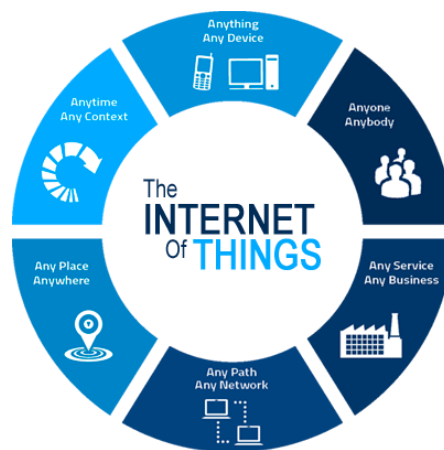


## BAB II

### LANDASAN TEORI

Dikarenakan tugas akhir ini membahas tentang perancangan sistem pengendali relay via internet dengan perintah suara menggunakan Google Assistant, maka dalam dasar teori akan dibahas tentang perangkat keras, perangkat lunak, dan komponen-komponen pendukungnya.

#### 2.1 IoT



Gambar 2.1 Ilustrasi Internet of Things

Internet untuk Segala (bahasa Inggris: *Internet of Things*, atau dikenal juga dengan singkatan IoT) merupakan sebuah konsep yang bertujuan untuk memperluas manfaat dari konektivitas internet yang tersambung secara terus-menerus. Adapun kemampuan seperti berbagi data, remote control, dan sebagainya, termasuk juga pada benda di dunia nyata. Contohnya bahan pangan, elektronik, koleksi, peralatan apa saja, termasuk benda hidup yang semuanya tersambung ke jaringan lokal dan global melalui sensor yang tertanam dan selalu aktif.

Pada dasarnya, IoT mengacu pada benda yang dapat diidentifikasi secara unik sebagai representasi virtual dalam struktur berbasis Internet. Istilah *Internet of Things* awalnya disarankan oleh Kevin Ashton pada tahun 1999 dan mulai terkenal melalui Auto-ID Center di MIT.

Pada bulan Juni 2009 Ashton berkomentar:

*"Hari ini komputer dan manusia, hampir sepenuhnya tergantung pada Internet untuk segala informasi yang semua terdiri dari sekitar 50 petabyte (satu petabyte adalah 1.024 terabyte) data yang tersedia pada Internet dan pertama kali digagaskan dan diciptakan oleh manusia. Dari mulai mengetik, menekan tombol rekam, mengambil gambar digital atau memindai kode bar.*

*Diagram konvensional dari Internet meninggalkan router menjadi bagian terpenting dari semuanya. Masalahnya adalah orang memiliki waktu, perhatian dan akurasi terbatas. Mereka semua berarti tidak sangat baik dalam menangkap berbagai data tentang hal di dunia nyata. Dan itu adalah masalah besar.*

*Dari segi fisik dan begitu juga lingkungan kita. Gagasan dan informasi begitu penting, tetapi banyak lagi hal yang penting. Namun teknologi informasi saat ini sangat tergantung pada data yang berasal dari orang-orang sehingga komputer kita tahu lebih banyak tentang semua ide dari hal-hal tersebut.*

*Jika kita memiliki komputer yang begitu banyak tahu tentang semua hal itu. Menggunakan data yang berkumpul tanpa perlu bantuan dari kita. Kita dapat melacak dan menghitung segala sesuatu dan sangat mengurangi pemborosan, kerugian, dan biaya. Kita akan mengetahui kapan hal itu diperlukan untuk mengganti, memperbaiki atau mengingat, dan apakah mereka menjadi terbaru atau melewati yang terbaik di sini serta nya!.*

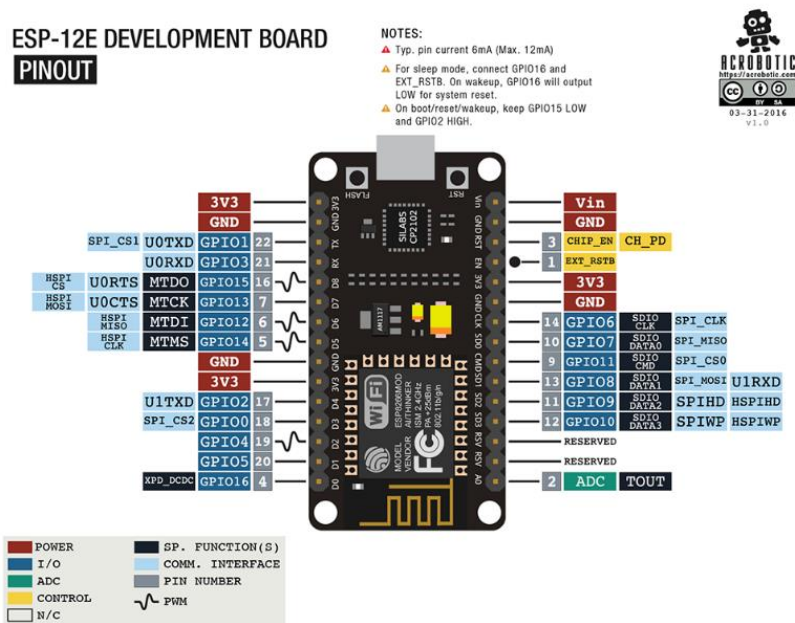
*Internet of Things memiliki potensi untuk mengubah dunia seperti pernah dilakukan oleh Internet, bahkan mungkin lebih baik.*

*Penelitian pada Internet of Things masih dalam tahap perkembangan. Oleh karena itu, tidak ada definisi standar dari Internet of Things. Terdapat juga berbagai definisi yang dirumuskan oleh peneliti yang berbeda serta tercantum dalam survei."*

## **2.2 ESP8266**

ESP8266 adalah chip Wi-Fi berbiaya rendah dengan kemampuan mengatur TCP/IP dan MCU (unit mikrokontroler) yang diproduksi oleh produsen China yang berbasis di Shanghai, Espressif Systems. Chip tersebut pertama kali mendapat

perhatian para pembuat Barat pada Agustus 2014 dengan modul ESP-01, dibuat oleh produsen pihak ketiga, Ai-Thinker. Modul kecil ini memungkinkan mikrokontroler untuk terhubung ke jaringan *Wi-Fi* dan membuat koneksi *TCP/IP* sederhana menggunakan perintah gaya *Hayes*. Namun, pada saat itu hampir tidak ada dokumentasi berbahasa Inggris tentang chip dan perintah yang diterimanya. Harga yang sangat rendah dan fakta bahwa hanya ada sedikit komponen eksternal pada modul yang memberi kesan bahwa pada akhirnya volume bisa sangat murah, menarik banyak *hacker* untuk mengeksplorasi modul, chip, dan perangkat lunak di dalamnya, dan juga untuk menerjemahkan dokumentasi Cina.



Gambar 2.2 ESP8266 Development Kit

Fitur-fiturnya antara lain:

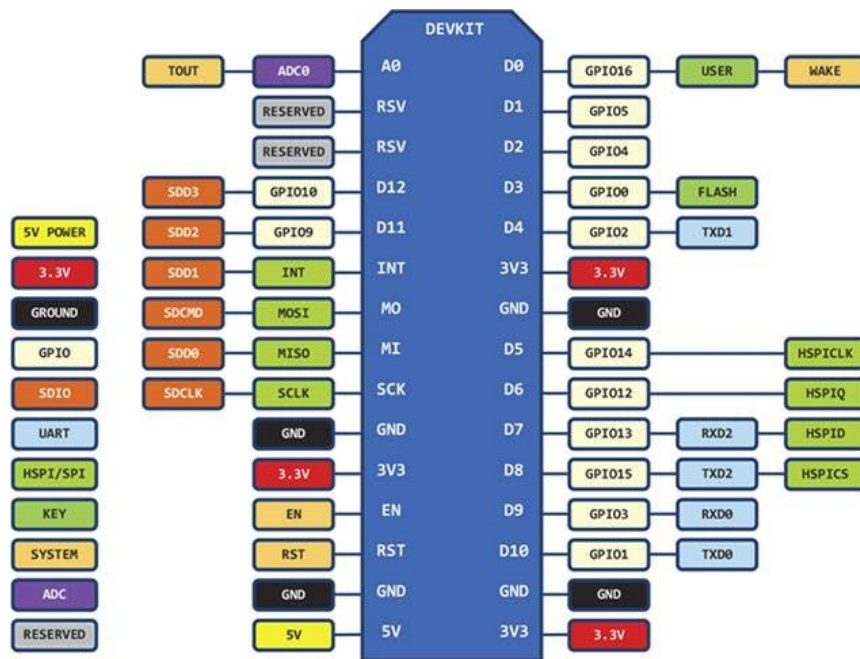
- Processor: L106 32-bit RISC microprocessor core based on the Tensilica Xtensa Diamond Standard 106Micro running at 80 MHz
- 64 KiB of instruction RAM, 96 KiB of data RAM
- External QSPI flash: 512 KiB to 4 MiB\* (up to 16 MiB is supported)
- IEEE 802.11 b/g/n Wi-Fi
  1. Integrated TR switch, balun, LNA, power amplifier and matching network
  2. WEP or WPA/WPA2 authentication, or open networks

- 16 GPIO pin
- SPI
- I<sup>2</sup>C (software implementation)
- I<sup>2</sup>S interfaces with DMA (sharing pins with GPIO)
- UART on dedicated pins, plus a transmit-only UART can be enabled on GPIO2
- 10-bit ADC (successive approximation ADC)

Pada akhir Oktober 2014, Espressif Systems merilis perangkat lunak *development kit* (SDK) yang memungkinkan *chip* diprogram, menghilangkan kebutuhan akan mikrokontroler yang terpisah. Sejak itu, telah banyak SDK resmi yang dirilis oleh Espressif. Espressif menyokong dua versi SDK, yang berbasis pada FreeRTOS dan yang lainnya berdasarkan *callback*. Alternatif selain SDK resmi Espressif adalah *open source* dari ESP-Open-SDK yang berbasis pada GCC toolchain. ESP8266 menggunakan mikrokontroler Cadence Tensilica L106 dan toolchain GCC terbuka dan disokong oleh Max Filippov. Alternatif lain adalah "*Unofficial Development Kit*" oleh Mikhail Grigorev.

### 2.3 NodeMCU

NodeMCU adalah platform IoT *open source*. Ini termasuk *firmware* yang berjalan di ESP8266 Wi-Fi SoC dari Espressif Systems, dan perangkat keras yang berbasis pada modul ESP-12. Istilah "NodeMCU" secara default mengacu pada *firmware* daripada perangkat *dev*. *Firmware* menggunakan bahasa *scripting* Lua. Hal ini didasarkan pada proyek eLua, dan dibangun di atas SDK Non-OS Espresso untuk ESP8266. Ini menggunakan banyak proyek *open source*, seperti *lua-cjson*, dan *spiffs*.



D0(GPIO16) can only be used as gpio read/write, no interrupt supported, no pwm/i2c/iow supported.

Gambar 2.3 NodeMCU Pinout

NodeMCU diciptakan tak lama setelah ESP8266 keluar. Pada tanggal 30 Desember 2013, Espressif Systems memulai produksi ESP8266. ESP8266 adalah *Wi-Fi SoC* yang terintegrasi dengan inti Tensilica Xtensa LX106, banyak digunakan dalam aplikasi *IoT*. NodeMCU dimulai pada 13 Okt 2014, ketika Hong melakukan file pertama *firmware* nodemcu ke GitHub. Dua bulan kemudian, proyek tersebut diperluas untuk menyertakan *open-hardware platform* saat pengembang Huang R membuat file *gerber* dari papan ESP8266, yang diberi nama devkit v0.9. Belakangan bulan itu, Tuan PM mem-*porting client library* MQTT (*Message Queue Telemetry Transport*) dari Contiki ke platform SoC ESP8266, dan berkomitmen pada proyek NodeMCU, kemudian NodeMCU bisa mendukung protokol MQTT *IoT*, menggunakan *Lua* untuk mengakses broker MQTT. Pembaruan penting lainnya dilakukan pada tanggal 30 Januari 2015, saat *Devsaurus* mem-*porting* proyek *u8glib* ke NodeMCU, memudahkan NodeMCU untuk menggerakkan LCD, Layar, OLED, bahkan display VGA. Di musim panas 2015 para pencipta meninggalkan proyek *firmware* ini dan sekelompok kontributor independen namun berdedikasi mengambil alihnya. Pada musim panas 2016, NodeMCU menyertakan lebih dari 40 modul yang berbeda. Karena keterbatasan

sumber daya, pengguna perlu memilih modul yang relevan untuk proyek mereka dan membuat *firmware* yang disesuaikan dengan kebutuhan mereka.

Beberapa fitur dari NodeMCU antara lain:

- **Perangkat keras IO (*input-output*) seperti Arduino**

API tingkat lanjut untuk perangkat keras IO, yang dapat secara dramatis mengurangi pekerjaan yang berlebihan untuk mengkonfigurasi dan memanipulasi perangkat keras. Kode seperti *arduino*, tapi interaktif dalam naskah *Lua*.

- **API (*application programming interface*) jaringan dengan gaya Nodejs**

*Event-driven* API untuk aplikasi jaringan, yang memudahkan pengembang menulis kode yang berjalan pada MCU (*memory controller unit*) berukuran 5mm\*5mm dengan gaya Nodejs, dan sangat mempercepat aplikasi pengembangan aplikasi IOT.

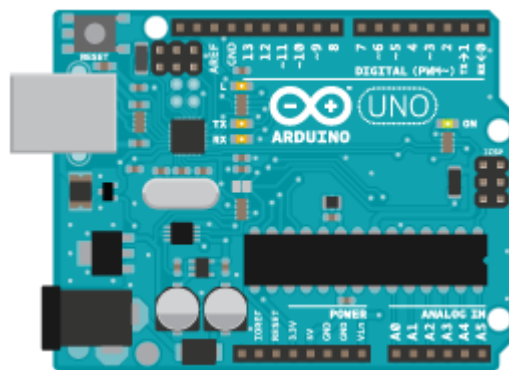
- **Perangkat WI-FI paling murah**

Perangkat WIFI MCU ESP8266 terintegrasi dan mudah untuk *prototyping development kit* dengan harga kurang dari \$2.

Saat Arduino.cc mulai mengembangkan papan MCU baru berdasarkan prosesor non-AVR seperti ARM / SAM MCU dan digunakan dalam Arduino Due, mereka perlu memodifikasi IDE Arduino, sehingga IDE akan mudah untuk mendukung alternatif *tool chains* sehingga memungkinkan Arduino C/C++ dikompilasi ke prosesor baru ini. Mereka melakukan ini dengan diperkenalkannya Board Manager dan SAM Core. Sebuah "inti" adalah kumpulan komponen perangkat lunak yang dibutuhkan oleh Board Manager dan Arduino IDE untuk mengkompilasi file sumber Arduino C/C++ ke bahasa mesin MCU. Beberapa penggemar kreatif ESP8266 telah mengembangkan inti Arduino untuk SoC WiFi ESP8266 yang tersedia di laman web inti GitHub ESP8266. Inilah yang populer disebut "ESP8266 Core untuk Arduino IDE" dan telah menjadi salah satu platform pengembangan perangkat lunak terkemuka untuk berbagai modul dan papan pengembangan berbasis ESP8266, termasuk NodeMCUs.

## 2.4 Arduino

Arduino adalah pengendali mikro single-board yang bersifat open-source, diturunkan dari *Wiring* platform, dirancang untuk memudahkan penggunaan elektronik dalam berbagai bidang. Perangkat kerasnya memiliki prosesor Atmel AVR dan perangkat lunaknya memiliki bahasa pemrograman sendiri. Arduino juga merupakan platform perangkat keras terbuka yang ditujukan kepada siapa saja yang ingin membuat purwarupa peralatan elektronik interaktif berdasarkan perangkat keras dan perangkat lunak yang fleksibel dan mudah digunakan. Mikrokontroler diprogram menggunakan bahasa pemrograman arduino yang memiliki kemiripan syntax dengan bahasa pemrograman C. Karena sifatnya yang terbuka maka siapa saja dapat mengunduh skema perangkat keras arduino dan membangunnya. Arduino menggunakan keluarga mikrokontroler ATmega yang dirilis oleh Atmel sebagai basis, namun ada individu/perusahaan yang membuat clone arduino dengan menggunakan mikrokontroler lain dan tetap kompatibel dengan arduino pada level perangkat keras. Untuk fleksibilitas, program dimasukkan melalui bootloader meskipun ada opsi untuk membypass bootloader dan menggunakan downloader untuk memprogram mikrokontroler secara langsung melalui port ISP.



Gambar 2.4 Arduino UNO

Program untuk Arduino dapat ditulis dengan bahasa pemrograman apapun untuk *compiler* yang menghasilkan kode mesin biner untuk prosesor target. Atmel menyediakan lingkungan pengembangan untuk *microcontrollers* mereka, AVR Studio dan Atmel Studio yang lebih baru.

Proyek Arduino menyediakan lingkungan pengembangan terpadu (Arduino IDE), yang merupakan aplikasi *cross-platform* yang ditulis dalam bahasa pemrograman Java. Ini berasal dari IDE untuk bahasa *Processing* dan *Wiring*. Ini termasuk editor kode dengan fitur seperti pemotongan dan penyisipan teks, pencarian dan penggantian teks, indentasi otomatis, penjepit, dan penyorotan sintaksis, dan menyediakan mekanisme satu klik sederhana untuk mengkompilasi dan mengunggah program ke papan Arduino. Ini juga berisi area pesan, konsol teks, toolbar dengan tombol untuk fungsi umum dan hirarki menu operasi. Program yang ditulis dengan IDE untuk Arduino disebut *sketch*. *Sketch* disimpan pada komputer pengembangan sebagai file teks dengan ekstensi file *.ino*.

Arduino IDE mendukung bahasa C dan C++ menggunakan aturan khusus penataan kode. Arduino IDE memasok perpustakaan perangkat lunak dari proyek *Wiring* yang menyediakan banyak prosedur *input* dan *output* yang umum. Kode yang ditulis pengguna hanya memerlukan dua fungsi dasar untuk memulai sketsa dan lingkaran program utama, yang disusun dan dihubungkan dengan program rintisan *main()* ke dalam program eksekutif siklik, yang dapat dieksekusi dengan *toolchain* GNU, juga disertakan dengan distribusi IDE. Arduino IDE menggunakan program *avrdude* untuk mengubah kode yang dapat dieksekusi menjadi file teks dalam pengkodean heksadesimal yang dimuat ke papan Arduino oleh program loader di papan firmware.

Sketsa Arduino C/C++ minimal seperti yang terlihat oleh programmer IDE Arduino, hanya terdiri dari dua fungsi:

1. *setup()*: Fungsi ini disebut satu kali saat sketsa dimulai setelah *power-up* atau *reset*. Ini digunakan untuk menginisialisasi variabel, mode pin *input* dan *output*, dan *libraries* lain yang dibutuhkan dalam sketsa.
2. *loop()*: Setelah *setup()* dipanggil, fungsi *loop()* dijalankan berulang kali dalam program utama. Ini mengendalikan papan sampai dimatikan atau diatur ulang.

Kebanyakan papan Arduino mengandung dioda LED dan resistor beban yang dihubungkan antara pin 13 dan *ground*, yang merupakan fitur yang mudah digunakan untuk banyak tes dan fungsi program.



```

#define LED_PIN 13 // Pin number attached to LED.

void setup() {
  pinMode(LED_PIN, OUTPUT); // Configure pin 13 to be a digital output.
}

void loop() {
  digitalWrite(LED_PIN, HIGH); // Turn on the LED.
  delay(1000); // Wait 1 second (1000 milliseconds).
  digitalWrite(LED_PIN, LOW); // Turn off the LED.
  delay(1000); // Wait 1 second.
}

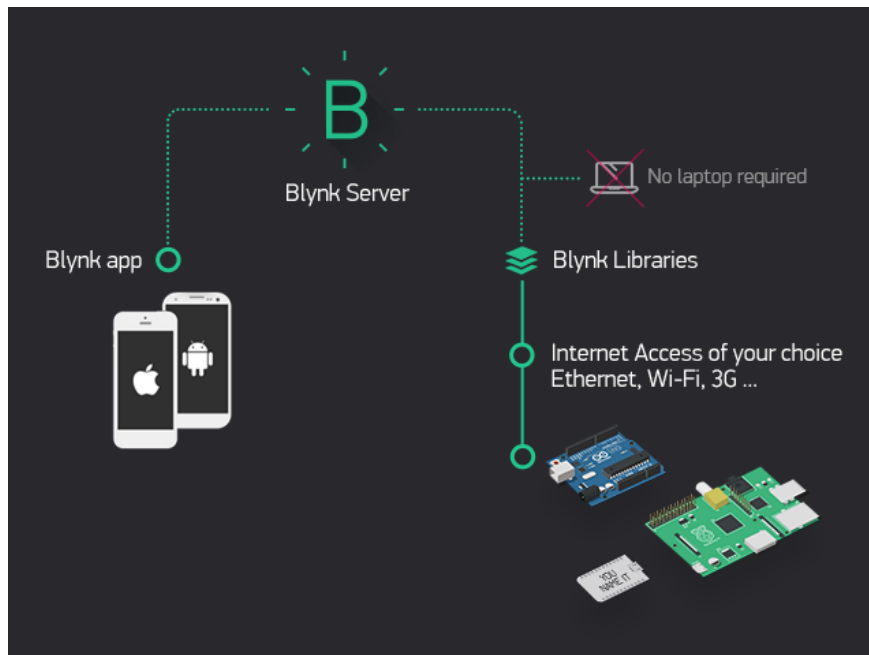
```

Program ini menggunakan fungsi *pinMode()*, *digitalWrite()*, dan *delay()*, yang disediakan oleh *libraries* internal yang termasuk dalam lingkungan IDE. Program ini biasanya dimuat di Arduino oleh pabrikan. Arduino IDE dan bahasa C memungkinkan pemrograman register tingkat rendah di *atmega328P*.

## 2.5 Blynk

Blynk dirancang untuk *Internet of Things* (IoT). Blynk dapat mengendalikan perangkat keras dari jarak jauh, bisa menampilkan data sensor, bisa menyimpan data, mengabadikannya dan melakukan banyak hal keren lainnya. Ada tiga komponen utama dalam platform:

1. Blynk App: memungkinkan kita membuat antarmuka yang menakjubkan untuk proyek kita dengan menggunakan berbagai widget yang disediakan.
2. Blynk Server: bertanggung jawab atas semua komunikasi antara smartphone dan perangkat keras. Kita bisa menggunakan Blynk Cloud atau menjalankan server Blynk pribadi secara lokal. Blynk bersifat *open source*, bisa dengan mudah menangani ribuan perangkat dan bahkan bisa diluncurkan di *Raspberry Pi*.
3. Blynk Libraries: bisa untuk semua platform perangkat keras yang populer - memungkinkan komunikasi dengan server dan memproses semua perintah yang masuk dan keluar.



Gambar 2.5 Cara Kerja Blynk

Blynk bekerja melalui Internet. Ini berarti *hardware* yang kita pilih harus bisa terhubung ke internet. Beberapa papan, seperti *Arduino Uno* memerlukan *Ethernet* atau *Wi-Fi Shield* untuk berkomunikasi, sedangkan papan yang lain sudah mengaktifkan Internet-nya; seperti *ESP8266*, *Raspberri Pi* dengan *dongle* WiFi, *Particle Photon* atau *SparkFun Blynk Board*. Tetapi bahkan jika Anda tidak memiliki *shield*, kita dapat menghubungkannya dengan USB ke laptop atau desktop. Aplikasi Blynk dirancang dengan program antarmuka yang baik, dapat bekerja pada iOS dan Android.

Kita dapat menjalankan aplikasi Blynk menggunakan perangkat NodeMCU, berikut adalah contoh *script* yang digunakan:

```
#define BLYNK_PRINT Serial

#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
char auth[] = "YourAuthToken";

// Your WiFi credentials.
// Set password to "" for open networks.
```

```

char ssid[] = "YourNetworkName";
char pass[] = "YourPassword";

void setup()
{
  // Debug console
  Serial.begin(9600);

  Blynk.begin(auth, ssid, pass);
  // You can also specify server:
  //Blynk.begin(auth, ssid, pass, "blynk-cloud.com", 8442);
  //Blynk.begin(auth, ssid, pass, IPAddress(192,168,1,100), 8442);
}

void loop()
{
  Blynk.run();
  // You can inject your own code or combine it with other sketches.
  // Check other examples on how to communicate with Blynk. Remember
  // to avoid delay() function!
}

```

## 2.6 IFTTT

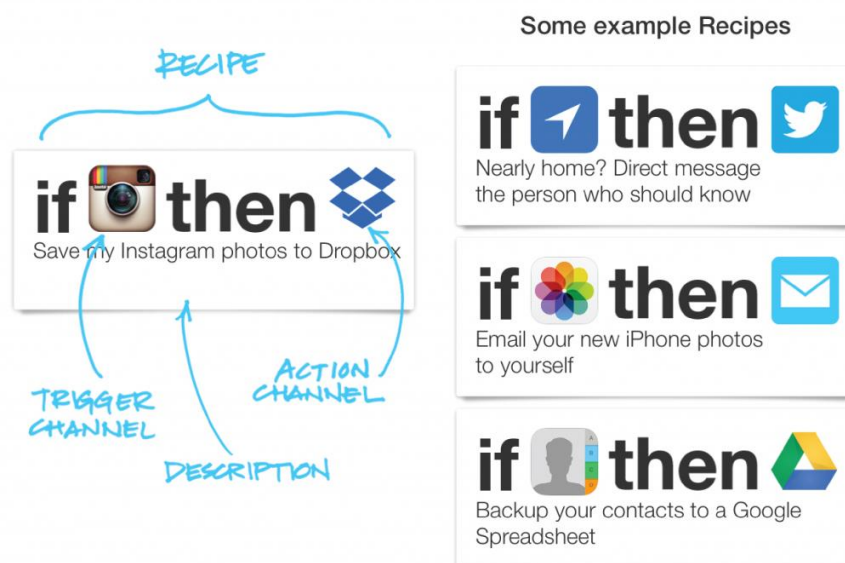
IFTTT adalah layanan berbasis *web* gratis untuk membuat rangkaian pernyataan bersyarat sederhana, yang disebut *applet*. *Applet* dipicu oleh perubahan yang terjadi dalam layanan *web* lain seperti Gmail, Facebook, Instagram, atau Pinterest. Misalnya, *applet* dapat mengirim pesan *e-mail* jika pengguna *posting* menggunakan *hashtag*, atau menyalin foto di Facebook ke arsip pengguna jika ada seseorang memberi *tag* ke pengguna di foto itu. Selain aplikasi berbasis *web*, layanan juga berjalan di perangkat iOS dan Android. Pada bulan Februari 2015, IFTTT mengganti nama aplikasi aslinya menjadi IF, dan merilis rangkaian aplikasi baru yang disebut Do yang memungkinkan pengguna membuat aplikasi dan *shortcut*. Pada tahun 2015, pengguna IFTTT membuat sekitar 20 juta resep setiap hari. IFTTT adalah kepanjangan dari “*If This Then That*”.

IFTTT menggunakan konsep berikut:

- *Services*; adalah blok-blok bangunan dasar dari IFTTT. Mereka terutama menggambarkan serangkaian data dari layanan web tertentu seperti *YouTube* atau *eBay*. *Services* juga dapat menggambarkan tindakan yang dikendalikan dengan *API*

tertentu, seperti *SMS*. Terkadang, mereka bisa mewakili informasi dalam hal cuaca atau persediaan. Setiap layanan memiliki serangkaian pemicu dan tindakan tertentu.

- *Triggers*; adalah bagian "*this*" dari *applet*. Mereka adalah *item* yang memicu aksi. Misalnya, dari umpan RSS, kita dapat menerima pemberitahuan berdasarkan kata kunci atau frase.
- *Actions*; adalah bagian "*that*" dari *applet*. Mereka adalah *output* yang dihasilkan dari *trigger*.
- *Applets*; adalah predikat yang dibuat dari *Triggers* dan *Actions*. Misalnya, jika Anda menyukai gambar di Instagram (*trigger*), aplikasi IFTTT dapat mengirim foto ke akun Dropbox Anda (*action*).
- *Ingredients*; adalah data dasar yang tersedia dari pemicu - dari pemicu email, misalnya; subjek, badan, lampiran, tanggal diterima, dan alamat pengirim.

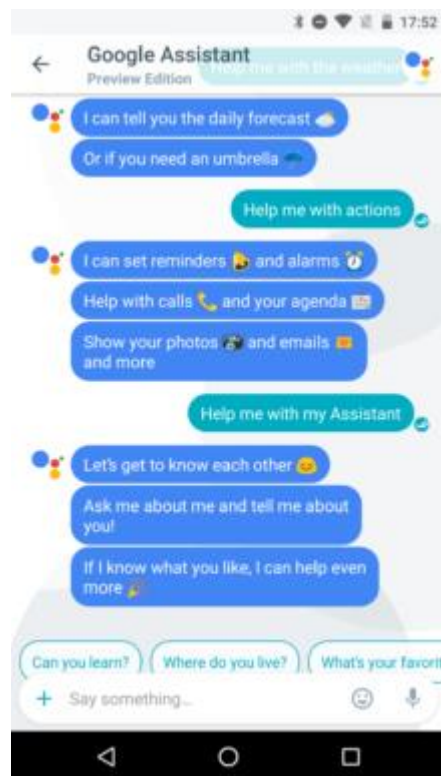


Gambar 2.6 Contoh Penggunaan IFTTT

IFTTT membantu kita menghubungkan semua aplikasi dan perangkat yang berbeda. Kita dapat mengaktifkan *Applet* yang membantu aplikasi bekerja sama untuk melakukan hal-hal spesifik yang tidak dapat mereka lakukan sebelumnya. Misalnya, kita dapat mencadangkan foto *Instagram* Anda ke *Dropbox*, menyalakan lampu saat memasuki rumah kita, atau secara otomatis. Ada jutaan *Applet* untuk dijelajahi.

## 2.7 GOOGLE ASSISTANT

Google Assistant adalah asisten pribadi virtual yang dikembangkan oleh Google dan diumumkan pengembangnya pada bulan Mei 2016. Tidak seperti Google Now, Asisten Google dapat terlibat dalam percakapan dua arah. Setelah masa eksklusivitas pada smartphone Pixel dan Pixel XL, aplikasi ini mulai disebarakan pada perangkat Android lainnya pada bulan Februari 2017, termasuk smartphone pihak ketiga dan Android Wear, dan diluncurkan sebagai aplikasi mandiri di sistem operasi iOS pada bulan Mei. Bersamaan dengan pengumuman pengembangan perangkat lunak pada bulan April 2017, Google Assistant telah, dan sedang dikembangkan lebih jauh untuk mendukung berbagai macam perangkat, termasuk mobil dan peralatan rumah tangga. Fungsionalitas Google Assistant juga dapat ditingkatkan oleh pengembang pihak ketiga.



Gambar 2.7 Contoh percakapan pada Google Assistant

Google Assistant dapat mencari di Internet, menjadwalkan acara dan alarm, menyesuaikan pengaturan perangkat keras di perangkat pengguna, dan menampilkan informasi dari akun Google pengguna. Hasil pencarian disajikan dalam format kartu yang dapat diketuk pengguna untuk membuka halaman.