

BAB II

DASAR TEORI

2.1 KAJIAN PUSTAKA

Beberapa contoh penelitian yang bisa dijadikan bahan pertimbangan yang bertujuan untuk membedakan antara penelitian ini dengan penelitian lain bertujuan memperkuat bahwasanya penelitian ini adalah penelitian asli, antara lain sebagai berikut:

Penelitian [4] menjelaskan mengenai pembuatan aplikasi android yang bertujuan untuk pengendali portable menggunakan smartphone android. Jaringan yang digunakan adalah jaringan lokal yang menggunakan raspberry sebagai servernya. Aplikasi android yang dibuat berfungsi untuk mengawasi dan mengontrol keranjang bayi secara real-time.

Penelitian [5] menjelaskan tentang penerapan internet of things pada sistem pengendali dan pengaman pintu berbasis android. Aplikasi yang dikembangkan dapat digunakan untuk memantau dan mengontrol aksesibilitas pintu dari mana saja melalui koneksi internet. Selain untuk mengontrol, aplikasi yang dikembangkan juga dapat digunakan untuk memonitoring.

Dari dua penelitian di atas, dapat dikaji bahwa media berbasis android memiliki kualitas yang baik digunakan untuk media atau platform untuk konfigurasi atau monitoring suatu alat kontroler jarak jauh tanpa kabel.

Penelitian [6] menjelaskan tentang bagaimana cara untuk mengetahui langkah – langkah dalam mengembangkan aplikasi. Pada penelitian tersebut, dilakukan pengembangan aplikasi cuaca untuk kegunaan sehari – hari. Pengembangan aplikasi tersebut dilakukan menggunakan *software* Android Studio, *software* yang sangat mudah digunakan untuk pengembangan aplikasi. Terdapat dua fitur utama pada *User Interface* aplikasi. Pertama, di bagian atas terdapat fitur yang menampilkan suhu terbaru serta ikon kondisi cuaca, di bagian bawahnya terdapat fitur yang berfungsi untuk melihat data angin, tekanan, kelembaban, matahari terbit, matahari terbenam di lokasi terdekat. Kedua, terdiri dari daftar yang menunjukkan perkiraan cuaca beberapa jam ke depan, dapat diatur setiap perjam hingga per enam jam.

Penelitian [7] menjelaskan mengenai pembuatan aplikasi android yang bertujuan untuk membantu menghubungkan para pemilik buku yang sudah tidak terpakai dengan para yang membutuhkan buku. Menurut penelitian yang dilakukan, aplikasi tersebut merupakan aplikasi donasi buku pertama yang dikembangkan di negara India dan menjadi aplikasi yang revolusioner. Terdapat tiga menu utama pada aplikasi yang dibuat, yaitu *Donate A Book*, *Find Donors*, dan *Make A Complaint* yang ditampilkan pada layar utama aplikasi tersebut.

Dua penelitian di atas menggunakan satu *software* yang sama yaitu android studio. Android studio menjadi salah satu *software* yang sering diandalkan ketika proses pengembangan aplikasi android. *Software* ini dapat digunakan untuk menulis program, *debugging*, serta simulasi aplikasi secara virtual. Berbagai fitur yang tersedia mampu membantu para *developer* untuk dapat menghasilkan aplikasi yang berkualitas. Dua penelitian di atas membuktikan bahwa android studio mampu memberikan kemudahan dalam membuat menu – menu pada aplikasi android dalam bentuk *activity* maupun *fragment*.

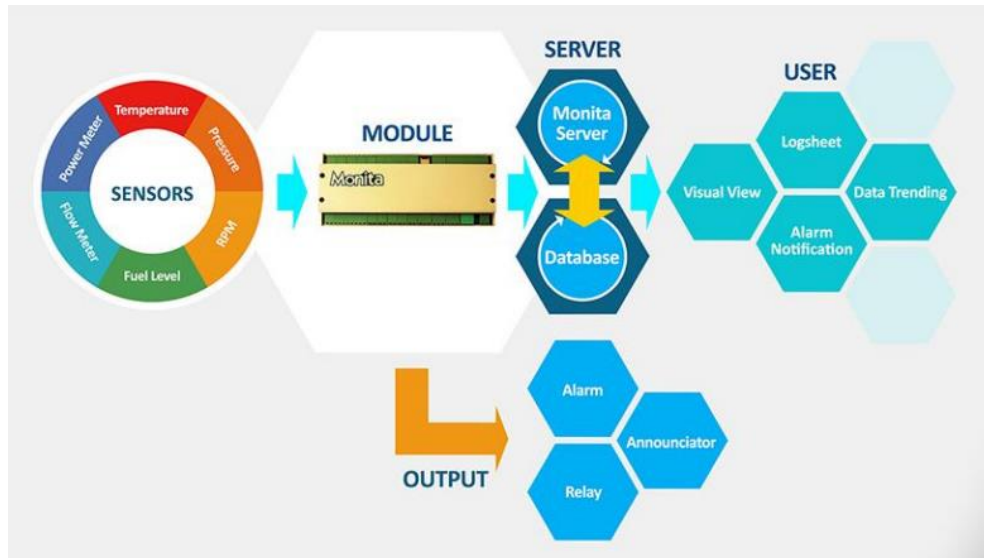
2.2 DASAR TEORI

2.2.1 Monita

Monita adalah sistem yang dirancang untuk kebutuhan data yang terus menerus (*continuous*). Sistem monita terdiri dari kombinasi *hardware* dan *software* untuk :

1. Pengumpulan,
2. Transmisi,
3. Penyimpanan,
4. Pemrosesan, dan
5. Pembuatan laporan [8].

Diagram untuk monita:



Gambar 2.1 Diagram Monita[8]

Online Monitoring System yang ditunjukkan pada Gambar 2.1 adalah sistem dasar Monita. Data equipment bisa dipantau dari jauh (*remote*) menggunakan berbagai sarana komunikasi dengan konsep “*monitoring only*”.

Komponen: [8]

1. Monita Analog *Module*



Gambar 2.2 Bentuk Fisik Modul Monita Analog[8]

Membaca data analog dan mengirimkannya ke server merupakan fungsi dari modul monita analog yang ditunjukkan pada Gambar 2.2, sensor yang bisa dibaca adalah:

- a. *Temperature*,
- b. *Pressure*,
- c. *Level Tangki*,
- d. *DC Voltage* [8].

2. Monita Binary Module



Gambar 2.3 Bentuk Fisik Modul Monita Binary[8]

Modul *binary* yang ditunjukkan pada Gambar 2.3 berfungsi untuk membaca data *binary*, *counter*, status (alarm) dan mengirim data ke server monita. Parameter yang bisa dibaca adalah:

- a. *Flow Meter*,
- b. *Running Hour*,
- c. Status Alarm,
- d. RPM (Mesin) [8].

3. Monita Hybrid Module



Gambar 2.4 Bentuk Fisik Modul Monita Hybrid[8]

Modul *hybrid* yang ditunjukkan pada Gambar 2.4 merupakan gabungan dari modul analog dan *binary*. Modul ini memiliki kemampuan komunikasi ke modem GPRS, modem satelit maupun *ethernet* (LAN). Dengan kemampuan ini, modul sangat cocok diaplikasikan untuk *monitoring* di area *remote* dengan parameter terbatas dan mengirimkan data secara langsung ke server [8].

4. Monita Server *Software*

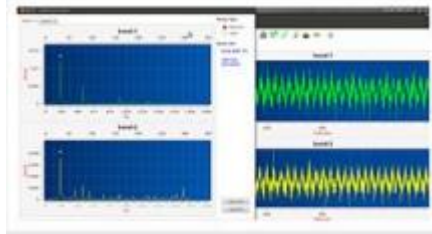


Gambar 2.5 Monita Server[8]

Server monita merupakan aplikasi server untuk satu *site/power plant* yang bisa diakses secara *online* ditunjukkan pada Gambar 2.5 dan berfungsi untuk:

- Pengambilan data,
- Pengiriman data,
- Penyimpanan data,
- Pembuatan *report* [8].

5. Monita Desktop *Software*



Gambar 2.6 Tampilan *Software* Monita Desktop[8]

Monita desktop seperti Gambar 2.6 merupakan aplikasi server untuk satu *site/power plant*, bisa diakses secara lokal dan berfungsi untuk:

- Pengambilan,
- Pengiriman,
- Penyimpanan,
- Pemrosesan,
- Pembuatan *report*, dan
- Simulasi data.

Software desktop ini sangat tepat untuk aplikasi *training center* [8].

Paket sistem monita :

1. Monita *Online Monitoring System*



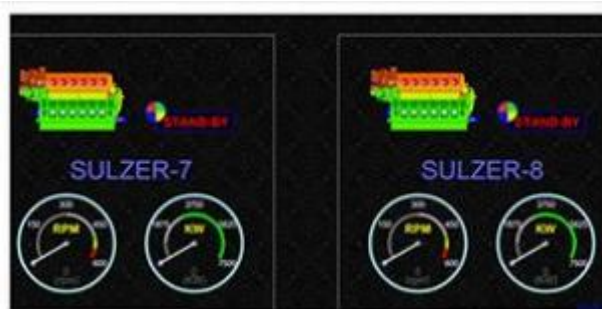
Gambar 2.7 Tampilan Monita *Online Monitoring System*[8]

Gambar 2.7 menunjukkan Sistem dasar dari monita yaitu *online monitoring system*. Data equipment bisa dipantau dari jauh (*remote*) menggunakan berbagai sarana komunikasi dengan konsep “*monitoring only*”.

Beberapa penggunaan monita *Online Monitoring System* adalah:

- a. Pemantauan PLTD,
- b. Pemantauan GI,
- c. Pemantauan level bahan bakar di area *remote*,
- d. Pemantauan kondisi *equipment* di area *remote* [8].

2. Monita *Engine Protection System*



Gambar 2.8 Tampilan *Engine Protection System*[8]

Sistem monita yang secara otomatis akan menghentikan *engine* jika parameter melebihi *level* tertentu. Level proteksi bisa diatur dari satu parameter ataupun multi parameter dan kondisi parameter bisa dimonitor secara online dengan sistem cerdas di modul monita. Gambar 2.8 menunjukkan tampilan *Protection System*. Beberapa penggunaan monita *protection system* adalah:

- a. Sistem proteksi *pad* mesin diesel,
- b. Sistem proteksi *exhaust* [8].

3. Monita AMR

Online monitoring system yang dikhususkan untuk pembacaan data meteran listrik dan kwh meter. Data ini disajikan dalam *software* AMR dalam bentuk file dengan format PDF/Excel yang siap untuk keperluan transaksi [8].

4. Monita Marine

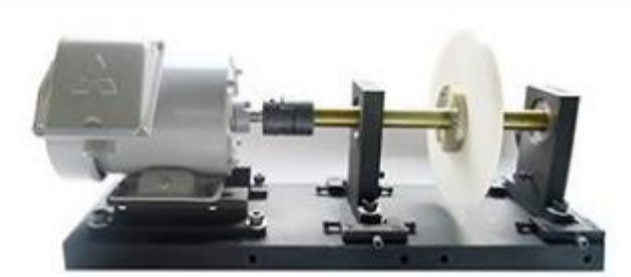


Gambar 2.9 Diagram Monita Marine[8]

Monita marine yang ditunjukkan pada Gambar 2.9 adalah sistem *monitoring vessel* yang membaca parameter – parameter operasi kapal dan mengirimkan ke pusat operasinya dengan menggunakan teknologi satelit. Data – data yang bisa dibaca dan dikirimkan adalah :

- a. RPM mesin,
- b. RPM *propeller*,
- c. *Flow meter* (konsumsi bahan bakar),
- d. KWH meter,
- e. Status generator,
- f. Level tangki bahan bakar,
- g. Lokasi (GPS) / *position tracking* [8].

5. Monita *Vibration Simulator*



Gambar 2.10 Bentuk Fisik Monita *Vibration Simulator*[8]

Monita *vibration simulator* pada Gambar 2.10 adalah sistem *hardware & software* yang berfungsi untuk melakukan simulasi vibrasi pada mesin – mesin *rotating* dan disesuaikan untuk keperluan *training* teknisi/engineer. Monita *vibration simulator* terdiri dari:

- a. *Rotating engine simulator*,
- b. Sensor set,
- c. *High speed vibration data module*,
- d. Desktop *software* [8].

2.2.2 Android

Pada tahun 2005, google mengembangkan sistem operasi untuk perangkat mobile berbasis linux yang diberi nama sistem operasi android. Pada Desember 2008, *PacketVideo, ARM Holdings, Atheros Communications, sustek Computer Inc, Garmin Ltd, Softbank, Sony Ericsson, Toshiba Corp, dan Vodafone Group PLC* diumumkan bergabung dengan proyek android[9]. Tabel 2.1 berikut menunjukkan perbandingan android dari versi *lollipop* hingga *snow cone*.

Tabel 2.1 Perbandingan Versi Android [10]

Versi	Tanggal Rilis	Fitur Untuk Pengguna	Fitur Untuk Pengembang	API	Nama Versi
Android 12	Q3 2021	<ul style="list-style-type: none">• Dapat menyesuaikan seluruh UI tergantung pada wallpaper yang dipilih• Peningkatan keamanan dan privasi seperti	<ul style="list-style-type: none">• Bahasa desain “<i>Material You</i>”• Inti komputasi pribadi android• Dukungan <i>Av1 Image</i>	31	<i>Snow Cone</i>

		garis waktu untuk mengetahui data apa yang diakses aplikasi	<i>File Format (AVIF)</i>		
Android 11	8 Sept 2020	<ul style="list-style-type: none"> • Perekaman layar • Pengaturan izin yang lebih baik • Gelembung pesan • Ethernet tethering 	<ul style="list-style-type: none"> • Dukungan aplikasi 5G • <i>Captive Portal</i> API • Datasets bersama • <i>Debugging</i> nirkabel 	30	<i>Red Velvet Cake</i>
Android 10	3 Sept 2019	<ul style="list-style-type: none"> • Dukungan privasi yang lebih baik • Akses pengaturan langsung menggunakan panel mengambang • Dukungan AV1, HDR10+, dan Opus • Dukungan Wi-Fi WPA3 • Tema gelap • Navigasi gerakan 	<ul style="list-style-type: none"> • MIDI API • Gelembung notifikasi • Autentikasi biometrik dalam aplikasi 	29	<i>Queen Cake</i>
Android 9	9 Agustus 2018	<ul style="list-style-type: none"> • Pembaruan UI : Sudut bulat diseluruh layar, bar notifikasi pindah ke kiri, dan transisi baru saat beralih aplikasi • <i>Power option</i> memiliki tangkapan layar • Autentikasi biometrik hanya dapat 	-	28	<i>Pie</i>

		dilakukan sekali			
Android 8	21 Agustus 2017	<ul style="list-style-type: none"> • <i>Picture-in-picture</i> dengan ukuran yang dapat diubah • Aplikasi instan android • Sistem notifikasi ditingkatkan • Pengaturan sistem ditingkatkan 	-	26	<i>Oreo</i>
Android 7	22 Agustus 2016	<ul style="list-style-type: none"> • Emoji 9.0 <i>unicode</i> • Multitasking yang lebih baik • <i>Mode Split Screen</i> 	<ul style="list-style-type: none"> • API SPM (<i>Sustained Performance Mode</i>) • API <i>rendering</i> 3D • <i>Virtual reality platform</i> 	24	<i>Nougat</i>
Android 6	5 Oktober 2015	<ul style="list-style-type: none"> • Dukungan USB tipe-C • Dukungan autentikasi sidik jari • Dukungan MIDI • Peningkatan Google Now 	<ul style="list-style-type: none"> • Pembaruan manajemen izin aplikasi 	23	<i>Marshmallow</i>
Android 5	17 Oktober 2014	<ul style="list-style-type: none"> • Desain baru (<i>Material Design</i>) • Peningkatan kecepatan • Peningkatan konsumsi baterai 	<ul style="list-style-type: none"> • Beberapa API baru • Melacak konsumsi baterai 	21	<i>Lollipop</i>

2.2.2.1 Aplikasi Android

Struktur aplikasi android atau fundamental aplikasi ditulis dalam bahasa pemrograman java. Kode java dikompilasi bersama dengan *resource file* yang dibutuhkan oleh aplikasi. Dimana prosesnya di *package* oleh *tools* yang dinamakan *apttools* kedalam paket android. Sehingga menghasilkan *file* dengan ekstensi apk. *File* apk ini yang disebut dengan aplikasi, dan nantinya dapat dijalankan pada peralatan *mobile*. Ada 4 komponen pada aplikasi android :

1. *Activities*

Activities merupakan komponen untuk menyajikan tampilan pemakai (*user interface*) kepada pengguna.

2. *Service*

Service merupakan komponen yang tidak memiliki tampilan pemakai (*user interface*), tetapi *service* berjalan secara *backgrounds*.

3. *Broadcast Receiver*

Broadcast receiver merupakan komponen yang berfungsi menerima dan bereaksi untuk menyampaikan notifikasi.

4. *Content Provider*

Content provider merupakan komponen yang membuat kumpulan aplikasi data secara spesifik, sehingga bisa digunakan aplikasi lain[11].

Tools pembangun android. Untuk membangun sebuah sistem operasi android dapat menggunakan MAC, Windows PC, ataupun Linux. *Tools* yang dibutuhkan gratis dan dapat di download dari web. Berikut adalah beberapa *tools* yang digunakan untuk membangun aplikasi android.

1. JDK (*Java Development Kit*),
2. Android SDK,
3. ADT (*Android Development Tools*)[11].

2.2.2.2 Java

Java merupakan bahasa OOP (*Object Oriented Programming*) yang dikembangkan di Sun Microsystems (yang sekarang anak perusahaan Oracle Corporation) dan dirilis pada tahun 1995. OOP adalah metodologi pemrograman yang membuatnya lebih nyaman untuk menggunakan kembali perangkat lunak. Java agak unik karena menggunakan kompiler dan penerjemah untuk mengonversi instruksi tingkat tinggi ke bahasa mesin. Kompiler digunakan untuk mengonversi Java menjadi bahasa tingkat menengah yang dikenal sebagai *bytecode*, dan kemudian dikonversi menjadi bahasa mesin menggunakan penerjemah. Gambar 2.11 merupakan contoh pemrograman bahasa java. Menampilkan teks *Hello World* pada java[12].

```

public class MyClass {
    public static void main(String[] args) {
        System.out.println("Hello World");
    }
}

```

Gambar 2.11 Contoh Pemrograman Bahasa Java Dasar[12]

Berkat fleksibilitasnya, java telah dijalankan di 13 miliar perangkat. Kehadiran java di tengah perkembangan teknologi ini karena sosok James Gosling di tahun 1995. Hingga saat ini, java menjadi bahasa pemrograman yang telah banyak dipakai oleh para *developer* karena keunggulannya. Di antaranya, mudah dijalankan di beberapa platform, memiliki *library* yang lengkap, berorientasi pada objek, hingga mudah dipelajari karena mirip bahasa C++[12].

2.2.2.3 XML

XML (*eXtensible Markup Language*) adalah bahasa yang didefinisikan oleh konsorsium W3C (World Wide Web Consortium), sebuah badan yang menetapkan standar untuk Web. Dengan demikian, XML merupakan evolusi dari bahasa markah lain seperti Hypertext Markup Language (HTML) yang elemen-elemennya sudah ditentukan sebelumnya tetapi tidak memadai. Sebaliknya, XML memungkinkan untuk membuat bahasa markup[13].

```

<?xml version="1.0"?> --> Deklarasi
<productlisting title:"Daftar Produk PT.A"> --> Atribut
  <product>
    <nama>Produk Pertama</nama> --> Elemen
    <deskripsi>Produk pertama meningkatkan daya tahan tubuh</deskripsi>
    <harga>Rp. 20.000</harga>
    <ongkir>Rp. 5000</ongkir>
  </product>
</productlisting>

```

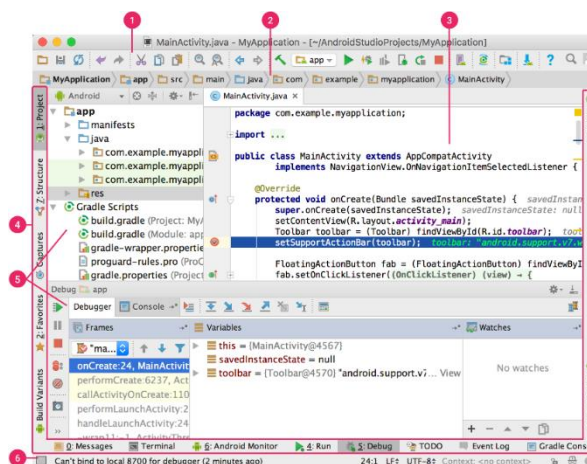
Gambar 2.12 Contoh Struktur XML[14]

XML dengan cepat diadopsi sebagai standar pertukaran data, khususnya untuk penggunaan lintas aplikasi dan platform. Kelebihan dari XML adalah karakteristiknya yang *extensible* dan platform independent. Berbeda dengan HTML, XML tidak mempunyai kosakata (berupa “*tag*”) yang baku, sebaliknya dengan XML bebas merancang *tag – tag* sendiri, sesuai dengan kebutuhan aplikasinya. Konsep dasar XML diantaranya:

- a. Karakter *unicode*, sebuah statement XML merupakan sekumpulan karakter. Hampir semua karakter *unicode* dapat tampil dalam dokumen XML.
- b. *Processor* dan *Application processor* menganalisa markup lalu menyerahkannya dalam bentuk dokumen terstruktur kepada *application*. Sebuah spesifikasi mengatur apa saja yang boleh dan tidak boleh dilakukan *processor*.
- c. *Markup* dan *content*, karakter – karakter yang membangun dokumen XML dibagi dalam *markup* dan *content*. *markup* adalah untaian karakter yang dimulai dengan karakter “<” dan diakhiri dengan karakter “>”. Untaian karakter selain *markup* adalah *content*.
- d. *Tag*, adalah *markup* yang bersifat deskriptif. *Tag* dapat dibagi menjadi tiga macam yaitu : *start-tags* <section> *end-tags* </section> dan *empty-element tags* <line-break/>.
- e. *Attribute*, setiap *element* XML dapat memiliki *attribute* yang berfungsi, untuk memberikan informasi tambahan terhadap *element* tersebut. Contoh : <file type="gif">computer.gif</file>.
- f. *XML declaration*, sebuah dokumen XML dapat dimulai dengan mendeklarasikan beberapa informasi. Contoh *declaration* : <?xml version="1.0" encoding="UTF-8"?>[14].

2.2.2.4 Android Studio

Android Studio adalah *Integrated Development Environment* (IDE) resmi untuk pengembangan aplikasi android, yang berdasarkan pada IntelliJ IDEA. Gambar 2.13 tampilan utama dari android studio.



Gambar 2.13 Tampilan Utama *Software Android Studio*[15]

1. Toolbar, memungkinkan melakukan berbagai tindakan, termasuk menjalankan aplikasi dan meluncurkan alat android.
2. Menu navigasi, membantu menjelajah *project* dan membuka file untuk diedit. Menu ini memberikan tampilan struktur yang lebih ringkas yang terlihat di jendela *project*.
3. Jendela editor, adalah tempat membuat dan memodifikasi kode. Bergantung pada jenis file yang ada, editor ini dapat berubah.
4. Panel jendela fitur, berada di sisi luar jendela IDE dan berisi tombol – tombol yang dapat diperluas dan diperkecil.
5. Jendela fitur, memberi akses ke tugas tertentu seperti pengelolaan *project*, penelusuran, kontrol versi, dan lainnya.
6. Status bar, menampilkan status *project* dan IDE, serta semua peringatan atau pesan.

Selain sebagai editor kode dan fitur *developer* IntelliJ yang andal, android studio menawarkan banyak fitur yang meningkatkan produktivitas dalam membuat aplikasi android, seperti :

- a. Sistem *build* berbasis *gradle* yang fleksibel,
- b. Emulator yang cepat dan kaya fitur,
- c. Lingkungan terpadu bisa mengembangkan aplikasi untuk semua perangkat android,
- d. Terapkan perubahan kode dan *resource* ke aplikasi yang sedang berjalan tanpa memulai ulang aplikasi,
- e. Template kode dan integrasi GitHub untuk membantu membuat fitur aplikasi umum dan mengimpor contoh kode,
- f. Dukungan bawaan untuk *Google Cloud Platform*, yang memudahkan integrasi *Google Cloud Messaging* dan *App Engine* [15].

2.2.2.5 Figma

Figma adalah sebuah platform *prototyping online* dan kolaboratif. *Figma* seperti campuran sketch dan photoshop, hanya saja *online*. *Figma* menarik karena sumber daya kolaboratifnya secara *realtime*. *Designer* dapat bekerja bersama dari jarak jauh. *Programmer* dapat menambah/menulis kode mereka

sendiri dalam proyek yang terdapat dalam *figma*. Kelebihan menggunakan *figma* adalah sebagai berikut :

1. *Figma* gratis untuk beberapa orang, namun untuk menggunakan bersama dengan tim ada biaya \$12 per bula per *editor*.
2. Siapa pun dapat membuka *file figma* tanpa perlu melakukan instalasi dan bisa langsung meninjau dan menulis komentar secara gratis.
3. Kecepatan dan performa bisa jauh lebih cepat saat mengedit dan melihat *file*.
4. Platform bersifat agnostik, sehingga dapat dijalankan di sistem operasi dan *browser* apapun.

Kolaborasi bisa berjalan secara *real-time*, demikian juga ketika melakukan pembaruan *file*[16].

2.2.3 *Bluetooth*

Teknologi *bluetooth* memungkinkan elektronik-elektronik untuk terhubung tanpa kabel atau kawat. *Bluetooth* dapat ditemukan di ponsel, komputer, perangkat medis, dan perangkat-perangkat hiburan. *Bluetooth* dinamai untuk menghormati raja Denmark abad ke-10 bernama Harald *Bluetooth*. *Bluetooth* menambah kenyamanan pada produk yang kita gunakan setiap hari. Biasanya, perangkat dihubungkan dengan kabel, tetapi kabel dapat kusut dan menghabiskan ruang di meja. *Bluetooth* menggantikan kabel, membuat perangkat lebih mudah digunakan dan lebih mobile.

Perangkat *bluetooth* menggunakan gelombang radio untuk terhubung. Gelombang radio berasal dari sebuah chip komputer. Ketika satu perangkat *bluetooth* mendekati perangkat *bluetooth* lainnya, perangkat tersebut akan mengirimkan sinyal radio satu sama lain. Sebagian besar perangkat *bluetooth* memiliki jangkauan sekitar 30 kaki (10 meter). Perangkat terhubung dan membuat PAN (*Personal Area Network*)[17].

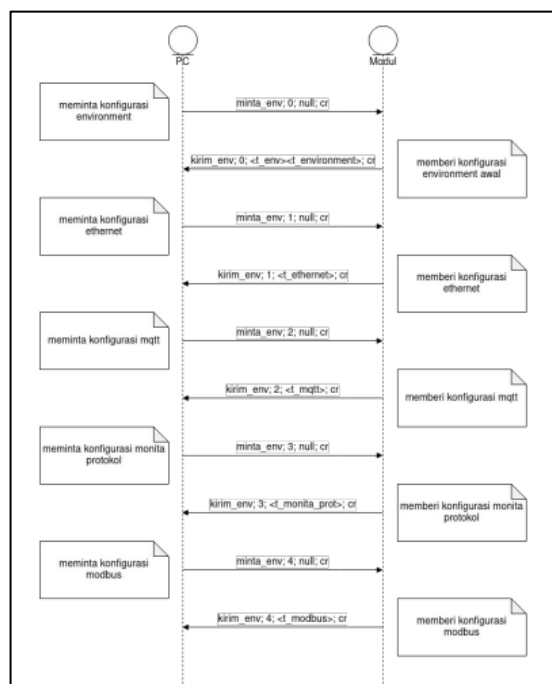


Gambar 2.14 Bentuk Fisik Modul *Bluetooth* JDY-31[18]

Gambar 2.14 menunjukkan bentuk fisik modul *Bluetooth* yang digunakan yaitu tipe JDY-31. JDY-31 bluetooth didasarkan pada desain bluetooth 3.0 SPP, yang dapat mendukung Windows, Linux, transmisi data android, frekuensi kerja 2,4 GHz, mode modulasi GFSK, data transmisi maksimum 8db, jarak transmisi maksimum 30 meter, dukungan pengguna untuk memodifikasi nama perangkat melalui perintah AT, nyaman dan cepat digunakan[18].

2.2.4 Protokol Komunikasi

Protokol komunikasi adalah deskripsi formal dari format dan aturan pesan digital. Protokol komunikasi penting dalam sistem telekomunikasi dan sistem lainnya karena menciptakan konsistensi dan universalitas untuk pengiriman dan penerimaan pesan. Protokol komunikasi dapat mencakup otentikasi, deteksi dan koreksi kesalahan, dan pensinyalan, serta dapat menggambarkan sintaksis, semantik, dan sinkronisasi komunikasi analog dan digital. Protokol komunikasi diimplementasikan dalam perangkat keras dan perangkat lunak. Ada ribuan protokol komunikasi yang digunakan di mana – mana dalam komunikasi analog dan digital [19]. Gambar 2.15 menunjukkan contoh implementasi protokol komunikasi antara CLI (*Command Line Interface*) dan monita [20].



Gambar 2.15 Contoh Protokol Komunikasi[19]

2.2.5 Standard Apptim Untuk Aplikasi Android

Ambang batas default yang ditetapkan di Apptim, yang didasarkan pada praktik terbaik Google (Android) dan praktik terbaik Apple (iOS), bersama dengan tolak ukur pasar lainnya yang memperimbangkan fragmentasi perangkat. Masing – masing akan memiliki warna sesuai dengan hasil terhadap penerapan ambang batas[21].

a. Batas Rendah

Saat nilai berada dalam rentang ambang batas default, ditandai dengan tanda warna hijau.

b. Batas Sedang

Saat nilai melebihi ambang batas “direkomendasikan” yang dapat menyebabkan masalah di beberapa perangkat, ditandai dengan tanda berwarna kuning.

c. Batas Peringatan

Jika nilai melebihi ambang batas “dapat diterima” dan kemungkinan akan menyebabkan masalah pada beberapa perangkat, ditandai dengan tanda warna merah[21].

Ambang batas default untuk aplikasi android :

Ambang batas performa perangkat dapat dilihat pada Tabel 2.2

Tabel 2.2 Ambang Batas Default Performa Perangkat Android[21]

Kategori	Batas Sedang	Batas Peringatan
Rata – rata penggunaan CPU	20%	50%
Penggunaan CPU	70%	90%
Rata – rata penggunaan memori	256 Mb	384 Mb
Penggunaan memori	384 Mb	512 Mb

Ambang batas performa aplikasi dapat dilihat pada Tabel 2.3

Tabel 2.3 Ambang Batas Default Performa Aplikasi Android[21]

Kategori	Batas Sedang	Batas Peringatan
Rata – rata FPS	30 FPS	15 FPS

2.2.6 Perbedaan Antara *White-Box Testing* dan *Black-Box Testing*

Istilah “White-Box” digunakan karena perspektif internal sistem. Kotak bening atau kotak putih, atau nama kotak transparan menunjukkan kemampuan untuk melihat melalui kulit luar perangkat lunak ke dalam cara kerjanya. Hal ini dilakukan oleh pengembang, dan kemudian perangkat lunak akan dikirim ke tim

penguji, di mana mereka melakukan pengujian kotak hitam. Tujuan utama pengujian white-box adalah untuk menguji infrastruktur aplikasi. Hal ini dilakukan pada tingkat yang lebih rendah, karena mencakup pengujian unit dan pengujian integrasi. Hal ini membutuhkan pengetahuan pemrograman, karena sebagian besar berfokus pada struktur kode, jalur, kondisi, dan cabang dari sebuah program atau perangkat lunak. Tujuan utama dari pengujian white-box adalah untuk fokus pada aliran input dan output melalui perangkat lunak dan memperkuat keamanan perangkat lunak [22].

Sumber utama “Black-Box” adalah spesifikasi persyaratan yang dinyatakan oleh pelanggan. Ini adalah teknik pengujian perangkat lunak yang memeriksa fungsionalitas perangkat lunak tanpa mengetahui struktur internal atau program. Ini tidak memerlukan pengetahuan pemrograman perangkat lunak. Semua kasus pengujian dirancang dengan mempertimbangkan input dan output dari fungsi tertentu. Dalam pengujian ini, insinyur penguji menganalisis perangkat lunak terhadap persyaratan, mengidentifikasi cacat atau bag, dan mengirimkannya Kembali ke tim pengembangan. Dalam metode ini, penguji memilih suatu fungsi dan memberikan nilai output untuk menguji fungsionalitasnya, dan memeriksa apakah fungsi tersebut memberikan output yang diharapkan atau tidak. Jika fungsi tersebut menghasilkan output yang benar, maka fungsi tersebut lolos dalam pengujian, jika tidak maka fungsi tersebut gagal. Use Case Technique yang digunakan untuk mengidentifikasi kasus pengujian dari awal hingga akhir sistem sesuai dengan penggunaan sistem. Dengan menggunakan teknik ini, tim penguji membuat scenario pengujian yang dapat menjalankan seluruh perangkat lunak berdasarkan fungsionalitas setiap fungsi dari awal hingga akhir [22].