

BAB II

TEORI PENUNJANG

2.1 Aquascape

Aquascape merupakan taman alami yang dibuat di dalam akuarium. Bukan sekedar taman biasa, tetapi lebih kepada pemaknaan hidup dari miniatur alam yang penuh dengan kesinergiaan. Dari dalam akuarium tanaman mendapatkan makanan dari media tanam, ikan pun menghasilkan CO₂ yang dibutuhkan tanaman. Timbal baliknya tanaman juga memproduksi oksigen yang diperlukan ikan dan juga udang pemakan lumut dan mikroorganisme yang ada di dalamnya. Biasanya aquascape mempunyai air terjun mini buatan untuk memperkuat kesan alam yang ada pada aquascape.



Gambar II-1 Aquascape [5]

2.2 LED RGB

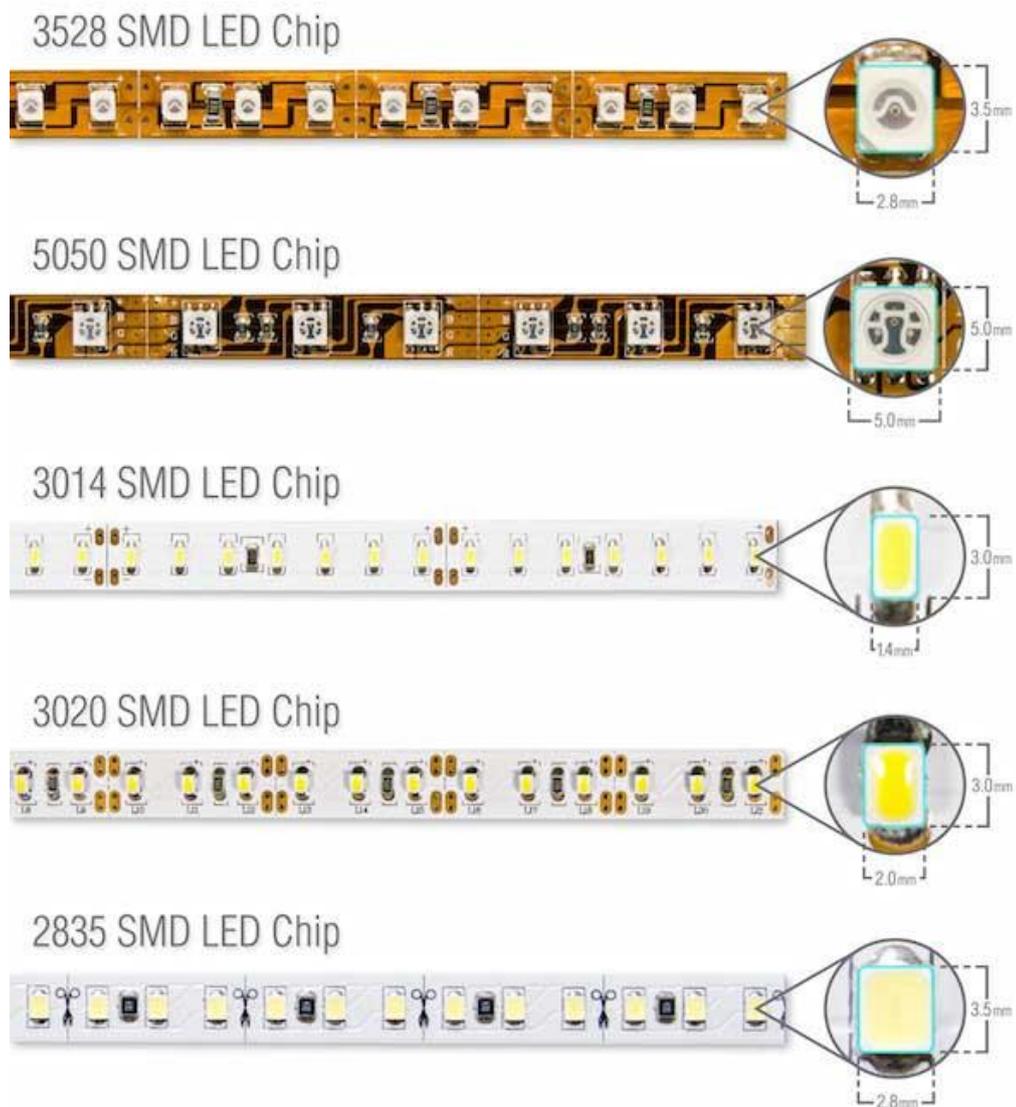
LED adalah singkatan dari Light Emitting Dioda, yaitu suatu semi konduktor yang memancarkan cahaya monokromatik yang tidak koheren ketika diberi tegangan maju[6]. LED merupakan keluarga Dioda yang terbuat dari bahan semikonduktor. Warna-warna cahaya yang dipancarkan oleh LED tergantung pada jenis bahan semikonduktor yang dipergunakannya[7]. LED banyak dipakai, seperti pada penggunaan lampu permainan untuk anak-anak, untuk trafic light, sebagai

indikator pada peralatan elektronik hingga mesin industri, penggunaan pada lampu emergency, pada televisi, dan berbagai perangkat elektronik lainnya sebagai tanda bahwa sistem dalam keadaan. LED RGB yaitu LED yang berisikan tiga warna LED yang terintegrasi menjadi satu lampu LED. LED RGB mengandung warna RED (merah), GREEN (hijau), dan BLUE (biru) [8].



Gambar II-2 RGB LED Strip 5050 [9]

LED Strip atau yang biasa disebut dengan LED Pita atau LED Flexible, bisa dipakai untuk berbagai macam kebutuhan. Walaupun LED Strip bisa saja digunakan untuk hampir semua pencahayaan tidak langsung (indirect lighting), Tiap jenis LED Strip lebih cocok digunakan untuk kebutuhan yang spesifik karena karakteristiknya yang unik. Jenis LED Strip 5050 SMD menggunakan energi antara 7 – 14.4 Watt per meternya, karena tingkat Cahaya yang lebih terang dibandingkan dengan 3528 SMD, Biasa digunakan untuk : Drop Ceiling (langit-langit bangunan), Etalase, Signage, Light Box, Back Drop lighting, toko emas dan tempat-tempat lain yang menginginkan cahaya mencolok untuk menarik perhatian.



Gambar II-3 Jenis LED Strip[10]

2.3 Mikrokontroler ESP32

ESP32 dikenalkan oleh Espressif System yang merupakan penerus dari mikrokontroler ESP8266. Mikrokontroler ESP32 memiliki keunggulan yaitu sistem yang berbiaya rendah, dan juga berdaya rendah dengan modul WiFi yang terintegrasi dengan chip mikrokontroler serta memiliki bluetooth dengan mode ganda dan fitur hemat daya menjadikannya lebih fleksibel. ESP32 kompatibel dengan perangkat seluler dan aplikasi IoT (Internet of Things). Mikrokontroler ini

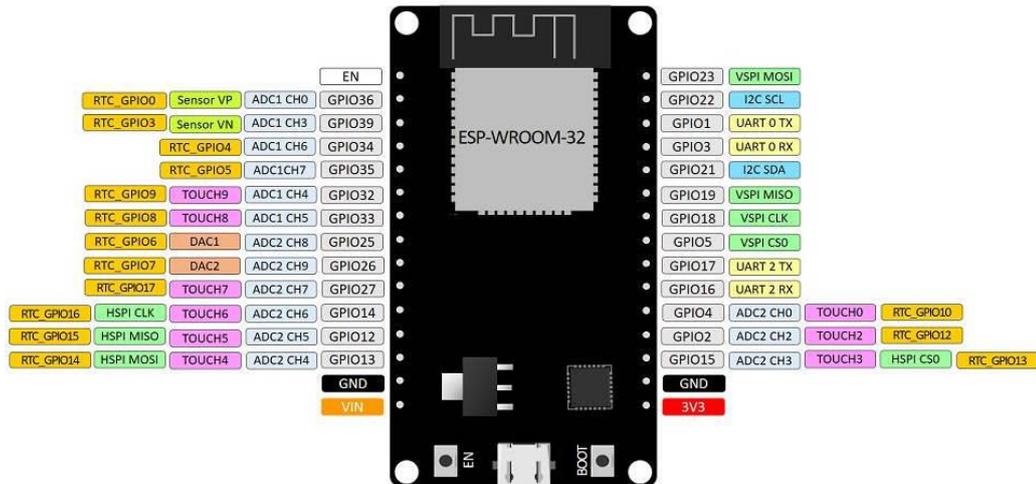
dapat digunakan sebagai sistem mandiri yang lengkap atau dapat dioperasikan sebagai perangkat pendukung mikrokontroler host[11].

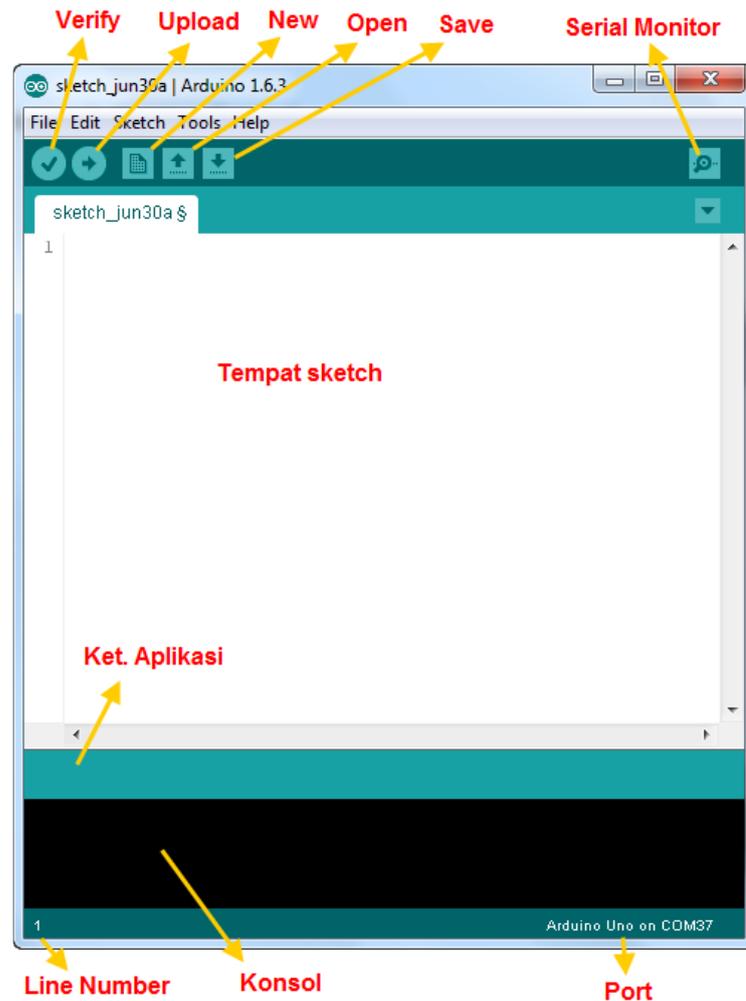
ESP32 menggunakan prosesor dual core yang berjalan di instruksi Xtensa LX16, ESP32 memiliki spesifikasi seperti yang ditampilkan pada tabel II-1[12].

Tabel II-1. Spesifikasi ESP32

No	Atribut	Detail
1	Tegangan	3.3 Volt
2	Prosesor	Tensilica L108 32 bit
3	Kecepatan prosesor	Dual 160MHz
4	RAM	520K
5	GPIO	34
6	ADC	7
7	Dukungan 802.11	11b/g/n/e/i
8	Bluetooth	BLE (Bluetooth Low Energy)
9	SPI	3
10	I2C	2
11	UART	3

Board ini memiliki dua versi, yaitu yang 30 dan 36 GPIO. Keduanya berfungsi dengan cara yang sama tetapi versi yang 30 GPIO dipilih karena memiliki dua pin GND. Semua pin diberi label di bagian atas board sehingga mudah untuk dikenali. Board ini memiliki interface USB-to-UART sehingga mudah diprogram dengan program pengembangan aplikasi seperti Arduino IDE atau yang lainnya. Sumber daya untuk board bisa diberikan melalui konektor micro-USB[13].





Gambar II-5 Interface Arduino IDE

Interface Arduino IDE tampak seperti gambar di atas. Dari kiri ke kanan dan atas ke bawah, bagian-bagian IDE Arduino terdiri dari[14]:

- **Verify** : pada versi sebelumnya dikenal dengan istilah Compile. Sebelum aplikasi diupload ke board Arduino, biasakan untuk memverifikasi terlebih dahulu sketch yang dibuat. Jika ada kesalahan pada sketch, nanti akan muncul error. Proses Verify / Compile mengubah sketch ke binary code untuk diupload ke mikrokontroler.
- **Upload** : tombol ini berfungsi untuk mengupload sketch ke board Arduino. Walaupun kita tidak mengklik tombol verify, maka sketch akan di-compile, kemudian langsung diupload ke board. Berbeda dengan tombol verify yang hanya berfungsi untuk memverifikasi source code saja.
- **New Sketch** : Membuka window dan membuat sketch baru

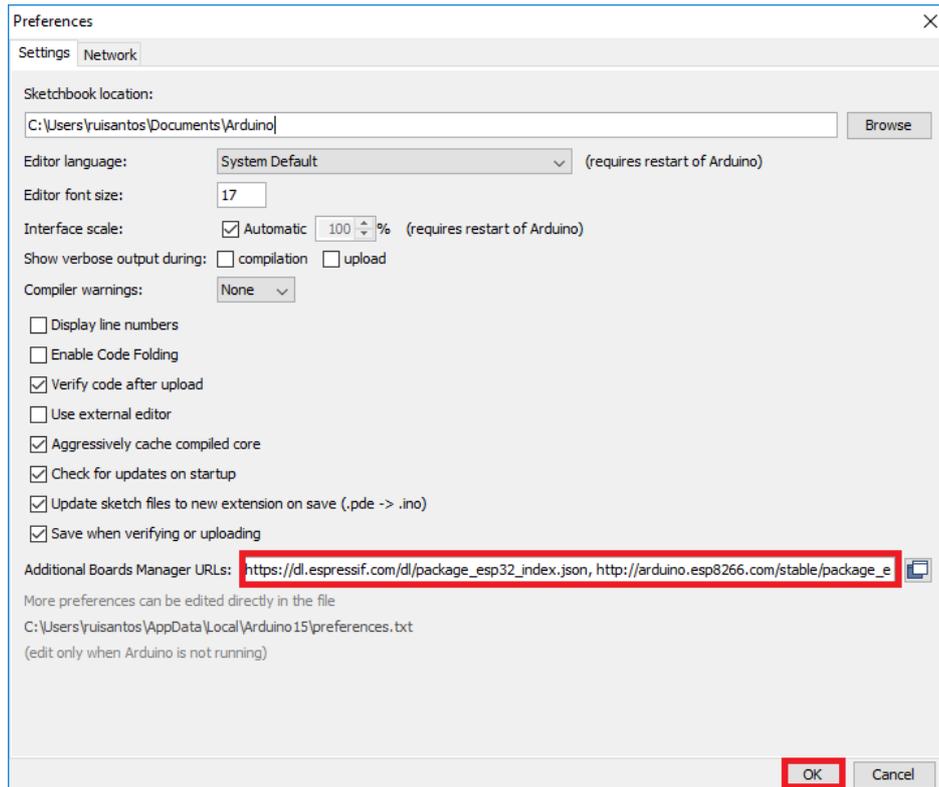
- Open Sketch : Membuka sketch yang sudah pernah dibuat. Sketch yang dibuat dengan IDE Arduino akan disimpan dengan ekstensi file .ino
- Save Sketch : menyimpan sketch, tapi tidak disertai mengcompile.
- Serial Monitor : Membuka interface untuk komunikasi serial, nanti akan kita diskusikan lebih lanjut pada bagian selanjutnya
- Keterangan Aplikasi : pesan-pesan yang dilakukan aplikasi akan muncul di sini, misal "Compiling" dan "Done Uploading" ketika kita mengcompile dan mengupload sketch ke board Arduino
- Konsol : Pesan-pesan yang dikerjakan aplikasi dan pesan-pesan tentang sketch akan muncul pada bagian ini. Misal, ketika aplikasi mengcompile atau ketika ada kesalahan pada sketch yang kita buat, maka informasi error dan baris akan diinformasikan di bagian ini.
- Baris Sketch : bagian ini akan menunjukkan posisi baris kursor yang sedang aktif pada sketch.
- Informasi Port : bagian ini menginformasikan port yang dipakai oleh board Arduino.

2.4.1 Instalasi Board ESP32 Devkit Pada Arduino IDE

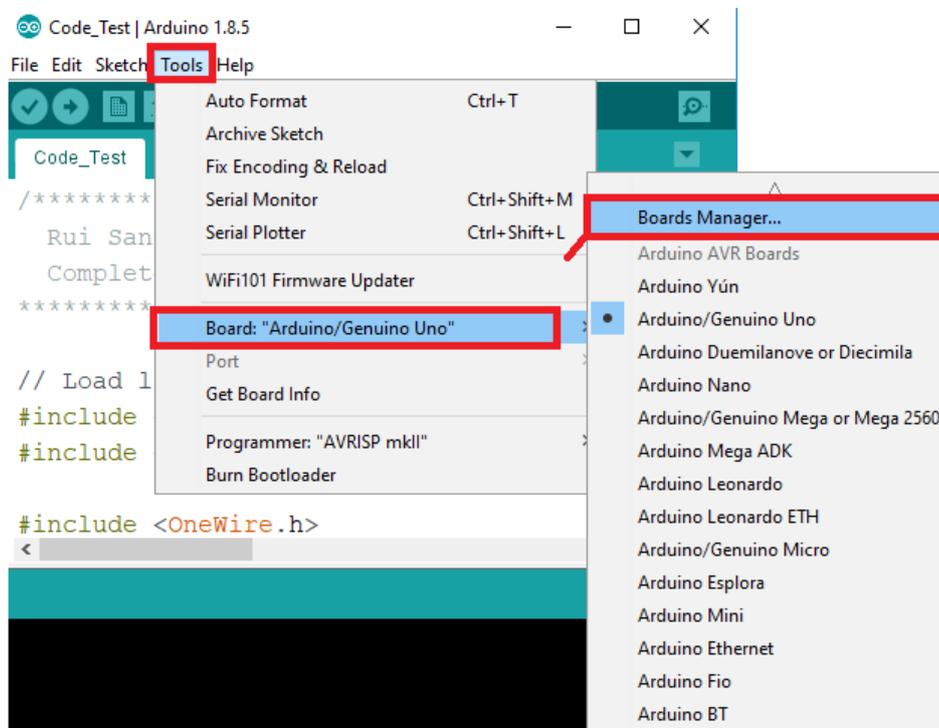
ESP32 saat ini sedang diintegrasikan dengan IDE Arduino seperti yang dilakukan untuk ESP8266. Pengaya untuk Arduino IDE ini memungkinkan untuk memprogram ESP32 menggunakan Arduino IDE dan bahasa pemrogramannya.

Untuk menginstal papan ESP32 di Arduino IDE, silahkan mengikuti instruksi berikut ini[15]:

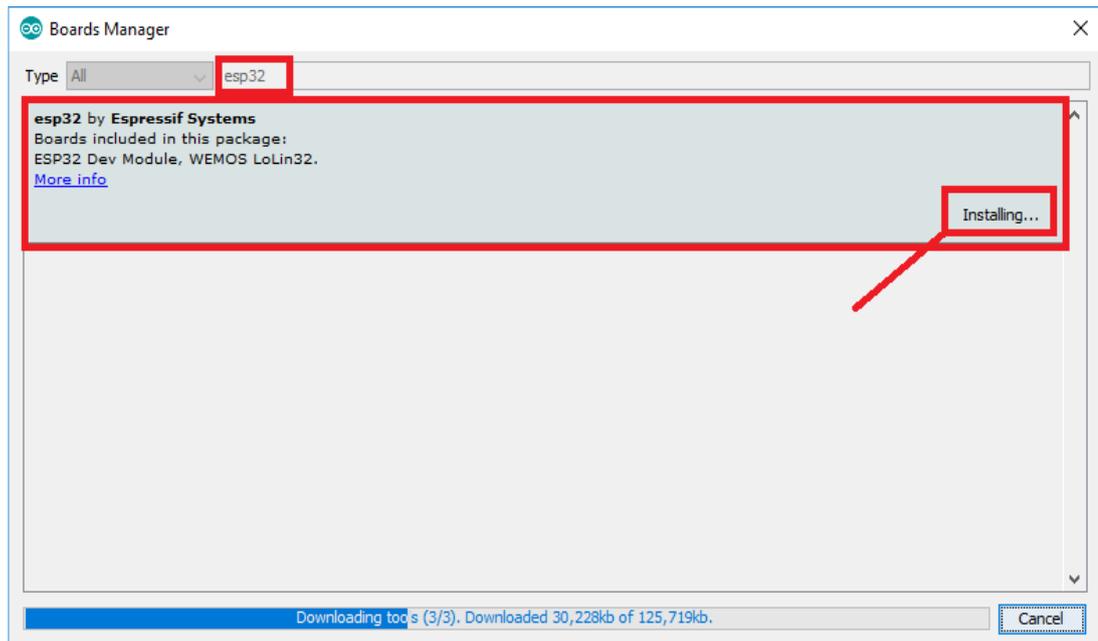
- 1) Buka jendela preferensi dari Arduino IDE. Pilih File - Preferences
- 2) Memasukkan https://dl.espressif.com/dl/package_esp32_index.json ke dalam "Additional Board Manager URLs". Seperti ditunjukkan pada gambar di bawah ini. Kemudian, klik tombol "OK".



3) Pergi ke boards manager. **Tools** ▶ **Board** ▶ **Boards Manager...**



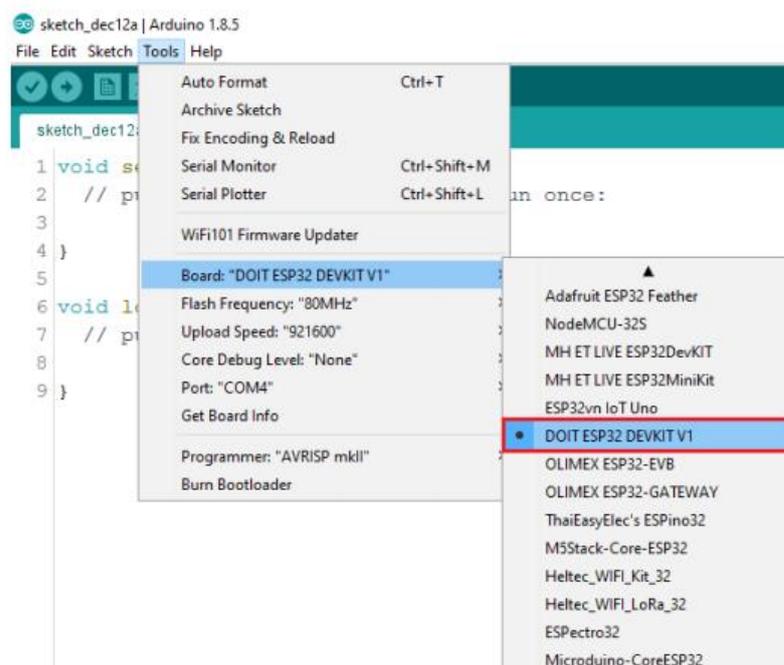
4) Cari ESP32 dan tekan tombol install untuk "ESP32 by Espressif Systems"



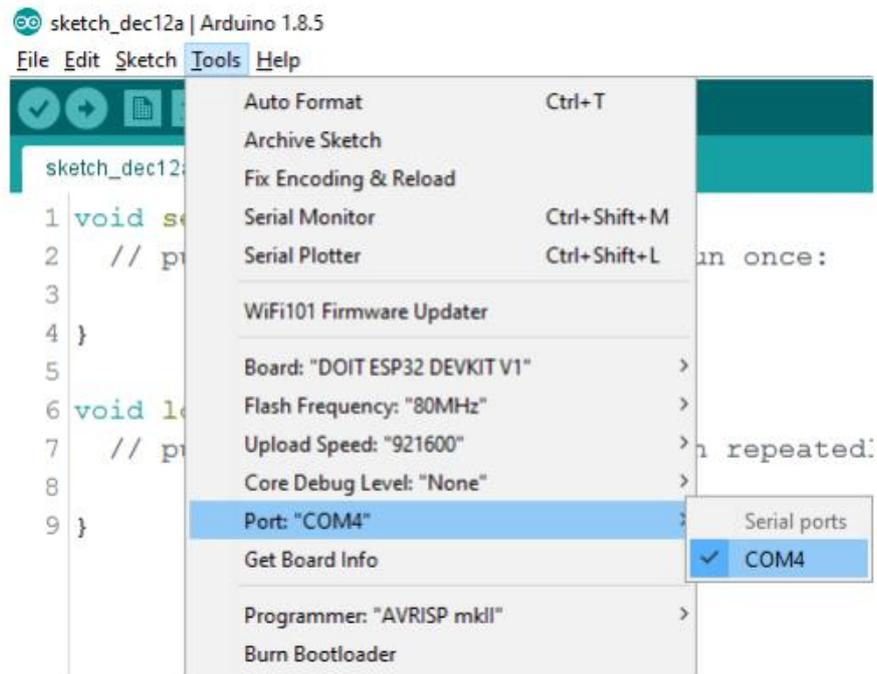
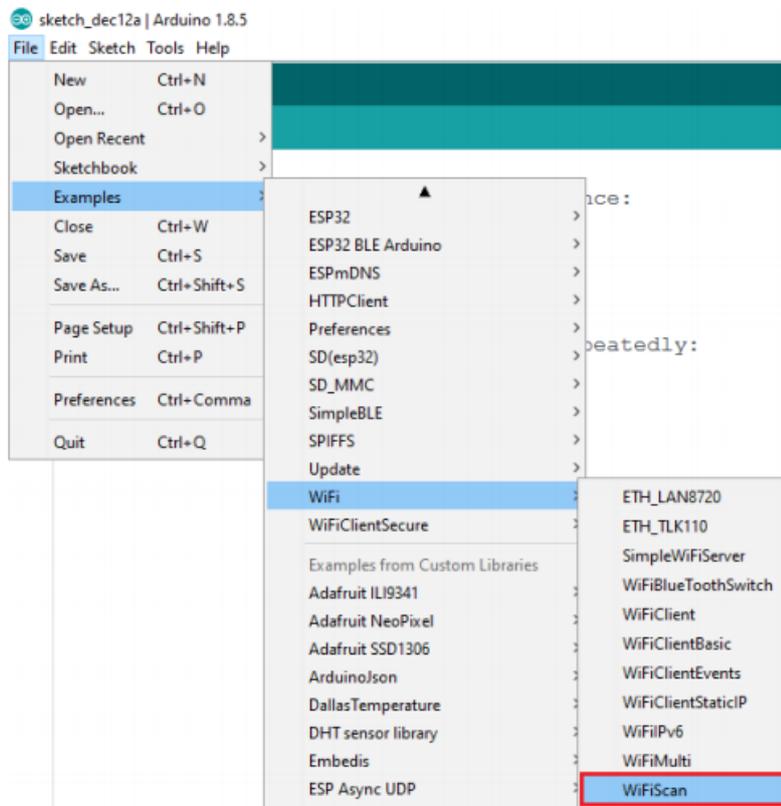
2.4.2 Menguji Instalasi

Pasang Papan ESP32 DOIT DEVKIT V1 ke PC. Kemudian, ikuti langkah-langkah ini:

- 1) Buka Arduino IDE
- 2) Pilih Board di menu Tools - Board



3) Pilih Port

4) buka contoh berikut **File** **Examples** **WiFi (ESP32)** **WiFi Scan**

5) Sketch baru terbuka:

```

WiFiScan | Arduino 1.8.5
File Edit Sketch Tools Help
WiFiScan
1 /*
2 * This sketch demonstrates how to scan WiFi networks.
3 * The API is almost the same as with the WiFi Shield library,
4 * the most obvious difference being the different file you need to include:
5 */
6 #include "WiFi.h"
7
8 void setup()
9 {
10   Serial.begin(115200);
11
12   // Set WiFi to station mode and disconnect from an AP if it was previousl
13   WiFi.mode(WIFI_STA);
14   WiFi.disconnect();
15   delay(100);
16
17   Serial.println("Setup done");
18 }
19
20 void loop()

```

6) Tekan tombol Upload di Arduino IDE. Tunggu beberapa detik saat kode dikompilasi dan diunggah ke papan. 

7) Jika semuanya berjalan seperti yang diharapkan, Anda akan melihat pesan "**Done uploading.**"

```

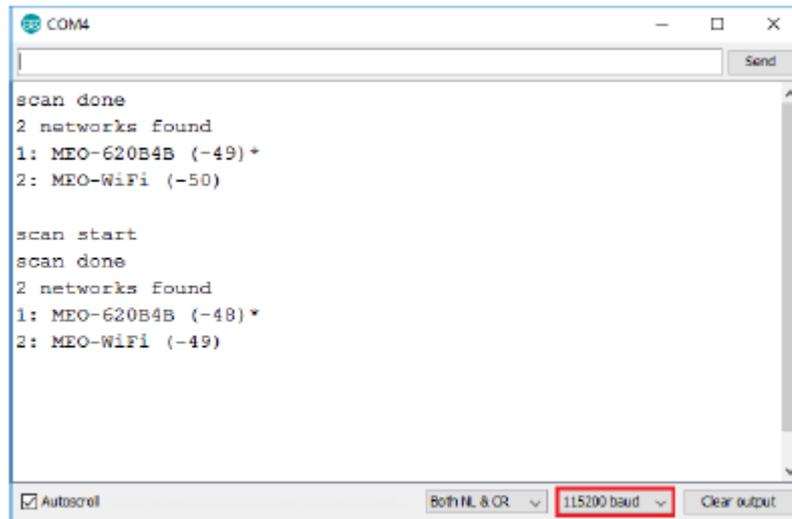
Done uploading
writing at 0x0004c000... (84 %)
Writing at 0x00050000... (89 %)
Writing at 0x00054000... (94 %)
Writing at 0x00058000... (100 %)
Wrote 481440 bytes (299651 compressed) at 0x00010000 in 4.7 seconds
Hash of data verified.
Compressed 3072 bytes to 122...

Writing at 0x00008000... (100 %)
Wrote 3072 bytes (122 compressed) at 0x00008000 in 0.0 seconds
Hash of data verified.

Leaving...
Hard resetting...

```

- 8) Buka Monitor Serial Arduino IDE dengan kecepatan rata-rata 115200: 
- 9) Tekan tombol ESP32 on-board Enable dan Anda akan melihat jaringan yang tersedia di dekat ESP32 Anda:

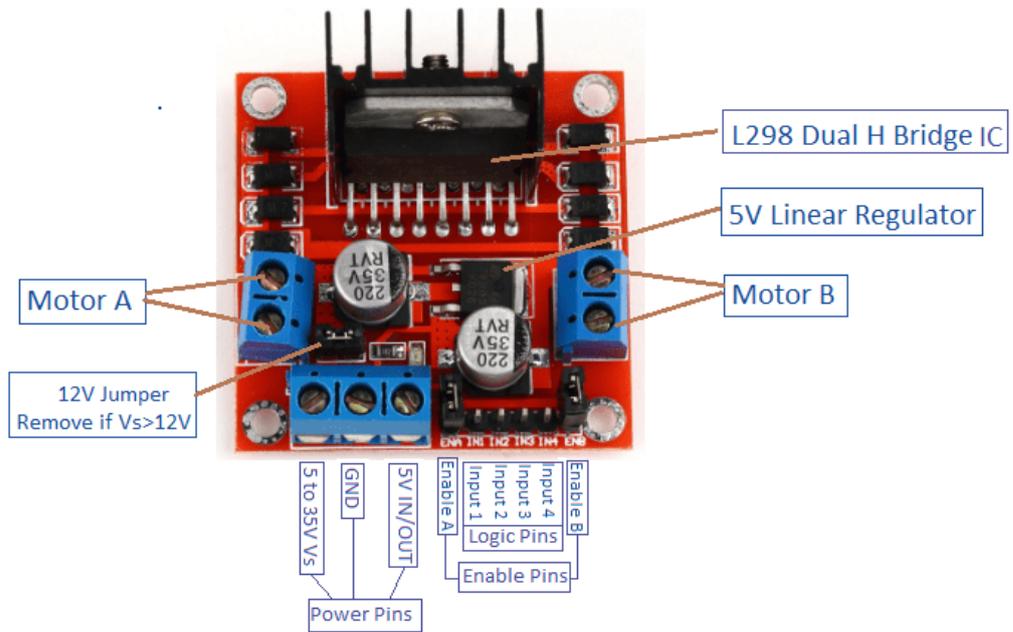


```
COM4
scan done
2 networks found
1: MEO-620B4B (-49)*
2: MEO-WiFi (-50)

scan start
scan done
2 networks found
1: MEO-620B4B (-48)*
2: MEO-WiFi (-49)
```

2.5 Driver Motor L298N

Driver motor L298N merupakan driver motor yang paling populer digunakan untuk mengontrol kecepatan dan arah pergerakan motor. Kelebihan dari driver motor L298N ini adalah cukup presisi dalam mengontrol motor. Selain itu kelebihan driver motor L298N adalah mudah untuk dikontrol. Untuk mengontrol driver motor L298N ini dibutuhkan 6 buah Pin mikrokontroler dua buah untuk Pin Enable (satu buah untuk motor A dan satu buah yang lain untuk motor B karena driver motor L298N ini dapat mengontrol dua buah motor dc) 4 buah untuk mengatur kecepatan motor tersebut. Pada prinsipnya rangkaian driver motor L298N ini dapat mengatur tegangan dan arus sehingga kecepatan dan arah motor dapat diatur[16].

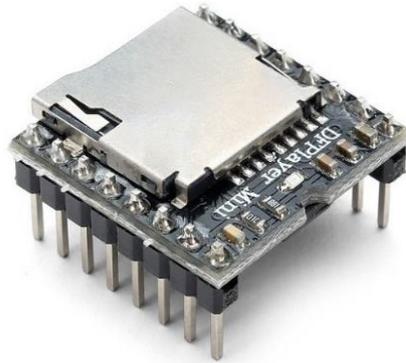


Gambar II-6 Driver motor L298N [17]

2.6 DFPlayer Mini

DFPlayer mini adalah modul mp3 yang outputnya sederhana, dapat langsung diaplikasikan pada pengeras suara speaker. DFPlayer mini dapat digunakan dengan cara berdiri tunggal menggunakan baterai, speaker, dan push button, juga dapat digunakan pada Arduino Uno ataupun dengan perangkat lain yang memiliki kemampuan receiver/transmitter.

DF Player Mini mampu menghubungkan modul decoding yang begitu rumit dengan baik, dengan memiliki format audio mp3, wav, wma, dan juga dapat support TF card dengan sistem file FAT16, FAT32. Melalui port serial sederhana, user bisa memutar suara audio yang dipilih tanpa melakukan perintah-perintah yang rumit untuk menjalankannya[18].



Gambar II-7 DFPlayer Mini [18]

2.7 Micro Mini Submersible Water Pump

Micro Mini Submersible Water Pump yaitu pompa air mini yang bertenagakan motor DC *brushless* yang bekerja pada tegangan 3 VDC - 6 VDC. Pompa air mini ini biasanya digunakan pada sistem pengairan tanaman hidroponik, robotika, dan lain sebagainya[19].

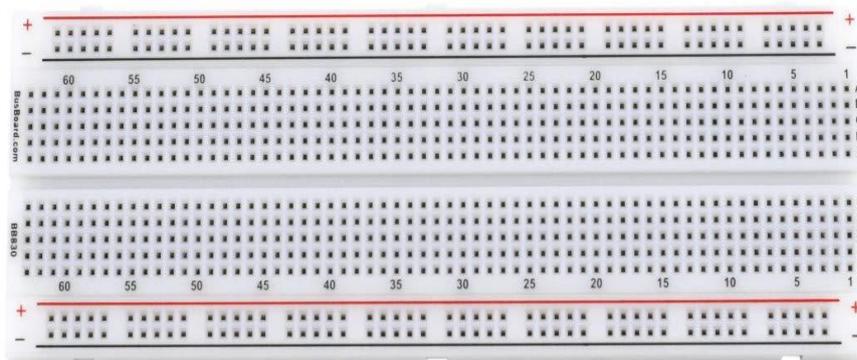


Gambar II-8 Micro Mini Submersible Water Pump[20]

2.8 Breadboard

Breadboard adalah salah satu bagian paling mendasar ketika belajar bagaimana membangun sebuah rangkaian elektronik sederhana. Breadboard elektronik sebenarnya mengacu pada breadboard tanpa solder. Ini adalah unit yang

bagus untuk membuat rangkaian sementara dan pembuatan prototipe, dimana kita sama sekali tidak memerlukan solder. Breadboard dapat menampung sirkuit yang paling sederhana maupun yang sangat kompleks. Jika rangkaian melebihi papan, kita juga dapat dengan mudah mengganti dan menambahkan komponen elektronika yang akan kita gunakan sehingga dapat menghemat waktu dan biaya[21]. Adapun breadboard ditunjukkan seperti pada Gambar 10 dibawah ini.



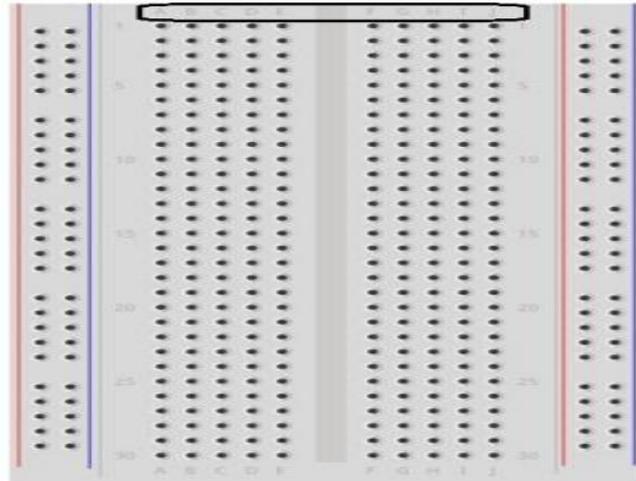
Gambar II-9 Breadboard [21]

2.8.1 Breadboard Layout

Hampir semua breadboard modern (ukuran penuh dan ukuran sedang) memiliki banyak lubang, huruf, angka, dan garis biru dan merah yang merupakan tampilan standar dari breadboard.

a. Huruf

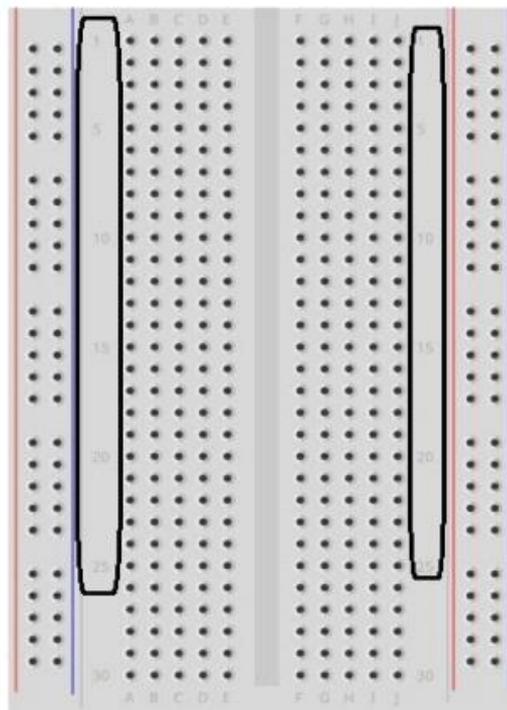
Pada bagian atas dan bawah breadboard terdapat huruf “A” hingga “J” yang tertera secara horizontal dan membagi lubang secara merata menjadi garis-garis vertikal.



Gambar II-10 Huruf Pada Breadboard [21]

b. Penomoran Baris

Untuk mengidentifikasi baris horizontal, angka dari 1 hingga 30 dicetak di tepi kiri dan kanan pada breadboard ukuran sedang, dan 1 hingga 60 atau 63 pada breadboard ukuran penuh. Beberapa breadboard menggunakan kelipatan 5, tetapi itu tidak mengubah konsep utama penomoran baris pada breadboard.



Gambar II-11 Penomoran Baris Breadboard [21]

c. Terminal Strips

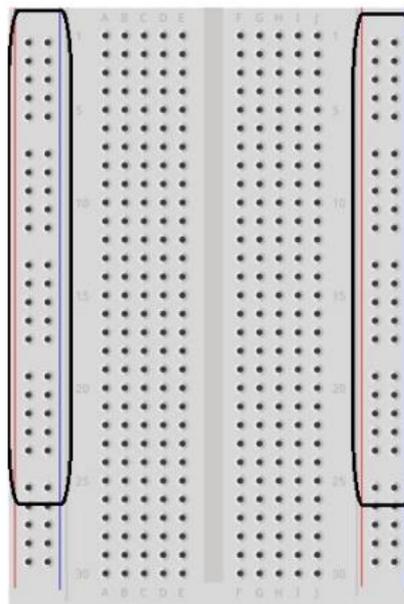
Baris atau "Terminal Strips" adalah tempat dimana seharusnya menghubungkan komponen elektronik. Baris diidentifikasi dengan angka, dan setiap baris berisi lima lubang di setiap sisi (kiri dan kanan) dipisahkan oleh pembagi tengah. Setiap lima lubang di baris yang sama terhubung dengan listrik, tetapi harus berada di sisi yang sama.

Misalnya, dalam baris nomor 10. Lubang 10A, 10B, 10C, 10D dan 10E saling terhubung, karena mereka berada di baris yang sama dan di sisi yang sama. Tetapi mereka tidak terhubung ke lubang F hingga J di sisi kanan papan, karena keduanya diisolasi oleh pembagi tengah.

d. Vertical bus

Ini terdiri dari dua kolom vertikal yang umumnya dikenal sebagai "Power rails." Setiap papan berisi dua power rail yang terletak di ujung kedua sisi di breadboard.

Tujuan dari power rails adalah untuk menghubungkan rangkaian ke catu daya eksternal. Satu kolom untuk menghubungkan sirkuit dengan tegangan positif (+) dan kolom lainnya adalah untuk menghubungkan sirkuit ke ground (-).



Gambar II-12 Power Rails [21]

e. Central Gutter

Pembagi tengah atau jurang yang terletak secara vertikal dan membagi papan secara simetris ke sisi kiri dan kanan. Pembagi ini memiliki fungsi yang sangat penting yaitu:

1. Memungkinkan paket dual-in-line (DIP) seperti IC agar pas di papan. Karena setiap pin dari IC memiliki tujuan unik, jurang ini menjaga pin diisolasi satu sama lain untuk menghindari korsleting.
2. Jurang ini memberikan pendinginan untuk komponen DIP dengan memungkinkan udara mengalir dari bawah.



Gambar II-13 Jurang Breadboard [21]

2.9 Jumper

Jumper pada komputer adalah connector atau penghubung sirkuit elektrik yang digunakan untuk menghubungkan atau memutus hubungan pada suatu sirkuit. Jumper juga digunakan untuk melakukan setting pada papan elektrik[22].



Gambar II-14 Kabel Jumper [23]

2.10 Adaptor

Adaptor yaitu piranti elektronik yang bisa mengubah tegangan listrik (AC) yang tinggi jadi tegangan listrik (DC) yang rendah, namun ada juga jenis adaptor yang bisa mengubah tegangan listrik yang rendah jadi tegangan listrik yang tinggi, dan ada banyak lagi macam-macam adaptor. Macam- macam adaptor [24] :

1. Adaptor DC Converter

DC Converter adalah adaptor yang bisa mengubah tegangan DC yang besar jadi tegangan DC yang kecil. Umpamanya : Dari tegangan 12 VDC jadi 6 VDC.

2. Adaptor Step Up serta Step Down

Adaptor Step Up yaitu adaptor yang bisa mengubah tegangan AC yang kecil jadi tegangan AC yang besar. Umpamanya : Dari Tegangan 110v jadi tegangan 220v.

Adaptor Step Down yaitu adaptor yang bisa mengubah tegangan AC yang besar jadi tegangan AC yang kecil. Umpamanya : Dari tegangan 220v jadi tegangan 110v.

Adaptor Step Up ataupun adaptor Step Down alatnya sama, tinggal bagaimana caranya kita memakainya.

3. Adaptor Inverter

Adaptor Inverter adalah adaptor yang bisa mengubah tegangan DC yang kecil jadi tegangan AC dengan ukuran besar. misal : Dari tegangan 12-v DC menjadi 220-v AC.

4. Adaptor Power Supply

Adaptor Power Supply adalah adaptor yang bisa mengubah tegangan listrik AC yang besar jadi tegangan DC yang kecil. Umpamanya : Dari tegangan 220v AC jadi tegangan 6v, 9v, atau 12 VDC.

Adaptor power supply di buat untuk menukar manfaat baterai atau accu supaya lebih ekonomis. Adaptor power supply ada yang di buat sendiri, namun ada yang di buat jadikan satu dengan rangkaian lain. Umpamanya dengan rangkaian Radio Tape, Tv, dan lain-lain.



Gambar II-15 Macam – Macam Adaptor [24]

2.11 Android

Menurut Teguh Arifianto, android merupakan perangkat bergerak pada sistem operasi untuk telepon seluler yang berbasis linux. Android menyediakan platform terbuka opensource bagi pengembang untuk menciptakan aplikasi[25].

Terdapat dua jenis distributor sistem operasi Android. Pertama yang mendapat dukungan penuh dari Google atau Google Mail Services(GMS) dan kedua adalah yang benar-benar bebas distribusinya tanpa dukungan langsung Google atau dikenal sebagai Open Handset Distribution (OHD). Android bersama Open Handset Alliance menyatakan mendukung pengembangan open source pada perangkat mobile. Diain pihak, Google merilis kode-kode Android dibawah lisensi Apache, sebuah lisensi perangkat lunak dan open platform perangkat selular.

Sistem operasi buatan Andy Rubin yang diakuisi oleh google pada 25 Agustus 2005 ini telah melakukan berbagai perbaikan dan cukup rajin dalam melakukan pembaharuan sistem. Pembaharuan sistem pada android biasanya semakin besar apabila versi dinaikan. Saat ini, android yang sejak pembaharuan ketiga menggunakan nama makanan penutup untuk setiap versinya telah memiliki beberapa tingkatan versi, yaitu:

Alpha (versi 1.0)

Beta (versi 1.1)

Cupcake (versi 1.5)

Donut (versi 1.6)

Eclair (versi 2.0 – 2.1)

Froyo (versi 2.2 – 2.2.3)

Gingerbread (versi 2.3 – 2.3.7)

Honeycomb (versi 3.0 – 3.2.6)

Ice Cream Sandwich (versi 4.0 – 4.0.4)

Jelly Bean (versi 4.1 – 4.3.1)

Kitkat (versi 4.4 – 4.4.2)

Lollipop (versi 5.0 – 5.01)

Marshmallow (versi 6.0)

Nougat (versi 7.0 – 7.1)

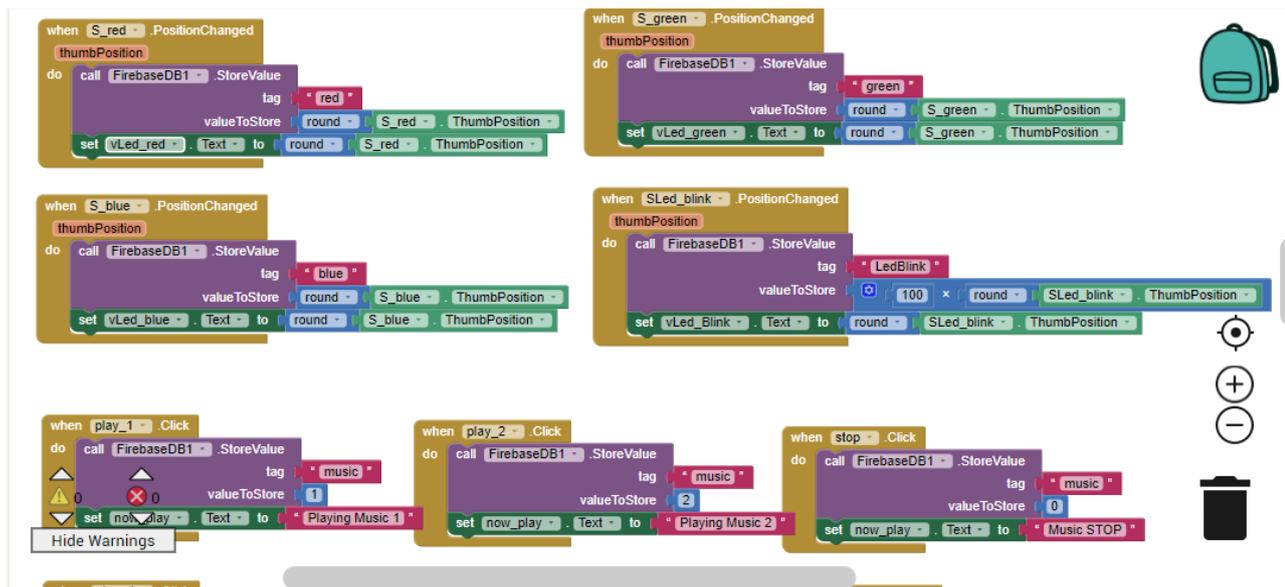
Oreo (versi 8.0 – 8.1)

Pie (versi 9.0)

Q (versi 10.0)

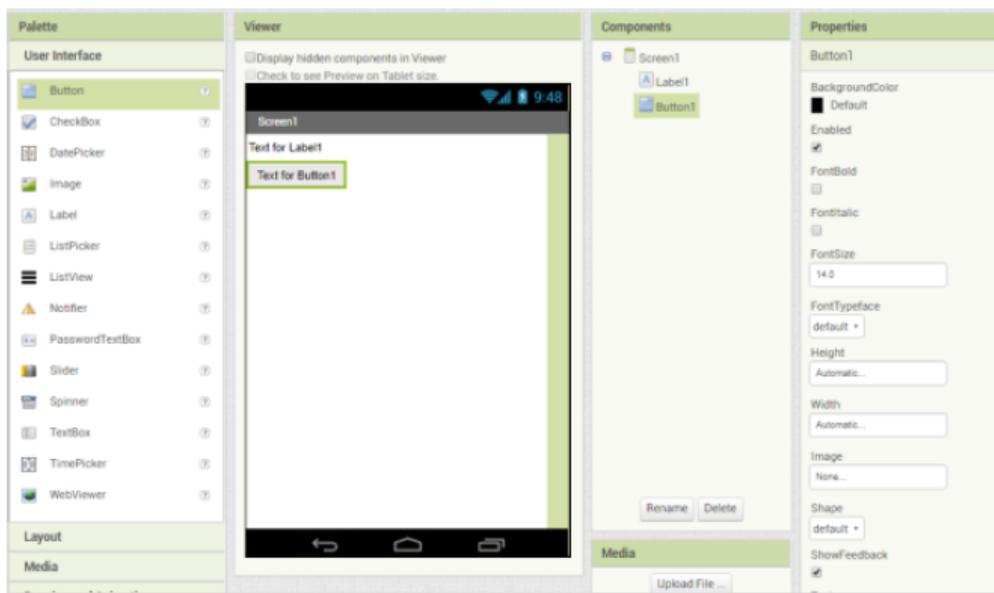
2.12 MIT App Inventor

App Inventor 2 (AI2) merupakan IDE generasi kedua dari App Inventor yang dikelola oleh Massachusetts Institute of Technology (MIT). AI2 berbasis cloud yang diakses menggunakan internet browser. Masuk kategori dalam visual programming, AI2 menggunakan block puzzle yang disusun untuk menjadi rangkaian kode.

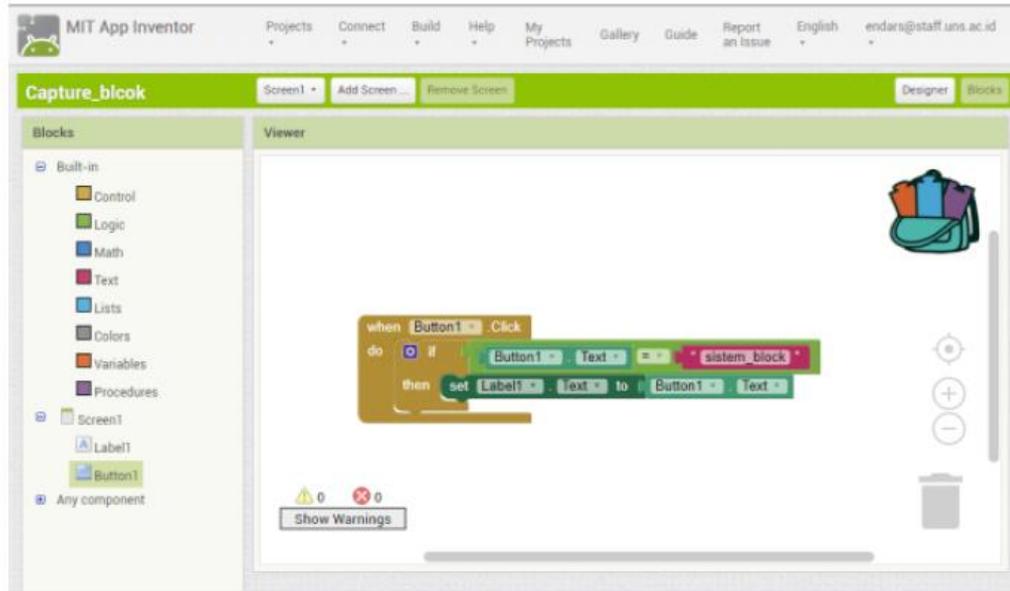


Gambar II-16 Block puzzle dalam AI2

AI2 memiliki 3 bagian utama, Component Designer, Block Editor dan Android Device yang digunakan untuk pengujian. Pengujian bisa menggunakan emulator maupun perangkat sebenarnya. Untuk perangkat sebenarnya bisa dihubungkan melalui jaringan wireless dan menggunakan USB. Component designer merupakan class dan method yang siap digunakan seperti halnya class dan method dalam bahasa pemrograman Java, hanya saja dalam AI2 dinamakan dengan komponen. Komponen tersebut adalah: (1) User Interface (2) Layout (3) Media (4) Drawing & Animation (5) Sensor (6) Social Component (7) Storage (8) Connectivity dan (9) Lego MindStorms. *Block Editor* merupakan sekumpulan blok berisi perintah untuk fungsi percabangan, perulangan, variable, array, serta beberapa kelas yang berfungsi seperti *Public Static Class*, jadi kita bisa langsung memakai metode tersebut tanpa perlu instansiasi (membuat objek) terlebih dahulu. Bila dilihat dari komponen yang telah tersedia, AI2 sudah cukup memadai untuk membangun aplikasi yang kompleks[26].



Gambar II-17 Component Designer dalam AI2



Gambar II-18 Block Editor dalam AI2

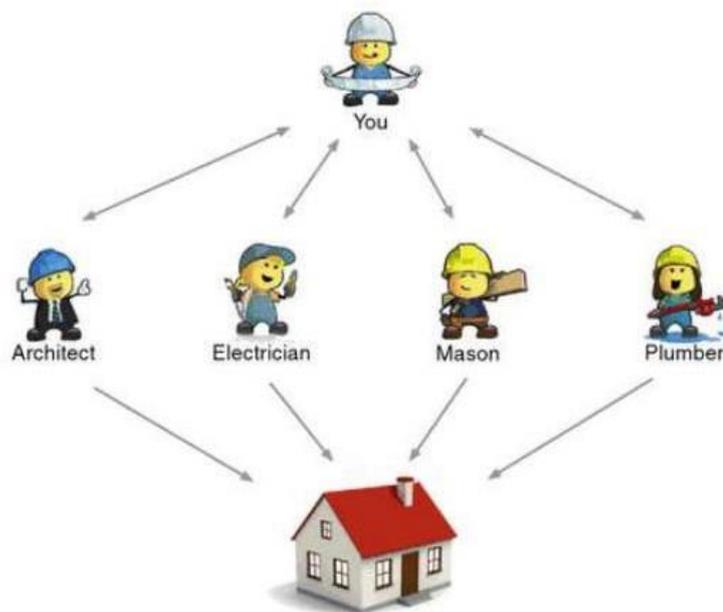
2.13 Java

Java merupakan bahasa berorientasi objek untuk pengembangan aplikasi mandiri, aplikasi berbasis internet, aplikasi untuk perangkat cerdas yang dapat berkomunikasi lewat internet/ jaringan komunikasi. Melalui teknologi java, dimungkinkan perangkat audio stereo dirumah terhubung jaringan komputer. Java tidak lagi hanya untuk membuat applet yang memerintah halaman web tapi java telah menjadi bahasa untuk pengembangan aplikasi skala interprise berbasis jaringan besar [27]. Java memiliki beberapa kelebihan, yaitu :

1. Sederhana. Java memiliki fitur-fitur singkat yang membuatnya mudah untuk dipelajari dan digunakan.
2. Aman. Java menyediakan firewall antara aplikasi jaringan dengan komputer.
3. Portable. Java dapat dijalankan di berbagai platform tanpa perlu di-compile ulang.
4. Object-oriented. Karena Java adalah bahasa pemrograman object-oriented, maka hal ini memudahkan dalam mendesain, membuat, mengembangkan dan mengalokasi kesalahan dalam program secara cepat, mudah dan terorganisir.
5. Multithreaded. Memungkinkan untuk menjalankan beberapa aktivitas di dalam Java.

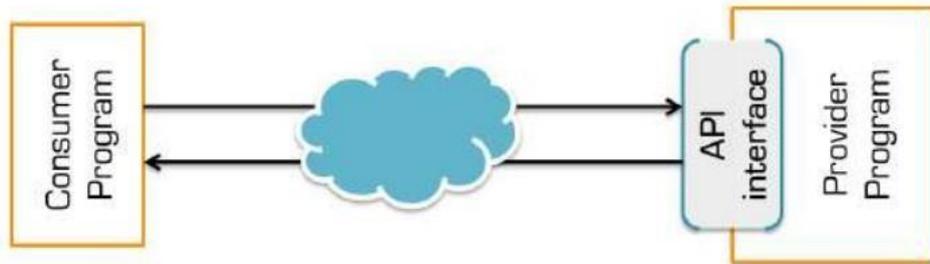
2.14 API (Application Programming Interface)

API merupakan software interface yang terdiri atas kumpulan instruksi yang disimpan dalam bentuk library dan menjelaskan bagaimana agar suatu software dapat berinteraksi dengan software lain [28]. Penjelasan ini dapat dicontohkan dengan analogi apabila akan dibangun suatu rumah. Dengan menyewa kontraktor yang dapat menangani bagian yang berbeda, pemilik rumah dapat memberikan tugas yang perlu dilakukan oleh kontraktor tanpa harus mengetahui bagaimana cara kontraktor menyelesaikan pekerjaan tersebut. Dari analogi tersebut, rumah merupakan software yang akan dibuat, dan kontraktor merupakan API yang mengerjakan bagian tertentu dari software tersebut tanpa harus diketahui bagaimana prosedur dalam melakukan pekerjaan tersebut.



Gambar II-19 Analogi API pada Pembangunan Rumah [29]

Interface pada software merupakan suatu entry points yang digunakan untuk mengakses seluruh resources yang terdapat di dalam software tersebut. Dengan adanya API, maka terdapat aturan bagaimana software dapat berinteraksi dengan software lain untuk mengakses resources melalui interface yang telah tersedia.

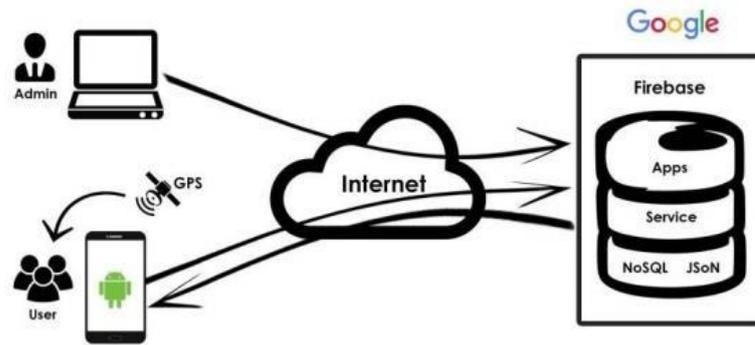


Gambar II-20 Skema Konektivitas API Antar Software [29]

Secara struktural, API merupakan spesifikasi dari suatu data structure, objects, functions, beserta parameter-parameter yang diperlukan untuk mengakses resource dari aplikasi tersebut. Seluruh spesifikasi tersebut membentuk suatu interface yang dimiliki oleh aplikasi untuk berkomunikasi dengan aplikasi lain, dan API dapat digunakan dengan berbagai bahasa programming, ataupun hanya dengan menggunakan URL (Uniform Resource Locator) yang telah disediakan oleh suatu website.

2.15 Firebase

Firebase Realtime Database adalah database yang di-host di cloud. Data disimpan sebagai JSON dan disinkronkan secara realtime ke setiap klien yang terhubung [30]. Firebase memiliki produk utama, yaitu menyediakan database realtime dan backend sebagai layanan (Backend as a Service). Layanan ini menyediakan pengembang aplikasi API yang memungkinkan aplikasi data yang akan disinkronisasi di klien dan disimpan di cloud Firebase ini. Firebase menyediakan library untuk berbagai client platform yang memungkinkan integrasi dengan Android, iOS, JavaScript, Java, Objective-C dan Node aplikasi Js dan dapat juga disebut sebagai layanan DbaaS (Database as a Service) dengan konsep realtime. Firebase digunakan untuk mempermudah dalam penambahan fitur-fitur yang akan dibangun oleh developer.



Gambar II-21 Firebase [31]

Semua data Firebase Realtime Database disimpan sebagai objek JSON. Bisa dianggap basis data sebagai JSON tree yang di-host di awan. Tidak seperti basis data SQL, tidak ada tabel atau rekaman. Ketika ditambahkan ke JSON tree, data akan menjadi simpul dalam struktur JSON yang ada. Meskipun basis data menggunakan JSON tree, data yang tersimpan dalam basis data bisa diwakili sebagai tipe bawaan tertentu yang sesuai dengan tipe JSON yang tersedia untuk membantu Anda menulis lebih banyak kode yang bisa dipertahankan.

2.16 Internet of Things

Internet of things (IOT) adalah jaringan benda fisik. Internet bukan hanya jaringan komputer, tetapi telah berkembang menjadi jaringan perangkat dari semua jenis dan ukuran, kendaraan, ponsel pintar, peralatan rumah tangga, mainan, kamera, instrumen medis dan sistem industri, hewan, manusia, bangunan, semua terhubung, semua berkomunikasi & berbagi informasi berdasarkan protokol yang ditetapkan untuk mencapai reorganisasi cerdas, penentuan posisi, penelusuran, aman & terkontrol & bahkan pemantauan online waktu nyata pribadi, peningkatan online, kontrol proses & administrasi. IoT dapat dikategorikan menjadi tiga, yaitu (1) *people to people*, (2) *people to machine*, dan (3) *machine to machine*.

Internet of Things adalah revolusi baru Internet. Objek membuat diri mereka dikenali dan mereka memperoleh kecerdasan dengan membuat atau mengaktifkan keputusan terkait konteks berkat fakta bahwa mereka dapat mengkomunikasikan informasi tentang diri mereka sendiri. Mereka dapat mengakses informasi yang telah dikumpulkan oleh hal-hal lain, atau mereka dapat menjadi komponen layanan

yang kompleks. Transformasi ini bersamaan dengan munculnya kemampuan komputasi awan dan transisi Internet menuju IPv6 dengan kapasitas pengalaman yang hampir tidak terbatas.

Tujuan dari Internet of Things adalah untuk memungkinkan segala sesuatu dapat terhubung kapan saja, di mana saja, dengan apa saja dan siapa saja yang secara ideal menggunakan jalur / jaringan dan layanan apa pun [32].

Cara kerja IoT, dengan memanfaatkan suatu argumentasi pemrograman, di mana tiaptiap perintah argumen tersebut dapat menghasilkan suatu interaksi antar mesin yang telah terhubung secara otomatis tanpa campur tangan manusia dan tanpa dibatasi oleh jarak yang jauh. Internet menjadi penghubung antara kedua interaksi mesin tersebut. Manusia dalam IoT tugasnya hanyalah menjadi pengatur dan pengawas dari mesin-mesin yang bekerja secara langsung tersebut.

Unsur-unsur pembentuk IoT yang mendasar adalah:

- Kecerdasan Buatan (Artificial Intelligence/AI), IoT membuat hampir semua mesin yang ada menjadi “Smart” (pintar). Ini berarti IoT bisa meningkatkan segala aspek kehidupan kita dengan pengembangan teknologi yang didasarkan pada AI. Pengembangan teknologi yang ada dilakukan dengan pengumpulan data, algoritma kecerdasan buatan, dan jaringan yang tersedia.
- Konektivitas dalam IoT, ada kemungkinan untuk membuat atau membuka jaringan baru, dan jaringan khusus IoT. Jaringan ini tidak lagi terikat hanya dengan penyedia utamanya saja. Jaringan ini tidak harus berskala besar dan mahal, bisa tersedia pada skala yang jauh lebih kecil dan lebih murah. IoT bisa menciptakan jaringan kecil di antara perangkat sistem.
- Sensor merupakan pembeda yang membuat IoT unik dibanding mesin canggih lainnya. Sensor ini mampu mendefinisikan instrumen, yang mengubah IoT dari jaringan standar dan cenderung pasif dalam perangkat, sehingga menjadi suatu sistem aktif yang dapat diintegrasikan ke dunia nyata dalam kehidupan sehari-hari.

- Keterlibatan Aktif (Active Engagement), IoT mengenalkan paradigma yang baru bagi konten aktif, produk, maupun keterlibatan layanan.
- Perangkat Berukuran Kecil. IoT memanfaatkan perangkat-perangkat kecil yang dibuat khusus agar menghasilkan ketepatan, skalabilitas, dan fleksibilitas yang baik[33].

2.17 UML

Unified Modelling Language merupakan alat perancangan sistem yang berorientasi pada objek. Secara filosofi kemunculan UML diilhami oleh konsep yang telah ada yaitu konsep permodelan Object Oriented (OO), karena konsep ini menganalogikan sistem seperti kehidupan nyata yang didominasi oleh obyek dan digambarkan atau dinotasikan dalam simbol-simbol yang cukup spesifik maka OO memiliki proses standard dan bersifat independen. UML diagram memiliki tujuan utama untuk membantu tim pengembangan proyek berkomunikasi, mengeksplorasi potensi desain, dan memvalidasi desain arsitektur perangkat lunak atau pembuat program[34].

2.17.1 Object Oriented Programming

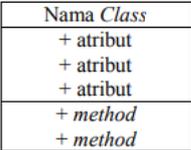
Berdasar artikel yang ditulis oleh Hartanto Kurniawan, Object Oriented Programming (OOP) merupakan paradigma pemrograman yang berorientasikan kepada object. Semua data dan fungsi pada paradigma ini dibungkus dalam class class atau object object [35]. Mengacu pada pengertian sebelumnya, OOP merupakan suatu konsep pemrograman yang mengacu pada object dimana semua data atau fungsi dibungkus dalam objek tersebut.

2.17.2 Beberapa Jenis Diagram UML

a. Class Diagram

Class diagram menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain dari suatu kelas. *Class* adalah sebuah spesifikasi yang jika di-instansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek [36]. *Class Diagram* juga bisa memperlihatkan sekumpulan *class*, *interface*, dan *collaborations* dan relasi yang ada didalamnya.

Tabel II-2. Simbol *Class Diagram*

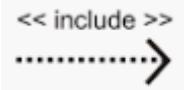
No	Simbol	Nama Komponen	Keterangan
1.		<i>Class</i>	Nama class ditempatkan pada bagian pertama (rata tengah, di-bold, dan Huruf besar), daftar atribut diletakan pada bagian kedua, dan tuliskan operasi-operasi pada class dibagian ketiga.
2.		<i>Association</i>	Associations adalah representasi/gambaran relasi statis diantara class-class. Tempatkan nama associations pada bagian atas, di, atau dibawah garis associations. Gunakan tanda anak panah yang berisi sebuah kata yang mengindikasikan relasi secara langsung. Letakan role (aturan/ketentuan) pada bagian akhir associations. Aturan merepresentasikan arah bagi kedua kelas untuk saling berhubungan satu sama lain
3.		<i>Composition</i>	Simbol yang digunakan apabila sebuah kelas tidak dapat berdiri sendiri, dan membutuhkan kelas lain, maka diperlukan relasi terhadap kelas induk.

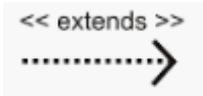
4.		<i>Dependency</i>	Simbol yang digunakan untuk menggambarkan sebuah kelas yang menggunakan kelas lain
5.		<i>Aggregation</i>	Simbol yang digunakan untuk mengindikasikan keseluruhan bagian relasi.

b. Use Case Diagram

Use case diagram merupakan pemodelan untuk kelakuan sistem informasi yang akan dibuat. *Use case* bekerja dengan mendeskripsikan tipikal interaksi antara user sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sistem itu dipakai[37].

Tabel II-3. Simbol Usecase Diagram

No	Simbol	Nama Komponen	Keterangan
1.		<i>Actor</i>	Aktor mewakili siapa pun atau apa saja yang harus berinteraksi dengan sistem. Aktor bisa didefinisikan sebagai berikut : Aktor hanya memberikan informasi kepada sistem, Aktor hanya menerima informasi dari sistem, Aktor memberikan dan menerima informasi ke dan dari sistem.
2.		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit

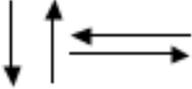
3.		<i>extends</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber
4.		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas
5.		<i>Usecase</i>	Use Case Model adalah dialog antara aktor dengan sistem yang akan menggambarkan fungsi yang diberikan oleh sistem.

c. Activity Diagram

Activity diagram merupakan diagram yang menggambarkan *workflow* atau aktivitas dari sebuah sistem yang ada pada perangkat lunak[37].

Tabel II-4. Simbol Activity Diagram

No	Simbol	Nama Komponen	Keterangan
1.		<i>Activity</i>	Notasi yang menggambarkan pelaksanaan dari beberapa proses dalam aliran pekerjaan.
2.		<i>Action</i>	<i>State</i> dari sistem yang mencerminkan eksekusi dari suatu aksi
3.		<i>Initial</i>	Titik awal, untuk memulai suatu aktivitas
4.		<i>Finish</i>	Titik akhir, untuk mengakhiri suatu aktivitas

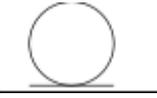
5.		<i>Decision</i>	Notasi yang menandakan kontrol cabang aliran berdasarkan decision point.
6.		<i>Life Connector</i>	Menghubungkan satu simbol dengan simbol lainnya

d. Sequence Diagram

Sequence diagram adalah diagram yang menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan output tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan.

Tabel II-5. Simbol Sequence Diagram

No	Simbol	Nama Komponen	Keterangan
1.		<i>Life line</i>	Objek entiti, antarmuka yang saling berinteraksi
2.		<i>Actor</i>	Digunakan untuk menggambarkan <i>user</i> / pengguna
3.		<i>Message</i>	Pesan yang dipertukarkan antar objek digambarkan dengan anak panah kemudian di atasnya diberikan label pesan.

4.		<i>Boundary</i>	Digunakan untuk menggambarkan sebuah form
5.		<i>Control Class</i>	Digunakan untuk menghubungkan <i>boundary</i> dengan tabel
6.		<i>Entity Class</i>	Digunakan untuk menggambarkan hubungan kegiatan yang akan dilakukan