

BAB II

TINJAUAN PUSTAKA

2.1 Jamur Tiram

Jamur tiram merupakan salah satu jenis jamur kayu dan memiliki nilai gizi yang lebih tinggi dibandingkan jamur kayu lainnya. Salah satu faktor yang perlu diperhatikan untuk memproduksi jamur tiram sebagai makanan pokok adalah ketersediaan *substrat* sederhana, salah satunya adalah penggunaan serbuk gergaji (Nasution, 2016).

Jamur tiram putih adalah salah satu jamur yang dapat dimakan yang hidup di pohon yang lapuk, dan selain kacang-kacangan, merupakan sumber *protein nabati* dan sumber biologis. Jamur tiram putih mengandung vitamin *B1, B2, C, niasin, biotin dan mineral K, P, Ca, Na, Mg dan Cu*. Kandungan Gizi Jamur Tiram (100 gr) meliputi : *protein 30.4% karbohidrat 56.6%*, lemak 2.2%, *kalsium 314 mg, fosfor 717 mg, natrium 837 mg*, besi 18.2 mg, *niacin 77.2 mg, riboflavin 4.95 mg dan thiamin 0.20 mg* (Suryani & Carolina, 2017).

2.2 Internet Of Things (IoT)

Internet of Things (IoT) adalah kerangka kerja di mana objek dan orang diidentifikasi secara eksklusif dan kemampuan untuk memindahkan data melalui jaringan tanpa memerlukan interaksi dua arah dari orang ke orang, dari sumber ke tujuan melakukan interaksi komputer. *Internet of Things* adalah perkembangan teknologi yang menjanjikan untuk mengoptimalkan kehidupan dengan sensor cerdas dan hal-hal yang terhubung ke jaringan dan bekerja sama di Internet (Sasmoko & Wicaksono, 2017).

Pada tahun 1989, Internet menjadi populer dan saya mulai bekerja secara *online*. Penelitian tentang Perangkat Kontrol Internet dilakukan pada tahun 1990 ketika John Romkey menciptakan pemanggang roti yang dapat dinyalakan dan dimatikan melalui Internet. Selain itu, berbagai penelitian tentang perangkat keras dan perangkat lunak untuk kendali jarak jauh melalui Internet sedang dilakukan. Kevin Ashton, direktur eksekutif MIT AutoID Labs, pertama kali menciptakan

istilah IoT (*Internet of Things*) pada tahun 1997 berdasarkan identifikasi frekuensi radio (RFID). Prinsip IoT adalah bahwa setiap objek harus memiliki alamat IP. Berbagai peralatan sehari-hari dengan sensor cerdas telah dibuat dan dikendalikan melalui internet. Melalui sensor cerdas, data analog diubah menjadi data digital dan selanjutnya dikirim ke prosesor secara realtime. Dengan demikian dapat dilakukan otomatisasi peralatan yang dikendalikan dari jarak jauh dalam arsitektur IoT. Alamat IP (*Protokol Internet*) adalah pengidentifikasi di jaringan yang memungkinkan anda memesan dari situs lain di jaringan yang sama. Alamat IP (*Protokol Internet*) dari objek-objek ini kemudian terhubung ke Internet (Wilianto & Kurniawan, 2018).

2.3 Logika Fuzzy

Salah satu metode komputasi terancang yang tersedia saat ini adalah sistem cerdas. Dalam teknologi informasi, sistem cerdas juga dapat digunakan untuk prediksi. Salah satu metode sistem cerdas yang dapat digunakan untuk prediksi adalah penggunaan logika *fuzzy* yang diperkenalkan oleh Lotfi A. Zadeh pada tahun 1965 (Setiawan et al., 2018).

Fuzzy secara *linguistik* di definisikan sebagai nilai kabur. Nilai bisa besar atau salah pada saat bersamaan. Dalam bentuk *fuzzy* diketahui derajat keanggotaan yang berkisar antara 0 (nol) sampai dengan 1 (satu). Berbeda dengan hard set dengan nilai 1 atau 0 (ya atau tidak). Logika *fuzzy* adalah logika dengan arti ambiguitas atau kabur antara benar atau salah. Dalam teori logika *fuzzy*, nilai *offset* adalah benar atau salah pada saat yang bersamaan. Namun, ketersediaan dan *margin of error* suatu produk tergantung pada bobot anggotanya. Logika *fuzzy* memiliki nilai keanggotaan dari 0 sampai 1. Tidak seperti logika seperti biasanya (Setia, 2019).

2.3.1 Himpunan Fuzzy

Himpunan *fuzzy* bukan hanya sebuah elemen, tetapi setiap elemen diberikan keanggotaan. Bentuk ini merupakan perpanjangan dari set klasik. Hal ini karena dalam himpunan klasik, setiap item yang terdaftar dalam himpunan tersebut memiliki nilai anggota 1, yang berarti merupakan anggota lengkap dari himpunan

tersebut. Tidak demikian dengan himpunan *fuzzy*, setiap anggota terkadang memiliki level keanggotaan setengah atau seperempat anggota, sebab derajat keanggotaan elemen himpunan *fuzzy* tidak harus penuh (1), maka satu elemen dapat artinya anggota lebih berasal satu himpunan *fuzzy* pada satu semesta. Elemen-elemen himpunan *fuzzy* dipetakan ke derajat keanggotaannya dengan fungsi keanggotaan ke dalam interval 0 dan 1 atau $\mu_A[x] \in [0,1]$ (Rusli, 2017)

Himpunan tegas memerlukan batas keanggotaan yang tegas, sehingga tidak dapat diterapkan dengan baik untuk masalah yang melibatkan unsur kekaburan (keambiguan). Unsur keambiguan yang ditimbulkan dalam kasus ini terkait dengan nilai-nilai kebahasaan atau *humanistik* seperti: lemah, sedang, tinggi, dingin, panas, dll. hampir semua orang akan ada ukuran yang berbeda dalam deklarasi nilai bahasa seperti itu, sehingga tidak ada batasan keanggotaan yang jelas. Konsep baru tentang himpunan kemudian diperkenalkan untuk mengatasi hal tersebut, yakni konsep tentang himpunan *fuzzy* (Handoyo & Sarwo Prasajo, 2017).

2.3.2 Operasi Himpunan Fuzzy

Operasi himpunan *fuzzy* digunakan untuk menggabungkan dan memodifikasi himpunan *fuzzy*. Nilai anggota yang dihasilkan dari dua operasi himpunan *fuzzy* sering disebut sebagai pangkat atau predikat termal. Himpunan *fuzzy* memiliki tiga operasi dasar: persimpangan, penyatuan, dan komplemen. Operasi lain yang umum digunakan adalah produk *Cartesian*. Definisi operasi potong (AND), join (OR), dan komplemen (NOT).

a. Irisan

Irisan dari himpunan *fuzzy* $A, C \subseteq X$ adalah himpunan *fuzzy* $A \cap C$ yang didefinisikan oleh fungsi keanggotaan berikut:

$$\mu_{A \cap C}(x) = \min(\mu_A(x), \mu_C(x)) \dots\dots\dots (2.1)$$

Untuk setiap $x \in X$. Fungsi keanggotaan dari irisan himpunan *fuzzy* A_1, A_2, \dots, A_n dirumuskan dengan:

$$\mu_{A_1 \cap A_2 \cap \dots \cap A_n}(x) = \min(\mu_1(x), \mu_2(x) \dots \mu_n(x)) \dots\dots\dots (2.2)$$

b. Gabungan

Gabungan dari himpunan *fuzzy* A dan himpunan *fuzzy* C adalah himpunan *fuzzy* $A \cup C$ yang didefinisikan oleh fungsi keanggotaan berikut:

$$\mu_{A \cup C}(x) = \max(\mu_A(x), \mu_C(x)) \dots\dots\dots (2.3)$$

Untuk setiap $x \in X$. Fungsi keanggotaan dari irisan himpunan *fuzzy* A_1, A_2, \dots, A_n dirumuskan dengan:

$$\mu_{A_1 \cap A_2 \cap \dots \cap A_n}(x) = \min(\mu_{A_1}(x), \mu_{A_2}(x), \dots, \mu_{A_n}(x)) \text{ untuk setiap } x \in X \dots\dots (2.4)$$

c. Komplemen

Komplemen dari himpunan *fuzzy* $A \subseteq X$ adalah himpunan *fuzzy* $A^c \subseteq X$ yang didefinisikan oleh fungsi keanggotaan berikut (Handoyo & Sarwo Prasajo, 2017):

$$\mu_{A^c}(x) = 1 - \mu_A(x) \text{ untuk setiap } x \in X \dots\dots\dots (2.5)$$

2.3.3 Fungsi Keanggotaan

Kelas keanggotaan $\mu_A[x]$ memetakan suatu objek atau properti (x) dari suatu objek ke bilangan riil positif dalam rentang [0,1]. Disebut fungsi keanggotaan karena properti tampilannya mirip dengan fungsi.

Fungsi keanggotaan adalah pemetaan elemen x dalam semesta nilai keanggotaan menggunakan bentuk teoritis dari fungsi tersebut. Beberapa fitur keanggotaan *fuzzy*:

1. Representasi Linear

Pemetaan input ke derajat keanggotaannya digambarkan sebagai suatu garis lurus. Terdapat 2 bentuk, yaitu representasi linear naik dan turun. Pada kurva representasi linear naik, himpunan dimulai pada nilai domain yang memiliki derajat keanggotaan [0].

Fungsi keanggotaannya adalah :

$$\mu_v(x) = \begin{cases} 0; & x \leq a \\ \frac{x-a}{b-a} & a \leq x \leq b \\ b-a & x \geq b \\ 1; & \end{cases} \dots\dots\dots (2.6)$$

Pada kurva representasi linear turun, dimulai pada nilai domain yang memiliki derajat keanggotaan [1] dan bergerak ke kanan menuju domain dengan derajat keanggotaan yang lebih rendah. Fungsi keanggotaannya adalah :

$$\mu_v(x) = \begin{cases} \frac{b-x}{b-a} & a \leq x \leq b \\ 0; & x \geq b \end{cases} \dots\dots\dots (2.7)$$

2. Representasi Kurva Segetiga

Representasi kurva segitiga merupakan kombinasi antara dua garis linear. Fungsi keanggotaannya adalah :

$$\mu_v(x) = \begin{cases} 0; & x \leq a \text{ atau } x \geq c \\ \frac{x-a}{b-a} & a \leq x \leq b \\ \frac{c-x}{c-b} & b \leq x \leq c \end{cases} \dots\dots\dots (2.8)$$

3. Representasi Kurva Trapesium

Representasi kurva trapesium menyerupai bentuk segitiga, namun memiliki beberapa titik dengan derajat keanggotaan 1. Fungsi keanggotaannya adalah (Muslihudin & Helmiyanto, 2020) :

$$\mu_v(x) = \begin{cases} 0; & x \leq a \text{ atau } \geq d \\ \frac{x-a}{b-a} & a \leq x \leq b \\ 1; & b \leq x \leq c \\ \frac{d-x}{d-c} & c \leq x \leq d \end{cases} \dots\dots\dots (2.9)$$

2.3.4 Inferensi Fuzzy Sugeno

Karena inferensi menggunakan metode *SUGENO* diperkenalkan oleh Takagi Sugeno Kang pada tahun 1985, maka metode ini sering disebut dengan metode TSK. Menurut Cox, metode TSK terdiri dari 2 jenis:

a. Model *Fuzzy* Sugeno Orde-Nol

Secara umum bentuk model *fuzzy* SUGENO Orde-Nol adalah :

$$\text{IF } (x_1 \text{ is } A_1) \text{ o } (x_2 \text{ is } A_2) \text{ o } (x_3 \text{ is } A_3) \text{ o } \dots \text{ o } (x_N \text{ is } A_N) \text{ THEN } z=k \dots\dots\dots (2.10)$$

dengan A_1 adalah himpunan *fuzzy* ke- i sebagai *anteseden*, dan k adalah suatu konstanta (tegas) sebagai konsekuensi.

b. Model *Fuzzy* Sugeno Orde-Satu

Secara umum bentuk model *fuzzy* SUGENO Orde-Satu adalah:

IF (x_1 is A_1) $\circ \dots \circ$ (x_N is A_N) THEN $z = p_1 * x_1 + \dots + p_N * x_N + q \dots \dots \dots (2.11)$

Dengan A_1 adalah himpunan *fuzzy* ke- i sebagai *anteseden*, dan p_1 adalah suatu konstanta (tegas) ke- i dan q juga merupakan konstanta dalam konsekuensi.

Apabila komposisi aturan menggunakan metode SUGENO, maka *defuzzifikasi* dilakukan dengan cara menilai rata-ratanya

1. *Fuzzifikasi*

Fuzzifikasi adalah langkah pertama yang bertanggung jawab untuk mengambil nilai input sebagai nilai bersih (*crisp*) dan menentukan sejauh mana *input* sehingga *input* dapat dikelompokkan ke dalam himpunan *fuzzy* yang sesuai. Penggabungan adalah proses menyebabkan bilangan bernilai tinggi memiliki nilai *fuzzy*. Nilai masukan sebagai nilai bersih akan diubah menjadi nilai *fuzzy*, sehingga dapat dikelompokkan ke dalam himpunan *fuzzy* tertentu.

2. Rule Evaluation

Pengambilan nilai input yang telah di *fuzzifikasi* dan mengaplikasikannya ke dalam aturan-aturan *fuzzy* lalu diimplikasikan dan jika sistem terdiri dari beberapa aturan, korelasi antara keluaran himpunan dan aturan adalah menghitung hasilnya.

3. *Defuzzifikasi*

Langkah terakhir dari proses inferensi *fuzzy* adalah untuk mengkonversi nilai *fuzzy* dari hasil agregasi aturan ke dalam sebuah bilangan dan jika sistem terdiri dari beberapa aturan, korelasi antara keluaran himpunan dan aturan adalah menghitung hasilnya (Batubara, 2017).

2.4 Mikrokontroler

Mikrokontroler adalah sistem komputer dengan fungsi pada sebuah chip yang berisi inti prosesor, memori (sejumlah kecil RAM, memori program atau keduanya), dan perangkat *input-output*. Dengan kata lain *Mikrokontroler* adalah

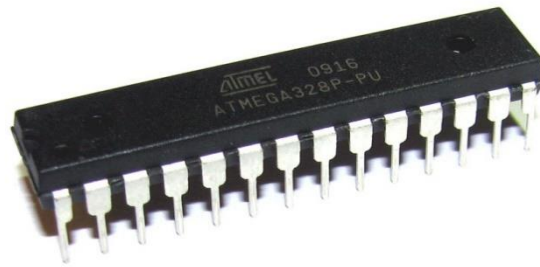
perangkat elektronik digital dengan *input* dan *output* serta control dengan program yang dapat ditulis dan dihapus dengan cara khusus, cara kerja *mikrokontroler* sebenarnya, control membaca dan menulis data. Sebagai contoh, jika mulai membaca dan menulis jika sudah bisa melakukannya dapat melakukan hal sebaliknya. Begitu juga jika mahir membaca dan menulis data, maka pasti dia bisa membuat program untuk membuat sistem kendali otomatis *mikrokontroler* sesuai dengan yang diinginkan (Sriani, 2019).

Mikrokontroler memiliki jalur-jalur masukan (port masukan) serta jalur-jalur keluaran (port keluaran) yang memungkinkan *mikrokontroler* tersebut untuk bisa digunakan dalam aplikasi pembacaan data, pengontrolan serta penyajian informasi. *Port* masukan digunakan untuk memasukkan informasi atau data dari luar ke *mikrokontroler* seperti kondisi saklar yang dihubungkan ke kaki *mikrokontroler*, apakah sedang terbuka atau sedang tertutup. Jalur masukan umumnya berupa jalur digital yang digunakan *mikrokontroler* untuk membaca keadaan digital (apakah logika 0 atau 1). *Port* keluaran digunakan untuk mengeluarkan data atau informasi dari mikrokontroler sebagai pengendali perangkat seperti LED, motor, relay dan menyajikan informasi melalui perangkat seperti *seven-segment* dan LCD. Untuk bisa bekerja, *mikrokontroler* perlu diberikan tegangan dari luar. IC *mikrokontroler* dapat bekerja pada tegangan 5V, namun sebagai IC *mikrokontroler* seperti *Atmega328* dapat dioperasikan dengan tegangan 5V (Dharmawan, 2017)

2.4.1. Mikrokontroler Atmega 328

Mikrokontroller Atmega328 salah satunya adalah mikrokontroller ARV(Alf and Vegard's Risc processor). Atmega328 yang menggunakan program Risc (Reduce Intrusion Set Computing) Program berjalan lebih cepat karena instruksi hanya membutuhkan satu siklus clock untuk dieksekusi.

ATmega328 adalah chip mikrokontroler 8-bit berdasarkan AVR-RISC Atmel. Chip ini memiliki kemampuan membaca dan menulis EEPROM 1KB dan Flash ISP 32KB dengan SRAM 2KB. Karena ukuran memori flash 32KB, chip ini disebut Atmega 328 (Saputra & Aswardi, 2018)



2.1 Arduino Atmega382

Sumber (Saputra & Aswardi, 2018)

2.4.2. Konfigurasi Pin Atmega 328

(PCINT14/RESET) PC6	1	28	PC5 (ADC5/SCL/PCINT13)
(PCINT16/RXD) PD0	2	27	PC4 (ADC4/SDA/PCINT12)
(PCINT17/TXD) PD1	3	26	PC3 (ADC3/PCINT11)
(PCINT18/INT0) PD2	4	25	PC2 (ADC2/PCINT10)
(PCINT19/OC2B/INT1) PD3	5	24	PC1 (ADC1/PCINT9)
(PCINT20/XCK/T0) PD4	6	23	PC0 (ADC0/PCINT8)
VCC	7	22	GND
GND	8	21	AREF
(PCINT6/XTAL1/TOSC1) PB6	9	20	AVCC
(PCINT7/XTAL2/TOSC2) PB7	10	19	PB5 (SCK/PCINT5)
(PCINT21/OC0B/T1) PD5	11	18	PB4 (MISO/PCINT4)
(PCINT22/OC0A/AIN0) PD6	12	17	PB3 (MOSI/OC2A/PCINT3)
(PCINT23/AIN1) PD7	13	16	PB2 (SS/OC1B/PCINT2)
(PCINT0/CLKO/ICP1) PB0	14	15	PB1 (OC1A/PCINT1)

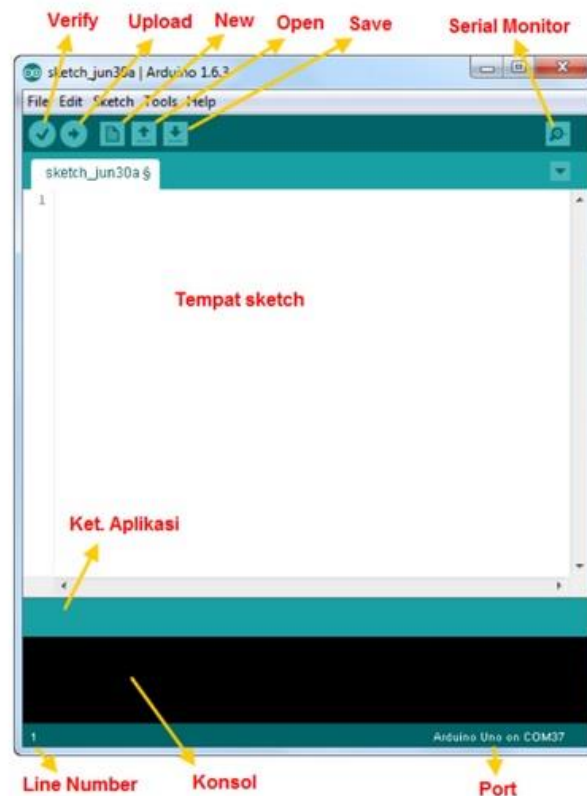
2.2 Konfigurasi Pin Atmega328

Sumber (Saputra & Aswardi, 2018)

Atmega328 Menyediakan total 28 pin *port*, dimana 23 diantaranya teridentifikasi sebagai *port B*, *port C* dan *port D*. Terdapat 6 channel untuk pemrosesan digital yang akan dikumpulkan dalam satu *port C*. Selain itu pada mikrokontroler terdapat tiga *timer/counter* untuk membandingkan kemampuan, proses utama terdiri dari 32 register, *watchdog timer* dan memori tipe SRAM.(Suryani & Carolina, 2017)

2.5 Arduino IDE

Arduino IDE (*Integrated Development Environment*) adalah perangkat lunak yang sangat berperan penting dalam penulisan program, kompilasi kode ke dalam biner dan memuatnya ke dalam memori mikrokontroler. (Samsir et al., 2020)



2.3 Tampilan Arduino IDE

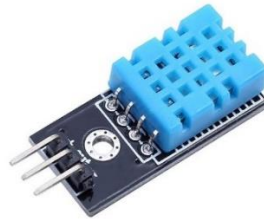
Sumber (Santoso, 2015)

Arduino IDE dibangun dalam bahasa Java, tetapi sintaks yang digunakan untuk Arduino IDE adalah C++. Arduino dapat mengenali perintah Arduino dan C++. (Ahyadi, 2018)

2.6 Sensor DHT11

Sensor DHT11 merupakan sebuah sensor dapat mengukur dua parameter lingkungan secara bersamaan, yaitu suhu dan kelembaban (humidity). Dari sensor tersebut terdapat termistor tipe NTC (*Negative Temperature Coefficient*) untuk

mengukur suhu, sensor kelembaban tipe resistif dan mikrokontroler 8 bit yang memproses kedua sensor dan mengirimkan hasilnya ke sensor kawat dua arah tunggal. (kawat tunggal dua arah)(Nugroho, 2018).



2.4. Sensor DHT11

Sumber (Yana et al., 2017)

2.7 ESP 8266

ESP8266 adalah modul Wifi yang baru-baru ini menjadi semakin populer di kalangan pengembang perangkat keras. Selain harganya yang sangat terjangkau, modul Wifi serbaguna ini sudah SOC (*System on Chip*), sehingga bisa langsung diprogram dengan ESP8266 tanpa perlu mikrokontroler tambahan. Kelebihan lainnya, ESP8266 ini dapat memainkan peran sebagai titik akses khusus dan klien.

Keunggulan lain dari ESP8266 adalah memiliki mode deep sleep, sehingga konsumsi daya akan relatif jauh lebih hemat dibandingkan modul Wifi. Catatan penting yang harus ditekankan adalah bahwa ESP8266 beroperasi pada tegangan 3.3V (Sasmoko & Wicaksono, 2017).



2.5 Module ESP 8266

Sumber (Shobrina et al., 2018)

2.8 Power Supply

Power supply adalah perangkat yang digunakan untuk memberi daya pada beban listrik. Perangkat ini sering digunakan Aktivitas aktual dalam berbagai aplikasi elektronik dan medan listrik. Berdasarkan kemampuan, paket daya dapat dibagi menjadi paket daya stabil, paket daya tidak stabil, dan paket daya yang dapat disesuaikan (Putranto et al., 2021).



2. 6 Power Supply 12V

Sumber (Suhantono, 2014)

2.9 Relay

Relay difungsikan sebagai saklar yang mematikan atau menghidupkan mikrokontroler. Relay ini dapat menghidupkan atau mematikan sistem mikrokontroler tergantung perintah dari *Automatic Control* (Gofur Saleh & Subijanto, 2013).



2. 7 Relay

Sumber (Saleh & Haryanti, 2017)

2.10 Pompa Air DC

Pompa adalah suatu mesin atau peralatan mekanis yang digunakan untuk mengangkat *fluida* dari *elevasi* rendah ke tinggi atau untuk mengeluarkan fluida dari daerah bertekanan rendah ke daerah bertekanan tinggi dan digunakan sebagai penguat aliran dalam sistem perpipaan. Prinsip kerja pompa adalah tekanan dan penghisapan cairan. Pada sisi hisap (*suction*) pompa, elemen pemompaan mengurangi tekanan di dalam ruang pompa, menciptakan perbedaan tekanan antara permukaan cairan hisap dan ruang pompa (Iqtimal & Devi, 2018).



2. 8 Pompa Air DC

Sumber (Yana et al., 2017)

2.11 Android

Android menurut Ed Brunete adalah “Sebuah *opensource software toolkit* untuk telepon genggam yang dibuat oleh Google dan *Open Handset Alliance*”. Android sendiri bukanlah sebuah bahasa pemrograman, tetapi android merupakan sebuah environment untuk menjalankan aplikasi. Android merupakan sistem operasi yang berbasis Linux dan dirancang untuk perangkat seluler layar sentuh seperti smartphone serta komputer tablet. Android pada awalnya dikembangkan oleh perusahaan bernama Android, Inc., dengan dukungan finansial yang berasal dari Google, yang kemudian Google pun membelinya pada tahun 2005. Sistem operasi android tersebut secara resmi dirilis pada tahun 2007, bersamaan dengan didirikannya sebuah perusahaan *Open Handset Alliance*, *konsorsium* dari beberapa perusahaan perusahaan perangkat keras, perangkat lunak, serta telekomunikasi yang memiliki tujuan untuk memajukan standar terbuka dari perangkat seluler.

Ponsel yang berbasis sistem operasi Android pertama dijual pada bulan Oktober 2008 (Alfarisi, 2019).

2.12 Kodular

Kodular adalah situs web yang menyediakan alat untuk membangun aplikasi Android dengan model pemrograman blok *drag-and-drop* yang mirip dengan MIT App Inventor untuk membangun aplikasi Android menggunakan pemrograman blok. Tidak perlu mengetikkan kode program secara manual untuk membuat aplikasi. Kodular menawarkan keuntungan, termasuk penyimpanan kodular dan ekstensi IDE kodular, yang memungkinkan pengembang untuk dengan mudah mengunggah aplikasi Android ke toko kodular, membuat blok program ekstensi IDE sesuai keinginan mereka (Kumala & Winardi, 2020).

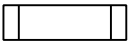
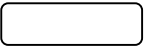
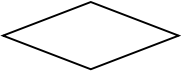
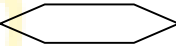



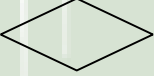
2.13 Flowchart

Flowchart adalah representasi grafis dari langkah-langkah prosedural dan urutan program. *Flowchart* membantu analis memecah masalah menjadi bagian-bagian yang lebih kecil dan menganalisis alternatif lain yang ada. Diagram blok memudahkan penyelesaian masalah secara umum, terutama yang memerlukan kajian dan evaluasi lebih lanjut. Suatu proses dalam lingkungan organisasi biasanya merupakan rangkaian kegiatan yang berulang. Setiap siklus operasi ini biasanya dapat dipecah menjadi beberapa langkah yang lebih kecil. Anda dapat menemukan langkah-langkah yang dapat ditingkatkan (*improved*) dalam deskripsi langkah-langkah ini.

Jadi, *flowchart* adalah diagram yang menyatakan aliran proses dengan menggunakan anotasi bidang-bidang geometri, seperti lingkaran, persegi empat, wajik, oval, dan sebagainya untuk merepresentasikan langkah-langkah kegiatan beserta urutannya dengan menghubungkan masing masing langkah tersebut menggunakan tanda panah. Programmer menggunakan *flowchart* program untuk menggambarkan urutan instruksi dari program komputer. Analis Sistem menggunakan *flowchart* program untuk menggambarkan urutan tugas-tugas pekerjaan dalam suatu prosedur atau operasi. Simbol-simbol *flowchart* yang biasanya dipakai adalah simbol-simbol *flowchart* standar yang dikeluarkan oleh

ANSI dan ISO. Simbol-simbol ini dapat dilihat pada Gambar 8. Simbol *flowchart* Standar berikut ini :

2.1 Simbol *Flowchart*

Simbol	Fungsi	Simbol	Fungsi
	Permulaan sub program		Permulaan/akhir program
	Perbandingan, pernyataan, penyeleksian data yang memberikan pilihan untuk langkah selanjutnya		Proses inialisasi/pemberian harga awal
	Penghubung bagianbagian flowchart yang berada pada satu halaman.		Proses penghitung/proses pengolahan data
	Proses input/output data		Belah ketupat menyatakan himpunan relasi merupakan hubungan antar entitas.

2.14 *Firebase*

Firebase adalah penyedia layanan database *real-time* dan *backend* sebagai layanan. Aplikasi yang memungkinkan pengembang membuat API yang disinkronkan untuk pelanggan yang berbeda dan dihosting di *cloud Firebase*. *Firebase* memiliki banyak *library* yang memungkinkan layanan ini terintegrasi dengan Android, iOS, Javascript, Java, dan NodeJS. Basis data *Firebase* juga dapat diakses melalui *REST API*. *REST API* menggunakan protokol *Server Sent Events*

dengan membuat koneksi HTTP untuk menerima pemberitahuan *push* dari server.
Karakteristik Alat Ukur.(Sonita & Fardianitama, 2018)

