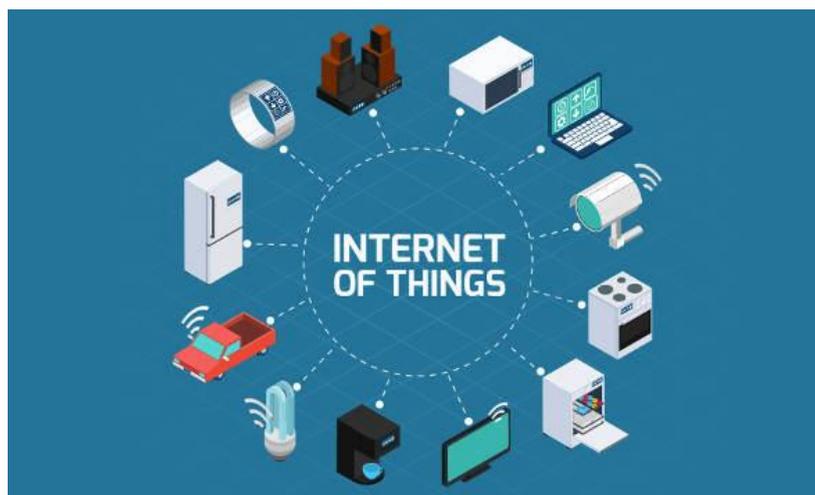


## BAB II TINJAUAN PUSTAKA

### 2.1 *Internet of Things (IoT)*

Internet of Things (IoT) adalah sebuah istilah yang muncul dengan pengertian sebuah akses perangkat elektronik melalui media internet. Akses perangkat tersebut terjadi akibat hubungan manusia dengan perangkat atau perangkat dengan perangkat dengan memanfaatkan jaringan internet. Akses perangkat tersebut terjadi karena keinginan untuk berbagi data, berbagi akses, dan juga mempertimbangkan keamanan dalam aksesnya.

Menurut (Peter Waher, 2015), Internet of Things (IoT) dimanfaatkan sebagai media pengembangan kecerdasan akses perangkat di dunia industri, di rumah tangga, dan beberapa sektor yang sangat luas dan beragam (contoh : sektor keamanan, dan sektor transportasi). Internet of Things (IoT) dapat dikembangkan dengan media perangkat elektronika yang umum seperti arduino untuk keperluan yang spesifik (khusus). IoT juga dapat dikembangkan aplikasi terpadu dengan sistem operasi android.



**Gambar 2.1.** Ilustrasi dari penggunaan IoT

## 2.2 NodeMCU ESP32

Menurut (Elga Aris Prasetyo, 2019), NodeMCU ESP32 dibuat oleh *Espressif Systems*, NodeMCU ESP32 adalah sistem dengan biaya yang rendah, berdaya rendah pada seri chip (SoC) dengan Wi-Fi & kemampuan Bluetooth dua mode. Keluarga ESP32 termasuk chip ESP32-D0WDQ6 (dan ESP32-D0WD), ESP32-D2WD, ESP32-S0WD, dan sistem dalam paket (SiP) ESP32-PICO-D4. Pada intinya, ada mikroprosesor Tensilica Xtensa LX6 *dual-core* atau *single-core* dengan *clock rate* hingga 240 MHz. ESP32 sudah terintegrasi dengan *built-in antenna switches, RF balun, power amplifier, low-noise receive amplifier, filters, and power management modules*. Didesain untuk perangkat seluler, perangkat elektronik yang dapat dipakai, dan aplikasi IoT, ESP32 juga bekerja dengan konsumsi daya sangat rendah melalui fitur hemat daya termasuk *fine resolution clock gating, multiple power modes, and dynamic power scaling*. Module ESP32 merupakan penerus dari module ESP8266 yang cukup populer untuk Aplikasi IoT. Pada ESP32 terdapat inti CPU serta Wi-Fi yang lebih cepat, GPIO yang lebih, dan mendukung *Bluetooth Low Energy*.



**Gambar 2.2** NodeMCU ESP32

## Fitur dan Spesifikasi ESP32:

### 1. Processors:

1. Main processor: Tensilica Xtensa 32-bit LX6 microprocessor
2. Cores: 2 or 1 (depending on variation)
3. Clock frequency: up to 240 MHz
4. Performance: up to 600 DMIPS
5. Ultra low power co-processor: Memungkinkan untuk melakukan konversi ADC, komputasi, dan ambang batas level saat tidak diaktifkan.

### 2. Wireless connectivity:

1. Wi-Fi: 802.11 b/g/n/e/i (802.11n @ 2.4 GHz up to 150 Mbit/s)
2. Bluetooth: v4.2 BR/EDR and Bluetooth Low Energy (BLE)

### 3. Memory:

#### a. Internal memory:

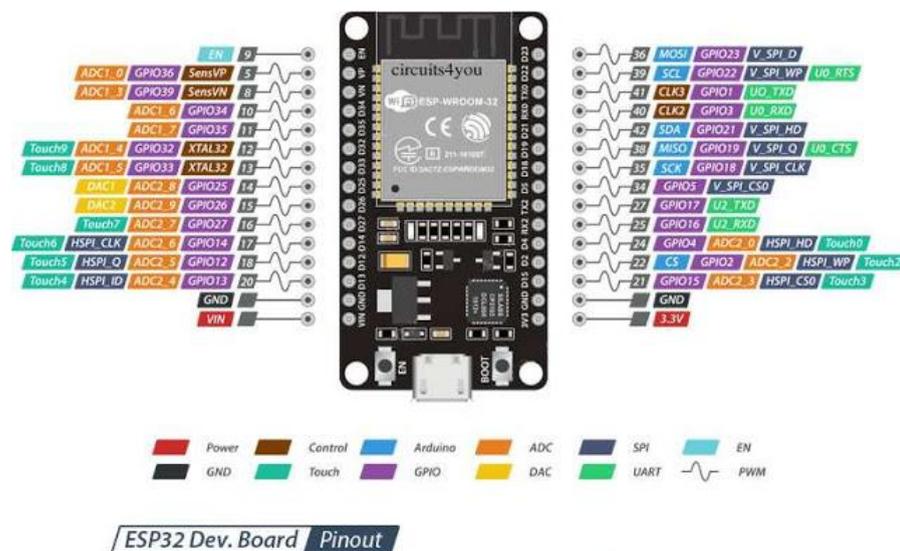
1. ROM: 448 KiB
2. SRAM: 520 KiB
3. RTC fast SRAM: 8 KiB
4. RTC slow SRAM: 8 KiB
5. eFuse: 1 Kibit

b. External flash & SRAM ESP32 mendukung hingga empat flash QSPI eksternal 16 MiB dan SRAM dengan enkripsi perangkat keras berdasarkan AES untuk melindungi program dan data. ESP32 dapat mengakses flash QSPI eksternal dan SRAM melalui cache berkecepatan tinggi.

c. Up to 16 MiB of external flash are memory-mapped onto the CPU code space, supporting 8-bit, 16-bit and 32-bit access. Code execution is supported.

d. Up to 8 MiB of external flash/SRAM memory are mapped onto the CPU data space, supporting 8-bit, 16-bit and 32-bit access. Data-read is supported on the flash and SRAM. Data-write is supported on the SRAM.

4. Peripheral input/output: Rich peripheral interface with DMA that includes capacitive touch, ADCs (analog-to-digital converter), DACs (digital-to-analog converter), I<sup>2</sup>C (Inter-Integrated Circuit), UART (universal asynchronous receiver/transmitter), CAN 2.0 (Controller Area Network), SPI (Serial Peripheral Interface), I<sup>2</sup>S (Integrated Inter-IC Sound), RMI (Reduced Media-Independent Interface), PWM (pulse width modulation), and more.
5. Security:
  1. IEEE 802.11 standard security features all supported, including WPA, WPA/WPA2 and WAPI
  2. Secure boot
  3. Flash encryption
  4. 1024-bit OTP, up to 768-bit for customers
  5. Cryptographic hardware acceleration: AES, SHA-2, RSA, elliptic curve cryptography (ECC), random number generator (RNG).



**Gambar 2.3** Pin NodeMCU ESP32

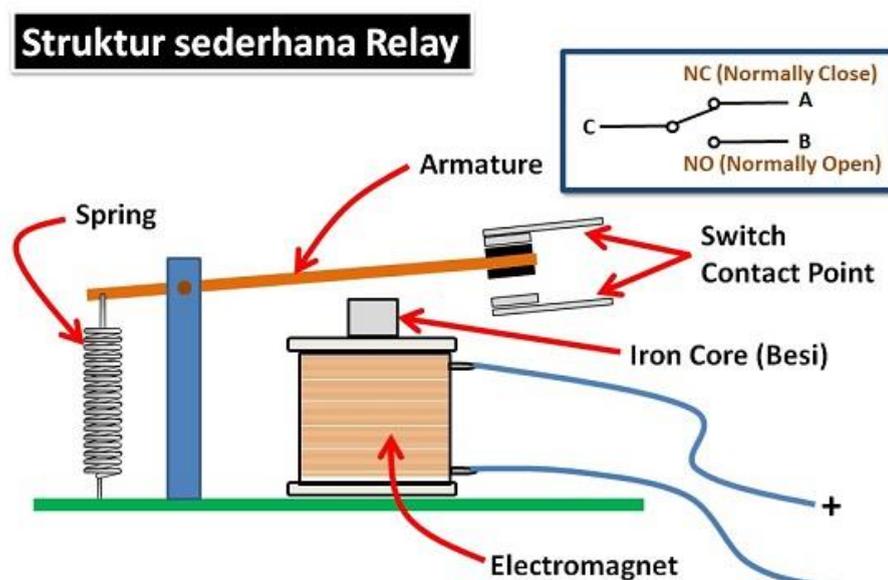
### 2.3 Relay

Menurut (Dickson kho, 2018), Relay adalah Saklar (*Switch*) yang dioperasikan secara listrik dan merupakan komponen Electromechanical (Elektromekanikal) yang terdiri dari 2 bagian utama yakni Elektromagnet (Coil) dan Mekanikal (seperangkat Kontak Saklar/Switch). Relay menggunakan Prinsip Elektromagnetik untuk menggerakkan Kontak Saklar sehingga dengan arus listrik yang kecil (*low power*) dapat menghantarkan listrik yang bertegangan lebih tinggi.

Pada dasarnya, Relay terdiri dari 4 komponen dasar yaitu :

1. Electromagnet (Coil)
2. Armature
3. Switch Contact Point (Saklar)
4. Spring

Berikut ini merupakan gambar dari bagian-bagian Relay:



**Gambar 2.4** Struktur sederhana Relay

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

1. Normally Close (NC) yaitu kondisi awal sebelum diaktifkan akan selalu berada di posisi CLOSE (tertutup)

2. Normally Open (NO) yaitu kondisi awal sebelum diaktifkan akan selalu berada di posisi OPEN (terbuka)

Berdasarkan gambar diatas, sebuah Besi (Iron Core) yang dililit oleh sebuah kumparan Coil yang berfungsi untuk mengendalikan Besi tersebut. Apabila Kumparan Coil diberikan arus listrik, maka akan timbul gaya Elektromagnet yang kemudian menarik Armature untuk berpindah dari Posisi sebelumnya (NC) ke posisi baru (NO) sehingga menjadi Saklar yang dapat menghantarkan arus listrik di posisi barunya (NO). Posisi dimana Armature tersebut berada sebelumnya (NC) akan menjadi OPEN atau tidak terhubung. Pada saat tidak dialiri arus listrik, Armature akan kembali lagi ke posisi Awal (NC). Coil yang digunakan oleh Relay untuk menarik Contact Point ke Posisi Close pada umumnya hanya membutuhkan arus listrik yang relatif kecil.

Karena Relay merupakan salah satu jenis dari Saklar, maka istilah Pole dan Throw yang dipakai dalam Saklar juga berlaku pada Relay. Berikut ini adalah penjelasan singkat mengenai Istilah Pole and Throw :

1. Pole : Banyaknya Kontak (*Contact*) yang dimiliki oleh sebuah relay
2. Throw : Banyaknya kondisi yang dimiliki oleh sebuah Kontak (*Contact*)

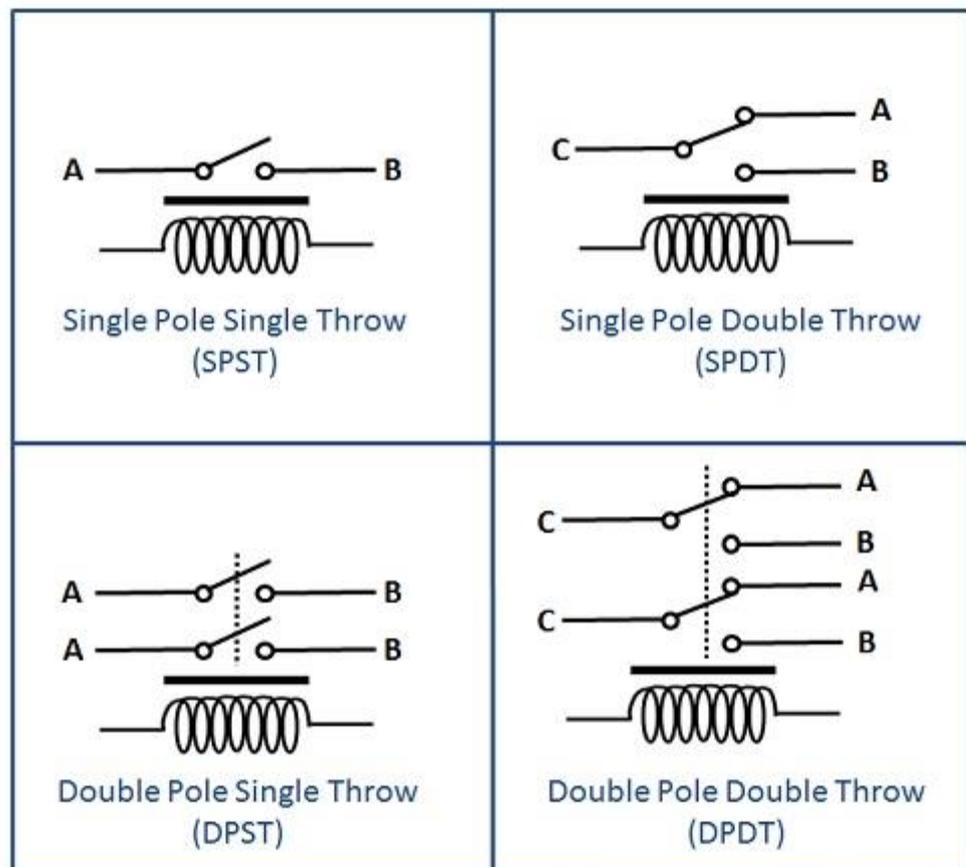
Berdasarkan penggolongan jumlah Pole dan Throw-nya sebuah relay, maka relay dapat digolongkan menjadi :

1. *Single Pole Single Throw (SPST)* : Relay golongan ini memiliki 4 Terminal, 2 Terminal untuk Saklar dan 2 Terminalnya lagi untuk Coil.
2. *Single Pole Double Throw (SPDT)* : Relay golongan ini memiliki 5 Terminal, 3 Terminal untuk Saklar dan 2 Terminalnya lagi untuk Coil.
3. *Double Pole Single Throw (DPST)* : Relay golongan ini memiliki 6 Terminal, diantaranya 4 Terminal yang terdiri dari 2 Pasang Terminal Saklar sedangkan 2 Terminal lainnya untuk Coil. Relay DPST dapat dijadikan 2 Saklar yang dikendalikan oleh 1 Coil.
4. *Double Pole Double Throw (DPDT)* : Relay golongan ini memiliki Terminal sebanyak 8 Terminal, diantaranya 6 Terminal yang merupakan 2 pasang Relay SPDT yang dikendalikan oleh 1 (single) Coil. Sedangkan 2 Terminal lainnya untuk Coil.

Selain Golongan Relay diatas, terdapat juga Relay-relay yang Pole dan Throw-nya melebihi dari 2 (dua). Misalnya 3PDT (Triple Pole Double Throw) ataupun 4PDT (Four Pole Double Throw) dan lain sebagainya.

Untuk lebih jelas mengenai Penggolongan Relay berdasarkan Jumlah Pole dan Throw, silakan lihat gambar dibawah ini:

### Jenis Relay berdasarkan Pole dan Throw



**Gambar 2.5** Penggolongan Relay berdasarkan Pole dan Throw

Beberapa fungsi Relay yang telah umum diaplikasikan kedalam peralatan Elektronika diantaranya adalah :

1. Relay digunakan untuk menjalankan Fungsi Logika (*Logic Function*)
2. Relay digunakan untuk memberikan Fungsi penundaan waktu (*Time Delay Function*)

3. Relay digunakan untuk mengendalikan Sirkuit Tegangan tinggi dengan bantuan dari Signal Tegangan rendah.
4. Ada juga Relay yang berfungsi untuk melindungi Motor ataupun komponen lainnya dari kelebihan Tegangan ataupun hubung singkat (Short).

## 2.4 Android

Menurut (Satya Komatineni dan Dave MacLean, 2012), android merupakan Operating System (OS) mobile open source yang tumbuh di tengah OS lainnya yang berkembang dewasa ini. Android menawarkan sebuah lingkungan yang berbeda untuk pengembang. Android merangkul semua ide mengenai komputasi serbaguna untuk perangkat genggam. Android merupakan Platform yang lengkap dimana OS berbasis linux menangani pengaturan kerja perangkat, memory, dan proses. Sementara Java Libraries Android menangani proses telephony, video, speech, graphic, connectivity, UI programming, dan beberapa aspek lain dari perangkat genggam tersebut.

Sedangkan menurut (Putra, 2013), di awal pembuatannya android ditargetkan bagi penggunaan perangkat kamera digital. Akan tetapi, para pencipta Android, yaitu Andy Rubin, Chris White, dan Nick Sears berpendapat bahwa pasar untuk kamera digital tidak terlalu besar. Maka dari itu, sistem operasi ini kemudian dialihkan penggunaannya pada ponsel pintar. Pada tahun 2004, Android mulai dipasarkan dan berhadapan dengan saingan *smartphone* berbasis sistem operasi Symbian dan Windows Mobile. Di awal pemasarannya ini, Andy Rubin dan *partner*-nya sulit mendapatkan investor. Hingga akhirnya, Android berhasil mendapatkan suntikan dana sebesar 10.000 dolar Amerika dari Steve Perlman, seseorang yang kala itu ingin membantu Andy Rubin. Di bulan Juli 2005, Google mengakuisisi Android Inc. dengan uang sebesar 50 juta dolar. Para pendiri Android kemudian bergabung dengan Google dan memimpin proyek ini. Setelah Google akhirnya berkompetisi juga dalam perangkat ponsel pintar yang dibelinya, yaitu Android, Google akhirnya membuat prototipe.

Prototipe tersebut merupakan *smartphone* yang memiliki *keyboard*, seperti

milik Blackberry. Hingga Desember 2006, berita mengenai prototipe Android ini terus tersiar. Tanpa disangka-sangka, pada tahun 2007, perusahaan Apple merilis iPhone dengan desain *smartphone* yang hampir seluruh permukaannya menggunakan layar sentuh. Mulai dari situ, Google memikirkan bagaimana perkembangan *smartphone* Android untuk ke depannya, mengingat prototipe awalnya menggunakan *keyboard* tanpa layar sentuh sama sekali.

Untuk menyaingi iPhone, Nokia dan Blackberry merilis ponsel dengan layar sentuh di tahun 2008. Tak ingin kalah dengan kompetitornya, Google juga merilis ponsel dengan layar sentuh, yaitu HTC Dream atau T-Mobile G1. Namun, tak hanya layar sentuh saja, *smartphone* ini juga tidak meninggalkan penggunaan keyboard.

#### **2.4.1 Android menurut para ahli**

Adapun definisi Android menurut para ahli sebagai berikut:

- 1 Menurut Teguh Arifianto (2011: 1), android yaitu suatu perangkat sistem operasi pada telepon seluler yang OS-nya linux.
- 2 Menurut Hermawan (2011: 1), Android yaitu OS (Operating System) Mobile yang berkembang ditengah-tengah OS lainnya seperti Iphone OS, Symbian, color OS, Windows Mobile, dan masih banyak jenis OS lainnya. Akan tetapi, OS yang berkembang saat ini berjalan dengan memprioritaskan aplikasi inti yang dibangun sendiri tanpa melihat potensi yang cukup besar dari aplikasi pihak ketiga. Oleh sebab itu, aplikasi pihak ketiga memiliki keterbatasan untuk mendapatkan data asli dari ponsel, keterbatasan berkomunikasi antar proses dan keterbatasan dalam distribusi aplikasi pihak ketiga untuk platform mereka.
- 3 Menurut Nazaruddin (2012: 1) yaitu sistem operasi yang diperuntukan pada telepon seluler dengan OS Linux. Pada OS android menyediakan platform yang terbuka untuk para pengembang dengan tujuan menciptaka aplikasi yang dibuat oleh para pengembang sendiri yang nantinya dapat digunakan oleh bermacam-macam device bergerak.

Android biasanya digunakan pada smartphone, tablet PC, smart TV, dan juga smartwatch. Fungsi dari android sama seperti sistem operasi Symbian, Iphone OS, Blackberry OS, Color OS, Windows Mobile, dan masih banyak jenis OS yang berkembang pada saat ini.

#### 2.4.2 Arsitektur Android

Sistem operasi Android dapat digambarkan sebagai tumpukan dari beberapa lapisan, dimana tiap lapisan terdiri dari sekelompok program, yang di dalamnya terdapat sistem operasi, *middleware*, dan aplikasi. Setiap lapisan memiliki fungsi yang berbeda. Berikut ini penjelasan dari tiap lapisan.

1. Linux kernel yang merupakan Lapisan paling bawah. Lapisan inilah yang terhubung langsung dengan perangkat keras dan menjadi penghubung antara perangkat keras tersebut dengan lapisan-lapisan di atasnya. Di dalamnya terdapat seluruh *driver* dari perangkat keras yang terdapat pada *smartphone*. Android menggunakan Linux kernel untuk mengatur fungsi-fungsi utamanya, seperti manajemen memori, manajemen proses, jaringan, pengaturan keamanan, dan lain-lain.
2. *Libraries* yang memungkinkan perangkat untuk menangani berbagai jenis data. Lapisan ini ditulis dengan bahasa pemrograman C/C++ dan dikhususkan untuk perangkat keras tertentu. Dimana fitur-fitur Android berada. Layer ini meliputi berbagai library C/C++ inti seperti Libc, SSL dan Libraries.
3. Android *Runtime* terdiri dari core libraries yang berfungsi untuk menterjemahkan bahasa java/c yang digunakan pada pemrograman aplikasi android dan dalvik virtual machine yang digunakan pada system operasi Android. Berbeda dengan Java Virtual Machine, Dalvik Virtual Machine tidak menjalankan file .class, melainkan file .dex. File .dex dibentuk dari file .class pada saat proses kompilasi.
4. *Application frameworks* yang merupakan lapisan dimana para pembuat aplikasi dapat melakukan pengembangan/pembuatan aplikasi yang dijalankan pada sistem operasi Android. Android mengizinkan pengembang untuk mengakses perangkat keras, mengakses informasi

*resources*, menjalankan *service background*, mengatur *alarm*, menambahkan status notifikasi, dan berbagai macam bentuk pengembangan lainnya.

5. *Applications and Widgets* adalah lapisan paling atas, dimana pengguna dapat menggunakan aplikasi-aplikasi utama, seperti email, sms, kalender, peta, *browser*, kontak, dan lain-lain.



**Gambar 2.6** Logo Android

## **2.5 Android Studio**

Menurut (Loki Fadilah, 2019), Android Studio adalah perangkat lunak (software) resmi dari Google untuk para pengembang aplikasi Android. Dengan Android studio, para pengembang dapat membuat aplikasi untuk Android hanya menggunakan satu perangkat lunak saja.



**Gambar 2.7** Logo Android Studio

Di Android Studio sudah tersedia beragam fitur yang memudahkan para pengembang, antara lain:

1. Android emulator
2. Editor teks
3. Cloud test lab integration
4. GPU debugger preview
5. Dan masih banyak lagi

Karena sudah memiliki fitur yang lengkap, Android Studio termasuk software IDE atau Integrated Development Environment. Atau dalam bahasa Indonesia berarti Lingkungan Pengembangan Terintegrasi.

Dari pengertian diatas fungsi utama Android studio adalah untuk mengembangkan aplikasi Android.

Namun selain itu, Android juga memiliki banyak fitur lain, diantaranya:

1. Mengembangkan aplikasi Android
2. Mengedit kode sumber aplikasi Android
3. Mengembangkan game sederhana
4. Dan masih banyak lagi

Namun intinya fungsi utama Android Studio adalah untuk mengembangkan aplikasi Android. Untuk dapat menggunakan Android Studio, kamu harus menguasai beberapa hal, antara lain:

1. Bahasa pemrograman
2. Bahasa markup XML
3. Data base
4. Algoritma

Bahasa pemrograman digunakan untuk “menggerakkan” aplikasi yang kamu buat. Misalnya, kamu menginginkan agar setelah tombol X disentuh akan keluar bunyi, dan seterusnya. Bahasa pemrograman yang di dukung di Android Studio adalah Java, C++, Go, dan Kotlin. Sedangkan XML digunakan untuk membuat tampilan. Sebenarnya untuk mengatur tampilan aplikasi yang ingin dibuat pada Android Studio sudah ada fitur drag and drop.

Namun akan sangat berguna jika menguasai XML dapat membuat tampilan yang lebih spesifik jika elemennya tidak ada di Android Studio. Berikutnya data base juga sangat penting. Digunakan jika aplikasi yang kamu buat membutuhkan penyimpanan file, informasi atau data dari pengguna. Algoritma juga wajib di pelajari agar aplikasi yang kamu buat dapat berjalan sesuai dengan urutan yang benar.

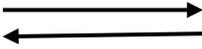
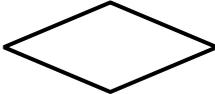
## 2.6 Flowchart

Menurut (Rusmawan, 2019), “*Flowchart* merupakan sebuah gambaran dalam bentuk diagram alur data algoritma-algoritma dalam suatu program yang menyatakan arah alur program”.

Berikut ini adalah simbol-simbol program *flowchart*:

**Tabel 2.1** Simbol-simbol pada *Flowchart*

No	Simbol	Keterangan
1.	Simbol input / output 	Digunakan untuk mewakili data masuk ,atau data keluar.

2.	Simbol proses 	Menyatakan kegiatan yang akan ditampilkandalam diagram alir
3.	Simbol garis alir 	Menunjukkan arah aliran proses atau algoritma
4.	Simbol keputusan 	Proses/langkah dimana perlu adanya keputusan atau adanya kondisi tertentu. Di titik ini selalu ada dua keluaran untuk melanjutkan aliran kondisi berbeda.
5.	Simbol titik terminal 	Menunjukkan awal atau akhir sebuah proses.