BAB II
DASAR TEORI

2.1 KAJIAN PUSTAKA

Agoes Santika Hyperastuty dan Yanar Mukhamad melaksanakan kajian ini pada tahun 2021 dengan judul "*Monitoring* Saturasi Oksigen Menggunakan SpO2 MAX30100 Berbasis Android" yang mencakup pengembangan sistem *monitoring* tingkat saturasi oksigen berbasis android. Membuat alat *pulse oxymeter* dengan dua parameter yaitu SpO2 dan *pulse rate* adalah tujuan dari penelitian ini. Serta membangun sistem *alarm* sebagai penanda kondisi tidak normal pada pasien dan aplikasi penampil sebagai alat *monitoring*. Pada perancangan ini menggunakan sensor MAX30100 dimana sensor ini mampu mengukur oksigen dalam darah hanya dengan ujung jari. Namun pada penelitian ini belum serta mengukur suhu tubuh pasien. Dimana suhu tubuh pasien juga sangat penting dalam me*monitoring* kesehatan pasien. Hasil dari 5 percobaan yang dilakukan dalam penelitian tersebut disebutkan bahwa perbedaan kadar oksigen dari 5 percobaan tersebut tidak terlalu jauh. Yang berarti sistem rancangan yang dibuat bekerja dengan baik [6].

Penelitian Della Rahmawarni dan Harmadi pada tahun 2021 yang berjudul "Sistem *Monitoring* Saturasi Oksigen dan Denyut Nadi dalam Darah Menggunakan Sensor MAX30100 via Telegram berbasis IoT" membahas tentang perancangan sistem *monitoring* menggunakan telegram. Mikrokontroler WeMos D1 Mini akan digunakan dalam sistem ini untuk memantau saturasi oksigen dan denyut nadi dari jarak jauh, dan Telegram akan diintegrasikan sebagai sistem *monitoring*nya. Namun penelitian ini juga belum serta me*monitoring* suhu tubuh pasien. Dimana suhu tubuh pasien sangat penting untuk informasi kesehatan pasien. Dan penggunaan telegram apabila aplikasi sedang bermasalah maka *monitoring* tidak bisa dilakukan. Menurut hasil penelitian, perhitungan persentase *error* dalam pengujian saturasi oksigen adalah 0.96%, untuk pengujian denyut nadi adalah 1.63%. [7].

Penelitian Ary Sulistyo U, Erda Hermono P N dan Mohamad Sofie pada tahun 2019 yang berjudul "*Monitoring Heart Rate* dan Saturasi Oksigen Melalui *Smartphone*" yang mengkaji perangkat pemantau detak jantung dan saturasi

oksigen yang dapat dilacak melalui *smartphone*. Tujuan dari sistem tersebut dibuat untuk memantau kondisi pasien setiap saat dengan menggunakan sensor MAX30100 serta mikrokontroler Arduino nano dan ESP01. Aplikasi blynk pada *smartphone* akan menampilkan data dari sistem. Namun pada penelitian ini belum mencakup *monitoring* suhu tubuh. Dimana suhu tubuh juga penting untuk dipantau, karena menunjukan juga kesehatan dari pasien. Menurut hasil penelitian, ada perbedaan 0.8 persen untuk detak jantung dan 1% untuk saturasi oksigen antara pembacaan nilai perangkat ini untuk parameter ini [8].

Dalam penelitian pada tahun 2021 yang berjudul "Rancang Alat Pengukur Suhu Tubuh dan Saturasi Oksigen dalam Darah pada Kondisi Pandemi COVID-19 Berbasis Mikrokontroler" Panji Wiratama, Randy Erfa S, dan Casi Setianingsih membahas tentang pembuatan alat pengukur suhu tubuh dan saturasi oksigen berbasis mikrokontroler. Menggunakan saturasi oksigen darah dan pengukuran suhu tubuh untuk mengidentifikasi gejala COVID-19 merupakan tujuan dari penelitian ini. Dalam mendeteksi suhu tubuh dan tingkat saturasi oksigenn menggunakan sensor MLX90614 dan sensor MAX30100 masing – masing digunakan dalam arsitektur sistem ini. Selain itu, juga menggunakan sensor buzzer sebagai peringatan dan sensor Ultrasonik HC-SR04 untuk mendeteksi jarak. Mikrokontroler ATMEGA328 berbasis Arduino Uno digunakan dalam penelitian ini. Pada LCD 16x2 akan ditampilkan hasil pengukuran yang dilakukan oleh sejumlah sensor. Penelitian ini memiliki beberapa kendala, antara lain *smartphone* dan hasil data yang masih belum terintegrasi sehingga tidak memungkinkan untuk dilakukan pemantauan secara real-time. Hasil yang didapatkan dari penelitian ini akurasi perbandingan sensor MLX90614 dan thermogun pada jarak 2cm sebesar 93%. Jarak 4cm sebesar 99,6%, jarak 6cm sebesar 93,2%. Serta akurasi perbandingan sensor MAX30100 dan *pulse oximeter* sebesar 99.7% [9].

Dalam penelitian 2018 yang berjudul "Sistem *Monitoring* Detak Jantung dan Suhu Tubuh Menggunakan Arduino", Tan Suryani S, Alamsyah, M. Bachtiar, Ardi Amir, dan Boenyamin Bontong melakukan penelitian terkait sistem *monitoring* detak jantung dan suhu tubuh. Membuat sistem *monitoring* untuk memudahkan tenaga medis dalam me*monitor* pasien merupakan tujuan dari penelitian ini. Sensor suhu tubuh dalam perancangan sistem ini menggunakan sensor DS18B20,

sedangkan sensor AD8232 mengukur detak jantung. Sedangkan Arduino Uno merupakan mikrokontroler yang digunakan. Hasil data yang diperoleh nantinya akan ditampilkan dalam LCD. Pada penelitian ini masih terdapat kelemahan bahwa tampilan data dari sistem *monitoring* ini belum dapat dipantau melalui *smartphone*. Selain itu, belum ada informasi tentang saturasi oksigen. Temuan penelitian menunjukkan bahwa sensor AD8232 dan DS18B20 dapat digunakan secara efektif, namun karena penempatan sensor yang tidak tepat, pengukuran suhu tubuh dan detak jantung menjadi kurang tepat. [10].

2.1 DASAR TEORI

Pada dasar teori penelitian sistem *monitoring* suhu tubuh dan saturasi oksigen akan dibahas beberapa teori yang mencakup beberapa komponen dan beberapa teori dasar tentang parameter yang akan diuji.

2.2.1 SATURASI OKSIGEN

Jumlah oksigen sesungguhnya yang terikat oleh hemoglobin atas kemampuan total hemoglobin darah mengikat oksigen disebut dengan saturasi oksigen (SpO2). [11]. Artinya saturasi oksigen dapat diartikan sebagai seberapa banyak persentase oksigen yang dapat dibawa oleh hemoglobin. Persentase hemoglobin yang terikat oksigen disebut dengan saturasi hemoglobin. Terdapat beberapa faktor yang mempengaruhi persentase saturasi oksigen. Salah satunya adalah tekanan parsial oksigen (PaO2). PaO2 menjadi faktor utama karena berkaitan dengan konsentrasi oksigen yang secara fisik larut dalam darah. Ketika PaO2 meningkat akan terjadi juga peningkatan pada persentase saturasi hemoglobin, begitu pula sebaliknya. Selain itu tekanan parsial karbondioksida (PaCO2), pH, suhu, hemoglobin, dan aktivitas sehari hari juga termasuk dalam faktor yang dapat mempengaruhi persentase saturasi oksigen [12].

Tabel 2.1 Klasifikasi hipoksemia berdasarkan nilai PaO2 dan SpO2 [12]

Derajat Hipoksemia	**PaO2 (mmHg)**	**SpO2**
Normal	97 – 100	95% – 97%
Kisaran normal	>79	>94%
Hipoksemia ringan	60 – 79	90% – 94%
Hipoksemia sedang	40 – 59	75% – 89%
Hipoksemia berat	<40	<75%

2.2.2 SUHU TUBUH

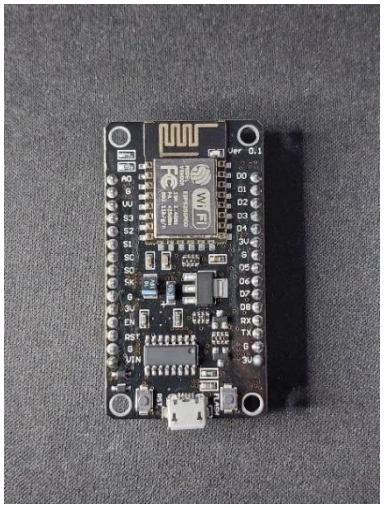
Suhu tubuh merupakan hasil dari kesetimbangan antara panas yang diproduksi dan digunakan tubuh, serta dinyatakan dengan satuan derajat. Suhu yang dimaksud merupakan panas atau dinginnya suatu materi. Selisih antara panas yang diproduksi tubuh dan panas yang keluar adalah suhu tubuh. Suhu tubuh yang merupakan refleksi dari panas tubuh adalah selisih antara panas yang diproduksi tubuh dengan panas yang dikeluarkannya. Meskipun suhu luar dapat berubah ketika tubuh melakukan aktivitas fisik yang berat, sistem pengaturan suhu tubuh bekerja untuk melindungi suhu inti atau jaringan relatif konstan. Namun, aliran darah dalam kulit dan jumlah panas yang keluar masih mempengaruhi suhu tubuh. Suhu tubuh yang bisa diterima berkisar antara 36˚C hingga 38˚C karena perubahan suhu lingkungan. Meskipun hasilnya berbeda – beda tergantung dimana pengujian dilakukan. Klasifikasi suhu tubuh manusia dapat dilihat pada Tabel 2.2 [13].

Tabel 2.2 Klasifikasi suhu tubuh [13]

Hipotermi	Normal	Panas	Hipertermi
< 36˚C	36˚C - 38˚C	38.1˚C - 40˚C	> 40˚C

2.2.3 NODEMCU ESP8266

NodeMCU adalah sebuah perangkat mikrokontroler yang dilengkapi dengan *chip* ESP8266 yang berfungsi sebagai penyedia jaringan WiFi. Pada perangkat ini terdapat beberapa pin input dan output yang dapat digunakan untuk mengembangkan suatu proyek *Internet of Things*. Selain itu perangkat ini juga dilengkapi dengan micro-USB port yang memiliki fungsi untuk memasukkan program dan dapat juga digunakan sebagai sumber tegangan. Software yang digunakan untuk memprogram perangkat ini dapat menggunakan Arduino IDE. [14].



Gambar 2.1 NodeMCU ESP8266

Tabel 2.3 Spesifikasi modul NodeMCU ESP8266 [14]

Spesifikasi NodeMCU	Keterangan
Chip	ESP8266-12E
Tegangan *input*	3.3V ~ 5V
GPIO	13 Pin
Kanal PWM	10 Kanal
10 bit ADC Pin	1 Pin
Flash memory	4MB
Clock speed	40/26/24 MHz
Wifi	IEEE 802.11 b/g/n
Frekuensi	2.4 GHz – 22.5 GHz
USB *Chip*	Micro USB
USB *Port*	CH340G

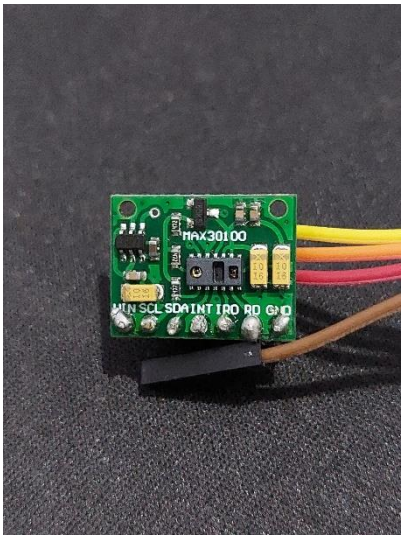
2.2.4 SENSOR SUHU DS18B20

Sensor suhu DS18B20 hanya menggunakan 1 pin untuk jalur komunikasi datanya karena merupakan sensor suhu digital dengan satu kabel. Sensor DS18B20 dapat terhubung dengan beberapa sensor pada GPIO yang sama, hal itu dikarenakan mempunyai nomor seri 64-bit yang berbeda. Hal ini sangat membantu saat pembacaan data untuk proyek monitoring suhu. Sensor DS18B20 menyediakan 9-bit hingga 12-bit untuk pembacaan pada saat pengukuran suhu. Selain itu, sensor DS18B20 juga dapat memperoleh daya langsung dari jalur data hal ini disebut dengan daya parasite, sehingga tidak memerlukan catu daya eksternal. Menurut *datasheet*, sensor ini memiliki akurasi sekitar 0,5˚C dari -10˚C hingga +85˚C dan dapat menguji suhu dari -55˚C hingga +125˚C [15].

Gambar 2.2 Sensor suhu DS18B20

2.2.5 SENSOR SATURASI OKSIGEN MAX30100

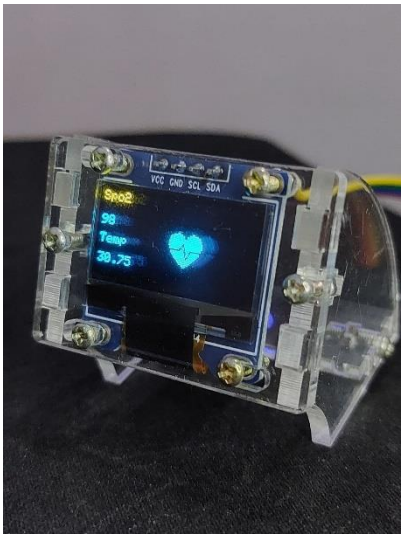
Sensor MAX30100 merupakan sensor yang menggabungkan pengukuran saturasi oksigen dan detak jantung. Sensor ini menggabungkan dua LED (*infrared* dan *red*), photodetektor optik yang sudah dioptimalkan, dan pemprosesan sinyal analog dengan *noise* rendah. Pada pembacaan saturasi oksigen terdapat 3 sub-sistem pada sensor MAX30100 yang digunakan, yaitu *Ambient Light Cancellation* (ALC) yang berfungsi untuk mengoptimalkan cahaya yang masuk, lalu yang kedua terdapat 16-bit sigma delta ADC yang berfungsi untuk mengkonversi sinyal analog ke digital, dan yang terakhir adalah filter waktu diskrit yang berfungsi untuk membatasi interferensi pada frekuensi 50Hz/60Hz serta membuang *residual noise* dengan frekuensi rendah. Sinyal sensor MAX30100 dapat dinonaktifkan melalui perangkat lunak dan dijalankan pada catu daya 1.8V dan 3.3V. Aplikasi alat ini yaitu sebagai alat *monitoring* kondisi tubuh sehari-hari. Perangkat ini juga digunakan sebagai alat pemantauan kebugaran dan medis [16].



Gambar 2.3 Sensor MAX30100

2.2.6 OLED SSD1306

SSD1306 adalah *driver* CMOS OLED/PLED *chip* tunggal dengan kontrol pemancar cahaya organik/polimer dan sistem tampilan grafis dioda dot-matrix. Pengontrol kontras, RAM tampilan, dan osilator ditanamkan ke dalam SSD1306 untuk mengurangi kebutuhan akan komponen dan daya eksternal. SSD1306 mempunyai 256 kontrol *step-brightness*. *Interface* serial kompatibel paralel 6800/8000, *interface* I2C atau *Serial Peripheral Interface* semuanya dapat digunakan untuk perintah yang dikirim melalui perangkat keras tertentu. SSD1306 sendiri memiliki resolusi panel dot-matrix 128x64. [17].



Gambar 2.4 OLED SSD1306

2.2.7 KY-012 ACTIVE PIEZO-BUZZER MODULE

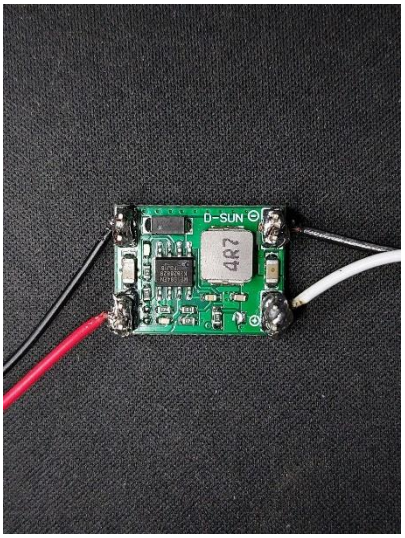
Modul buzzer aktif dengan kemampuan menghasilkan suara bip merupakan modul KY-012. Modul buzzer ini dapat memiliki suara hingga 85 dB dan frekuensi resonansi 2,5 kHz hingga 300 Hz. Dua pin pada komponen ini beroperasi pada rentang tegangan DC 3 – 5.5V. Berbeda dengan modul pasif (KY-006), modul buzzer aktif ini tidak membutuhkan gelombang persegi untuk menghasilkan suara. Buzzer ini secara otomatis akan menghasilkan gelombang persegi jika pin sinyal menerima tegangan minimal 3V. [18].

Gambar 2.5 Modul Buzzer KY-012

2.2.8 MP1584 STEP DOWN MODULE

MP1584 merupakan *step-down switching regulator* yang memiliki frekuensi tinggi, yang terintegrasi pada sisi dengan tegangan daya MOSFET yang tinggi. Hal ini menghasilkan keluaran 3A dengan kontrol mode saat ini untuk *respons loop* cepat dan kompensasi yang mudah. Rentang masukan modul ini antara 4,5V hingga 28V dengan output tetap tegangan yang dikeluarkan oleh modul ini sebesar 5V. Arus statis operasional yang digunakan pada modul ini ialah 100µA, sehingga memungkinkan untuk menggunakan masukan berupa baterai. Efisiensi konversi daya yang tinggi pada rentang beban yang lebar didapat dengan menurunkan frekuensi *switching* pada kondisi beban ringan untuk mengurangi kerugian *switching* dan penggerak gerbang. Pembalikan frekuensi membantu mengatasi arus induktor mengalir selama pengaktifan dan pematian termal menyediakan operasi yang baik dan toleran terhadap kesalahan. Dengan *switching* pada 1,5MHz,

MP1584 dapat mencegah masalah kebisingan EMI (*Electromagnetic Interference*), seperti yang ditemukan pada aplikasi radio AM dan ADSL. MP1584 tersedia dalam paket SOIC8E yang ditingkatkan secara termal. Switching disini memiliki fungsi untuk memutus pada tegangan DC yang belum teregulasi dengan frekuensi yang sangat tinggi. Tujuan dari frekuensi tinggi ini untuk memperoleh tingkat efisiensi yang tinggi [19].



Gambar 2.6 MP1584 Step Down Module

2.2.9 INTERNET OF THINGS (IoT)

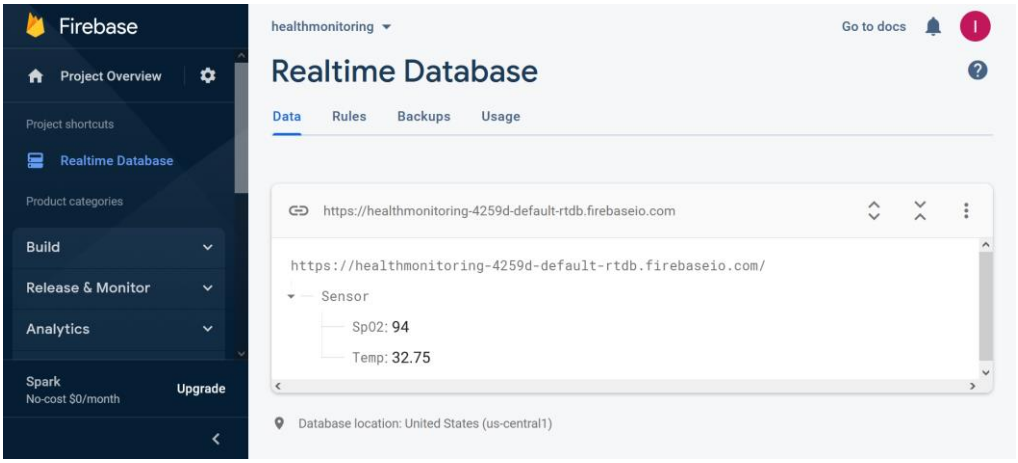
Internet of Things (IoT) didefinisikan oleh ITU-T (*International Telecommunication Union Telecommunication Standardization Sector*) sebagai infrastruktur global dengan seperangkat informasi yang memungkinkan layanan berkelanjutan dengan menghubungkan hal-hal fisik (baik nyata maupun *virtual*) berdasarkan informasi dan komunikasi terkini serta teknologi yang sedang berkembang saat ini [20]. Interaksi manusia-ke-manusia maupun manusia-ke-komputer tidak diperlukan dalam proses ini. Program mengoperasikan semuanya secara otomatis. Sedangkan Kevin Ashton mendefinisikan IoT sebagai sensor yang terkoneksi ke internet dan memiliki perilaku seperti internet dengan berbagi data dengan bebas, menjaga koneksi terbuka setiap waktu, dan mengaktifkan aplikasi tidak terduga sehingga komputer bisa mengerti dunia disekitarnya dan menjadi bagian kehidupan sehari-hari.

Ungkapan "*Internet of Things*" tersusun dari dua unsur utama. Yang pertama adalah *Internet*, yaitu sistem jaringan komputer yang memakai protokol internet

(TCP/IP) dalam berbagi dan mengkomunikasikan data informasi. Disisi lain, *Things* didefinisikan sebagai objek nyata yang dibawa melewati sensor dan ditransmisikan melewati internet. IoT berfungsi melalui penggunaan argumen algoritma bahasa pemrograman yang dikompilasi. Dimana dalam setiap argumen yang dibuat akan berinteraksi dengan argumen lain untuk membantu perangkat keras menjalankan beberapa tugas atau fungsi. Agar mesin dapat dikendalikan secara otomatis dan tanpa memerlukan bantuan manusia. Jaringan internet, yang berfungsi sebagai jembatan antara sistem perangkat keras, adalah elemen paling penting dalam perkembangan program. Dalam skenario ini, tugas utama manusia adalah bertindak sebagai supervisor, mengawasi setiap tindakan dan perilaku mesin saat beroperasi [21].

2.2.10 GOOGLE FIREBASE

Saat ini, Google adalah pemilik Google Firebase, sebuah perusahaan yang menawarkan layanan *backend* dan *real-time database*. Untuk memudahkan pekerja *Mobile Apps Developer*, Google menawarkan Google Firebase sebagai solusinya. Pembuat aplikasi tidak lagi harus menghabiskan banyak waktu di *backend* berkat Google Firebase, dan sebaliknya dapat berkonsentrasi untuk membuat aplikasi baru. Fungsionalitas *real-time database* ditawarkan oleh Google Firebase. *Database* NoSQL yang dapat diakses pengguna secara *real-time* disediakan oleh fitur *Firebase Real Time Database*. Fitur tersebut memiliki keuntungan karena bisa menyinkronkan data dengan instan setelah terkoneksi akses internet, sekaligus mengumpulkan data secara lokal pada saat akses internet tidak tersedia [22].



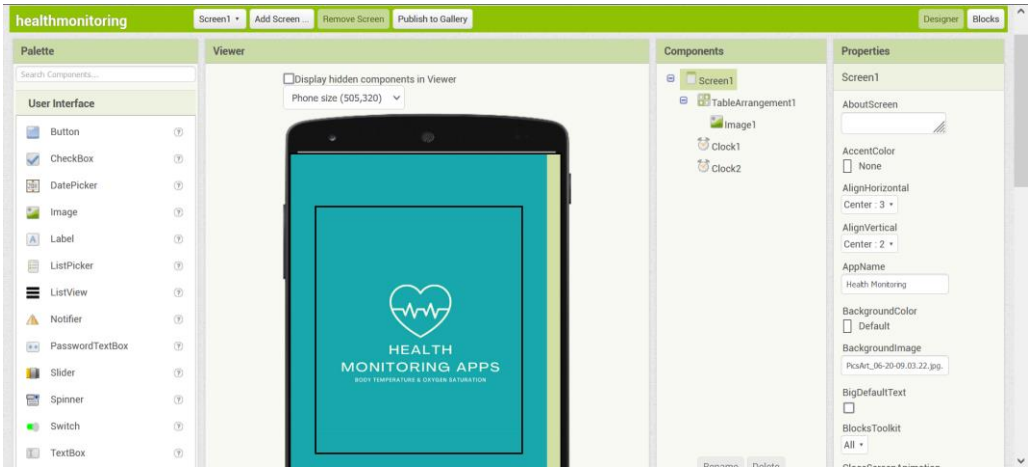
Gambar 2.7 Google Firebase

2.2.11 APP INVENTOR

Aplikasi web *open source* bernama App Inventor for Android dibuat oleh Google dan saat ini dijalankan oleh Massachusetts *Institute of Technology* (MIT). Pengguna baru dapat mempelajari pemrograman dengan menggunakan App Inventor dalam mengembangkan aplikasi pada sistem operasi Android [23]. Mirip dengan Scratch, antarmuka pengguna grafis App Inventor ini memungkinkan pengembang untuk *drag-and-drop* objek visual dalam menyusun program yang akan beroperasi di perangkat Android. Mirip dengan menyusun teka-teki, pengkodean hanya membutuhkan operasi *drag-and-drop* daripada kode program yang panjang. Menurut Google, App Inventor mempermudah dan mempercepat seseorang yang sangat awam dengan bahasa pemrograman dalam mengembangkan aplikasi Android. Biasanya, App Inventor terdiri dari dua susunan utama, yaitu:

1. *Design View*, fitur dari komponen ini digunakan untuk mendesain aplikasi.
2. *Block Editor*, adalah komponen yang memungkinkan untuk menggabungkan blok dalam membuat aplikasi Android.

Pendistribusian *Open Block* oleh MIT *Scheller Teacher Education Program* App Inventor 2 yang memiliki komponen kuat, terhubung pada bahasa pemrograman Scratch dari MIT pada khususnya [24].



Gambar 2.8 App Inventor

2.2.12 QUALITY OF SERVICE

Mekanisme jaringan yang disebut *Quality of Service* (Qos) dapat memutuskan apakah layanan jaringan bisa beroperasi sesuai standar kualitas layanan yang sudah ditetapkan. *Throughput*, *delay*, dan *packet loss* adalah contoh parameter QoS. Terdapat standar QoS, salah satunya adalah TIPHON

(*Telecomunications and Internet Protocol Harmonization Over Network*) TR.101329.V2.1.1.1999-06. Nilai standar QoS ini ditunjukkan dalam Tabel 2.4. [25].

Tabel 2.4 Kategori standar nilai QoS [25]

Nilai Indeks	Persentase (%)	Kategori
3.8 – 4	95 – 100	Sangat Bagus
3 – 3.79	75 – 94.75	Bagus
2 – 2.99	50 – 74.75	Sedang
1 – 1.99	25 – 49.75	Jelek

2.2.12.1 Parameter *Quality of Service*

a. *Throughput*

Kecepatan transfer data dikenal sebagai *throughput*. *Throughput* dihitung dengan membagi keseluruhan paket datang yang diamati di tujuan dalam periode waktu yang telah ditentukan dengan panjang periode waktu tersebut. Tabel 2.5 mencantumkan klasifikasi *throughput* menurut TIPHON.

Tabel 2.5 Standar *throughput* [25]

Kategori	***Throughput***	Indeks
Sangat Bagus	100	4
Bagus	75	3
Sedang	50	2
Jelek	<25	1

Persamaan perhitungan *throughput*:

$$Throughput = \frac{\text{Paket data terkirim}}{\text{Waktu pengiriman data}} \times 8 \qquad (2.1)$$

b. *Delay*

Jumlah waktu yang diperlukan paket data dalam perjalanannya dari asal menuju tujuannya dikenal sebagai *delay*. Media fisik, jarak, kongesti, dan waktu pemrosesan yang lambat semuanya dapat berkontribusi pada *delay*. Pada Tabel 2.6, merupakan klasifikasi *delay* menurut TIPHON.

Tabel 2.6 Standar *delay* [25]

Kategori	***Delay***	Indeks
Sangat Bagus	< 150 ms	4
Bagus	150 – 300 ms	3
Sedang	300 – 450 ms	2
Jelek	> 450 ms	1

Persamaan perhitungan *delay*:

$$Delay(s) = \frac{\text{Total } delay}{\text{Paket data yang diterima}} \qquad (2.2)$$

c. *Packet loss*

Jumlah paket yang tidak sampai pada destinasi yang diinginkan dikenal sebagai *packet loss*. Pada Tabel 2.7, merupakan kategori *packet loss* menurut TIPHON [25].

Tabel 2.7 Standar *packet loss* [25]

Kategori	Packet loss	Indeks
Sangat bagus	0%	4
Bagus	3%	3
Sedang	15%	2
Jelek	25%	1

Persamaan perhitungan *packet loss*:

$$Packet\ loss = \frac{(\text{Paket data dikirim} - \text{Paket data diterima})}{\text{Paket data diterima}} \times 100\% \qquad (2.3)$$

2.2.13 PERSENTASE *ERROR*

Selisih persen antara nilai percobaan dan nilai yang benar digunakan untuk menyatakan persentase *error*. Dalam sains, ini digunakan untuk mendokumentasikan varians antara nilai yang benar atau tepat dan yang telah diukur atau diperoleh melalui eksperimen. Menggunakan perbedaan antara nilai yang diukur atau eksperimental dan nilai yang diterima atau diketahui, dibagi dengan nilai yang diketahui dengan nilai yang diketahui, dikalikan dengan 100%, adalah cara umum untuk menghitung persentase error. Persentase *error* terkait dengan absolut *error* dan relatif *error*. Perbedaan antara nilai eksperimental dan nilai yang diketahui adalah absolut *error*. Saat membagi angka itu dengan nilai yang diketahui, maka akan mendapatkan relative *error*. Persentase *error* adalah kesalahan relatif dikalikan 100%. Dalam semua kasus, laporkan nilai menggunakan jumlah digit signifikan yang sesuai. Absolut *error* menggambarkan seberapa jauh nilai yang diukur (atau diklaim) terletak dari nilai sebenarnya. Berikut merupakan persamaan absolut *error:*

$$\text{Absolut}\ error = Measured\ value - True\ value \qquad (2.4)$$

Sedangkan relatif *error* membandingkan ukuran absolut *error* dengan nilai sebenarnya dan sering dinyatakan sebagai persentase. Berikut merupakan persamaan relatif *error*:

$$\text{Relatif}\ error = \frac{(Nilai\ yang\ diukur - Nilai\ sebenarnya)}{Nilai\ sebenarnya} \times 100\% \qquad (2.5)$$

Dimana dari perhitungan persentase *error* tersebut didapatkan nilai akurasi dengan persamaan sebagai berikut [26].

$$\text{Akurasi} = 100\% - \text{Relatif } error \quad (2.6)$$