

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Karakteristik Anak Usia Dini**

Anak adalah asset bangsa yang paling berharga. Karena anak adalah penerus generasi bangsa itu sendiri. Dalam perkembangannya sangat diperlukan sekali perhatian yang ekstra guna memperoleh anak yang memiliki sumber daya manusia yg berkompeten. Maka dari itu, dalam perjalanannya, mengasuh anak bukan Cuma merawat dan memenuhi kebutuhan sehari-hari, tetapi juga memberikan pengertian serta pembelajaran yang efektif. Medidik anak sejak dini menjadi suatu kewajiban orang tua di rumah sejak dari kandungan hingga beranjak dewasa.

Dalam UU NO. 20 TAHUN 2003 tentang Sistem Pendidikan Nasional Bab 1, Pasal 1, Butir 14 dinyatakan bahwa "Pendidikan Anak Usia Dini adalah suatu upaya pembinaan yang ditujukan kepada anak sejak lahir sampai dengan usia 6 tahun yang dilakukan melalui pemberian rangsangan pendidikan untuk membantu pertumbuhan dan perkembangan jasmani dan rohani agar anak memiliki kesiapan dalam memasuki pendidikan lebih lanjut". Sedangkan pada pasal 28 tentang Pendidikan Anak Usia Dini dinyatakan bahwa "(1) Pendidikan Anak usia dini diselenggarakan sebelum jenjang pendidikan dasar, (2) Pendidikan anak usia dini dapat diselenggarakan melalui jalur pendidikan formal, non formal, dan/atau informal, (3) Pendidikan anak usia dini jalur pendidikan formal: TK, RA, atau bentuk lain yang sederajat, (4) Pendidikan anak usia dini jalur pendidikan non formal: KB, TPA, atau bentuk lain yang sederajat, (5) Pendidikan usia dini jalur pendidikan informal: pendidikan keluarga atau pendidikan yang diselenggarakan oleh lingkungan, dan (6) Ketentuan mengenai pendidikan anak usia dini sebagaimana dimaksud dalam ayat (1), ayat (2), ayat (3), dan ayat (4) diatur lebih lanjut dengan peraturan pemerintah."

## **2.2 Pengertian *Android***

Android merupakan OS (Operating System) Mobile yang tumbuh ditengah OS lainnya yang berkembang dewasa ini. OS lainnya seperti Windows Mobile, i-Phone OS, Symbian, dan masih banyak lagi (Hermawan, 2011).

### **2.2.1 Versi Android**

#### **1. Apple Pie V1.0**

Untuk OS pertama yang dimiliki oleh Android adalah Apple Pie, OS ini dirilis pada tanggal 23 September 2008.

#### **2. Banana Bread V1.1**

Dan untuk OS Android yang selanjutnya adalah Banana Bread yang dirilis pada bulan Februari 2009 yang digunakan oleh Smartphone pengadopsi Android, HTC dengan merk Dream.

#### **3. CupCake V1.5**

Untuk OS yang selanjutnya adalah Cupcake, nah mulai dari sini versi Android CupCake mulai menunjukkan kecanggihannya sebagai Sistem Operasi Android, dan ini kali pertama Google menjadikan nama makanan.

#### **4. Donut V1.6**

Versi Donut ini dirilis pada bulan September 2009, di versi ini Google mulai memperbaiki Bug yang membuat Android ini lebih sempurna ditambah lagi dengan fitur navigasi turn-by-turn

## **5. Eclair V2.0-2.1**

Versi Donut ini dirilis pada bulan September 2009, di versi ini Google mulai memperbaiki Bug yang membuat Android ini lebih sempurna ditambah lagi dengan fitur navigasi turn-by-turn

## **6. Froya V2.2**

Pada bulan Mei 2010 Paduan antara Google Chrome dengan Android mulai nampak dengan ditambahkan script Chrome dengan upaya untuk meningkatkan kecepatan kinerja. Frozen Yogurt sendiri yakni memiliki makna Yogurt yang dibekukan, jadi seperti Ice Cream.

## **7. GingerBread V2.3**

Bulan Desember 2010 seiring dirilisnya Gingerbread, Android kedatangan salah satu merk Platform raksasa Smartphone yakni Samsung yang meluncurkan Samsung Nexus. Di android versi 2.3 ini Google menambahkan fitur baru yakni NFC, internet Calling, Download Manager.

## **8. HoneyComb V3.0-3.2**

Lanjutan dari Huruf berabjad H adalah Honeycomb yang dirilis pada Februari 2014 , untuk versi ini Android diperuntukkan dan spesialis Tablet dimana era Tablet sedang menjadi-jadi di tahun ini, Update tak hanya di versi 3.0 saja namun juga dilanjutkan ke versi 3.1 dan 3.2.

## **9. Ice Cream Sandwich V4.0**

Revormasi Android mulai terlihat dengan adanya Ice Cream Sandwich dimana OS versi ini mulai bekerja disemua jenis platform dan smartphone seperti tablet , fiturnya juga menjadi bertambah diantaranya adalah ekstra multitasking dan notifikasi yang lebih banyak.

## **10. Jelly Bean V4.1.2**

Android versi Jelly Bean ini dirilis pada 27 Juni 2014 lewat konferensi I/O Google. Jelly Bean menjadi versi Android yang juga banyak mendapatkan update, tercatat 2 kali sudah update dilakukan di Jelly Bean yakni versi 4.1.2 dimana perbedaan dibanding versi sebelumnya adalah segi User Interface yang lebih elegan serta penambahan fitur Google Search.

### **11. Kitkat V4.4.2**

Android Versi KitKat ini dirilis pada tahun 2013 dan paling banyak dipakai pada smartphone masa kini.

### **12. Lollipop V5.0**

Android Lollipop adalah Android versi terbaru yang diluncurkan Google pada tahun 2014. Versi Lollipop ini pertama kali ditanamkan di Smartphone Google Nexus 6

### **13. Marsmellow V6.0**

Rilis Terbaru Android 6.0 Bernama Marshmallow | Pasti sobat sudah pada tahu nih OS Smartphone terlaris jaman ini selalu menghadirkan nama nama unik setiap versi yang dirilisnya untuk versi lengkap dari awal sudah pernah saya bahas di Urutan Nama Nama Versi Android Dari Banyak yang mengira bahwa versi android seteleah versi 5.0 Lollipop adalah Milkshake, namun google membantah akan dugaan itu. Google secara resmi mengeluarkan Android versi 6.0 yang diberi nama Marshmallow. Selain itu Android Marshmallow juga akan menambah fitur fitur terbarunya.

### **14. Nougat V7.0**

Nougat adalah versi Android termutakhir yang baru diperkenalkan pada ajang kumpul developer Google I/O, pertengahan 2016 ini. Beberapa lama setelahnya, Google menghadirkan Nougat secara resmi untuk publik. Pembaruan paling mendasar pada versi Nougat adalah kehadiran Google Assistant yang menggantikan Google Now. Asisten digital tersebut lebih bisa diandalkan untuk menjalankan berbagai fungsi. Fitur-fitur baru lainnya mencakup layar split-screen saat dipakai multitasking, serta fitur Doze yang telah dikenalkan di versi Android Marshmallow namun telah ditingkatkan. Android Nougat juga memiliki dukungan terhadap platform virtual reality terbaru Google.

### **15. Oreo V8.0**

untuk pengguna android sekarang lagi booming booming nya nih tentang OS terbarunya, ya itu adalah OS android versi 8.0 atau yang sering juga disebut sebagai Android O atau Android Oreo. Versi android ini resmi diperkenalkan oleh Google pada tanggal 22 Agustus 2017 yang lalu dan juga sudah secara resmi bisa diluncurkan langsung ke lapangan, tapi sebelum versi android ini diresmikan oleh google, nama "Oreo" sendiri sudah terendus sejak Android O pertama kali diperkenalkan di ajang Google I/O 2017 pada Mei 2017 lalu.

### **16. Pie V9.0**

Os android versi kesembilan ini secara resmi dirilis pada 6 Agustus 2018, android p di kenal dengan nama "*Android Pie*". Os terbaru android ini pertama kali di rilis untuk perangkat atau smartphone google pixel.

## **2.3 Android Studio**

Android Studio merupakan sebuah *Integrated Development Environment (IDE)* khusus untuk membangun aplikasi yang berjalan pada platform android. Android studio ini berbasis pada IntelliJ IDEA, sebuah IDE untuk Bahasa pemrograman Java. Bahasa pemrograman utama yang digunakan adalah Java, sedangkan untuk

membuat tampilan atau layout, digunakan bahasa XML. Android studio juga terintegrasi dengan *Android Software Development Kit (SDK)* untuk deploy ke perangkat android. Android Studio juga merupakan pengembangan dari eclipse, dikembangkan menjadi lebih kompleks dan professional yang telah tersedia didalamnya Android Studio IDE, Android SDK tools. Setiap proyek di Android Studio berisi satu atau beberapa modul dengan file kode sumber dan file sumber daya. Jenis-jenis modul mencakup:

- Modul aplikasi Android
- Modul Pustaka
- Modul Google App Engine

Secara default, Android Studio akan menampilkan file proyek Anda dalam tampilan proyek Android, seperti yang ditampilkan dalam gambar 2.1. Tampilan disusun berdasarkan modul untuk memberikan akses cepat ke file sumber utama proyek Anda. Semua file versi terlihat di bagian atas di bawah Gradle Scripts dan masing-masing modul aplikasi berisi folder berikut:

1. manifests: Berisi file *AndroidManifest.xml*.
2. java: Berisi file kode sumber Java, termasuk kode pengujian JUnit.
3. res: Berisi semua sumber daya bukan kode, seperti tata letak XML, string UI, dan gambar bitmap.

#### **2.4 Java Development Kit (JDK)**

*Java Development Kit (JDK)* merupakan perangkat lunak yang digunakan untuk manajemen dan membangun berbagai aplikasi Java. JDK merupakan superset dari JRE, berisikan segala sesuatu yang ada di JRE (*Java Runtime Environment*) ditambahkan *compiler* dan *debugger* yang diperlukan untuk membangun aplikasi.

## **2.5 Pengertian Multimedia**

Multimedia adalah penggunaan komputer untuk menyajikan dan menggabungkan teks, suara, gambar, animasi dan video dengan alat bantu (*tool*) dan koneksi (*link*) sehingga pengguna dapat bernavigasi, berinteraksi, berkarya dan berkomunikasi (Darmawan, 2011).

## **2.6 Unified Modeling Language (UML)**

*Unified Modeling Language (UML)* sebuah teknik pengembangan sistem yang menggunakan bahasa grafis sebagai alat pendokumentasian dan melakukan spesifikasi pada sistem (mulyani, 2016).

Indrajani dalam bukunya menyebutkan bahwa, diagram – diagram yang termasuk di dalam UML, yaitu:

1. *Use Case Diagram*
2. *Activity Diagram*
3. *Statechart Diagram*
4. *Component Diagram*
5. *Deployment Diagram Object Diagram*
6. *Class Diagram (Static Structure)*
7. *Sequence Diagram*
8. *Collaboration Diagram*

Diagram UML yang ada begitu banyak, namun pada umumnya dalam proses pembangunan ataupun pengembangan sistem, diagram yang harus ada yaitu, *usecase, activity, sequence, dan class diagram* (Indrajani, 2015).

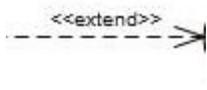
### 2.6.1 Use Case Diagram

*Use case diagram* adalah suatu diagram yang di antaranya berisi, *use case*, *actor*, serta *relationship*. *Use case* diagram merupakan titik awal yang baik dalam memahami dan menganalisis kebutuhan sistem pada saat perancangan. *Use case diagram* dapat digunakan untuk menentukan kebutuhan apa saja yang diperlukan dari suatu sistem, jadi dapat digambarkan dengan detail bagaimana suatu sistem memproses atau melakukan sesuatu, bagaimana cara aktor akan menggunakan sistem, serta apa saja yang dapat dilakukan terhadap suatu sistem. Notasi yang digunakan *use case* adalah persegi panjang yang merupakan *system boundary*, *oval* yang merupakan suatu proses, dan gambar *actor* yang berinteraksi dalam proses tersebut (Indrajani, 2015).

Table 2.1 Simbol Use Case Diagram (Sugiarti, 2013).

Nama	Simbol	Deskripsi
<i>Use Case</i>		Fungsionalitas yang disediakan sistem sebagai unit – unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal di frase nama <i>use case</i> .
<i>Actor</i>		Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri. Aktor hanya memberikan informasi ke sistem, aktor hanya menerima informasi dari sistem, aktor memberikan dan menerima informasi ke sistem dan dari sistem.

<i>Association</i>		Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor. Asosiasi merupakan hubungan statis antar elemen yang menggambarkan elemen yang memiliki atribut berupa elemen lain, atau elemen yang harus mengetahui eksistensi elemen lain.
--------------------	--	--

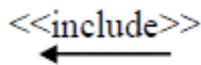
<b>Nama</b>	<b>Simbol</b>	<b>Deskripsi</b>
<i>Extend</i>		Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walaupun tanpa <i>use case</i> tambahan itu, mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek. <i>use case</i> tambahan biasanya memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan. Arah panah mengarah pada <i>use case</i> yang ditambahkan, biasanya <i>use case</i> yang menjadi <i>extend</i> -nya merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya.
<i>eneralization</i>		Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i>

dimana fungsi yang satu adalah fungsi yang lebih umum dari yang lainnya, misalnya: arah panah mengarah pada *use case* yang menjadi generalisasinya (umum). Generalisasi merupakan hubungan hirarkis antara elemen. Elemen dapat mewarisi semua atribut dan metode

---

elemen asalnya dan menambah fungsionalitas baru.

*Include*



Relasi *use case* tambahan ke sebuah *use case* yang ditambahkan memerlukan *use case* ini untuk menjalankan fungsinya atau sebagai syarat.

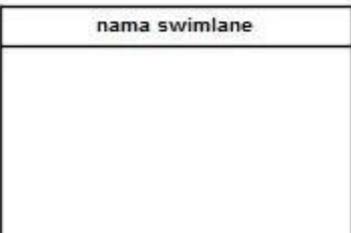
---

### 2.6.2 Activity Diagram

*Activity diagram* digunakan untuk menganalisis *behavior* dengan *use case* yang lebih kompleks dan menunjukkan intraksi – interaksi di antara mereka satu sama lain. *Activity diagram* sebenarnya memiliki kesamaan dengan *statechart diagram*, dalam hal ini menggambarkan aliran data pada model bisnis, tetapi *activity diagram* biasanya digunakan untuk menggambarkan aktivitas bisnis yang lebih kompleks, di mana digambarkan hubungan antar satu *use case* dengan *use case* lain (Indrajani, 2015).

Tabel 2.2 Simbol Activity Diagram (Sugiarti, 2013).

Nama	Simbol	Deskripsi
<i>InitialState</i>		Status awal aktivitas sistem, sebuah aktivitas memiliki sebuah status awal.
<i>ActionState</i>		Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
<i>Decision</i>		Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
<i>Join</i>		Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Nama	Simbol	Deskripsi
<i>FinalState</i>		Status akhir yang dilakukan sistem, sebuah diagram

		aktivitas memiliki sebuah status akhir.
<i>Swimlane</i>		Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

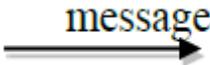
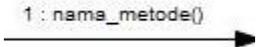
### 2.6.3 Sequence Diagram

*Sequence diagram* adalah suatu diagram interaksi yang menggambarkan bagaimana objek – objek berpartisipasi dalam bagian interaksi (*particular interaction*) dan pesan yang ditukar dalam urutan waktu.

*Sequence diagram* merupakan peralatan untuk interaksi berkomunikasi diagram. Sebuah interaksi didesain antara objek atau sistem yang berpartisipasi dalam sebuah kolaborasi. Interaksi dijelaskan oleh pesan – pesan yang diletakkan pada sebuah waktu, atau lebih dari dua pesan yang akan dikirim pada saat yang sama. Interaksi merupakan peran komunikasi yang penting pada kenyataannya. Konsep yang dapat diterapkan pada konteks bervariasi. *Sequence diagram* menggambarkan interaksi antar objek secara beraturan sesuai dengan waktu. *Sequence diagram* dapat digambarkan dalam *level* secara detail dan untuk tujuan *lifecycle*, ketika pesan dikirim pada sebuah objek akan meminta sebuah operasi dari objek, namun pesan biasanya sesuai dengan operasi yang akan diminta. Sebuah pesan diterima, operasi yang telah diminta akan melaksanakan pesan tersebut. *Sequence diagram* pada beberapa tahap selama operasi yang dilakukan tadi, disebut aktivitas (Indrajani, 2015).

Tabel 2.3 Simbol Sequence Diagram

Nama	Simbol	Deskripsi
------	--------	-----------

<i>Actor</i>		Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar
<b>Nama</b>	<b>Simbol</b>	<b>Deskripsi</b>
		sistem informasi yang akan dibuat itu sendiri.
<i>Lifeline</i>		Menyatakan kehidupan suatu objek, untuk menggambarkan kelas dan objek.
<i>Object</i>		Menyatakan objek yang berinteraksi (pesan).
Waktu aktif		Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif.
Pesan tipe <i>create</i>		Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
Pesan tipe <i>call</i>		Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri, sesuai dengan kelas objek yang berinteraksi.

#### 2.6.4 Class Diagram

*Class diagram* digunakan untuk menggambarkan perbedaan yang mendasar antara *class-class*, hubungan antar – *class*, dan di mana sub-sistem *class* tersebut. *Class diagram* sendiri terdapat sebuah nama *class*, *attributes*, *operations*, serta *association* (hubungan antar – *class*) (Indrajani, 2015).

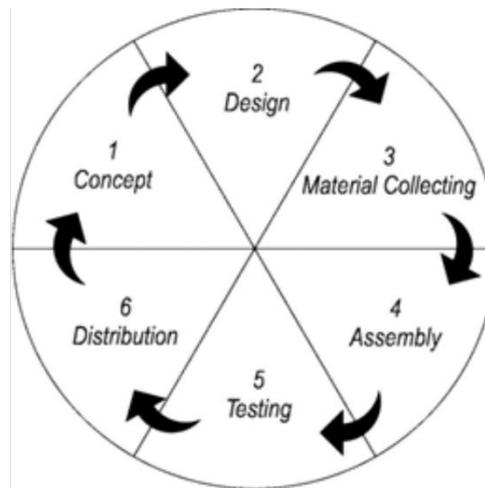
Tabel 2.4 Simbol Class Diagram

Nama	Simbol	Deskripsi
<i>Generalization</i>		Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).
<i>Nary Association</i>		Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
<i>Class</i>		Himpunan dari objek – objek yang berbagi atribut serta operasi yang sama.

<i>Collaboration</i>		Deskripsi dari urutan aksi – aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i> .
<i>Realization</i>		Operasi yang benar-benar dilakukan oleh suatu objek.
<i>Dependency</i>		Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
<i>Association</i>		Apa yang menghubungkan antara objek satu dengan objek lainnya.

## 2.7 Multimedia Development Life Cycle

*Multimedi Development Life Cycle* merupakan salah satu metodologi perangkat lunak. Menurut Luther dalam Binanto, metdologi pengembangan perangkat lunak sendiri terdapat enam tahap, yaitu *concept* (pengonsepan), *design* (pendedainan), *material collecting* (pengumpulan materi), *assembly* (pembuatan), *testing* (pengujian), *distribution* (pendistribusian). Keenam tahap ini tidak harus saling berurutan dalam pelaksanaannya, tahap-tahap tersebut juga dapat saling bertukar posisis. Meski begitu, tahap *concept* memang harus menjadi hal yang pertama kali dikerjakan. Berikut ini adalah gambar struktur metodologi sekaligus penjelasan dari keenam tahap diatas :



Gambar 2.1 Metodologi Pengembangan Multimedia

1. *Concept*

Tahap *concept* (pengonsepan) adalah tahap untuk menentukan tujuan dan siapa pengguna program (identifikasi audiens).

2. *Design*

*Design* (perancangan) adalah tahap pembuatan spesifikasi mengenai arsitektur program, gaya, tampilan, dan kebutuhan material/bahan untuk program. Spesifikasi dibuat serinci mungkin sehingga pada tahap berikutnya, yaitu *material collecting* dan *assembly*, pengambilan keputusan baru tidak perlu dilakukan lagi, cukup menggunakan keputusan yang sudah ditentukan pada tahap ini. Meskipun demikian, pada praktiknya, pengerjaan proyek pada tahap awal masih akan sering mengalami penambahan bahan atau pengurangan bagian aplikasi, atau perubahan-perubahan lain.

3. *Material Collecting*

*Material Collecting* (pengumpulan bahan) adalah tahap pengumpulan bahan yang sesuai dengan kebutuhan yang dikerjakan. Bahan-bahan tersebut antara lain gambar *clip art*, foto, animasi, audio, dan lain-lain yang dapat

diperoleh secara gratis atau dengan pemesanan kepada pihak lain sesuai dengan rancangannya.

4. *Assembly*

Tahap *Assembly* adalah tahap pembuatan semua objek atau bahan multimedia. Pembuatan aplikasi didasarkan pada tahap *design*.

5. *Testing*

Tahap *testing* (pengujian) dilakukan setelah menyelesaikan tahap pembuatan (*assembly*) dengan menjalankan aplikasi program dan melihatnya apakah ada kesalahan atau tidak. Tahap pertama pada tahap ini disebut tahap pengujian alpha (*alpha test*) yang pengujiannya dilakukan oleh pembuat atau lingkungan pembuatnya sendiri. Setelah lolos dari pengujian alpha, pengujian beta yang melihatkan pengguna akhir akan dilakukan.

6. *Distribution*

Pada tahap ini, aplikasi akan disimpan dalam suatu media penyimpanan. Jika media penyimpanan tidak cukup untuk menampung aplikasinya, kompresi terhadap aplikasi akan dilakukan. Tahap ini juga dapat disebut tahap evaluasi untuk mengembangkan produk yang sudah jadi supaya menjadi lebih baik. Hasil evaluasi ini dapat digunakan sebagai masukan untuk tahap *concept* pada produk selanjutnya.

## 2.8 Waktu dan Objek Penelitian

Waktu penelitian dilakukan 22 April 2019 sampai dengan selesai dengan Objek Penelitian yang penulis gunakan adalah PAUD Mahkota Bunda Labuhan Ratu Bandar Lampung.

Tabel 2.5 Jadwal Penelitian

No	Kegiatan	Waktu Penelitian
----	----------	------------------

		Bulan Ke-									
		I	II	III	IV	V	VI	VII	VIII	IX	X
1.	Penyusunan dan pengujian Proposal										
2.	Pengumpulan Data										
3.	Pengerjaan Program										

## 2.9 Analisa Kebutuhan

### 2.9.1 Analisis Kebutuhan Perangkat Keras

Kebutuhan perangkat keras yang digunakan dalam membangun aplikasi ini menggunakan sebuah *laptop* dengan spesifikasi sebagai berikut:

1. Intel Core i5-5200U CPU 2.20 GHz 2.19 GHz
2. RAM 4GB
3. NVIDIA GEFORCE 930M
4. HDD 1TB
5. Perangkat input output

Kemudian untuk perangkat keras agar dapat menjalankan aplikasi ini menggunakan *smartphone android* dengan spesifikasi minimal sebagai berikut:

1. Android V7.0 Nougat

### **2.9.2 Kebutuhan Perangkat Lunak**

Kebutuhan perangkat lunak yang digunakan dalam membangun aplikasi ini adalah sebagai berikut:

1. Sistem Operasi *Windows 10 64bit*
2. *Android Studio*
3. *Google Chrome*
4. *Adobe Photoshop*
5. *Adobe Illustrator*
6. *Adobe Audition*
7. *Audio Recorder*
8. *CorelDRAW*