

BAB II

TINJAUAN PUSTAKA

2.1 Pengertian *NodeMCU*

NodeMCU adalah sebuah platform IoT yang bersifat *opensource*. Terdiri dari perangkat keras berupa *System On Chip ESP8266*. dari *ESP8266* buatan *Espressif System*, juga *firmware* yang digunakan, yang menggunakan bahasa pemrograman *scripting* Lua. [Sumardi, 2016] Istilah *NodeMCU* secara *default* sebenarnya mengacu pada firmware yang digunakan dari pada perangkat keras development kit *NodeMCU* bisa dianalogikan sebagai board arduino-nya *ESP8266*.

Sejarah lahirnya *NodeMCU* berdekatan dengan rilis *ESP8266* pada 30 Desember 2013, *Espressif Systems* selaku pembuat *ESP8266* memulai produksi *ESP8266* yang merupakan SoC Wi-Fi yang terintegrasi dengan prosesor Tensilica Xtensa LX106. Sedangkan *NodeMCU* dimulai pada 13 Oktober 2014 saat Hong me-commit file pertama *nodemcu-firmware* ke Github. Dua bulan kemudian project tersebut dikembangkan ke platform perangkat keras ketika Huang R meng-commit file dari board *ESP8266* , yang diberi nama *devkit v.0.9*.

Berikutnya, di bulan yang sama. Tuan PM memporting pustaka *client MQTT* dari Contiki ke platform SOC *ESP8266* dan di-commit ke project *NodeMCU* yang membuatnya mendukung protokol IOT MQTT melalui Lua. Pemutakhiran penting berikutnya terjadi pada 30 Januari 2015 ketika Devsaurus memporting *u8glib* ke project *NodeMCU* yang memungkinkan *NodeMCU* bisa mendrive display *LCD*, *OLED*, hingga *VGA*. Demikianlah, *project NodeMCU* terus berkebang hingga kini berkat komunitas open source dibalikny, pada musim panas 2016 *NodeMCU* sudah terdiri memiliki 40 modul fungsionalitas yang bisa digunakan sesuai kebutuhan developer.



Gambar 2.1 NodeMCU

(Sumber : <https://www.google.com/>)

Karena jantung dari *NodeMCU* adalah *ESP8266* (khususnya seri *ESP-12*, termasuk *ESP-12E*) maka fitur – fitur yang dimiliki *NodeMCU* akan kurang lebih sama *ESP-12* (juga *ESP-12E* untuk *NodeMCU v.2* dan *v.3*) kecuali *NodeMCU* telah dibungkus oleh *API* sendiri yang dibangun berdasarkan bahasa pemrograman *eLua*, yang kurang lebih cukup mirip dengan *javascript*. Beberapa fitur tersebut antara lain

1. 10 Port GPIO dari D0 – D10
2. Fungsionalitas PWM
3. Antarmuka I2C dan SPI
4. Antarmuka 1 Wire
5. ADC

Modul ini membutuhkan daya sekitar 3.3v dengan memiliki tiga mode *wifi* yaitu *Station*, *Access Point* dan *Both* (Keduanya). Modul ini juga dilengkapi dengan prosesor, memori dan GPIO dimana jumlah pin bergantung dengan jenis *ESP8266* yang kita gunakan. Sehingga modul ini bisa berdiri sendiri tanpa menggunakan mikrokontroler apapun karena sudah memiliki perlengkapan layaknya mikrokontroler.

2.1.1 Versi *NodeMCU*

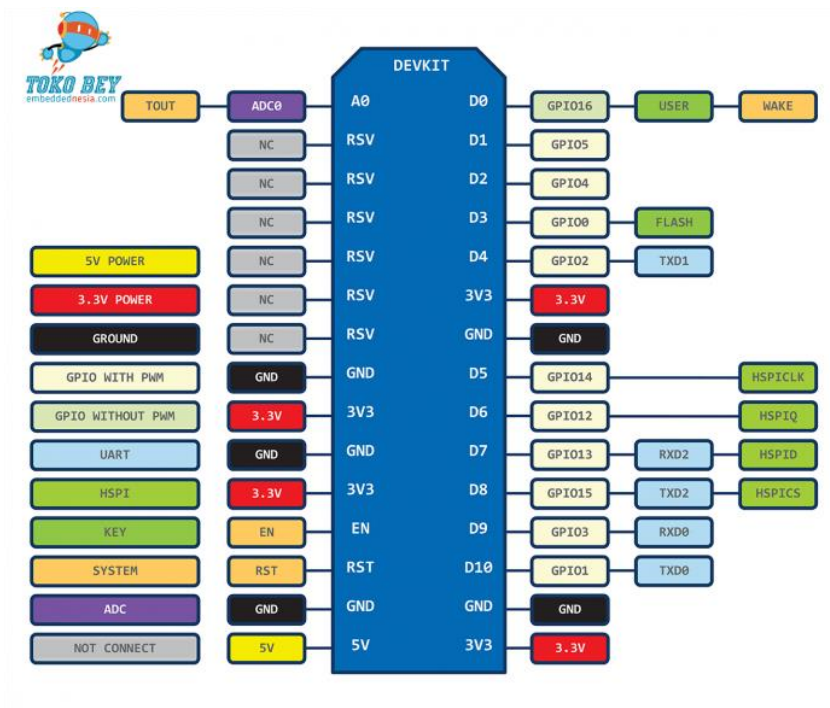
Beberapa pengguna awal masih cukup bingung dengan beberapa kehadiran board *NodeMCU*. Karena sifatnya yang *open source* tentu akan banyak produsen yang memproduksinya dan mengembangkannya. Secara umum ada tiga produsen *NodeMCU* yang produknya kini beredar di pasaran: Amica, DOIT, dan Lolin/WeMos. Dengan beberapa varian board yang diproduksi yakni V1, V2 dan V3.

a. Generasi pertama / board v.0.9 (Biasa disebut V1)



Gambar 2.2 Generasi Pertama *NodeMCU*
(Sumber : <https://tutorkeren.com/>)

Board versi 0.9 sering disebut di pasar sebagai V.1 adalah versi asli yang berdimensi 47mm x 31mm. Memiliki inti ESP-12 dengan flash memory berukuran 4MB. Berikut adalah pinout dari board v.0.9



Gambar 2.3 Skematik Posisi Pin *NodeMCU* Devkit V1
(Sumber : <https://tutorkeren.com/>)

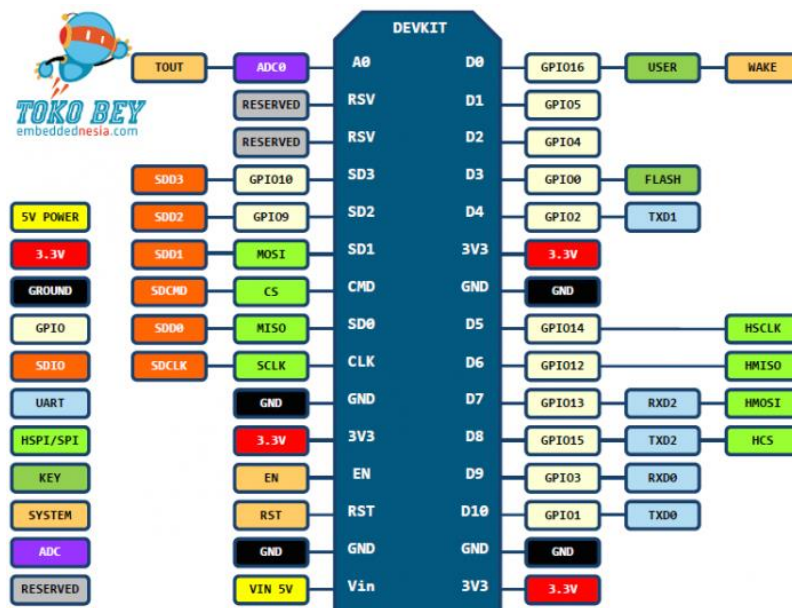
Namun beberapa produk juga ada yang menggunakan chip ESP-12E sebagai inti dari board v.0.9 dengan tampilan board berubah menjadi hitam.

b. Generasi kedua / board v 1.0 (biasa disebut V2)



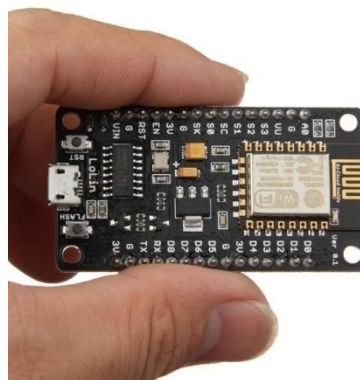
Gambar 2.4 *NodeMCU* Devkit V2
(Sumber : <https://tutorkeren.com/>)

Generasi kedua adalah pengembangan dari versi sebelumnya, dengan chip yang ditingkatkan dari sebelumnya ESP12 menjadi ESP12E. Dan IC Serial diubah dari CHG340 menjadi CP2102



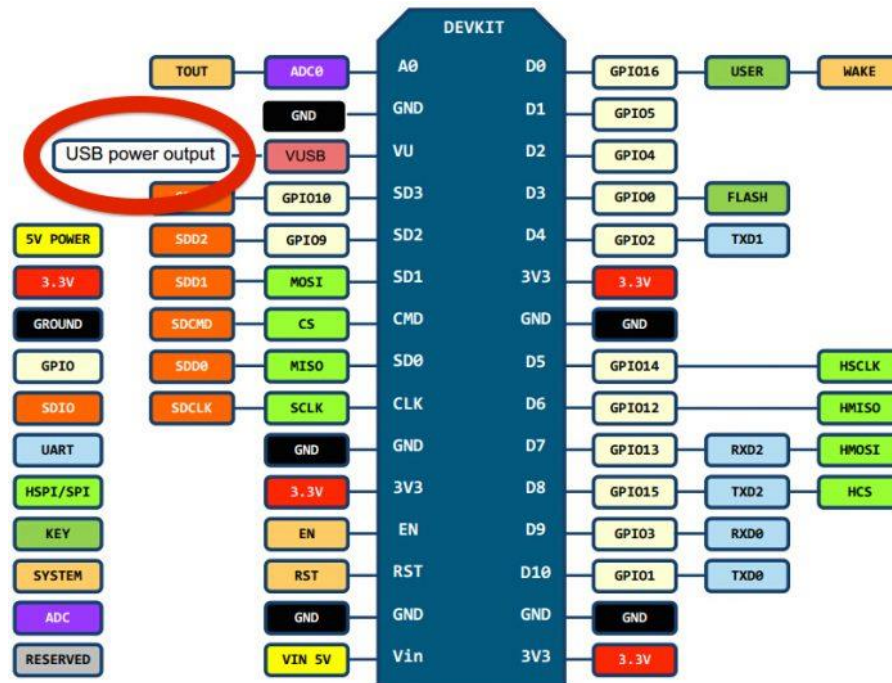
Gambar 2.5 Skematik Posisi Pin NodeMCU Dekvit V2
(Sumber : <https://tutorkeren.com/>)

c. Generasi ketiga / *board v 1.0* (biasa disebut V3 Lolin)



Gambar 2.6 NodeMCU Dekvit V3
(Sumber : <https://tutorkeren.com/>)

Sedangkan untuk V3 sebenarnya bukanlah versi resmi yang dirilis oleh *NodeMCU*. Setidaknya sampai posting ini dibuat, belum ada versi resmi untuk V3 *NodeMCU*. V3 hanyalah versi yang diciptakan oleh produsen LoLin dengan perbaikan minor terhadap V2. Diklaim memiliki antarmuka *USB* yang lebih cepat.



Gambar 2.7 Skematik Posisi Pin *NodeMCU* Dekvit V3
(Sumber : <https://tutorkeren.com/>)

Jika anda bandingkan dengan versi sebelumnya, dimensi dari board V3. akan lebih besar dibanding V2. Lolin menggunakan 2 pin cadangan untuk daya *USB* dan yang lain untuk GND tambahan.

Tentu 3 jenis versi ini akan berkembang dan bertambah seiring dengan waktu karena sifatnya yang *opensource*. Mungkin beberapa bulan atau beberapa tahun setelah tulisan ini dibuat akan muncul versi- versi lain yang beredar.

2.2 Sensor Ultrasonik

Sensor ultrasonik adalah sebuah sensor yang memiliki fungsi untuk mengubah besaran fisis alias bunyi menjadi besaran listrik, begitupun sebaliknya. [Yakob Lilikwati, 2012] Prinsip kerja sensor ultrasonic ini cukup simpel, yakni berdasarkan pantulan suatu gelombang suara sehingga dapat digunakan untuk mendefinisikan eksistensi suatu jarak suatu benda dengan frekuensi tertentu. Gelombang ultrasonic sendiri memiliki frekuensi yang sangat tinggi, mencapai 20.000 Hz yang tidak bisa didengar oleh telinga manusia. Bunyi dengan frekwensi setinggi itu hanya bisa didengar oleh hewan-hewan tertentu seperti kucing, anjing, kelelawar, sampai dengan lumba-lumba.

Bunyi dari sensor ultrasonic sendiri dapat merambat melalui benda padat, cair, atau gas. Namun yang paling bagus adalah benda cair.

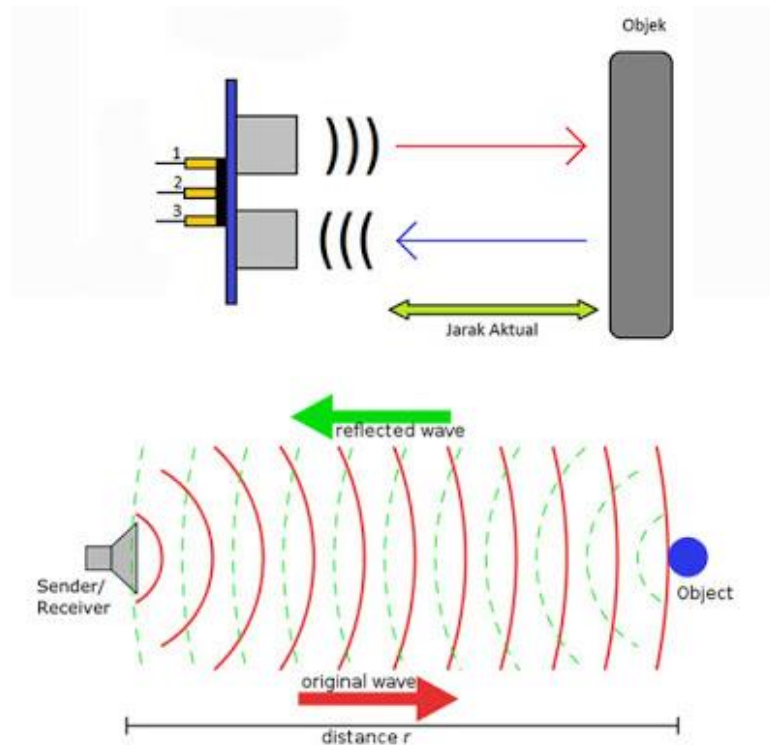


Gambar 2.8 Sensor Ultrasonik HC-SRF04

(Sumber : <https://www.elangsakti.com>)

Pada sensor ultrasonik, gelombang ultrasonik dibangkitkan melalui sebuah alat yang disebut dengan piezoelektrik dengan frekuensi tertentu. Piezoelektrik ini akan menghasilkan gelombang ultrasonik (umumnya berfrekuensi 40kHz) ketika sebuah osilator diterapkan pada benda tersebut. Secara umum, alat ini akan menembakkan gelombang ultrasonik menuju suatu area atau suatu target. Setelah gelombang menyentuh permukaan target, maka target akan memantulkan kembali

gelombang tersebut. Gelombang pantulan dari target akan ditangkap oleh sensor, kemudian sensor menghitung selisih antara waktu pengiriman gelombang dan waktu gelombang pantul diterima.



Gambar 2.9 Gambar Cara Kerja Sensor Ultrasonik Dengan Transmitter dan Receiver

(Sumber : <https://www.elangsakti.com>)

Secara detail, cara kerja sensor ultrasonik adalah sebagai berikut:

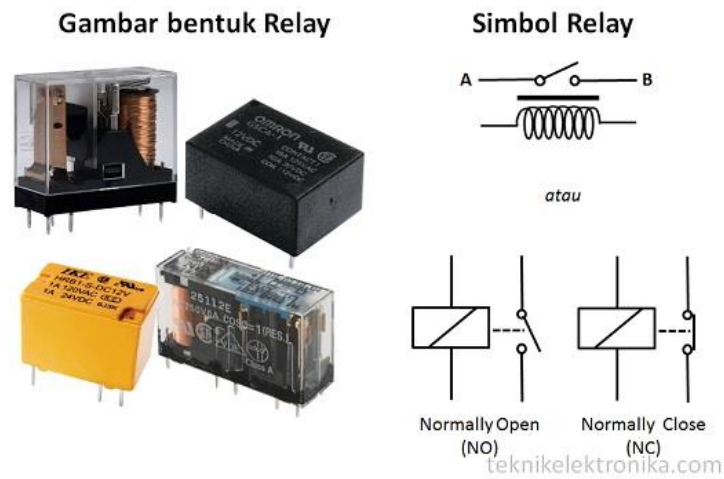
- Sinyal dipancarkan oleh pemancar ultrasonik dengan frekuensi tertentu dan dengan durasi waktu tertentu. Sinyal tersebut berfrekuensi diatas 20kHz. Untuk mengukur jarak benda (sensor jarak), frekuensi yang umum digunakan adalah 40kHz.

- Sinyal yang dipancarkan akan merambat sebagai gelombang bunyi dengan kecepatan sekitar 340 m/s. Ketika menumbuk suatu benda, maka sinyal tersebut akan dipantulkan oleh benda tersebut.
- Setelah gelombang pantulan sampai di alat penerima, maka sinyal tersebut akan diproses untuk menghitung jarak benda tersebut. Jarak benda dihitung berdasarkan rumus : $S = 340.t/2$ (2-1)

dimana S merupakan jarak antara sensor ultrasonik dengan benda (bidang pantul), dan t adalah selisih antara waktu pemancaran gelombang oleh transmitter dan waktu ketika gelombang pantul diterima receiver.

2.3 *Relay*

Relay adalah Saklar (*Switch*) yang dioperasikan secara listrik dan merupakan komponen *Electromechanical* (Elektromekanikal) yang terdiri dari 2 bagian utama yakni Elektromagnet (*Coil*) dan Mekanikal (seperangkat Kontak Saklar/Switch). [Yakob Lilikwati, 2012] *Relay* menggunakan Prinsip *Elektromagnetik* untuk menggerakkan Kontak Saklar sehingga dengan arus listrik yang kecil (*low power*) dapat menghantarkan listrik yang bertegangan lebih tinggi. Sebagai contoh, dengan *Relay* yang menggunakan Elektromagnet 5V dan 50 mA mampu menggerakkan *Armature Relay* (yang berfungsi sebagai saklarnya) untuk menghantar kan listrik 220V 2A.

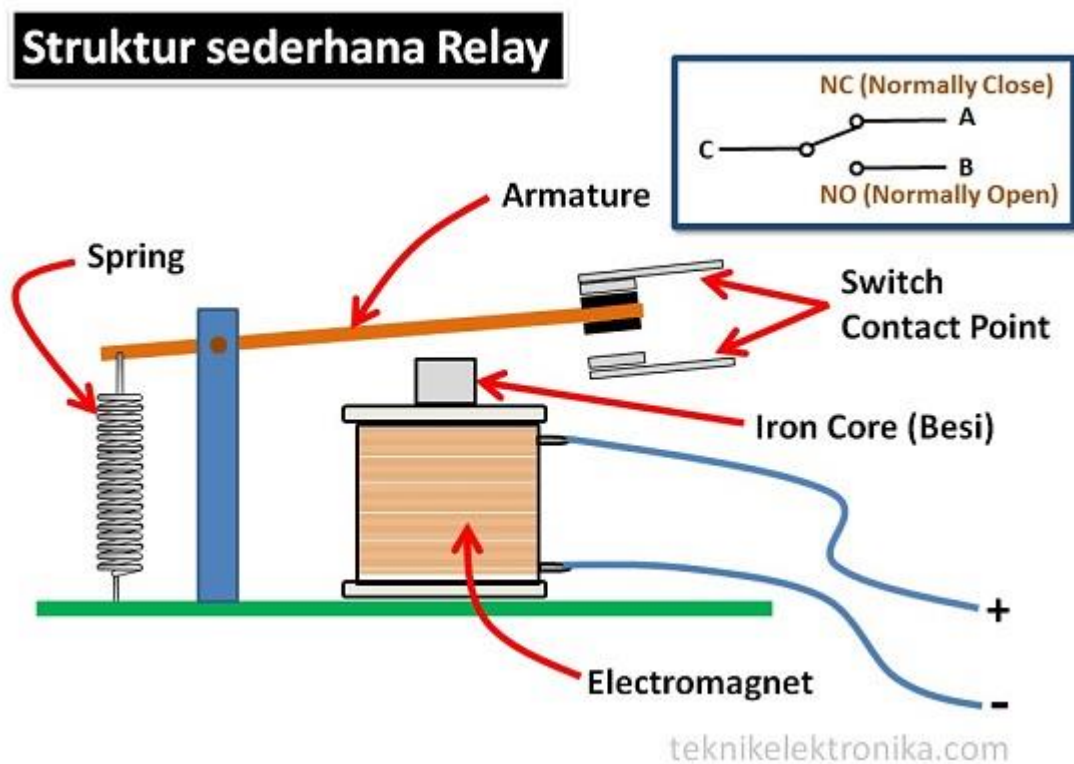


Gambar 2.10 Relay
(Sumber : <https://teknikelektronika.com>)

Pada dasarnya, *Relay* terdiri dari 4 komponen dasar yaitu :

- a) *Electromagnet* (Coil)
- b) *Armature*
- c) *Switch Contact Point* (Saklar)
- d) *Spring*

Berikut ini merupakan gambar dari bagian-bagian Relay :Struktur dasar Relay



Gambar 2.11 Struktur Sederhana Relay
 (Sumber : <https://teknikelektronika.com>)

Kontak Poin (Contact Point) Relay terdiri dari 2 jenis yaitu :

- *Normally Close (NC)* yaitu kondisi awal sebelum diaktifkan akan selalu berada di posisi *CLOSE* (tertutup)
- *Normally Open (NO)* yaitu kondisi awal sebelum diaktifkan akan selalu berada di posisi *OPEN* (terbuka)

Berdasarkan gambar diatas, sebuah Besi (*Iron Core*) yang dililit oleh sebuah kumparan *Coil* yang berfungsi untuk mengendalikan Besi tersebut. Apabila Kumparan *Coil* diberikan arus listrik, maka akan timbul gaya Elektromagnet yang kemudian menarik *Armature* untuk berpindah dari Posisi sebelumnya (*NC*) ke posisi baru (*NO*) sehingga menjadi Saklar yang dapat menghantarkan arus listrik di posisi

barunya (NO). Posisi dimana Armature tersebut berada sebelumnya (NC) akan menjadi OPEN atau tidak terhubung. Pada saat tidak dialiri arus listrik, Armature akan kembali lagi ke posisi Awal (NC). Coil yang digunakan oleh *Relay* untuk menarik *Contact Poin* ke Posisi *Close* pada umumnya hanya membutuhkan arus listrik yang relatif kecil.

2.4 Aplikasi Telegram

Telegram adalah sebuah aplikasi layanan pengirim pesan instan multiplatform berbasis awan yang bersifat gratis dan nirlaba. Klien Telegram tersedia untuk perangkat telepon seluler (Android, iOS, Windows Phone, Ubuntu Touch) dan sistem perangkat komputer (Windows, OS X, Linux). Para pengguna dapat mengirim pesan dan bertukar foto, video, stiker, audio, dan tipe berkas lainnya. Telegram juga menyediakan pengiriman pesan ujung keujung terenkripsi opsional.

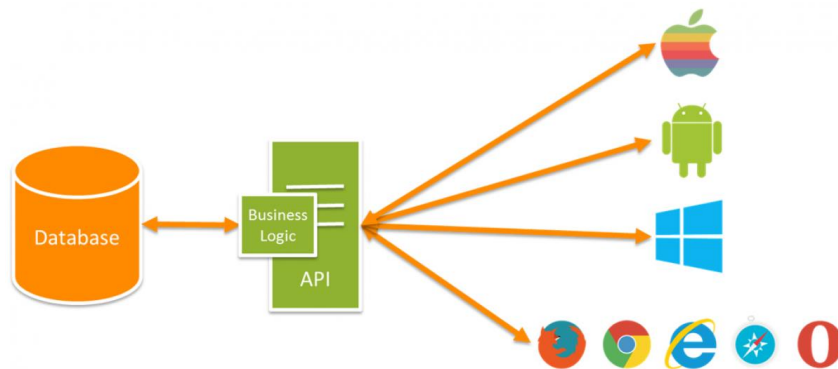
Telegram dikembangkan oleh Telegram Messenger LLP dan didukung oleh wirausahawan Rusia Pavel Durov. Kode pihak kliennya berupa perangkat lunak sistem terbuka namun mengandung blob binari, dan kode sumber untuk versi terbaru tidak selalu segera dipublikasikan, sedangkan kode sisi servernya bersumber tertutup dan berpaten. Layanan ini juga menyediakan API kepada pengembang independen. Pada Februari 2016,



Gambar 2.12 Aplikasi Telegram
(Sumber : <https://www.google.com/telegram>)

2.5 API (*Application Programming Interface*)

API adalah singkatan dari *Application Programming Interface*, dan memungkinkan *developer* untuk mengintegrasikan dua bagian dari aplikasi atau dengan aplikasi yang berbeda secara bersamaan. *API* terdiri dari berbagai elemen seperti *function*, *protocols*, dan *tools* lainnya yang memungkinkan *developers* untuk membuat aplikasi. [Abdul Kadir, 2016] Tujuan penggunaan *API* adalah untuk mempercepat proses *development* dengan menyediakan *function* secara terpisah sehingga *developer* tidak perlu membuat fitur yang serupa. Penerapan *API* akan sangat terasa jika fitur yang diinginkan sudah sangat kompleks, tentu membutuhkan waktu untuk membuat yang serupa dengannya. Misalnya: integrasi dengan *payment gateway*. Terdapat berbagai jenis sistem *API* yang dapat digunakan, termasuk sistem operasi, *library*, dan web.



Gambar 2.13 Sistem API
(Sumber : <https://www.codepolitan.com/api>)

API yang bekerja pada tingkat sistem operasi membantu aplikasi berkomunikasi dengan *layer* dasar dan satu sama lain mengikuti serangkaian protokol dan spesifikasi. Contoh yang dapat menggambarkan spesifikasi tersebut adalah **POSIX** (*Portable Operating System Interface*). Dengan menggunakan standar *POSIX*, aplikasi yang di-*compile* untuk bekerja pada sistem operasi tertentu juga dapat bekerja pada sistem lain yang memiliki kriteria yang sama. *Software library*

juga memiliki peran penting dalam menciptakan *compatibility* antar sistem yang berbeda.

Aplikasi yang berinteraksi dengan *library* harus mengikuti serangkaian aturan yang ditentukan oleh *API*. Pendekatan ini memudahkan *software developer* untuk membuat aplikasi yang berkomunikasi dengan berbagai *library* tanpa harus memikirkan kembali strategi yang digunakan selama semua *library* mengikut *API* yang sama. Kelebihan lain dari metode ini menunjukkan betapa mudahnya menggunakan *library* yang sama dengan bahasa pemrograman yang berbeda.

Seperti namanya, Web *API* dalam diakses melalui protokol **HTTP**, ini adalah konsep bukan teknologi. Kita bisa membuat Web *API* dengan menggunakan teknologi yang berbeda seperti *PHP*, *Java*, *.NET*, dll. Misalnya *Rest API* dari **Twitter** menyediakan akses *read* dan *write* data dengan mengintegrasikan *twitter* kedalam aplikasi kita sendiri.

2.5.1 Fitur Web API

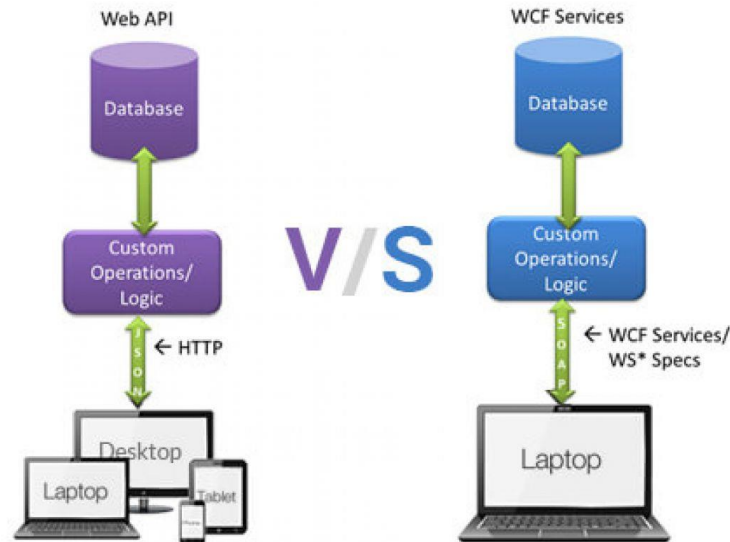
Untuk membuat Web API, beberapa hal yang harus disediakan adalah:

1. Mendukung fungsi **CRUD** yang bekerja melalui *HTTP* protocol dengan method **GET, POST, PUT dan DELETE**
2. Memiliki *response Accept Header* dan *HTTP status code*
3. *Response* dengan format *JSON*, *XML* atau format apapun yang kamu inginkan. Akan tetapi kebanyakan digunakan kedalam format *JSON*.
4. Mendukung fitur MVC seperti *routing*, *controllers*, *action results*, *filter*, *model*, *IOC container*, dll.
5. Web *API* dapat berjalan di *Apache* atau *web server* lainnya yang didukung sesuai bahasa pemrograman yang digunakan.

Web *API* seperti sebuah alamat web (*end point*) yang dibuat untuk menangani beberapa *task* sesuai *request* yang diterima, juga terkadang memiliki *parameter* sebagai data yang dibutuhkan agar dapat menampilkan hasil yang diinginkan, juga pada beberapa kasus untuk mengakses *API* dibutuhkan kode otentikasi yang telah diizinkan untuk melihat data yang diinginkan. Semua *rule* ini ditentukan oleh *programmer* yang membuatnya.

2.5.2 Perbedaan Web API dan Web Service

1. Semua *web service* menggunakan *API* tapi tidak semua *API* digunakan sebagai *web service*
2. *Web service* memfasilitasi untuk melakukan interaksi antara dua perangkat atau aplikasi melalui jaringan. Sedangkan *API* bertindak sebagai penghubung antara dua aplikasi berbeda sehingga bisa berkomunikasi satu sama lain baik dengan ataupun tanpa jaringan.
3. *Web service* hanya menggunakan 3 *style* yaitu *SOAP*, *REST*, atau *XML-RPC* untuk berkomunikasi sedangkan *API* dapat menggunakan *style* apapun.
4. *Web service* selalu membutuhkan jaringan untuk pengoperasiannya sedangkan *API* tidak selalu memerlukan jaringan untuk operasinya.



Gambar 2.14 Perbedaan Web Api dan Web Service
(Sumber : <https://www.codepolitan.com/api>)

2.6 Chatterbot

Chatterbot (disebut juga **chatbot** atau **bots**) adalah sebuah program komputer yang dirancang untuk menyimulasikan percakapan intelektual dengan satu atau lebih manusia baik secara audio maupun teks. [Jubile Enterprise, 2017]

Pada mulanya, program komputer (*bots*) ini diuji melalui Turing Test, yaitu dengan merahasiakan identitasnya sebagai mesin sehingga dapat mengelabui orang yang bercakap-cakap dengannya. Jika pengguna tidak dapat mengidentifikasi *bots* sebagai suatu program komputer, maka *chatterbot* tersebut dikategorikan sebagai kecerdasan buatan (atau *artificial intelligence*). Dewasa ini, *chatterbot* telah dimanfaatkan untuk tujuan praktis seperti bantuan online, layanan personal, atau akuisisi informasi, dalam hal ini dapat dilihat fungsi program sebagai suatu jenis agen percakapan (atau *conversational agent*). Yang membedakan *chatterbot* dengan sistem pemrosesan bahasa alami (atau *Natural Language Processing System*) adalah kesederhanaan algoritme yang digunakan.

Sejarah klasik dari *chatbot* awal adalah ELIZA (1966) dan PARRY (1972). Program yang baru-baru saja dikembangkan yaitu A.L.I.C.E, Jabberwacky dan D.U.D.E. Pada masanya, ELIZA dan PARRY digunakan untuk menstimulasi percakapan tertulis, namun banyak *chatbot* kini mendukung fitur fungsional seperti permainan dan kemampuan pencarian website. Tahun 1984, sebuah buku berjudul *The Policeman's Beard is Half Constructed* dipublikasikan. Buku ini diduga ditulis oleh sebuah *chatbot* Racter –walaupun program ini dirilis untuk tidak mampu melakukannya.

Salah satu penelitian penting di bidang kecerdasan buatan (AI) adalah pemrosesan bahasa alami (atau *Natural Language Processing*). Biasanya, bidang AI lemah memberdayakan perangkat lunak (atau *software*) khusus atau bahasa pemrograman yang dibuat secara spesifik dengan fungsi yang lebih sempit. Contohnya, A.L.I.C.E menggunakan bahasa pemrograman yang disebut sebagai AIML yang fungsinya spesifik yaitu sebagai agen percakapan, yang selanjutnya banyak diadopsi oleh pengembang *Alicebots* lain. Meskipun demikian, A.L.I.C.E masih murni berdasarkan teknik pencocokan pola tanpa kemampuan penalaran – teknik yang sama yang digunakan ELIZA pada tahun 1966. Berbeda dengan AI kuat, yang membutuhkan cita rasa dan kemampuan penalaran logis.

Jabberwacky mempelajari respons baru dan berbasis pada konteks interaksi pengguna waktu nyata (atau *real-time*), bukan dengan digerakan basis data statis. Beberapa *chatbot* terbaru juga mengkombinasikan pembelajaran waktu nyata dengan algoritme evolusioner yang mengoptimalkan kemampuan komunikasi berbasis percakapannya, dengan contoh populernya yaitu Kyle, pemenang Penghargaan Leodis AI 2009. Meskipun, saat ini belum ada tujuan umum percakapan kecerdasan buatan, dan beberapa pengembang perangkat lunak berfokus pada aspek praktis, pengambilan informasi (atau *information retrieval*).

2.7 *Intergrated Development Environment (IDE) Arduino*

IDE merupakan kependekan dari *Integrated Development Environment*, atau secara bahasa mudahnya merupakan lingkungan terintegrasi yang digunakan untuk melakukan pengembangan. [Muhammad Syahwil, 2015] Disebut sebagai lingkungan karena melalui software inilah Arduino dilakukan pemrograman untuk melakukan fungsi-fungsi yang dibenamkan melalui sintaks pemrograman. Arduino menggunakan bahasa pemrograman sendiri yang menyerupai bahasa C. Bahasa pemrograman Arduino (*Sketch*) sudah dilakukan perubahan untuk memudahkan pemula dalam melakukan pemrograman dari bahasa aslinya. Sebelum dijual ke pasaran, IC mikrokontroler Arduino telah ditanamkan suatu program bernama *Bootlader* yang berfungsi sebagai penengahantara *compiler* Arduino dengan mikrokontroler.

Arduino IDE dibuat dari bahasa pemrograman JAVA. Arduino IDE juga dilengkapi dengan library C/C++ yang biasa disebut *Wiring* yang membuat operasi input dan output menjadi lebih mudah. Arduino IDE ini dikembangkan dari software Processing yang dirombak menjadi Arduino IDE khusus untuk pemrograman dengan Arduino.