

BAB II TINJAUAN PUSTAKA

2.1 PERBANDINGAN PENELITIAN SEJENIS

Seperti yang telah dipaparkan pada latar belakang bahwa penulis menggunakan beberapa jurnal yang sejenis pada penelitian ini sebagai pembanding juga referensi. Disini penulis membandingkan data jurnal yang diambil dari sisi keunggulan juga kelemahan masing-masing jurnal, untuk keterangan lebih lanjut bisa dibaca pada tabel dibawah ini.

Tabel 2.1 Tabel Perbandingan Penelitian Sejenis

| No | Judul Referensi | Nama Peneliti dan / Tahun | Keunggulan | Kekurangan |
|-----------|--------------------------------------------------------------------------------------------|----------------------------------|---------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | Rancang Bagun Pendeteksi Kematangan Buah jeruk Dengan Sensor TCS 3200 Berbasis Atmega 8535 | Ginting, R. / 2021 | Dapat mendeteksi otomatis kematangan pada buah jeruk berdasarkan tingkat warna. | Pada pendeteksi warna yang memiliki bentuk bulat, sebaiknya tidak hanya dilakukan pada bidang datar, agar sensor dapat mendeteksi secara keseluruhan pada objek tersebut. |

| | | | | |
|----|---------------------------------------------------------------------------------------------------------------|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------|
| 2 | Internet Of Things: Sejarah Teknologi dan Penerapannya | Farhan dan Salma / 2020 | Uraian masalah pada jurnal ini menunjukkan bahwa semakin berkembangnya teknologi internet dan memaparkan sejarah, teknologi dan penerapan IoT. | - |
| 3. | Perangkat Lunak Alat Pendeteksi Asam Manis dan Timbang Otomatis Buah Jeruk Berbasis <i>Internet of Things</i> | Siti Meirina / 2022 | Dapat mendeteksi otomatis asam manis pada buah jeruk tanpa mengupas kulitnya dengan metode warna yaitu menggunakan sensor TCS3200 dan juga dapat langsung timbang berat dan harga otomatis. | Pendeteksi warna pada objek tidak banyak memiliki variasi warna sehingga tidak kompleks. |

2.2 PERANGKAT LUNAK (*SOFTWARE*)

2.2.1 Pengertian Perangkat Lunak (*Software*)

Menurut Melwin (2007:22) , definisi perangkat lunak (*software*) adalah berfungsi sebagai pengatur aktivitas kerja komputer dan semua intruksi yang mengarah pada sistem komputer. Perangkat lunak menjembatani interaksi *user* dengan *computer* yang hanya memahami bahasa mesin.

Deskripsi buku teks tentang perangkat lunak dapat berbentuk sebagai berikut : “Perangkat lunak adalah (1) instruksi (program komputer) yang ketika dijalankan memberikan fungsi dan kinerja yang diinginkan, (2) struktur data yang memungkinkan program untuk memanipulasi informasi secara memadai, dan (3) dokumen yang menggambarkan operasi dan penggunaan program”.^[17]

2.2.2 Jenis - jenis Perangkat Lunak (*Software*)

Jika dilihat secara umum, perangkat lunak terbagi menjadi tiga jenis yakni sebagai berikut :

1. Sistem Operasi (*Operating System*)

Operating system software merupakan perangkat lunak yang berfungsi untuk mengatur setiap perangkat keras yang sudah terhubung dengan CPU komputer. Selain itu, ia juga berfungsi dalam menerjemahkan aktivitas serta mengatur seluruh proses yang terjadi dan diperintahkan oleh pengguna pada CPU sehingga perintah pengguna tersebut dapat dikerjakan komputer dengan baik.

2. Bahasa Pemrograman (*Programming Language*)

Jenis perangkat lunak yang satu ini adalah bahasa yang digunakan dalam pembuatan sebuah program. Dalam bahasa pemrograman ini terdapat beberapa macam bahasa, seperti JAVA, PHP, HTML, C, C++, dan masih banyak lainnya. Dalam membuat bahasa pemrograman ini nantinya masih memerlukan perangkat lunak khusus untuk merancang program sesuai metode dan struktur yang ada pada bahasa pemrograman itu sendiri.

3. Program Aplikasi (*Application Software*)

Jenis perangkat lunak ini merupakan sebuah perangkat lunak yang umumnya sudah banyak digunakan dalam menyelesaikan tugas - tugas tertentu. Dalam hal ini, software dapat berfungsi sesuai dengan tujuan pembuatannya. Salah satu contohnya adalah *Microsoft Word, Excel*, dan yang lainnya.

2.3 *INTERNET OF THINGS (IOT)*

2.3.1 Pengertian *Internet of Things (IoT)*



Gambar 2.1 Ilusi *Internet of Things*^[3]

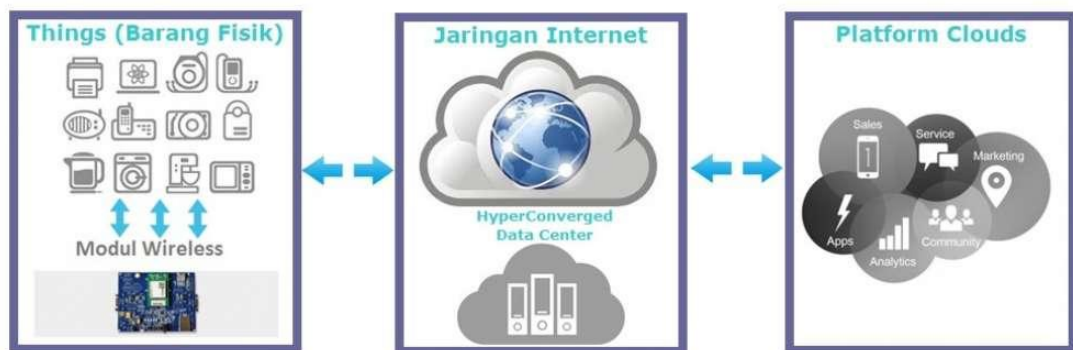
Internet of Things (IoT) adalah konsep komputasi tentang objek sehari-hari yang terhubung ke internet dan mampu mengidentifikasi diri ke perangkat lain. Koneksi internet adalah hal yang luar biasa, bisa memberi kita segala macam manfaat yang sebelumnya mungkin sulit untuk didapat. Ambil ponsel kamu sebelum menjadi smartphone sebagai contoh. Kamu bisa menelpon dan mengirim pesan teks dengan ponsel lama. Tapi, sekarang kamu bisa membaca buku, menonton film, atau mendengarkan musik lewat smartphone kamu yang terhubung dengan internet. Istilah "*Internet of Things*" terdiri atas dua bagian utama yaitu internet yang mengatur konektivitas dan *Things* yang berarti objek atau perangkat. Secara sederhana, kamu memiliki "*Things*" yang memiliki kemampuan untuk mengumpulkan data dan mengirimkannya ke internet. Data ini dapat diakses oleh "*Things*" lainnya juga. Jadi, *Internet of Things* sebenarnya

adalah konsep yang cukup sederhana, yang artinya menghubungkan semua objek fisik di kehidupan sehari-hari ke internet.^[3]

2.3.2 Sistem pada *Internet of Things (IoT)*

Sistem dasar pada *Internet of Things (IoT)* terdiri dari tiga hal, yakni sebagai berikut :

1. Barang fisik yang dilengkapi modul IoT.
2. Perangkat koneksi ke internet seperti modem dan router wireless speedy.
3. *Cloud Data Center* sebagai tempat untuk menyimpan data dan menjalankan aplikasi.



Gambar 2.2 Cara Kerja *Internet of Things (IoT)*^[4]

Seluruh penggunaan barang yang terhubung ke internet akan menyimpan data. Data tersebut terkumpul sebagai “big data” yang kemudian dapat di olah untuk di analisa baik oleh pemerintah, perusahaan, maupun negara asing untuk kemudian di manfaatkan bagi kepentingan masing - masing.^[4]

Adapun didapat banyak sekali contoh penerapan teknologi dari *Internet of Things (IoT)* , yakni sebagai berikut :

1. Internet Industri, yaitu untuk monitoring dan controlling peralatan serta proses industri.
2. Kesehatan, yaitu untuk monitoring kondisi kesehatan seseorang.
3. Smart Home, yaitu sebuah sistem keamanan rumah berbasis internet yang dapat mengetahui keadaan rumah serta mengontrol peralatan rumah tangga melalui jaringan internet.

4. Transportasi, yaitu untuk memanajemen dan pemberi informasi seputar lalu lintas.

2.3.3 *Adafruit I/O*



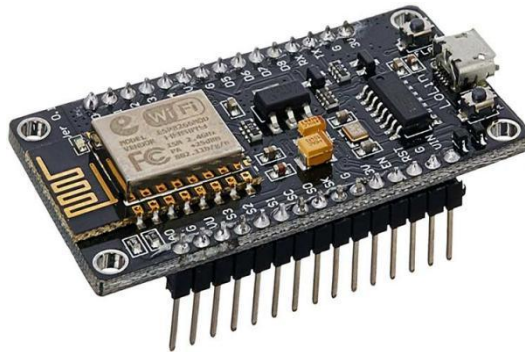
Gambar 2.3 Tampilan Menu *Adafruit I/O*^[5]

Adafruit I/O adalah platform IoT yang dibangun menggunakan prinsip Protokol MQTT (*Message Queue Telemetry Transport*). MQTT adalah protokol ringan yang memungkinkan beberapa perangkat untuk terhubung ke server bersama, yang disebut MQTT *Broker*, dan berlangganan atau menulis ke topik yang ditentukan pengguna. Ketika suatu perangkat berlangganan suatu topik, broker akan mengirimkannya pemberitahuan kapan pun topik itu berubah. MQTT paling cocok untuk aplikasi dengan jecepatan data rendah, kendala daya yang ketat atau koneksi internet yang tidak begitu cepat.^[5]

Adapun terdapat beberapa fungsi dasar yang dapat dilakukan *Adafruit I/O* :

1. Menampilkan data secara *real time*
2. Membuat project internet - connected : kontrol motor, membaca sensor data dan fungsi lainnya.
3. Hubungkan proyek ke layanan web seperti Twitter, RSS feed, layanan cuaca, dan sebagainya.
4. Hubungkan proyek Anda ke perangkat lain yang mendukung internet.

2.4 NODEMCU ESP8266



Gambar 2.4 NodeMCU ESP8266^[6]

NodeMCU ESP8266 merupakan modul mikrokontroler yang didesain dengan ESP8266 di dalamnya. ESP8266 berfungsi untuk konektivitas jaringan Wifi antara mikrokontroler itu sendiri dengan jaringan Wifi. NodeMCU berbasis bahasa pemrograman Lua namun dapat juga menggunakan Arduino IDE untuk pemrogramannya ^[6].

Alasan pemilihan NodeMCU ESP8266 karena mudah diprogram dan memiliki pin I/O yang memadai dan dapat mengakses jaringan internet untuk mengirim atau mengambil data melalui koneksi Wifi ^[6].

Modul ini membutuhkan daya sekitar 3,3 V dengan memiliki tiga mode wifi yaitu Station, Access Point dan 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.

Firmware default yang digunakan oleh perangkat ini menggunakan *AT Command*, selain itu ada beberapa *Firmware SDK* yang digunakan oleh perangkat ini berbasis open source yang diantaranya adalah sebagai berikut :

1. NodeMCU dengan menggunakan *basic programming lua*.
2. MicroPython dengan menggunakan *basic programming python*.
3. AT Command dengan menggunakan perintah AT Command.

Menurut (Widyaman Tresna, 2016), untuk pemrogramannya sendiri bisa menggunakan ESPlorer untuk Firmware berbasis NodeMCU dan menggunakan putty sebagai terminal control untuk AT Command. Selain itu kita bisa memprogram perangkat ini dengan Arduino IDE. Dengan menambahkan library ESP8266 pada board manager kita dapat dengan mudah memprogram dengan basic program arduino. Ditambah lagi dengan harga yang cukup terjangkau, kamu dapat membuat berbagai proyek dengan modul ini. Maka dari itu, banyak orang yang menggunakan modul ini untuk membuat proyek yang berbasis *Internet of Things* (IoT).

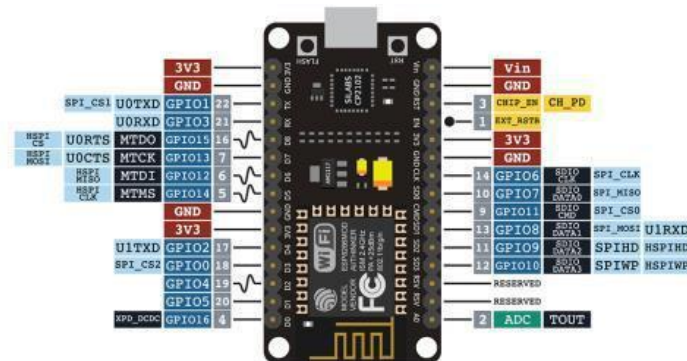
2.4.1 Spesifikasi Umum NodeMCU ESP8266

Spesifikasi yang dimiliki oleh NodeMCU ESP8266 sebagai berikut :

Tabel 2.2 Spesifikasi Umum NodeMCU ESP8266^[6]

| | |
|----------------|--------------------|
| Mikrokontroler | ESP8266 |
| Tegangan Input | 3.3~5V |
| GPIOg | 17 Pin |
| Flash Memory | 16 MB |
| RAM | 32KB + 80KB |
| Konsumsi Daya | 10 μ A~170mA |
| Frekuensi | 2.4 GHz - 22.5 Ghz |
| USB Port | Micro USB |
| Wifi | IEEE 802.11b/g/n |
| Kanal PWM | 10 Kanal |
| USB Chip | CH340G |
| Clock Speed | 40/26/24 MHz |

NodeMCU memiliki 17 Pin GPIO yang dapat diintegrasikan dengan komponen elektronika lainnya. Bekerja pada tegangan 3.3V - 5V, dengan konsumsi daya 10 μ A~170mA. Kecepatan prosessor berkisar 80 - 160MHz dan memiliki RAM sebesar 32KB + 80KB serta flash memory hingga 16 MB membuat NodeMCU V1 lebih efisien dari versi sebelumnya.^[6]



Gambar 2.5 Pinout NodeMCU ESP8266^[6]

2.5 ANDROID

2.5.1 Pengertian Android



Gambar 2.6 Logo Android^[7]

Android adalah sistem operasi yang dirancang oleh *Google* dengan basis kernel *Linux* untuk mendukung kinerja perangkat elektronik layar sentuh, seperti tablet atau smartphone. Jadi, *android* digunakan dengan sentuhan, gesekan ataupun ketukan pada layar gadget anda. Android bersifat open source atau bebas digunakan, dimodifikasi, diperbaiki dan didistribusikan oleh para pembuat ataupun pengembang perangkat lunak. Dengan sifat open source perusahaan teknologi bebas menggunakan OS ini diperangkatnya tanpa lisensi alias gratis^[7].

Seperti software lain, *Android* memiliki nomor versi. Selain nomor versi, Google juga menyematkan nama julukan untuk setiap versinya yang berbasiskan

nama makanan penutup. Jika anda sudah lama mengenal Android, tentu hal ini bukanlah hal aneh.

2.5.2 Sejarah Android

Android adalah sistem operasi yang berbasis *Linux* untuk telepon seluler seperti telepon pintar dan komputer tablet. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak [8].

Awalnya, Google Inc. Membeli Android Inc. Yang merupakan pendatang baru yang membuat peranti lunak untuk ponsel atau smartphone. Kemudian untuk mengembangkan Android, dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan peranti keras, peranti lunak dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia [9].

Android dirilis pertama kali pada tanggal 5 November 2007, *Android* bersama *Open Handset Alliance* menyatakan mendukung pengembangan *open source* pada perangkat *mobile*. Di lain pihak, *Google* merilis kode-kode *Android* di bawah lisensi *Apache*, sebuah lisensi perangkat lunak dan open platform perangkat lunak.

2.5.3 Perkembangan Sistem Operasi *Android* Dari Masa ke Masa

Perkembangan *Android* sendiri saat ini cukup pesat, *Android* selalu melakukan *upgrade* melalui perbaikan bug dan penambahan fitur-fitur baru. Berikut ini adalah perkembangan sistem operasi *android* dari masa ke masa :

1. Android 1.0 Astro (Alpha)

Android pertama kali lebih dikenal dengan Android Alpha. Dirilis pada 23 Desember 2008, yang sebenarnya versi awal dari Android ini akan dinamakan Astro, tetapi karena ada masalah hak cipta atas nama Astro, sehingga berubah nama menjadi Android Alpha.

2. Android 1.1 Blender

Android versi kedua ini dirilis pada 30 Februari 2009. Sama dengan sebelumnya, versi ini punya masalah penamaan karena Bender ternyata

sudah ada yang punya dan dipatenkan. Awalnya, versi ini dirilis cuma untuk perangkat T-Mobile G1.

3. Android 1.5 Cupcake

Android Cupcake pertama kali dirilis pada 30 April 2009 dan pada versi ini ada peningkatan pada sisi pencarian dan UI yang lebih bersahabat dan mendukung teknologi CDMA.

4. Android 1.6 Donut

Pada awal perilisan Android banyak memiliki bug, dan perlu pengembangan serta perbaikan. Hal ini dilakukan pada Android 1.6 Donut yang dirilis pada 15 September 2009, yang artinya belum genap setahun semenjak perilisan Android 1.5 Cupcake atau hanya berselang lima bulan saja. Android pun menambahkan beberapa pembaruan, terutama dukungan pada layar ponsel yang lebih besar..

5. Android 2.0 dan 2.1 Eclair

Pada tahap ini terjadi penambahan fitur untuk peningkatan Google Maps, perubahan UI ditambah dengan browser baru, daftar kontak baru, dukungan flash untuk kamera 3.2MP dan bluetooth 2.1. Versi ini dirilis pada tanggal 9 Desember 2009. Sama seperti versi sebelumnya, Android 2.0 & 2.1 Eclair masih berfungsi untuk menutupi bug yang masih ditemukan pada sistem operasi mobile ini. Android 2.0 & 2.1 Eclair digunakan pada perangkat seperti HTC Nexus One.

6. Android 2.2 Frozen Yoghurt (Froyo)

Pada versi ini Android sudah mendukung penggunaan SD Card sebagai penyimpanan kedua dan sudah mendukung Adobe Flash Player 10.1. Froyo dirilis pada 20 Mei 2010 pada Google Nexus One. Android 2.2 Froyo memberikan peningkatan pada kecepatan kerja, fitur USB tethering, WiFi hotspot, serta fitur keamanan.

7. Android 2.3 Gingerbread

Ada banyak peningkatan yang terjadi pada versi ini ketimbang sebelumnya. Gingerbread dirancang untuk memaksimalkan kemampuan aplikasi dan game dan mulai digunakannya NFC (Near Field Communication). Versi ini dirilis pada 6 Desember 2010 dan memiliki pengguna terbanyak dibanding versi sebelum-sebelumnya.

8. Android 3.0 dan 3.2 Honeycomb

Pada versi ini, Android memfokuskan pada pengoptimalan pada tablet PC. Pertama kali diperkenalkan ke publik pada 22 Februari 2011 dan smartphone yang pertama kali mencicipinya adalah Motorola Xoom.

9. Android 4.0 Ice Cream Sandwich

Pertama kali dirilis pada 19 Oktober 2011 dan Ice Cream Sandwich (ICS) langsung menjadi versi Android yang paling anyar. ICS juga pertama kali muncul di salah satu produk Samsung, Galaxy Nexus 7.

10. Android 4.1 dan 4.3 Jelly Bean

Pada versi ini Android membawa pembaruan yang cukup signifikan. Fitur-fitur baru pun ditambahkan seperti input keyboard, desain baru dari fitur pencarian, UI yang baru dan pencarian via suara menjadi lebih cepat. Sistem operasi ini sendiri pertama kali dirilis pada Juni 2012 dengan membawa sejumlah peningkatan terutama di sektor pengolahan grafis.

11. Android 4.4 KitKat

Google butuh waktu setahun untuk bisa melahirkan KitKat. Salah satu alasannya karena Google merombak habis-abisan UI versi Jelly Bean dan pada tahap ini juga Android memiliki teknologi "Ok, Google" yang mendapat banyak pujian dari para pakar teknologi.

Versi Android boleh dibilang yang terbaik karena menjadi favorit yang mendukung hampir seluruh perangkat smartphone di dunia. Sebab,

Android 4.4 KitKat dapat memberikan optimalisasi yang baik, termasuk pada perangkat ponsel yang memiliki spesifikasi kurang mumpuni alias cukup rendah saat itu. Android 4.4 KitKat pertama kali dirilis pada Oktober 2013.

12. Android 5.0 dan 5.1 Lollipop

Android dan Google pun mulai secara rutin memperbarui sistem operasi mereka dalam selang waktu setahun. Termasuk Android 5.0 dan 5.1 Lollipop yang dirilis dan diresmikan pada Juni 2014. Versi Android ini sudah mendukung arsitektur 64-bit yang sudah memungkinkan penggunaan RAM di atas 3GB. Salah satunya Asus Zenfone 2 yang sudah mengusung RAM 4GB saat itu.

13. Android 6.0 Marshmallow

Android 6.0 Marshmallow juga menjadi suksesor dari versi Android sebelumnya. Sistem operasi ini sendiri pertama kali diperkenalkan pada Mei 2015 dan mulai dirilis pada Oktober 2015. Sistem operasi ini secara jelas memberikan peningkatan pada sistem keamanan dengan dihadirkannya fingerprint sensor sebagai sistem keamanan biometrik yang digunakan.

14. Android 7.0 dan 7.1 Nougat

Android 7.0 dan 7.1 Nougat pertama kali diperkenalkan pada Juni 2016 dengan menampilkan ikon robot Android dengan batangan Nougat. Sistem operasi Android 7.0 dan 7.1 Nougat mengalami perubahan dari segi tampilan antarmuka. Selain itu ada juga fitur splitscreen untuk membagi tampilan layar untuk dua aplikasi sekaligus.

15. Android 8.0 dan 8.1 Oreo

Android 8.0 dan 8.1 Oreo menjadi sistem operasi Android paling terbaru yang banyak digunakan saat ini. Sistem operasi ini dirilis secara

stabil mulai Agustus 2017 sudah mengalami pembaruan lewat versi Android 8.1 Oreo.

Sistem operasi ini menawarkan pengalaman multitasking yang makin mumpuni dibanding versi sebelumnya. Selain itu, ada juga Project Treble yang memungkinkan pengguna mendapat pembaruan lebih cepat.

16. Android 9.0 Pie

Android 9.0 Pie yang secara resmi diperkenalkan pada Agustus 2018. Sistem operasi Android ini memberi banyak ubahan, terutama untuk HP dengan desain baru. Misal Android 9.0 Pie memberikan navigasi berupa gesture yang menggantikan tombol fisik Home, Back, dan Recent Apps.

17. Android 10 (Android Q Beta)

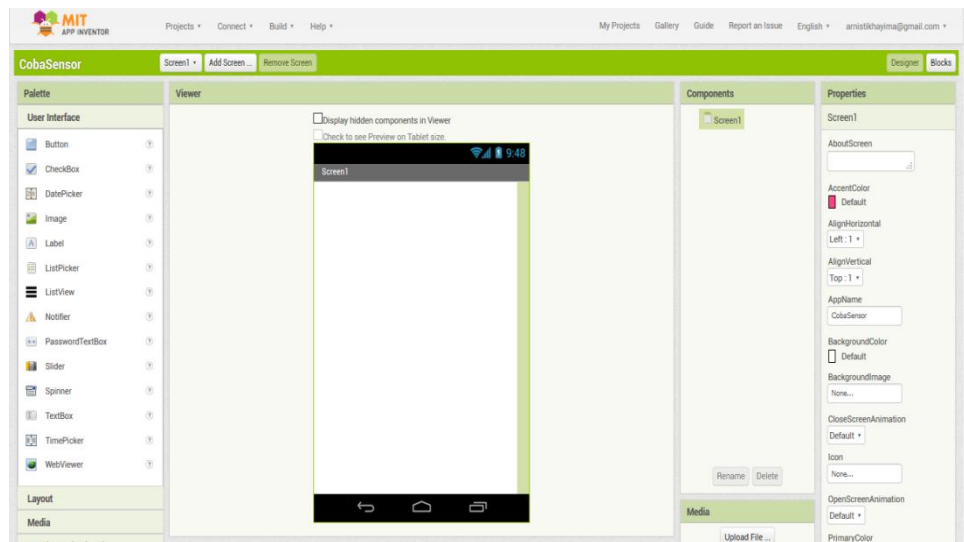
Android Q Beta yang diluncurkan pada 13 Maret 2019 dan saat ini masih terbatas pada beberapa perangkat HP Android saja, seperti pada seri smartphone Google, yakni Google Pixel, Google Pixel XL, Google Pixel 2, Google Pixel 2 XL, Google Pixel 3, Google Pixel 3 XL, dan Google Pixel 3 Lite. Salah satu fitur Android Q Beta adalah Dark Mode alias mode gelap yang diklaim mampu meningkatkan performa baterai.

18. Android 11

Android 11 adalah rilis utama kesebelas dan versi ke-18 Android, sistem operasi telepon genggam yang dikembangkan oleh Open Handset Alliance yang dipimpin oleh Google. Sistem operasi ini dirilis pada 8 September 2020.^[10]

2.6 MIT APP INVENTOR

2.6.1 Pengertian MIT App Inventor



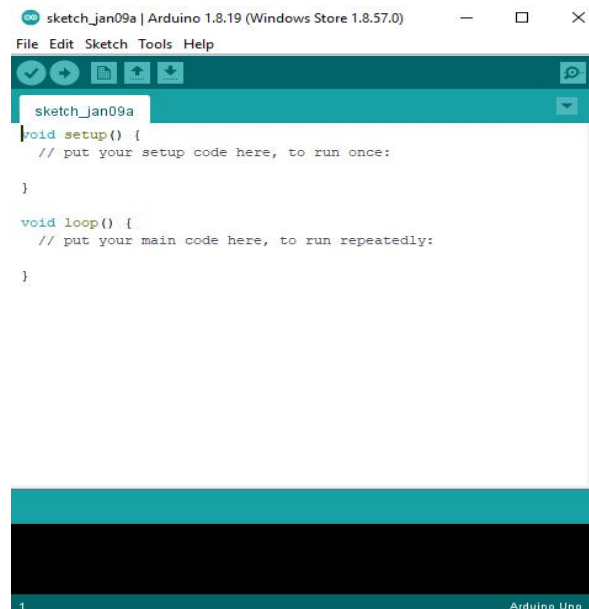
Gambar 2.7 Tampilan Awal MIT App Inventor^[11]

App Inventor merupakan aplikasi *opensource* berbasis web yang disediakan oleh google dan sekarang dikelola oleh Massachusetts Institute of Technology (MIT). MIT App Inventor (*Integrated Development Environment*) Merupakan sebuah aplikasi IDE berbasis *drag and drop visual block programming* yang memungkinkan kita untuk membuat dan mengembangkan aplikasi android tanpa harus memiliki pengalaman dalam bahasa pemrograman. Fitur *Visual Block Programming* yang ada pada MIT App Inventor ini mampu mengubah bahasa pemrograman berbasis teks menjadi sebuah block puzzle yang dapat kita susun sesuai dengan tindakan yang kita inginkan.

Dengan app inventor, pengguna bisa melakukan pemrograman komputer untuk menciptakan aplikasi perangkat lunak dengan sistem operasi berbasis android. App inventor ini berbasis visual block programming karena memungkinkan pengguna bisa menggunakan, melihat, menyusun dan men-drag and drops block yang merupakan simbol perintah dan fungsi event handler untuk menciptakan sebuah aplikasi yang bisa berjalan di sistem android. App inventor ini juga tidak hanya bisa membantu dalam menciptakan sebuah aplikasi namun juga bisa membantu anda dalam mengasah logika. Beberapa contoh aplikasi yang bisa dibuat dengan app inventor seperti game dengan grafis 2D dan animasi, utility,

aplikasi multimedia, navigasi dan GPS, sistem pengamanan, dan masih banyak lagi [12].

2.7 ARDUINO IDE (*Integrated Development Environment*)









Gambar 2.8 Tampilan *Sketch Software* Arduino IDE^[13]

Arduino IDE (*Integrated Development Environment*) adalah *software open source* yang dikembangkan oleh Arduino untuk memrogram Arduino. Arduino IDE ini berguna sebagai text editor untuk membuat, mengedit, dan juga memvalidasi kode program. Bisa juga digunakan untuk meng-upload ke board Arduino. Kode program yang digunakan pada Arduino disebut dengan istilah Arduino “sketch” atau disebut juga source code arduino, dengan ekstensi file source code. ino. 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 penengah antara *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. Arduino IDE terdiri dari:

1. Editor program, sebuah window yang memungkinkan pengguna menulis dan mengedit program dalam bahasa C++ yang disederhanakan, yang merupakan turunan dari proyek *open source* Wiring.
2. Compiler, sebuah modul yang mengubah kode program menjadi kode biner. Bagaimanapun sebuah mikrokontroller tidak akan bisa memahami bahasa C++. Yang bisa dipahami oleh mikrokontroller adalah kode biner. Itulah sebabnya compiler diperlukan dalam hal ini.
3. Uploader, sebuah modul yang memuat kode biner dari komputer ke dalam memori di dalam papan Arduino.^[14]

Tabel 2.3 Menu *Software* Arduino IDE

| Simbol | Keterangan |
|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------|
|  | Verify, sebagai pemeriksa kode Anda untuk kesalahan kompilasi itu. |
|  | Upload, untuk mengkompilasi kode Anda dan upload ke papan dikonfigurasi. Lihat upload di bawah ini untuk rincian. |
|  | New, untuk membuat sketsa baru. |
|  | Open, untuk membuka file yang sudah ada. |
|  | Save, untuk mengamankan sketsa Anda yang sudah dibuat. |
|  | Serial Monitor, untuk membuka monitor serial. |

2.7.1 Pemrograman Arduino IDE

Pemrograman arduino menggunakan struktur Bahasa C. Mekanisme pemrogramannya arduino sama dengan mikrokontroler pada umumnya. Mulai dari membuat sketsa program, meng-compile, selanjutnya proses *upload* pada papan arduino. Pengisian program dengan metode upload ialah mengisi papan arduino dengan program yang sudah berbentuk Hex atau hasil *compile* dari bahasa C ke bahasa mesin ^[15].

Program Arduino dapat dibagi dalam tiga bagian utama : struktur, nilai - nilai (variabel dan konstanta) dan fungsi.

1. Struktur Utama

a. Setup()

Fungsi `setup()` dipanggil ketika sketsa program dimulai. Fungsi ini digunakan untuk menginisialisasikan variabel, mode pin, penggunaan librari, dll. Fungsi `setup()` hanya akan berjalan sekali, setelah power arduino dinyalakan atau saat mereset papan Arduino.

Contoh : Program 1.1

```
int ledPin = 13;
void setup () {
    pinMode (ledPin, OUTPUT) ;
    digitalWrite (ledPin, HIGH) ;
    delay (5000) ;
    digitalWrite (ledPin, HIGH) ;
}
void loop () {
    // ...
}
```

Program 1.1 akan menyalakan LED pada pin 13 selama 5 detik lalu mati. Eksekusi ini dilakukan hanya sekali.

b. Loop ()

Setelah membuat fungsi setup(), maka berikutnya adalah fungsi loop(). Fungsi loop () akan melakukan loop berturut - turut dimana program akan dijalankan terus menerus secara berututan dan loop untuk mengontrol papan Arduino.

Contoh : Program 1.2

```
const int buttonPin = 3 ;
void setup () {
    Serial.begin(9600) ;
    pinMode (buttonPin, INPUT) ;
}
void loop () {
    if (digitalRead (buttonPin) == HIGH)
        Serial.write ('H') ;
    else
        Serial.write ('L') ;
    delay (1000) ;
}
```

Program 1.2 pada serial monitor akan menampilkan huruf H ketika tombol pada pin 3 ditekan dan bila dilepaskan akan tampil huruf L.

c. //Komentar

Komentar digunakan untuk memberikan keterangan pada program yang dibuat. Komentar tidak dieksekusi maka komentar tidak menambah ukuran file hasil *compile*. Cara membuat komentar ialah sebagai berikut :

```
//komentar segaris diawali dengan dua garis miring
/*komentar untuk lebih dari satu baris diawali dengan garis
miring lalu tanda bintang serta diakhiri dengan bintang lalu garis
miring*/
```

2. Ekspresi Bila

Dalam pemrograman bahasa C pada Arduino, bilangan tepat diekspresikan dalam beberapa format, yaitu :

a. Biner

Ditulis dengan awalan huruf '0b'. Contoh : b11110010

b. Desimal

Ditulis biasa tanpa awalan. Contoh : 435

c. Oktal

Ditulis dengan awalan angka '0'. Contoh : 0753

d. Heksadesimal

Diawali dengan '0x'. Contoh : 0x5A

3. Struktur Kontrol

Setiap program yang dibuat membutuhkan suatu kontrol. Tak hanya perulangan namun suatu eksekusi dengan syarat - syarat tertentu juga diperlukan.

Pengujian Kondisi :

a. if

Digunakan untuk mengecek suatu kondisi. Jika benar maka perintah didalam if akan dikerjakan.

```
if (kondisi) {  
Pernyataan / perintah  
}  
Contoh :  
If (x==6) {  
a=a+5 ;  
}
```

b. if - else

Seperti dengan if, hanya saja ada 2 pilihan pernyataan / perintah. Jika kondisi benar maka perintah didalam if akan dikerjakan, jika kondisinya salah maka pernyataan didalam else lah yang akan dikerjakan.

| | |
|---------------------------------|---------------------------|
| if(kondisi){ | Contoh : if(x==1) {a=1; } |
| Pernyataan / perintah 1 }else { | else { |
| Pernyataan / perintah 2 } | a=0; } |

c. if - else if

Untuk melakukan pengecekan suatu kondisi lebih dari satu maka bisa menggunakan if - else if.

| | |
|-------------------------------------|-------------------|
| if(kondisi1){ | Contoh : if(x==1) |
| Pernyataan / perintah 1 }else | {a=1; } |
| if(kondisi2){ Pernyataan / perintah | else |
| 2 }else if(kondisi ke-n){ | if(x==2){a=2; } |
| Pernyataan / perintah ke-n } | else if(x==3){ |
| | a=3; } |

d. Switch case

Pernyataan ini digunakan untuk memilih kondisi yang sesuai untuk kemudian mengerjakan perintahnya. Bedanya adalah kondisi yang diuji berupa sebuah nilai variabel.

| | |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> case 1 : //pernyataan/perintah 1 break; case 2 : //pernyataan/perintah 2 break; case n : //pernyataan/perintah n break; default : //pernyataan/perintah default } </pre> | <pre> Contoh : switch(a){ case 1 : digitalWrite(pin1,HIGH) break; case 2 : digitalWrite(pin2,HIGH) break; case 3 : digitalWrite(pin3,HIGH) break; default : digitalWrite(pin4,LOW) } </pre> |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Jika variabel memenuhi syarat dari salah satu case maka dia akan mengerjakan pernyataan / perintah tersebut. Misal nilai variabel = 2, maka dia akan mengerjakan pernyataan / perintah 2. Jika tidak memenuhi maka dia akan mengerjakan default.

4. Perulangan

a. While

Perulangan ini digunakan untuk membuat perulangan yang tidak terbatas selama kondisi selama kondisi dalam while benar.

| | |
|-------------------------|------------------------|
| while(kondisi){ | Contoh : while(a<200){ |
| //pernyataan/perintah } | a++; } |

Perulangan while akan berhenti atau keluar setelah a mencapai angka 200.

b. do ... while

Perulangan ini akan melakukan pernyataan / perintah lalu akan melihat kondisi dalam while. Jika benar maka pernyataan / perintah akan dieksekusi kembali.

| | |
|-------------------------|-----------------|
| do{ | Contoh : do{ |
| //pernyataan/perintah } | a++; } |
| while (kondisi) ; | while (a<200) ; |

Perulangan penambahan a + 1 akan dilakukan sampai nilai a = 200.

c. For

Digunakan untuk perulangan yang sifatnya terbatas.

| | |
|---------------------------------|-----------------------------|
| for(inisialisasi;kondisi;step){ | Contoh :for(a=0;a<=10;a++){ |
| //pernyataan/perintah } | Serial.println(a); } |

Inisialisasi : nilai awal suatu variabel untuk proses perulangan.

Kondisi : kondisi yang menentukan proses perulangan, jika benar perulangan dikerjakan.

Step : tahap perulangan bisa dalam bentuk perkalian, penambahan, pengurangan, dan pembagian. Program tersebut akan menampilkan nilai a dari 0 sampai 10.

d. Goto

Perintah ini digunakan untuk melompat atau menuju perintah yang telah diberi lael.

```
goto label ;
Contoh :
while (1) { digitalWrite(pin0, HIGH) ;
delay (1000) ;
digitalWrite(pin0, LOW) ;
delay (1000) ;
if(digitalRead(pin1)==HIGH) ;
{ goto keluar;} } keluar:
```

e. Return

Digunakan untuk memberikan nilai baik dari sebuah fungsi.

```
Contoh : int
data() { if(analogRead(A0)>100)
{ return 1 ;
else
return 0 ; }
```


f. Continue

Untuk melewati perulangan yang tersisa dari struktur looping (do, for, atau while).

Contoh :

```
for (a=0;a<=255;a+10) { if(digitalRead(pin0)--HIGH) { continue ;
}
digitalWrite(pwm1,a) ;
delay(100) ;
}
```

g. Break

Perintah “keluar” dari pernyataan perulangan do, for atau while. Juga digunakan untuk mengakhiri pernyataan dalam switch - case.

5. Syntax

a. Semicolon

Digunakan untuk mengakhiri sebuah pernyataan.

b. {} (curly braces)

Bagian utama dari bahasa pemrograman C yang digunakan dalam beberapa konstruksi yang berbeda dalam beberapa fungsi.

c. #define

Komponen C yang berguna yang memungkinkan *programmer* untuk memberi nama untuk nilai konstan sebelum program dikompilasi.

d. #include

Digunakan untuk memasukkan perpustakaan atau *library* di luar di sketsaprogram.

6. Operasi Aritmatika

Tabel 2.4 Operasi Aritmatika

| Operasi Aritmatika Operator | Keterangan |
|-----------------------------|-----------------|
| = | Pemberian Nilai |
| + | Penjumlahan |
| - | Pengurangan |
| * | Perkalian |
| / | Pembagian |
| % | Sisa Bagian |

7. Operator Perbandingan

Tabel 2.5 Operasi Perbandingan

| Operator | Keterangan |
|----------|------------------------------------------------------------------------------|
| == | Persamaan. Jika kedua nilai yang dibandingkan sama maka hasilnya „true“ |
| != | Pertidaksamaan. Jika kedua nilai yang dibandingkan tidak samahasilnya „true‘ |
| > | Lebih besar |
| < | Lebih kecil |
| >= | Lebih besar atau sama dengan |
| <= | Lebih kecil atau sama dengan |

8. Operator Boolean

Tabel 2.6 Operator Boolean

| Operator | Keterangan |
|----------|------------|
| && | AND |
| | OR |
| ! | NOT |

9. Operator Bitwise

Bitwise Operator digunakan untuk operasi bit per bit pada nilai integer. Terdiri dari operator NOT, AND, OR, XOR, Shl, Shr. Type : int atau char.

Bitwise operator, dari namanya sudah jelas diketahui pasti berhubungan dengan bit. Biasanya digunakan untuk memanipulasi data bertipe bit dari tipe bit data integer atau byte.

Tabel 2.7 Operator Bitwise

| Operator | Keterangan |
|----------|-------------|
| << | Geser Kiri |
| >> | Geser Kanan |
| & | AND |
| | OR |
| ^ | XOR |
| ~ | NOT |

10. Operator Pertambahan dan Pengurangan

Tabel 2.8 Operator Pertambahan dan Pengurangan

| Operator | Keterangan | Contoh | Keterangan |
|----------|----------------------------------|--------|------------|
| ++ | Pertambahan 1 / <i>increment</i> | a++ | a = a + 1 |
| -- | Pengurangan 1 / <i>decrement</i> | a-- | a = a - 1 |

11. Variabel

Variabel adalah suatu wadah untuk menyimpan atau menampung data. Nama variabel dibebaskan namun ada peraturan tersendiri seperti tidak boleh ada spasi, maksimal 32 karakter dan tidak boleh menggunakan istilah baku dalam bahasa C arduino karena dapat tersaji program yang eror. Cara mendeklarasikan variabel sebelum digunakan yaitu sebagai berikut :

```
[tipe data][spasi][nama variabel][=][nilai]
```

```
Int nilai_1;
```

Contoh :

```
Int nilai_1=17;
```

```
//variabel bilangan tipe integer diisi nilai 17
```

12. Tipe Data

Tipe data yang berbeda - beda memiliki kapasitas penyimpanan yang berbeda - beda pula. Berikut tipe data tersebut :

Tabel 2.9 Tipe - tipe Data

| Tipe data | Lebar Data | Jangkauan |
|---------------|------------|--------------|
| Char | 1 byte | -128 s/d 127 |
| unsigned char | 1 byte | 0 s/d 255 |
| Byte | 1 byte | 0 s/d 255 |

| | | |
|---------------|--------|----------------------------------|
| Word | 2 byte | 0 s/d 65535 |
| Int | 2 byte | -32768 s/d 32767 |
| unsigned int | 2 byte | 0 s/d 65535 |
| Long | 4 byte | -2147438648 s/d 2147438647 |
| unsigned long | 4 byte | 0 s/d 4294967295 |
| Float | 4 Byte | -3.4028235E+38 s/d 3.4028235E+38 |

13. Pin Input dan Output

Pada papan Arduino Uno terdapat 20 pin I/O yaitu 14 pin digital dan 6 pin analog.

1. Inisialisasi Fungsi Pin I/O

Pada saat yang sama, sebuah pin hanya bisa memiliki satu fungsi saja baik input maupun output. Inisialisasi ini dilakukan pada fungsi `setup()`, dengan cara :

`pinMode(pin, mode)`

- a. Pin : nomor pin yang dikonfigurasi dari papan arduino.
- b. Mode : INPUT, INPUT_PULLUP, OUTPUT

Sebagai contohnya, jika pin no 3 akan dibuat menjadi input, maka :
`pinMode(3,INPUT);` bila pin 3 menjadi output, maka :
`pinMode(3,OUTPUT);`

Catatan : penulisan besar dan kecilnya huruf sangat berpengaruh. Perhatikan dengan seksama saat menulis program.

2. Menulis Data Digital di Pin Output

Setelah membuat pin sebagai digital output, selanjutnya untuk menulis atau mengeluarkan logika data digital dengan perintah sebagai berikut :

| | |
|------------------------------------|------------------------------|
| digitalWrite(pin,value); | Sebagai Contoh : |
| a. Pin : nomor pin digital output. | pinMode(3,OUTPUT); |
| b. Value : HIGH atau LOW. | digitalWrite(3,HIGH); |

3. Membaca Data Digital pada Pin Input

Jika sebuah pin dibuat sebagai input, maka kita harus menentukan aktif HIGH atau aktif LOW. Jika aktif HIGH maka dibutuhkan resistor *pull down*. Jika memilih aktif LOW, cukup dengan memanggil resistor internal dengan *pull up* pada setiap pin arduino. Sebelum melakukan pembacaan maka perlu diatur untuk masukkannya.

pinMode(pin, mode)

- a. Pin : nomor pin yang dikonfigurasi dari papan arduino.
- b. Mode : INPUT, INPUT_PULLUP.

Setelah itu, dilanjutkan dengan :

digitalRead(pin)

- c. Pin : nomor pin arduino yang digunakan sebagai input.

Contoh : Pin 3 digunakan sebagai input *pull up*.

```
int baca ;
pinMode(3, INPUT_PULLUP);
baca = digitalRead(3) ;
serial.println(baca);
```

Hasil pembahasan pin 3 maka disimpan pada variabel baca.

4. Menulis Data Analog di Pin Output PWM

Untuk emnggunakan `analogWrite()`, tidak perlu menggunakan `pinMode()` untuk mengatur pin sebagai output.

Cara menggunakannya sebagai berikut :

```
analogWrite(pin,value);
```

- a. Pin : nomor pin arduino yang digunakan sebagai output.

Contoh : Pin 5 digunakan sebagai output pwm
`analogWrite(5,100);`

- b. Value : nilai pwm mulai dari 0 - 255.

5. Membaca Data Analog di Pin Input ADC (*Analog to Digital Converter*)

Untuk menggunakan `analogRead()`, tidak perlu menggunakan `pinMode()` untuk mengatur pin sebagai input.

Cara menggunakannya sebagai berikut :

```
analogRead(analogPin);
```

- a. `analogPin` : nomor pin arduino yang digunakan sebagai input analog (A0, A1, A2, A3, A4, A5).

Contoh : membaca nilai analog pada analogPin A0 dan ditampilkan pada Serial.

```
Int val = analogRead(A0); Serial.println(val);
```

14. Time

a. Millis()

Menghitung dengan satuan milliseconds sejak papan Arduino mulai menjalankan program hingga 50 hari setelah itu akan kembali ke nol begitupun selanjutnya.

```
unsigned long time = millis();  
Serial.println(time);
```

b. Micros()

Menghitung dengan satuan microseconds sejak papan Arduino mulai menjalankan program hingga 70 menit setelah itu akan kembali ke nol begitupun selanjutnya.

```
unsigned long time = millis();  
Serial.println(time);
```

c. Delay()

Jeda program untuk jumlah waktu (dalam milidetik). (Ada 1000 milidetik dalam satu detik).

```
digitalWrite (ledPin, TINGGI); //set LED on  
delay (1000) ; //menunggu untuk kedua  
digitalWrite (ledPin, LOW) ; set LED off  
delay (1000) ; //menunggu untuk kedua
```


d. delayMicroseconds()

Jeda program untuk jumlah waktu (dalam mikrodetik). Ada seribu mikrodetik di milidetik, dan satu juta mikrodetik dalam detik.

```
digitalWrite (ledPin, TINGGI); //set LED on
delayMicroseconds(50) ; //menunggu
digitalWrite (ledPin, LOW) ; set LED off
delayMicroseconds(50) ; //menunggu
```

15. External Interrupts

1. attachInterrupt()

Anda harus menggunakan digitalPinToInterrupts (pin) untuk menerjemahkan pin digital sebenarnya untuk jumlah interrupt tertentu.

Misalnya, jika Anda terhubung ke pin 3, menggunakan digitalPinToInterrupt(3) sebagai parameter pertama yang attachInterrupt.

Tabel 2.10 Digital Pins Usable for Interrupts pada attachInterrupt

| Board | Digital Pins Usable For Interrupts |
|-----------------------------------|------------------------------------|
| Uno, Nano, Mini, other 328-based | 2, 3 |
| Mega, Mega2560, MegaADK | 2, 3, 18, 19, 20, 21 |
| Micro, Leonardo, other 32u4-based | 0, 1, 2, 3, 7 |
| Zero | all digital pins, except 4 |
| MKR1000 Rev.1 | 0, 1, 4, 5, 6, 7, 8, 9, A1, A2 |
| Due | all digital pins |
| 101 | all digital pins |

Sintaknya adalah :

Tabel 2.11 Sintak attachInterrupt

| | |
|--------------------------------------------------------------|----------------------------------------------------|
| attachInterrupt (digital Pin To Interrupt (pin), ISR, mode); | <i>(Disarankan)</i> |
| attach Interrupt (interrupt, ISR, mode); | <i>(Tidak disarankan)</i> |
| attachInterrupt (pin, ISR, mode); | <i>(Hanya Arduino Due, Zero, MKR1000,101 saja)</i> |

Parameternya :

Tabel 2.12 Parameter attachInterrupt

| | | |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------|
| Interupsi : | Jumlah Interrupt (int) | |
| Pin : | Nomor pin | <i>(Arduino Zero, MKR1000 saja)</i> |
| ISR : | ISR untuk panggilan ketika interupsi terjadi ; fungsi ini harus ada parameter dan kembali apa - apa. Fungsi ini kadang - kadang disebut sebagai <i>rutin layanan interrupt</i> . | |
| Modus : | Interrupt harus dipice. Empat konstanta yang telah ditetapkan sebagai nilai - nilai yang valid : 1. LOW untuk memicu interupsi setiap kali pin rendah, 2. CHANGE memicu interupsi setiap kali pim perubahan nilai, 3. RISING untuk memicu ketika pin ganti adri rendah ke tinggi. 4. FALLING ketika pin ganti dari tinggi ke rendah. | |

| | | |
|--|------------------------------------------------------------------------------------------------------------|--------------------------------------|
| | The papan Arduino Due memungkinkan untuk : HIGH untuk memicu interupsi setiap kali pin yang tinggi. | (<i>Arduino Zero, MKR1000</i> saja) |
|--|------------------------------------------------------------------------------------------------------------|--------------------------------------|

Contoh :

```

const byte ledPin = 13 ;
const byte interruptPin = 2;
volatile byte state = LOW ;
void setup()
{ pinMode(ledPin, OUTPUT) ;
  pinMode(interruptPin, INPUT_PULLUP) ;
attachInterrupt(digitalPinToInterrupt(interruptPin),blink,
CHANGE) ; }
void loop() {
  digitalWrite(ledPin, state) ; }
void blink() {
  state = !state: }

```

2. detachInterrupt()

Mematikan interupsi yang diberikan

Sintak :

Tabel 2.13 Sintak detachInterrupt

| | |
|------------------------------------------------|-------------------------------------|
| detachInterrupt (<i>interrupt</i>) | |
| detachInterrupt (digitalPinToInterrupt (pin)); | |
| detachInterrupt (<i>pin</i>) | (<i>Arduino Karena, Nol saja</i>) |

16. Interrupts

1. Interrupts()

Mengaktifkan kembali interupsi (setelah dinonaktifkan oleh `noInterrupts()`). Interupsi memungkinkan melakukan tugas - tugas penting tertentu di belakang program utama dan diaktifkan secara default. Beberapa fungsi tidak akan bekerja saat interupsi dinonaktifkan dan komunikasi yang masuk dapat diabaikan.

Contoh :

```
void setup() {}  
void loop() {  
  noInterrupts() ; //critical, time-sensitive code here  
  Interrupts() ; //other code here }
```

2. noInterrupts()

Menonaktifkan interupsi (dapat mengaktifkan kembali mereka dengan `interrupts()`). Interupsi memungkinkan melakukan tugas - tugas penting tertentu di belakang program utama dan diaktifkan secara default.

Contoh :

```
void setup() {}  
void loop() {  
  noInterrupts() ;  
  //critical, time-sensitive code here  
  interrupts() ;  
  // other code here }
```

17. Communication

Komunikasi serial merupakan komunikasi dua arah dari transmitter dengan receiver dan sebaliknya. Kita bisa melakukan komunikasi serial dengan memanfaatkan pin Rx dan Tx pada arduino maupun pada USB. Cara menggunakan serial sebagai berikut.

```
Contoh :   void setup() {  
           Serial.begin(9600) ; }  
  
           void loop() {  
           Unsigned char a = a++ ;  
           Serial.println(a);  
           }
```