFDM, 54Mbps		21		dB
HT20, MCS0		37		dB
HT20, MCS7		20		dB

3.4. Flashing ESP8266

Chip ESP8266 adalah modul wifi yang berfungsi sebagai perangkat tambahan dalam mikrokontroler (misalnya Arduino) agar dapat terhubung langsung dengan perangkat wifi lainnya dan dapat membuat koneksi TCP/IP.

Modul ini membutuhkan tegangan sekitar 3.3V, dengan memiliki tiga mode wifi yaitu Station, Access Point dan Both (Keduanya). Modul ini juga sudah dilengkapi dengan prosesor, memori dan GPIO, jumlah pin bergantung dengan jenis ESP8266 yang digunakan. Sehingga modul ini dapat bersifat *stand alone* tanpa menggunakan mikrokontroler apapun karena sudah memiliki kelengkapan layaknya mikrokontroler.

Firmware default yang dipakai oleh ESP8266 ini menggunakan AT Command, selain itu ada beberapa *Firmware* SDK yang digunakan oleh perangkat ESP8266 ini berbasis *opensource* diantaranya adalah:

- NodeMCU dengan menggunakan basic programming lua
- MicroPython dengan menggunakan basic programming python
- AT Command dengan menggunakan perintah perintah AT command

Untuk pemrogramannya sendiri digunakan ESPlorer untuk *Firmware* berbasis NodeMCU dan menggunakan putty sebagai terminal control untuk AT Command. Selain itu untuk memprogram perangkat ini dapat menggunakan Arduino IDE. Dengan menambahkan library ESP8266 pada board manager dapat dengan mudah dilakukan pemrogram dengan basic program arduino.

ESP8266 diberi nilai 3.3V dan pin Arduino UNO RX dan TX diberi nilai 5V. Sirkuit berikut dapat bekerja tanpa merusak ESP8266, Arduino atau laptop tapi voltase yang lebih tinggi dapat merusak perangkat ini.

3.4.1. Material yang dibutuhkan

Hardware :

1. Arduino Uno
2. USB cable untuk daya Arduino dari komputer
3. ESP8266-01 Module
4. Kabel Jumper (male to male and female to male)

5. Breadboard (atau prototype shield) – bersifat optional
6. 2x atau 3x Resistors (untuk pembagi tegangan 3.3V/5V)

Software :

1. Arduino IDE
2. Python 2, PIP, esptool
3. Firmware ESP8266, Micropython

3.4.2. Instalasi Software Yang Dibutuhkan

Beberapa software yang dibutuhkan untuk melakukan flash ESP8266.

- Arduino IDE
- Python 2
- PIP
- esptool (installed lewat PIP)
- MicroPython (ESP8266 Firmware)

Instalasi IDE Arduino dan Python 2 untuk sistem operasi yang ada. Bila versi PIP lama hanya dengan Python 2. Untuk hal itu buka terminal/console untuk mengupgrade PIP versi baru:

```
1 pip install --upgrade esptool
```

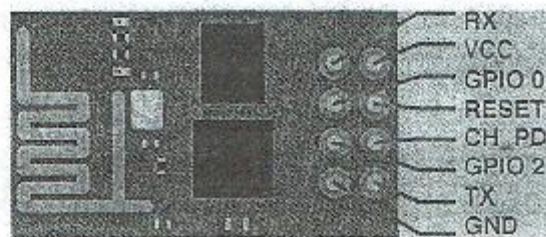
Jika mendapatkan kesalahan di Linux, coba sudo:

```
1 sudo pip install --upgrade esptool
```

Untuk saat ini, hanya diperlukan mendownload MicroPython. Setelah mendownload file `esp8266-20161110-v1.8.6.bin` - ganti namanya menjadi `micropy.bin`.

3.4.3. Bagaimana Menghubungkan ESP8266 dengan Arduino

Kita pasang rangkaian untuk mengubah Arduino menjadi konverter *USB-to-Serial* yang kemudian digunakan untuk meng-flash ESP8266 dengan firmware baru. Berikut adalah pinout ESP-01:



Gambar 3.3. Beard ESP8266 dan Pinout

Jalur Pin Arduino dan ESP8266

UNO — ESP8266

RX — TX

TX — RX

GND — GND

GND — GPIO 0

3.3V — VCC

3.3V — CH_PD

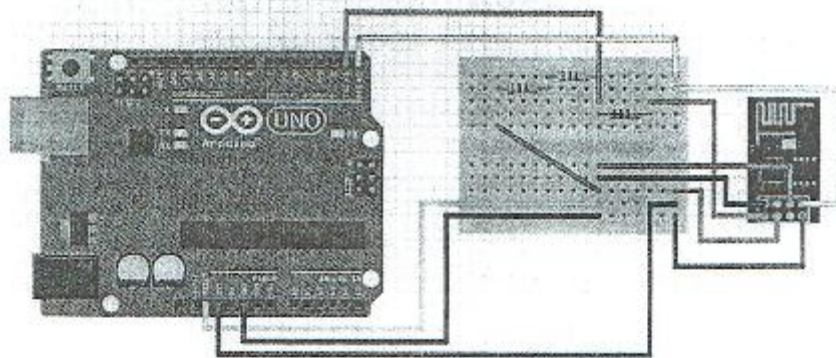
Dapat menghubungkan ESP8266 ke Arduino Uno dengan atau tanpa pembagi tegangan. Pembagi tegangan mengubah Arduinos 5V TX/RX ke 3.3V yang dibutuhkan oleh

port ESP8266 TX/RX. Anda memerlukan tiga Ohm 1k Ohm atau tiga resistor Ohm.

Selain itu, dihubungkan jumper dari RESET ke GND di Arduino. Hal ini akan mengubah Uno menjadi konverter TTL-serial-to-USB. Setelah menyetel jumper, unggah sketsa kosong (program kosong ke Arduino).

3.4.4. Rangkaian Pembagi Tegangan

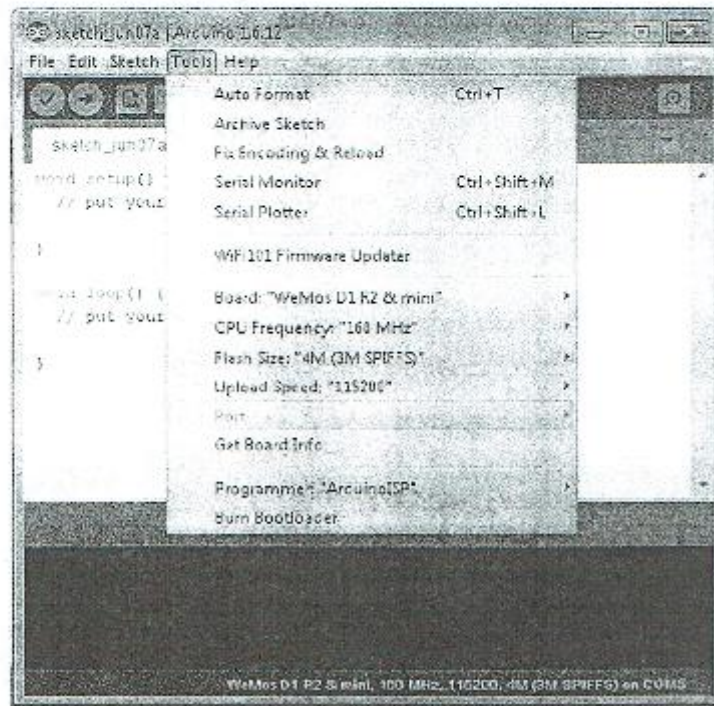
Pertama menghubungkan ESP TX / RX, yang bekerja pada 3.3 Volt, dengan Arduino TX / RX, yang bekerja pada 5 Volts, melalui pembagi tegangan tiga buah resistor.



Gambar 3.4. Koncksi Arduino UNO dengan ESP8266

3.4.5. Hubungkan Komputer dengan Arduino

Pastikan bahwa Arduino terhubung dengan komputer /laptop yang ditunjukkan pada menu Tools → Port



Gambar 3.5. Menu untuk menghubungkan Komputer dengan Arduino

3.4.6. Menguji Koneksi ESP8266 dengan Arduino

ESP8266 diisi dengan firmware dari AI-Thinker, untuk mengirim perintah AT yang digunakan memeriksa apakah modul dan konfigurasi tes berfungsi seperti yang diharapkan sebelum flash modul dengan firmware baru.

Buka **Tools > Serial Monitor** pada IDE Arduino dan set **"Both NL & CR"** dan **Baud rate 9600**.

Kirim perintah untuk proses pengujian :



Respon setelah perintah AT ini adalah OK. Jika tidak ada yang muncul, cobalah baud rate yang berbeda seperti 115200 atau 57600 dan tes ulang. Sekarang coba perintah berikut ini:

```
1 AT+GMR
```

Respon dari perintah diatas adalah versi firmware dan release dari ESP8266 saat ini:

```
1 AT version:0.40.0.0(Aug 8 2015 14:45:58)
2 SDK version:1.3.0
3 AI-Thinker Technology Co.,Ltd.
4 Build:1.3.0.2-Sep-11-2015 11:48:04
```

Jika Anda ingin mempelajari lebih lanjut tentang perintah AT, daftar perintah dapat ditemukan dibawah ini.

3.4.7. Upload Skecth Kosong Arduino

Menyeting pin GND dan pin RESET, kemudian mengunggah sketch kosong ke Arduino :

```
1 void setup() {
2   // put your setup code here, to run once
3 }
4 void loop() {
5   // put your main code here, to run repeatedly.
6 }
```

3.4.8. Menghapus Flash Memori

Jika perintah AT di atas bekerja seperti yang diharapkan, kita siapkan ESP8266 dan hapus memori flashnya.

Buka program terminal / konsol dan masukkan perintah berikut-ingat, `/dev / ttyACM0` merupakan port USB yang dilihat pada IDE Arduiono sebelumnya, jadi kemungkinan berbeda!

```
esptool.py --port /dev/ttyACM0 erase_flash
```

Keluaran dari esptool adalah

```
1 esptool.py v1.2.1
2 Connecting..
3 Running Cesanto floshe stub..
4 Erasing flash (this may take a while)..
5 Erase took 0.9 seconds
```

Jika muncul kesalahan, lepaskan kabel USB dan pasang kembali. Dengan cara ini Anda me-reboot ESP8266 Anda. Dalam 5 detik berikutnya ulangi perintahnya.

3.4.9. Flash Firmware ESP8266

Lepaskan dan sambungkan kembali kabel USB untuk me-reboot modul ESP8266 Anda. Sekarang masukkan perintah berikut:

```
esptool.py --port /dev/ttyACM0 --baud 115200
write_flash --flash_size=detect 0
/home/myuser/downloads/micropy.bin -verify
```

ESPtool akan mengeluarkan output sebagai berikut :

```

1  esptool.py v1.2.1
2  Connecting..
3  Auto-detected Flash size: 8m
4  Running Cesanta flasher stub..
5  Flash params set to 0x0020
6  Writing 569344 @ 0x0. 569344 (100 %)
7  Wrote 569344 bytes at 0x0 in 48.4 seconds (94.1 kbit/s)..
8  Leaving..
9  Verifying just-written flash..
10 Verifying 0x8a2dc (565980) bytes @ 0x00000000 in flash against /home/vi
11 - verify OK (digest matched)

```

3.4.10. Menghubungkan ke wifi ESP8266

Setelah berhasil mentransmisikan ESP dengan MicroPython, modul akan menampilkan *Access Point* (AP) wifi baru dengan SSID yang mirip dengan "*MicroPython-103432*". Saat menggunakan jumper untuk menghubungkan RESET dan GND pada Arduino, lepaskan jumper dan sambungkan kembali kabel USB dan AP, wifi baru harus muncul dalam daftar. Kata sandi untuk wifi adalah **micropython** (perhatikan huruf besar N) dan alamat IP-nya adalah 192.168.4.1.

3.5. Command AT Dasar

Modul Wifi ESP8266 dapat dikendalikan melalui serial interface dengan perintah standar "*AT command*". Berikut ini adalah perintah AT command yang dapat digunakan.

Dasar	
Command	Diskripsi
AT	Menguji Konektivitas
AT+RST	Restart Modul
AT+GMR	Info Versi modul
AT+GSLP	Enter deep-sleep mode
ATE	Command AT echo atau tidak
AT+RESTORE	Reset Factory

AT+UART	Konfigurasi UART
AT+UART_CUR	Konfigurasi UART yang sedang berjalan
AT+UART_DEF	Konfigurasi Default UART
AT+SLEEP	Mode Sleep
AT+RFPOWER	Set nilai maksimum daya TX RF
AT+RFVDD	Set daya TX RF sama dengan VDD33

3.5.1. Perintah AT

Perintah ini digunakan untuk menguji fungsi penyiapan modul WiFi nirkabel yang akan digunakan.

AT – Test AT startup	
Response	OK
Parameters	null

3.5.2. Restart Modul

Perintah ini berfungsi untuk merestart modul, bila terjadi 'hang' atau kesalahan di saat proses terjadi dan ingin kembali ke awal program.

AT+RST – Restart module	
Response	OK
Parameters	null

3.5.3. Melihat Info Versi AT Command

Perintah AT ini digunakan untuk memeriksa versi perintah AT dan SDK yang digunakan.

AT+GMR—View version info	
Response	<AT version info> <SDK version info> <compile time> OK
Parameters	<AT version info> information about AT version <SDK version info> information about SDK version <compile time> time of the bin was compiled

3.5.4. Enter Deep-sleep Mode

Perintah ini digunakan untuk memanggil mode deep-sleep dalam modul, yang tipenya "set". Penyesuaian kecil harus dilakukan sebelum modul memasuki mode *deep sleep* ini, yaitu menghubungkan XPD_DCDC dengan EXT_RSTB melalui resistor 0 ohm.

AT+GSLP=<time>	
Response	<time> OK
Parameters	The time unit of <time> is ms ESP8266 will wake up after deep sleep <time>ms

3.5.5. AT Command Echo

Perintah ATE ini adalah perintah trigger AT echo. Artinya perintah yang masuk bisa diulang kembali ke pengirim saat perintah ATE digunakan. Perintah mengembalikan "OK" dalam kasus normal dan "ERROR" bila parameter selain 0 atau 1 telah ditentukan.

ATE—AT commands echo	
Response	OK
Parameters	ATE0: Switch echo off ATE1: Switch echo on

3.5.6. Reset Factory

Perintah ini digunakan untuk mengatur ulang semua parameter yang tersimpan dalam flash, mengembalikan pengaturan default pabrik. Chip akan di restart saat perintah ini dijalankan.

AT+RESTORE – Factory reset	
Response	OK
Notes	Restore factory default settings. The chip will restart.

3.5.7. Konfigurasi UART

Perintah ini mengatur konfigurasi UART dan menulis konfigurasi baru ke memori flash. Ini disimpan sebagai parameter default dan juga akan digunakan sebagai default baudrate untuk selanjutnya.

AT+UART=<baudrate>,<data bits>,<stopbits>,<parity>,<flow control> This command is deprecated, please use AT+UART_DEF or AT+UART_DEF instead.		
Example	AT+UART=115200,8,1,0,3	
Response	OK	
Parameters	<baudrate>	Baudrate range: 110 to 115200*40 (4.800 Mega)
	<data bits>	5: 5 bits data 6: 6 bits data 7: 7 bits data 8: 8 bits data
	<stopbits>	1: 1 bit stop bit 2: 1.5 bit stop bit 3: 2 bit stop bit
	<parity>	0: None 1: Odd 2: EVEN
	<flow control>	0: disable flow control 1: enable RTS 2: enable CTS 3: enable both RTS and CTS

Notes	<ul style="list-style-type: none"> This configuration will also store the baudrate as the default rate in the user parameter area in the Flash for boot up Flow control needs hardware support: MTCK is UART0 CTS and MTDO is UART0 RTS.
--------------	--

3.5.8. Konfigurasi UART yang sedang berjalan

Perintah ini mengatur konfigurasi UART saat ini; Hal ini tidak menulis ke memori plash Makanya tidak ada perubahan pada baudrate default.

AT+UART_CUR=<baudrate>,<databits>,<stopbits>,<parity>,<flow control>		
Example	AT+UART_CUR=115200,8,1,0,3	
Response	OK	
Parameters	<baudrate>	Baudrate range: 110 to 115200*40 (4.608 Mega)
	<databits>	5: 5 bits data 6: 6 bits data 7: 7 bits data 8: 8 bits data
	<stopbits>	1: 1 bit stop bit 2: 1.5 bit stop bit 3: 2 bit stop bit
	<parity>	0: None 1: Odd 2: EVEN
	<flow control>	0: disable flow control 1: enable RTS 2: enable CTS 3: enable both RTS and CTS
Notes	<ul style="list-style-type: none"> This configuration will NOT store in the Flash. Flow control needs hardware support: MTCK is UART0 CTS and MTDO is UART0 RTS 	

3.5.9. Konfigurasi UART Default

Perintah ini mengatur konfigurasi UART dan yang tersimpan di memori flash. Ini disimpan sebagai parameter default dan juga akan digunakan sebagai default baudrate untuk selanjutnya.

AT+UART_DEF=<baudrate>,<databits>,<stopbits>,<parity>,<flow control>		
Example	AT+UART_DEF=115200,8,1,0,3	
Response	OK	
Parameters	<baudrate>	Baudrate range: 110 to 115200*40 (4.608 Mega)
	<databits>	5: 5 bits data 6: 6 bits data 7: 7 bits data 8: 8 bits data
	<stopbits>	1: 1 bit stop bit 2: 1.5 bit stop bit 3: 2 bit stop bit
	<parity>	0: None 1: Odd 2: EVEN
	<flow control>	0: disable flow control 1: enable RTS 2: enable CTS 3: enable both RTS and CTS
Notes	<ul style="list-style-type: none"> This configuration will be stored in user parameter area in the Flash for boot up Flow control needs hardware support: MTCK is UART0 CTS and MTDO is UART0 RTS. 	

3.5.10. Mode Sleep

Perintah ini mengatur mode sleep ESP8266. Ini hanya bisa digunakan dalam pusat mode, defaultnya adalah modem mode sleep.

AT+SLEEP=<sleep mode>	
Command	AT+SLEEP?
Response	+SLEEP: <sleep mode> OK
Parameters	<sleep mode> 0: disable sleep mode 1: light-sleep mode 2: modem-sleep mode
Command	AT+SLEEP=<sleep mode>
Response	OK
Parameters	The same as above.

3.5.11. Set Nilai Maksimum Daya untuk TX RF

Perintah ini menetapkan nilai maksimum daya RF TX RF8266.

AT+RFPOWER – set RF TX Power	
Example	AT+RFPOWER=50
Command	AT+RFPOWER=<TX power>
Response	OK
Parameters	<TX power> maximum value of RF TX power, range 0 ~ 82, unit: 0.25dBm

3.5.12. Set Daya RF TX sama dengan VDD33

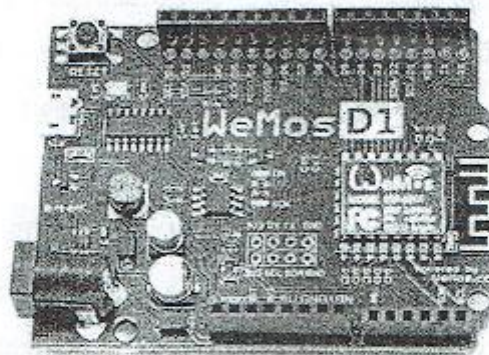
Perintah ini mengatur daya ESP8266 RF TX sesuai dengan VDD33. Untuk mendapatkan tegangan listrik VDD33 ESP8266, pin TOUT harus ditahan. Pin TOUT harus ditahan untuk mengukur VDD33.

AT+RFVDD – set RF TX power according to VDD33	
Command	AT+RFVDD?
Response	+RFVDD <VDD33> OK
Parameters	<VDD33> power voltage of ESP8266 VDD33, unit: 1/1024 V TOUT pin has to be suspended to measure VDD33
Command	AT+RFVDD=<VDD33>
Response	OK
Parameters	<VDD33> power voltage of ESP8266 VDD33, range [1800, 3300]
Command	AT+RFVDD
Response	OK
Note	*AT+RFVDD* will automatically set RF TX power according to VDD33. TOUT pin has to be suspended.

BAB IV

BOARD WEMOS DAN INSTALASI

Board ini terlihat seperti board Arduino yang sering dipakai kebanyakan orang. Dimensi dan tata letak pin sama persis. Jadi, board ini kompatibel dengan semua *shield* yang ada untuk Arduino. Tapi jangan berharap bisa bekerja dengan mudah dengan modul ini, karena *library* yang tersedia untuk chip ESP8266 tidak banyak. Board ini tidak menggunakan chip ATMEGA seperti board Arduino standar, modul ini menggunakan chip WiFi ESP8266 yang mengesankan.



Gambar 4.1. Modul Wemos D1 R2

Chip ESP8266EX yang digunakan pada board WeMos D1:

- ✓ 32 bit RISC CPU running at 80MHz
- ✓ 64Kb RAM instruksi dan 96Kb RAM data
- ✓ 4MB flash memory
- ✓ Wi-Fi
- ✓ 16 GPIO pins
- ✓ I2C,SPI
- ✓ I2S
- ✓ 1 ADC

Ini adalah prosesor 32bit yang berjalan pada frekuensi yang jauh lebih tinggi. Perbedaan kinerja dengan standar Arduino jauh lebih besar dibandingkan dengan Arduino Uno, menawarkan 64KB memori RAM instruksi dan 96KB RAM data. Memori Flash perangkat juga 100 kali lebih besar (4 MB) dari Arduino. Tentu saja, ini merupakan alasan utama bahwa setiap orang menggunakan chip ini, adalah bahwa pada chip ini sudah dapat terkoneksi langsung ke Wi-Fi.

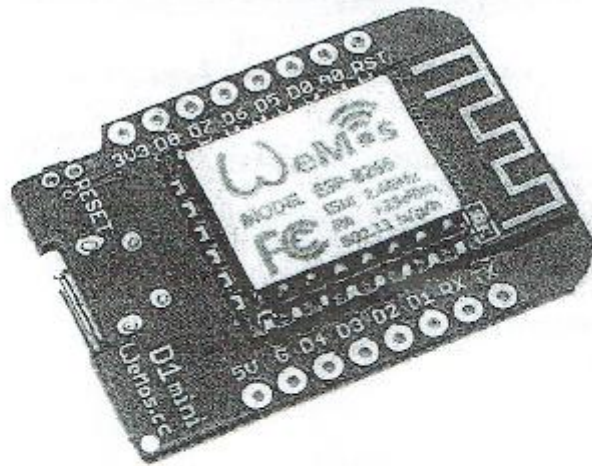
Agar bisa menggunakan board system ini, diharuskan mendownload add-on untuk Arduino IDE. Cari file *json* di link berikut ini <https://github.com/esp8266/Arduino>, kemudian cari *path*-nya http://arduino. esp8266.com/stable/package_esp8266com_index.json copi dan nanti *dipaste*-kan di Arduino IDE. Buka File -> Preferences dan masukkan link diatas kesini. Lalu ke menu Tools -> Board dan *load the Boards Manager*.

Yang harus dilakukan adalah mendownload file untuk board ESP8266. Setelah itu bisa memilih board WeMos D1(ESP 8266) dan siap menggunakannya. Untuk mencoba memulai menggunakan board

di upload program led berkedip standar. Buka **File -> Example -> ESP866 -> Blink** untuk memuat program dan kemudian tekan tombol upload. Setelah beberapa saat LED pada chip ESP8266 akan mulai berkedip.

Perhatikan, bahwa ini bukan sketsa Blink standar yang kami gunakan sejauh ini. Ini serupa yang dikembangkan untuk ESP8266. Karena arsitektur perangkat keras yang berbeda dari kebanyakan program ESP8266 tidak akan bekerja tanpa modifikasi. Juga sebagian besar *library* perlu ditulis ulang agar bisa bekerja dengan baik. Hal itu banyak pekerjaan yang perlu dilakukan.

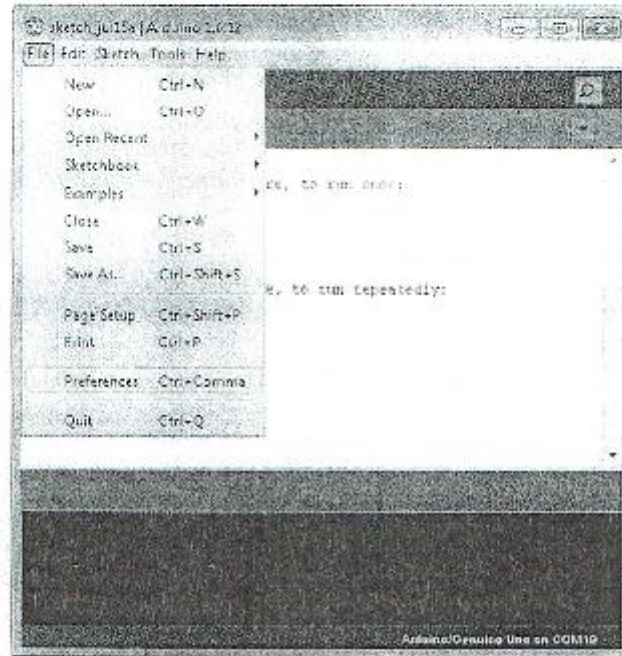
Hal pertama yang ingin diketahui adalah kemudahan terhubung ke internet dengan board ini. Jadi akan mudah mengembangkan sebuah program sederhana yang terhubung ke jaringan Wi-Fi dimana saja.



Gambar 4.2. Wemos D1 Mini

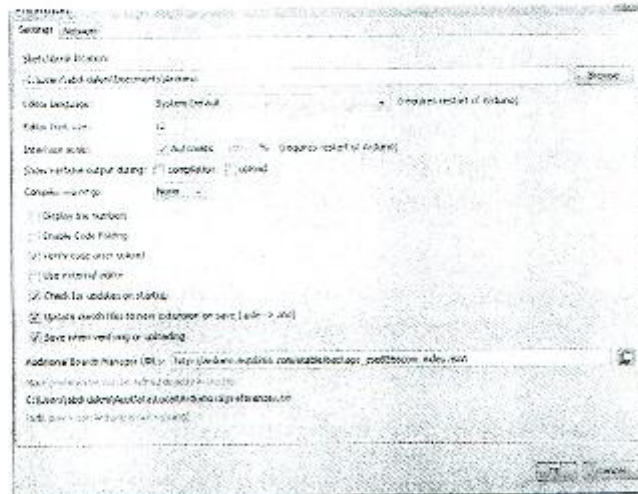
4.1. Langkah-langkah Instalasi Wemos (ESP8266)

1. Jalankan IDE arduino, lalu pilih menu *File -> Preferences* atau tekan tombol *Ctrl* dan tanda koma pada keyboard seperti ini:



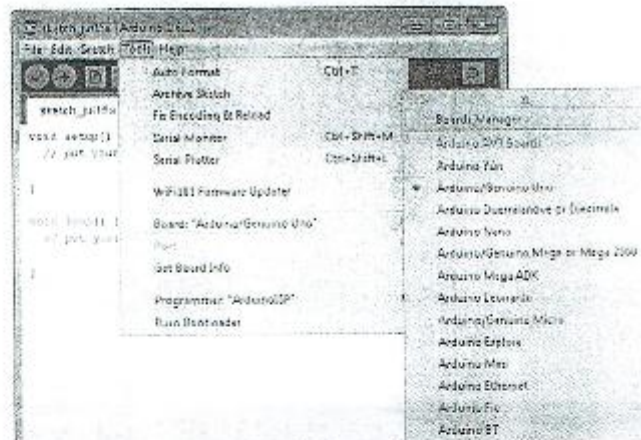
Gambar 4.3. Menu untuk mengisi path json melalui preferences

2. Pada dialog *Preferences*, masukkan alamat http://arduino.esp8266.com/stable/package_esp8266com_in_dex.json pada bagian *Additional Boards Manager URLs* seperti pada gambar di bawah ini:



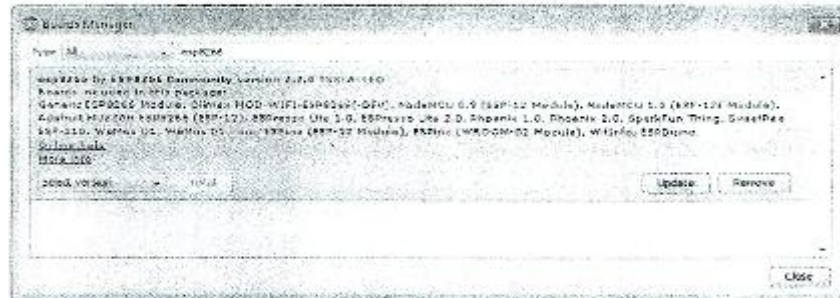
Gambar 4.4. Form Dialog Preferences

3. Selanjutnya, pilih menu *Tools* -> *Board* -> *Boards Manager...* seperti pada gambar berikut ini:



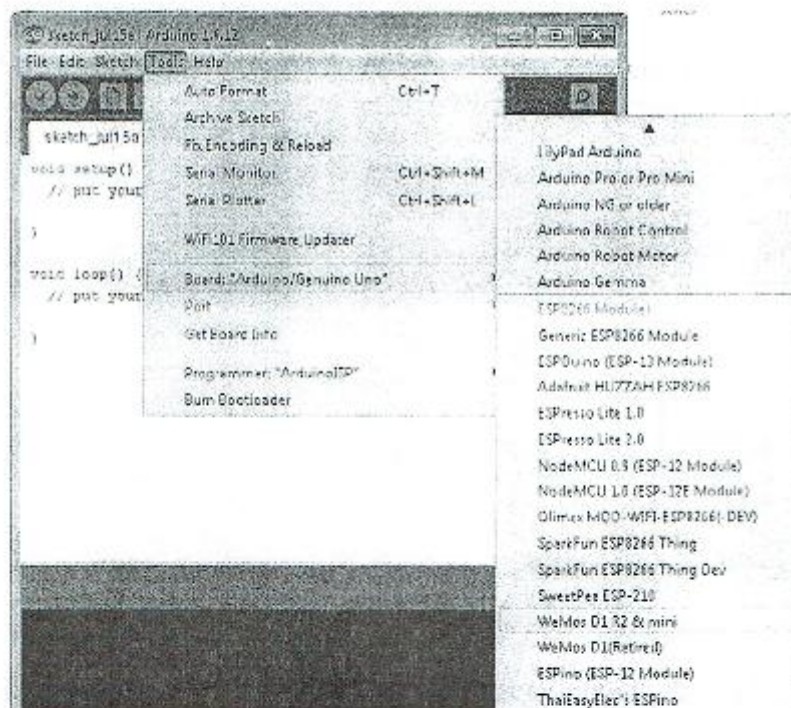
Gambar 4.5. Mencari Board Wemos di Board Manager

4. Pada dialog *Boards Manager*, masukkan *esp8266* lalu lakukan instalasi. Setelah proses instalasi selesai, hasilnya seperti ini:



Gambar 4.6. Form Dialog Search Komponen

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



Gambar 4.7. Menu Untuk Menentukan Kontroler Yang digunakan

4.2. Fungsi Pin pada Wemos D1 & R2 dan Mini

Pin dari chip wemos ini berfungsi sebagai saluran multiplek yaitu satu pin dapat berfungsi beberapa, misalnya dapat berfungsi sebagai saluran digital, dan analog atau saluran data dan control, ataupun input dan output. Pin wemos terdiri dari 16 pin disebelah kanan dan kiri dari chip esp8266. Pin ini sudah dibundling dengan software IDE Arduino, jadi dapat diatur menggunakan software melalui program inisialisasi.

Tabel 4.1. Fungsi Pin Wemos D1 Mini (<https://wiki.wemos.cc/downloads>)

Pin	Function	ESP-8266 Pin
TX	TXD	TXD
RX	RXD	RXD
A0	Analog input, max 3.3V input	A0
D0	IO	GPIO16
D1	IO, SCL	GPIO5
D2	IO, SDA	GPIO4
D3	IO, 10k Pull-up	GPIO0
D4	IO, 10k pull-up, BUILTIN_LED	GPIO2
D5	IO, SCK	GPIO14
D6	IO, MISO	GPIO12
D7	IO, MOSI	GPIO13
D8	IO, 10k pull-down, SS	GPIO15
G	Ground	GND
5V	5V	—
3V3	3.3V	3.3V
RST	Reset	RST

BAB V

ARDUINO WEMOS

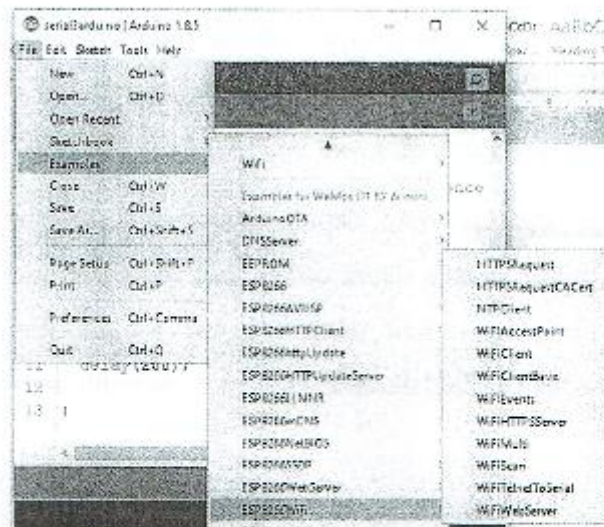
Setelah dilakukan instalasi board wemos dengan mengcopi path json di github selanjutnya akan muncul board di software IDE Arduino, yang dapat dimanfaatkan untuk eksperimen dengan board nya Wemos . Untuk mengetahui apakah board wemos telah terinstall dengan melihat di menu tools pada sosftware Arduino dan di check pada submenu board selanjutnya ada sekumpulan modul yang tergabung dalam grup ESP8266 modul, cek board “Wemos D1 mini” seperti terlihat pada gambar dibawah ini.

Beberapa contoh yang dapat digunakan untuk eksperimen menggunakan board wemos dapat dilihat dan dicoba di menu file → examples , pilih yang terkait dengan board wemos yang terlebih dahulu boardnya harus diaktifkan dengan cara memilih board “wemos d1 R2 & Mini”.



Gambar 5.1. Menu Board Modul ESP8266

Dalam contoh-contoh program dapat dilakukan untuk eksperimen I/O pada umumnya ataupun aplikasi seperti pada modul Arduino yang lainnya. Kelebihannya board wemos ini adalah chipnya sudah termasuk wifi sehingga dapat terhubung dengan jaringan dan dapat dimanfaatkan untuk aplikasi IoT. Contoh-contoh program yang dapat dimanfaatkan untuk bereksperimen seperti gambar dibawah ini.



Gambar 5.2. Contoh Program Untuk ESP8266WiFi

5.1. WiFiScan

Sketsa ini menunjukkan cara memindai jaringan WiFi. API hampir sama dengan pustaka WiFi Shield, perbedaan paling jelas adalah file berbeda yang perlu disertakan:

(<https://github.com/esp8266/Arduino/blob/master/libraries/ESP8266WiFi/examples/WiFiScan/WiFiScan.ino>)

```
/*
 * This sketch demonstrates how to scan WiFi networks.
 * The API is almost the same as with the WiFi Shield
 * library,
 * the most obvious difference being the different file
 * you need to include:
 */
#include "ESP8266WiFi.h"

void setup() {
  Serial.begin(115200);

  // Set WiFi to station mode and disconnect from an AP if
  // it was previously connected
  WiFi.mode(WIFI_STA);
  WiFi.disconnect();
  delay(100);

  Serial.println("Setup done");
}

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

  // WiFi.scanNetworks will return the number of networks
  // found
  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) {
      // Print SSID and RSSI for each network found
      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) ? " " : "+");
        delay(10);
    }
}
Serial.println("");

// Wait a bit before scanning again
delay(5000);
}

```

Hasil dari program ini adalah mendeteksi keberadaan sinyal wifi disekitar board wemos ini yang terjangkau beserta kekuatan sinyal wifinya.

```

COM10
scan done
3 networks found
1: TAMU PIKONARO (-49)*
2: sky fall (-83)*
3: Unnar Jaya Wifi Zone (-90)

scan start
scan done
4 networks found
1: TAMU PIKONARO (-49)*
2: sky fall (-77)*
3: Unnar Jaya Wifi Zone (-80)
4: Unnar Jaya Wifi Zone (-86)

```

Gambar 5.3. Tampilan Serial Monitor Sinyal Wifi

5.2. Access Point Dengan Wemos

Difinisi access point adalah sebagai pengatur lalu lintas data, dimana dapat menghubungkan beberapa perangkat dalam sebuah akses, yang memungkinkan banyak client dapat saling berhubungan dan tukar menukar data. Contoh program diambil dari website (atau contoh aplikasi di Arduino). <https://github.com/esp8266/Arduino/blob/master/libraries/ESP8266WiFi/examples/WiFiAccessPoint/WiFiAccessPoint.ino>

```
/*
```

```
    Copyright (c) 2015, Majenko Technologies
    All rights reserved.
    Redistribution and use in source and binary
    forms, with or without modification,
    are permitted provided that the following
    conditions are met:
    * * Redistributions of source code must retain
    the above copyright notice, this
    list of conditions and the following
    disclaimer.
    * * Redistributions in binary form must reproduce
    the above copyright notice, this
    list of conditions and the following
    disclaimer in the documentation and/or
    other materials provided with the
    distribution.
    * * Neither the name of Majenko Technologies nor
    the names of its
    contributors may be used to endorse or
    promote products derived from
    this software without specific prior written
    permission.
    THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT
    HOLDERS AND CONTRIBUTORS "AS IS" AND
    ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING,
    BUT NOT LIMITED TO, THE IMPLIED
    WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
    PARTICULAR PURPOSE ARE
    DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
    HOLDER OR CONTRIBUTORS BE LIABLE FOR
    ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
    EXEMPLARY, OR CONSEQUENTIAL DAMAGES
```

```

        (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
        SUBSTITUTE GOODS OR SERVICES;
        LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
        INTERRUPTION) HOWEVER CAUSED AND ON
        ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
        STRICT LIABILITY, OR TORT
        (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN
        ANY WAY OUT OF THE USE OF THIS
        SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
        SUCH DAMAGE.
*/

```

```

/* Create a WiFi access point and provide a web
server on it. */

```

```

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

```

```

/* Set these to your desired credentials. */
const char *ssid = "ESPap";
const char *password = "thereisnospoon";

```

```

ESP8266WebServer server(80);

```

```

/* Just a little test message. Go to
http://192.168.4.1 in a web browser
connected to this access point to see it.
*/

```

```

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

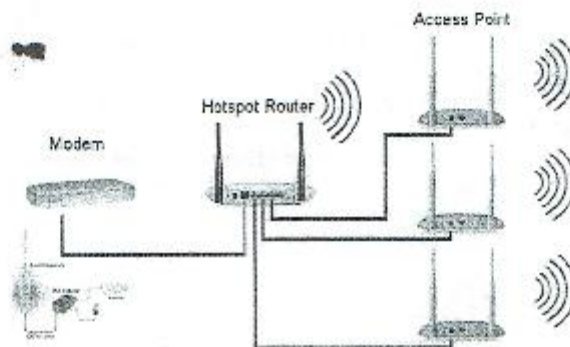
```

```

void setup() {
    delay(1000);
    Serial.begin(115200);
    Serial.println();
    Serial.print("Configuring access point...");
    /* You can remove the password parameter if you
want the AP to be open. */
    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();  
}
```



Gambar 5.4. Access Point (<http://sewuweb.com/cara-kerja-dan-fungsi-access-point/>)

5.3. Wifi Web Server dengan Wemos

Web server merupakan sebuah software layanan berbasis data serta mempunyai fungsi menerima permintaan dari HTTP atau HTTPS pada klien yang dikenal dan biasanya kita kenal dengan nama web browser (Mozilla Firefox, Google Chrome) dan untuk mengirimkan kembali yang hasilnya dalam bentuk beberapa halaman web dan pada umumnya akan berbentuk dokumen HTML. (<https://idcloudhost.com/pengertian-web-server-dan-fungsinya/>)

Dalam program wifi webserver ini yang diterima oleh modul wemos sebagai server adalah permintaan http request. Dengan memberikan no IP dan permintaan data yang diinginkan. Contoh berikut ini berupa permintaan http untuk menghidup matikan lampu yang terhubung internal board Wemos tersebut.

```

/*
   This sketch demonstrates how to set up a simple
   HTTP-like server.
   The server will set a GPIO pin depending on the
   request
   http://server_ip/gpio/0 will set the GPIO2
   low,
   http://server_ip/gpio/1 will set the GPIO2
   high
   server_ip is the IP address of the ESP8266
   module, will be
   printed to Serial when the module is connected.
*/
#include <ESP8266WiFi.h>

const char* ssid = "your-ssid";
const char* password = "your-password";

// Create an instance of the server
// specify the port to listen on as an argument
WiFiServer server(80);

void setup() {
  Serial.begin(115200);
  delay(10);

  // prepare GPIO2
  pinMode(2, OUTPUT);
  digitalWrite(2, 0);

  // Connect to WiFi network
  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);

```

```
while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");

// Start the server
server.begin();
Serial.println("Server started");

// Print the IP address
Serial.println(WiFi.localIP());
}

void loop() {
  // Check if a client has connected
  WiFiClient client = server.available();
  if (!client) {
    return;
  }

  // Wait until the client sends some data
  Serial.println("new client");
  while (!client.available()) {
    delay(1);
  }

  // Read the first line of the request
  String req = client.readStringUntil('\r');
  Serial.println(req);
  client.flush();

  // Match the request
  int val;
  if (req.indexOf("/gpio/0") != -1) {
    val = 0;
  } else if (req.indexOf("/gpio/1") != -1) {
    val = 1;
  } else {
    Serial.println("invalid request");
    client.stop();
    return;
  }

  // Set GPIO2 according to the request
  digitalWrite(2, val);
}
```

```

    client.flush();

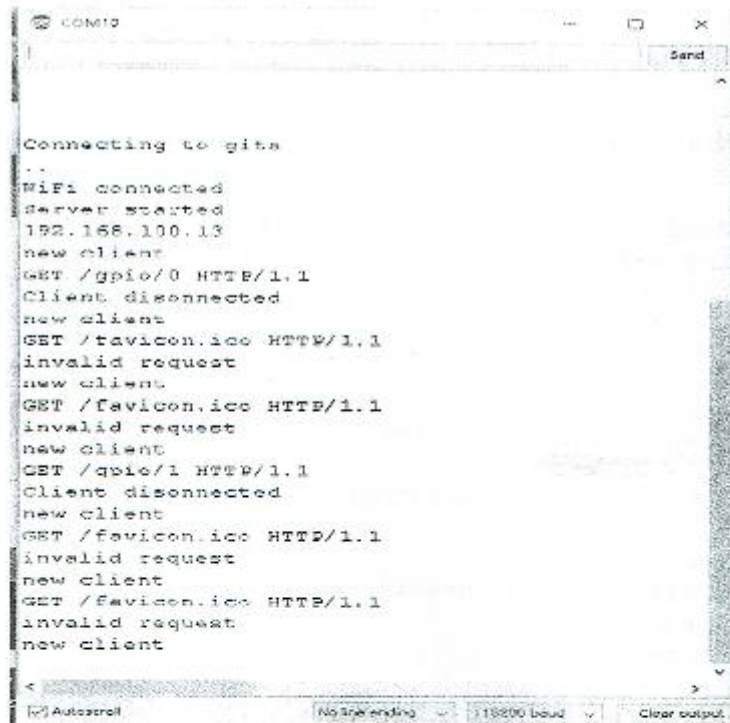
    // Prepare the response
    String s = "HTTP/1.1 200 OK\r\nContent-Type:
text/html\r\n\r\n<!DOCTYPE HTML>\r\n<html>\r\nGPIO
is now ";
    s += (val) ? "high" : "low";
    s += "</html>\n";

    // Send the response to the client
    client.print(s);
    delay(1);
    Serial.println("Client disconnected");

    // The client will actually be disconnected
    // when the function returns and 'client' object
    is destroyed
}

```

Hasil percobaan untuk http request dari program diatas adalah sebagai berikut :



```

COM10
Connecting to gita
..
WiFi connected
Server started
192.168.100.13
new client
GET /gpio/0 HTTP/1.1
Client disconnected
new client
GET /favicon.ico HTTP/1.1
invalid request
new client
GET /favicon.ico HTTP/1.1
invalid request
new client
GET /gpio/1 HTTP/1.1
Client disconnected
new client
GET /favicon.ico HTTP/1.1
invalid request
new client
GET /favicon.ico HTTP/1.1
invalid request
new client

```

Gambar 5.5 Hasil Percobaan Http Request

5.4. Wifi Client

Modul Wemos ini selain dapat berfungsi seperti yang telah dijelaskan diatas dapat juga berfungsi sebagai client. Client adalah komputer dalam jaringan yang menggunakan sumber daya yang disediakan oleh server. Lebih jelasnya Client adalah Pemakai layanan. Pada prinsipnya Client dan server merupakan suatu system yang merupakan aplikasi jaringan komputer yang saling terhubung. (<https://www.temukanpengertian.com/2013/01/pengertian-client.html>)

Sketsa ini mengirim data melalui permintaan GET HTTP ke layanan data.sparkfun.com. Anda perlu mendapatkan streamId dan privateKey di data.sparkfun.com dan tempelkan di bawah. Atau hanya menyesuaikan skrip ini untuk berbicara dengan server HTTP lain.

```
/*  
    This sketch sends data via HTTP GET requests to  
    data.sparkfun.com service.  
  
    You need to get streamId and privateKey at  
    data.sparkfun.com and paste them  
    below. Or just customize this script to talk to  
    other HTTP servers.  
*/  
  
#include <ESP8266WiFi.h>  
  
const char* ssid      = "your-ssid";  
const char* password = "your-password";  
  
const char* host = "data.sparkfun.com";  
const char* streamId = ".....";  
const char* privateKey = ".....";
```

```
void setup() {
  Serial.begin(115200);
  delay(10);

  // We start by connecting to a WiFi network

  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  /* Explicitly set the ESP8266 to be a WiFi-client,
  otherwise, it by default,
  would try to act as both a client and an
  access-point and could cause
  network-issues with your other WiFi-devices on
  your WiFi-network. */
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

int value = 0;

void loop() {
  delay(5000);
```

```
++value;

Serial.print("connecting to ");
Serial.println(host);

// Use WiFiClient class to create TCP connections
WiFiClient client;
const int httpPort = 80;
if (!client.connect(host, httpPort)) {
  Serial.println("connection failed");
  return;
}

// We now create a URI for the request
String url = "/input/";
url += streamId;
url += "?private_key=";
url += privateKey;
url += "&value=";
url += value;

Serial.print("Requesting URL: ");
Serial.println(url);

// This will send the request to the server
client.print(String("GET ") + url + "
HTTP/1.1\r\n" +
             "Host: " + host + "\r\n" +
             "Connection: close\r\n\r\n");
unsigned long timeout = millis();
while (client.available() == 0) {
  if (millis() - timeout > 5000) {
    Serial.println(">>> Client Timeout!");
    client.stop();
    return;
  }
}
```

```
}  
  
    // Read all the lines of the reply from server and  
    print them to Serial  
    while (client.available()) {  
        String line = client.readStringUntil('\r');  
        Serial.print(line);  
    }  
  
    Serial.println();  
    Serial.println("closing connection");  
}
```

BAB VI

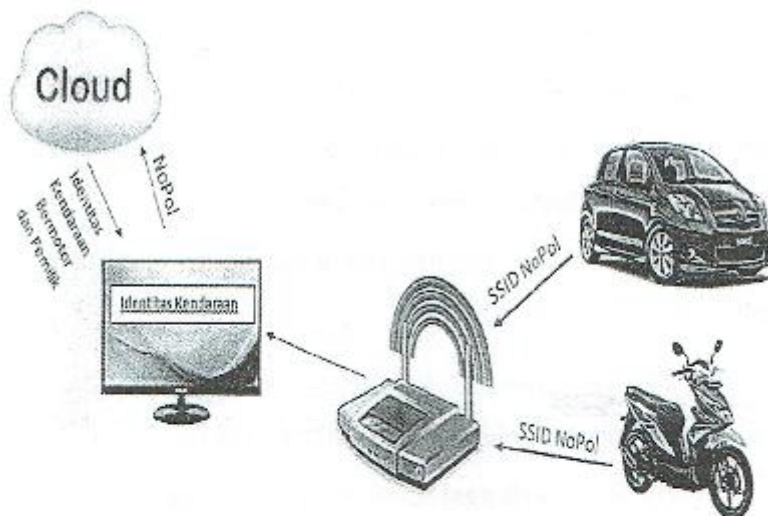
IDENTIFIKASI KENDARAAN BERMOTOR SECARA DIGITAL

Sistem identifikasi kendaraan bermotor ini menggunakan sebuah chip yang ditanamkan dalam sebuah kendaraan bermotor untuk melengkapi identitas kendaraan secara digital. Sistem yang dibangun ini terdiri dari tiga tahapan mulai dari client sebagai identitas kendaraan, scanner sebagai penangkap identitas kendaraan, dan web service untuk menampilkan identitas kendaraan di server dan sekaligus menyimpan data kendaraan bermotor.

Client sebagai identitas kendaraan dibangun menggunakan chip wemos yang didalam chip tersebut di tanamkan sebuah program dan data, program yang berfungsi untuk membaca data yang tersimpan dan memancarkan lewat sinyal wifi setiap kendaraan bermotor tersebut dinyalakan. Data yang tersimpan dalam sebuah chip tersebut berupa data identitas kendaraan yang tertuang dalam STNK setiap kendaraan bermotor, sehingga kendaraan tersebut dapat diidentifikasi secara

digital oleh system komputerisasi yang terkoneksi dengan jaringan internet.

Scanner sebagai penangkap sinyal SSID wifi kendaraan bermotor dibangun juga menggunakan chip mikrokontroler Wemos namun dilengkapi dengan sebuah display LCD untuk menampilkan informasi nomor kendaraan bermotor yang terdeteksi dan terkoneksi ke scanner tersebut. Scanner ini akan selalu aktif untuk menerima sinyal dari client dan menampilkan di LCD , kapan kendaraan bermotor itu masuk dan kapan keluar (meninggalkan) lokasi yang terpasang scanner. Scanner ini akan menampilkan plat nomor yang terakhir terdeteksi saja, yang selanjutnya akan meneruskan datanya ke server untuk dicatat dan disimpan dalam database.



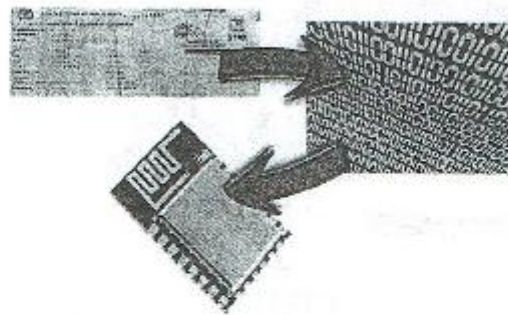
Gambar 7.1. Sistem Identifikasi Kendaraan Bermotor

Web Service (Server) sebagai pusat data yang akan menerima informasi dari scanner berupa data nomor kendaraan yang melewati scanner tersebut. Server akan mencatat dan menyimpan nomor polisi

kendaraan bermotor tersebut dalam sebuah database, sekaligus akan mencatat tanggal dan jam kendaraan bermotor tersebut berada disekitar scanner tersebut. Jika terjadi sesuatu terhadap kendaraan tersebut dapat dicek di database komputer tersebut.

6.1. Client Identifikasi Kendaraan Bermotor

Pada kendaraan bermotor akan dipasang sebuah chip Wemos yang berisi plat nomor kendaraan dan sekaligus berfungsi sebagai SSID kendaraan bermotor tersebut. Pada saat kendaraan bermotor dinyalakan maka secara otomatis modul client ini akan bekerja karena mendapat supply tegangan dari accu kendaraan bermotor dan akan memancarkan SSID tersebut dengan sinyal wifi dari modul wemos tersebut. Yang pertama data diinputkan ke dalam chip tersebut dan disimpan dalam sebuah chip. Untuk memasukkan data dan program client tersebut menggunakan software IDE Arduino.



Gambar 7.2. Digitalisasi Data STNK Kendaraan Bermotor

Program untuk client yang dipasang pada kendaraan bermotor, contoh program tersebut dipasang pada sebuah sepeda motor.

```
#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 Q2";
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);

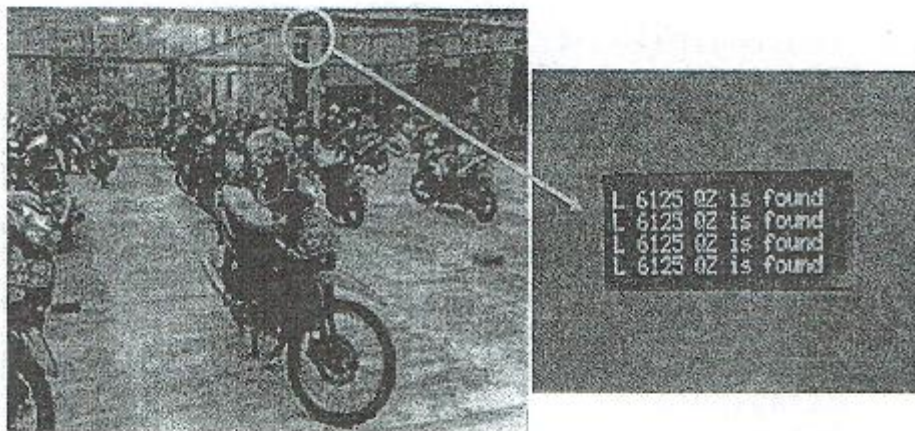
  IPAddress myIP = WiFi.softAPIP();
  Serial.print("AP IP address: ");
  Serial.println(myIP);
  server.on("/", handleRoot);
  server.begin();
  Serial.println("HTTP server started");
}

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

6.2. Scanner Identifikasi Kendaraan Bermotor

Untuk menangkap sinyal wifi kendaraan bermotor dengan SSID masing-masing diperlukan sebuah alat, yang disebut dalam

system ini adalah scanner. Scanner ini akan selalu siap bila di sekitar terdapat sebuah client yang terpasang pada kendaraan bermotor dan memancarkan sinyal wifi dengan SSID nomor polisi kendaraan bermotor tersebut dan secara otomatis akan merekam nomor polisi tersebut dan menampilkannya dalam sebuah display LCD dengan ukuran 128X64 pixel dan juga meneruskan ID kendaraan bermotor tersebut ke server untuk disimpan menjadi sebuah database identifikasi kendaraan bermotor. Hasil tampilan LCD pada scanner dalam menangkap nomor polisi kendaraan bermotor dan menampilkannya.



Gambar 7.3. Scanner dan Display LCD

Untuk program scanner identitas kendaraan bermotor dan menampilkan di LCD sebagai berikut.

```
#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_nopcl;

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_nopcl = "";
```

```
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/api/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);  
}
```

6.3. Web Service Identifikasi Kendaraan Bermotor

Untuk menampung data kendaraan bermotor diperlukan sebuah komputer server, sekaligus menampilkan data plat nomor di web sehingga memudahkan dalam proses pengecekan identifikasi kendaraan bermotor, karena dapat dilakukan dimana saja dan kapan pun. SSID kendaraan bermotor yang dikirimkan oleh scanner diterima oleh server dan kemudian diidentifikasi kendaraan tersebut dalam sebuah web service dan hasil identifikasi ditampilkan dalam sebuah halaman web. Tampilan identifikasi kendaraan di web service sebagai berikut:

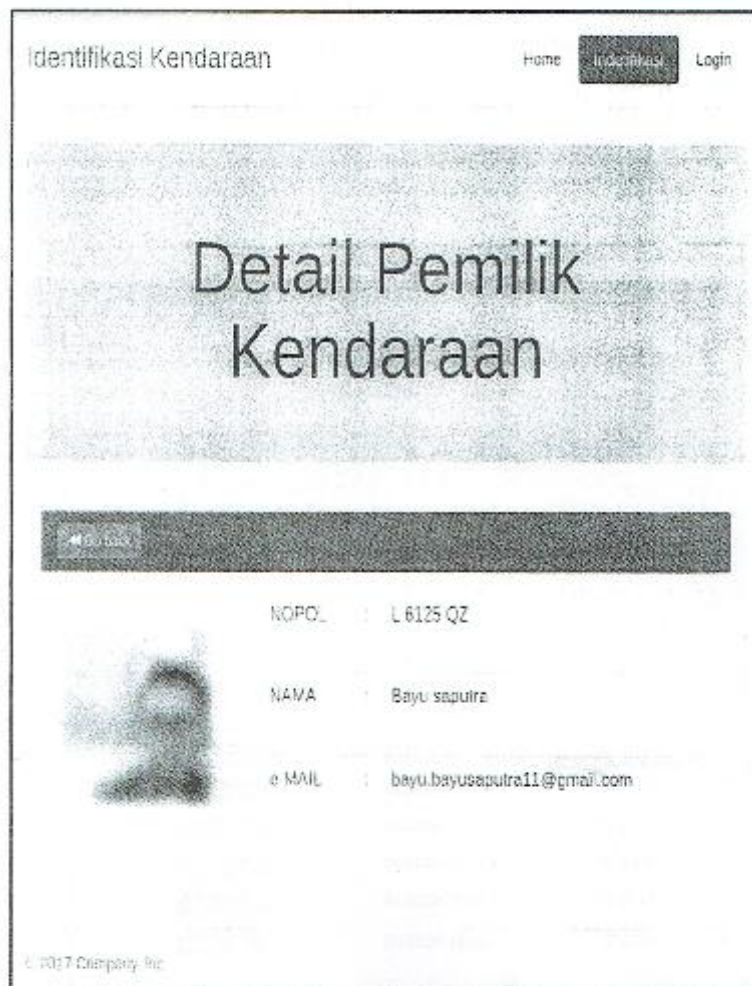


The screenshot shows a web page titled "Identifikasi Kendaraan". At the top right, there are links for "Home", "Identifikasi", and "Login". The main heading reads "Daftar Kendaraan yang teridentifikasi". Below this is a table with the following data:

NO	NOPOL	LAST SEEN	DETAIL
1	L 6125 QZ	2018-02-21 00:20:12	Detail Kendaraan
2	L 6125 QZ	2018-02-21 00:26:27	Detail Kendaraan
3	L 6125 QZ	2018-02-21 00:26:28	Detail Kendaraan
4	L 6125 QZ	2018-02-21 00:26:28	Detail Kendaraan
5	L 6125 QZ	2018-02-21 00:26:29	Detail Kendaraan
6	L 6125 QZ	2018-02-21 00:26:30	Detail Kendaraan

Gambar 7.3. Tampilan di Web Service

Bila ingin ditampilkan detail pemilik kendaraan dapat diklik tombol "Detail Pengguna", Sistem akan menampilkan data pemilik kendaraan beserta foto dan lainnya. Program didesain menggunakan PHP dan databasenya menggunakan MySQL. Berikut tampilan detail pemilik kendaraan bermotor.



Gambar 7.3. Tampilan Data Pemilik Kendaraan Bermotor

Program Webservice untuk identifikasi kendaraan bermotor seperti pada list berikut ini.

```
<?php

namespace App\Http\Controllers;

use Illuminate\Foundation\Bus\DispatchesJobs;
use Illuminate\Routing\Controller as
BaseController;
use
Illuminate\Foundation\Validation\ValidatesReque
sts;
use
Illuminate\Foundation\Auth\Access\AuthorizesReq
uests;

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;
```



```
        $status = $this->retrofitResponse("success", "success", 200, false);
        return response()->json(array("status" => $status, "data" => Pengguna::all()), 200);
    }

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

    public function store(Request $request) {
        $pengguna = Pengguna::create($request->all());
        $status = $this->retrofitResponse("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->retrofitResponse("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

- Direktorat Jendral Hubungan Darat Kementrian Perhubungan Republik Indonesia, "*Lalu Lintas dan Angkutan Jalan*" <http://hubdat.dephub.go.id/uu/288-uu-nomor-22-tahun-2009-tentang-lalu-lintas-dan-angkutan-jalan>, diakses Mei 2018
- Direktorat Jendral Hubungan Darat Kementrian Perhubungan Republik Indonesia, "*Pemeriksaan Kendaraan Bermotor di Jalan*", <http://hubdat.dephub.go.id/peraturan-pemerintah/81-pp-no-42-tahun-1993-ttg-pemeriksaan-kendaraan-bermotor-di-jalan>, diakses Mei 2018
- Ivan Grokhotkov, "*ESP8266 Arduino Core Documentation Release 2.4.0*", May 14, 2017, https://media.readthedocs.org/pdf/arduino-esp8266/docs_to_readthedocs/arduino-esp8266.pdf
- 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., Prasetijo, J., Harmanto, D., Chip Number Vehicle Applications as Part of Internet of Thing (IoT), *Sustainable Transportat*
- 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.
- Neil Kolban, "ESP8266", Texas, USA, Nopember 2016, <http://neilkolban.com/tech/esp8266/>
- 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
- Smartkomputer lab, "Guide to practical classes in IoT design and integration for students in Komputer Science", <http://www.smartkomputerlab.org/M1/IoT.practical.guide.pdf>, akses 2018
- Steve Rackley, "Wireless Networking Technology : From Principles to Succesful Implementation", Elsevier, Great Britain, 2007, ISBN 13: 978-0-7506-6788-3
- Winardi, S., Kamisutara, M., Purworusmiardi, T., Sukoco, A., Mudjanarko, S. W., "Internet of Things (IoT) as Green City Economic Development Smart Transportation System", *MATEC Web of Conferences*, (2017), EDP Sciences , Volume 138, pages 07015

Sumber Website/Internet

<http://wahyusmekers.blogspot.co.id/>, 2011, "Pengetahuan Tentang Kendaraan Bermotor"

<https://www.temukanpengertian.com/2013/01/pengertian-client.html>, "Pengertian Client "

[https://id.wikipedia.org/wiki](https://id.wikipedia.org/wiki/Kendaraan_Bermotor), "Kendaraan Bermotor"

<https://github.com/esp8266/Arduino>, "ESP8266 core for Arduino"

http://arduino.esp8266.com/stable/package_esp8266com_index.json

<http://scwuweb.com/cara-kerja-dan-fungsi-access-point/>, "Cara Kerja dan Fungsi Access Point"

<https://idcloudhost.com/pengertian-web-server-dan-fungsinya/>, "Pengertian Web Server dan Fungsinya"

<http://www.arduino.cc>, "Arduino Reference"



Universitas Narotama Surabaya