

BAB II

LANDASAN TEORI

2.1. Tinjauan Pustaka

Sampai saat ini telah banyak dilakukan penelitian dengan tema pengembangan aplikasi pembelajaran dengan menggunakan platform android. Pada tahun 2017, Nur Maemanah dalam penelitian berjudul Perancangan Aplikasi *Mobile* “Aplikasi pembelajaran *Vocabulary* menggunakan *TTS*” Berbasis Android, mengembangkan sebuah aplikasi berbasis android untuk anak-anak dalam mempelajari kosakata Bahasa Inggris. Pada penelitian ini dalam pembuatan aplikasi tersebut, penulis menggunakan *Android Studio*.

Penelitian ini bertujuan untuk membantu anak-anak untuk mengenalkan materi Bahasa Inggris serta kosakata sebab komponen tersebut penting dalam dunia pendidikan.

Penelitian Suyono et al (2020) dalam penelitian yang berjudul “Aplikasi Belajar Cepat Pengenalan Kosakata Bahasa Inggris bagi Anak Usia Dini”. Dalam bidang Pendidikan saat ini sudah memasuki perkembangan teknologi yang sangat pesat. Di bidang Pendidikan atau pembelajaran khususnya anak usia dini pembelajaran dituntut mengikuti perkembangan jaman dimana cara paling efektif untuk meningkatkan minat belajar dengan cara guru harus bisa mengembangkan ide atau kreatifitasnya dalam menggunakan media pembelajaran yang lebih menarik. Maka dari itu penulis membuat terobosan dengan merancang aplikasi pembelajaran dengan android agar lebih mudah dipahami oleh anak-anak. Untuk pengembangan aplikasinya penulis menggunakan metode *System Development Life Cycle* yang dapat membantu dalam pengembangan aplikasi android tersebut.

Tahapan metode adalah perencanaan sistem, analisis sistem, perancangan sistem, implementasi, uji coba dan maintenance atau pemeliharaan. Hasil akhir dari aplikasi ini adalah membuat aplikasi belajar cepat pengenalan kosakata bahasa inggris bagi anak usia dini berbasis android yang nantinya dapat mempermudah anak-anak dalam memahami dan melafalkan dengan baik dan mudah.

Penelitian Santosa et al (2021) mengembangkan sebuah aplikasi pembelajaran berbasis android yang berjudul “Tren pemanfaatan teknologi pembelajaran dalam pengajaran kosakata bahasa inggris pada sekolah dasar”. Keterbatasan dari tinjauan ini adalah penggunaan teknologi pembelajaran yang digabungkan dengan pengajaran kosa kata untuk sekolah dasar. Metode Tinjauan Literatur sistematis digunakan dalam penelitian ini dengan menerapkan pendekatan empat fase yang serupa yaitu pencarian, filter, dan kelayakan, dan kueri/pencarian, filter, kelayakan, dan inklusi. Berdasarkan pendekatan empat tahap, 31 artikel dari 996 telah disaring yang memenuhi persyaratan penggunaan teknologi pembelajaran dalam pembelajaran Kosakata dari sekolah dasar. hasil tinjauan penelitian sebelumnya menyatakan bahwa bentuk teknologi yang diterapkan dalam pembelajaran kosakata adalah perangkat pembelajaran yang meliputi video game, AR, tablet, dan aplikasi aplikasi yang dibentuk oleh peneliti dengan tujuan untuk meningkatkan pembelajaran kosakata. Selanjutnya dalam bentuk sumber belajar, YouTube merupakan sumber belajar yang digunakan dalam pembelajaran kosakata bahasa asing di sekolah dasar. Hal menarik lainnya, tipe lingkungan belajar yang menerapkan teknologi sangat jarang diterapkan. Satu studi membahas bagaimana kahoot dan duolingo diterapkan sebagai lingkungan belajar dalam pembelajaran kosa kata. Dan tipe yang terakhir yaitu teknologi dalam metode pembelajaran tidak

ada dalam proses pembelajaran kosakata bahasa asing. Hasil penelitian ini dapat memberikan pandangan baru dalam penggunaan audiovisual pada pembelajaran kosakata Bahasa asing disekolah dasar.

Penelitian Hamidah et al (2020) membuat penelitian yang berjudul “Pemanfaatan *e-Dictionary* Berbasis Android pada *Vocabulary for Beginner* dalam Meningkatkan Kualitas Pengajaran di Mosaic English Training Center Kampoeng Inggris Pare”. Pada sistem pembelajaran kata dalam Bahasa Inggris dan makna dalam Bahasa Indonesia terdapat banyak jumlah kata dan frase yang dipelajari oleh masing-masing individu. Oleh karena itu, banyak ahli Bahasa Inggris yang mengemukakan deskripsi rinci tentang bentuk *Vocabulary* dalam Bahasa Inggris baik English-Indonesia dan Indonesia- Inggris, yang setidaknya dapat dengan mudah dipahami oleh lingkungan pengguna Bahasa Inggris. Untuk mempelajari *vocabulary* tentunya ada level yang mempelajari beberapa macam *vocabulary*, salah satu pembelajar *vocabulary* adalah beginners. Hal ini, para instruktur dan learners di *Mosaic English Training Center* perlu diperkenalkan pada *e-dictionary* untuk mempelajari *vocabulary*. Para instruktur sering kali menggunakan kamus atau *dictionary* manual sehingga diperlukan aplikasi yang lebih modern yakni *e-vocabulary dictionary* dalam belajar vocab di *Mosaic English Trainng Center*. Tujuan diadakan pengabdian masyarakat ini adalah untuk meningkatkan kemampuan para pengajar dan kualitas pengajaran sehingga dapat diaplikasikan pada proses belajar mengajar dan memudahkan pembelajar atau learners dalam mempelajari *vocabulary* di *Mosaic English English Training Center*.

2.2. Pengertian *Vocabulary*

Kosakata atau perbendaharaan (*vocabulary*) kata adalah himpunan kata yang dimiliki oleh seseorang atau entitas lain, atau merupakan bagian dari suatu bahasa tertentu. Kosakata seseorang didefinisikan sebagai himpunan semua kata-kata yang dimengerti oleh orang tersebut atau semua kata-kata yang kemungkinan akan digunakan oleh orang tersebut untuk menyusun kalimat baru. Kekayaan kosakata seseorang secara umum dianggap merupakan gambaran dari kecerdasan atau tingkat pendidikannya. Oleh karena itu, banyak ujian standar, seperti SAT, yang memberikan pertanyaan yang menguji kosakata. Dengan demikian, jika suatu kata sudah masuk dalam kosakata yang dikuasai, ketika mendengar atau melihat tulisan kata tersebut sudah mengerti maknanya. Sedangkan kata-kata yang belum dikuasai atau belum paham maknanya belum disebut sebagai *vocabulary*.

Vocabulary merupakan awal untuk mempermudah kita memahami dan mengaplikasikan Bahasa Inggris dalam dunia edukasi maupun komunikasi. Anak yang memasuki masa emas mereka sangat cocok untuk diperkenalkan *vocabulary* agar membantu mereka nantinya dalam menumbuhkan minat belajar Bahasa Inggris. Dampak lain adalah anak bisa menguasai skill atau berbagai ketrampilan berbahasa Inggris serta komponen pendukungnya (Hamidah et al., 2020).

2.3. Media Pembelajaran

Media Pembelajaran adalah alat bantu pembelajaran, yaitu segala sesuatu yang dapat dipergunakan untuk merangsang pikiran, perasaan, perhatian dan

kemampuan atau ketrampilan peserta didik sehingga dapat mendorong terjadinya proses belajar. Batasan ini masih cukup luas dan mendalam mencakup pengertian sumber, lingkungan, manusia dan metode yang digunakan untuk tujuan pembelajaran. Media pembelajaran adalah sarana pendidikan yang dapat digunakan sebagai perantara dalam proses pembelajaran untuk mempertinggi efektifitas dan efisiensi dalam mencapai tujuan pengajaran. Secara umum dapat dikatakan bahwa substansi dari media pembelajaran adalah (Rozi & Kristari, 2020):

1. Bentuk saluran, yang digunakan untuk menyampaikan pesan, informasi atau bahan pelajaran kepada penerima pesan atau pembelajar
2. Berbagai jenis komponen dalam lingkungan pembelajar yang dapat merangsang pembelajar untuk belajar
3. Bentuk alat fisik yang dapat menyajikan pesan serta merangsang pembelajar untuk belajar
4. Bentuk-bentuk komunikasi yang dapat merangsang pembelajar untuk belajar, baik cetak maupun audio, visual, dan audio-visual

2.4. Materi Pembelajaran Bahasa Inggris Kelas V Sekolah Dasar

Dalam kurikulum 2013 Materi bahasa Inggris untuk kelas V Sekolah Dasar sudah tidak menggunakan KTSP namun menggunakan Muatan Lokal, menurut RPP di SDN 5 Bumi Waras Materi yang dipelajari adalah sebagai berikut :

- a) *Fruit and Vegetables*

Merupakan materi yang memperkenalkan Buah dan Sayuran dalam bahasa Inggris, dan bertujuan agar siswa mengerti perbendaharaan kata dan mengucapkannya dengan benar.

b) Sports

Merupakan materi yang memperkenalkan macam-macam olahraga dalam bahasa Inggris, dan bertujuan agar siswa mengerti perbendaharaan kata dan mengucapkannya dengan benar.

c) Daily Activities

Merupakan materi yang memperkenalkan kegiatan sehari-hari dalam bahasa Inggris, dan bertujuan agar siswa mengerti perbendaharaan kata dan mengucapkannya dengan benar.

d) Vehicles.

Merupakan materi yang memperkenalkan kendaraan dalam bahasa Inggris, dan bertujuan agar siswa mengerti perbendaharaan kata dan mengucapkannya dengan benar.

e) Clothes

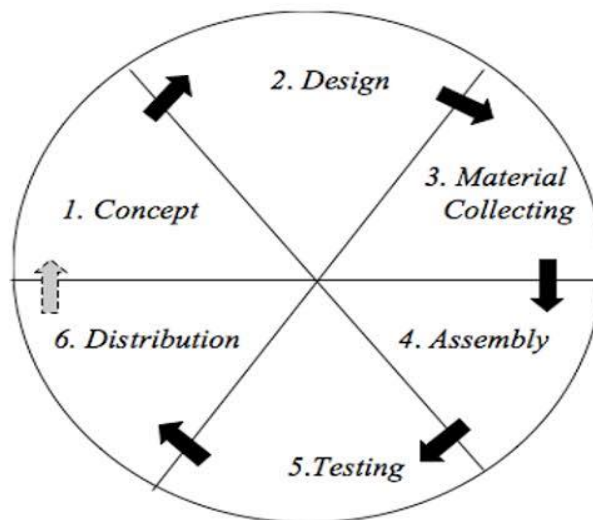
Merupakan materi yang memperkenalkan pakaian dalam bahasa Inggris, dan bertujuan agar siswa mengerti perbendaharaan kata dan mengucapkannya dengan benar.

f) Human Body

Merupakan materi yang memperkenalkan bagian dalam tubuh manusia dalam bahasa Inggris, dan bertujuan agar siswa mengerti perbendaharaan kata dan mengucapkannya dengan benar.

2.5. Metode Pengembangan *Multimedia Development Life Cycle* (MDLC)

Multimedia Development Life Cycle (MDLC) merupakan salah satu metodologi pengembangan perangkat lunak. Menurut Luther dalam Binanto, metodologi pengembangan multimedia sendiri terdapat enam tahap, yaitu *concept* (pengonsepan), *design* (pendesainan), *material collecting* (pengumpulan materi), *assembly* (pembuatan), *testing* (pengujian), dan *distribution* (pendistribusian). Keenam tahap ini tidak harus berurutan dalam praktiknya, tahap-tahap tersebut dapat saling bertukar posisi. Berikut ini adalah gambar tahapan pengembangan MDLC yaitu :



Gambar 1. 1. Pengembangan MDLC

Berikut ini adalah penjelasan dari enam tahap di atas :

1. *Concept*

Tahap *concept* (pengonsepan) adalah tahap untuk menentukan tujuan dan siapa pengguna program (identifikasi audiens). Tujuan dan pengguna akhir program berpengaruh pada nuansa multimedia sebagai pencerminan dari identitas organisasi yang menginginkan informasi sampai pada pengguna akhir. Karakteristik pengguna termasuk kemampuan pengguna juga perlu dipertimbangkan karena dapat mempengaruhi pembuatan desain.

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 adalah tahap penggumpulan bahan yang sesuai dengan kebutuhan yang dikerjakan. Bahan-bahan tersebut antara lain gambar clip art, foto, animasi, video, audio, dan lain-lain yang dapat diperoleh secara gratis atau dengan pemesanan kepada pihak lain sesuai dengan rancangannya. Tahap ini dapat dikerjakan secara paralel dengan tahap *Assembly*. Namun, pada beberapa kasus, tahap *Material Collecting* dan tahap *Assembly* akan dikerjakan secara linear dan tidak parallel

4. *Assembly*

Tahap *Assembly* adalah tahap pembuatan semua objek atau bahan multimedia. Pembuatan aplikasi didasarkan pada tahap design, seperti storyboard, bagan alir (flowchart), dan/atau stuktur navigasi.

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.

2.6. Pemrograman Android

2.6.1. Sejarah Android

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 (Rozi & Kristari, 2020)

Sejarah android pada mulanya berasal dari perusahaan bernama Android, Inc. didirikan tempatnya di Palo Alto, California, pada Oktober tahun 2003 oleh Andy Rubin (pendiri Danger), Rich Miner seorang pendiri Wildfire Communications, Inc., Nick Sears seorang mantan VP T-Mobile, dan Chris White seorang kepala desain dan pengembangan antarmuka WebTV untuk mengembangkan sebuah "perangkat seluler pintar yang lebih sadar tentang lokasi dan preferensi penggunanya".

ujian awal dari perkembangan tersebut pada mulanya diperuntukkan bagi kamera digital, namun disadari bahwa pasar dari kamera digital tidak besar potensinya, dan pengembangan Android lalu dialihkan pada pasar telepon pintar atau smartphone untuk menyaingi Symbian serta Windows Mobile (iPhone Apple pada saat itu belum dirilis). Meskipun parapengembang Android tersebut merupakan pakar-pakar teknologi yang berpengalaman, Android Inc. dijalankan secara diam-diam dan hanya diungkapkan bahwa para pengembang tersebut sedang berusaha menciptakan sebuah perangkat lunak yang dapat diperuntukkan untuk telepon seluler. Masih pada tahun yang sama, Andy Rubin kehabisan uang. Steve Perlman adalah seorang teman

dekat Andy Rubin dan meminjaminya \$10.000 tunai serta menolak tawaran saham di perusahaan.

Google mengakuisisi perusahaan Android Inc. pada tanggal 17 Agustus 2005 dan menjadikannya sebagai anak perusahaan yang dimiliki oleh Google. Pendiri Android Inc. yaitu Rubin, Miner, serta White tetap bekerja pada perusahaan tersebut setelah diakuisisi oleh Google. Di Google, tim yang dipimpin oleh Andy Rubin mulai untuk mengembangkan sebuah platform perangkat seluler dengan menggunakan kernel Linux.

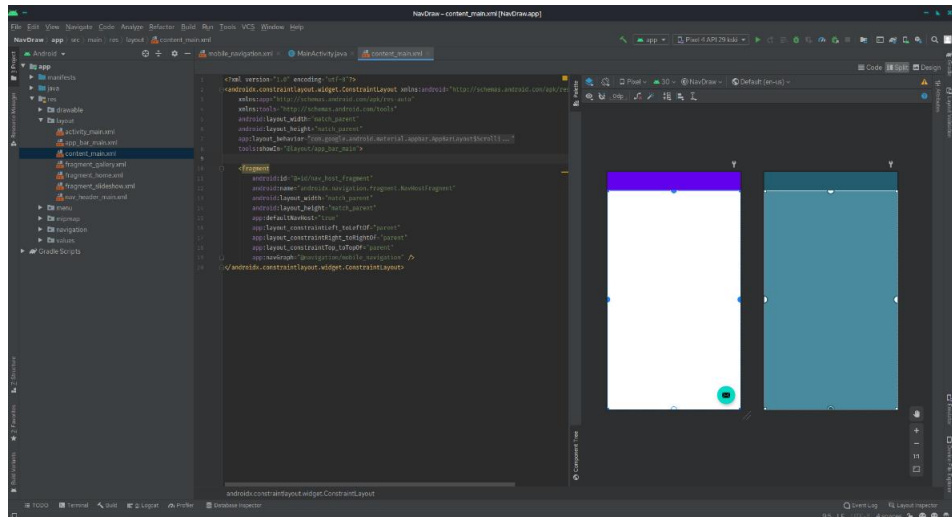
Sejak tahun 2008, Android mulai secara bertahap melakukan sejumlah pembaruan atau update untuk meningkatkan kinerja dari sistem operasi tersebut dengan menambahkan fitur baru, memperbaiki bug pada versi android yang sebelumnya. Setiap versi yang dirilis dinamakan secara alfabetis dengan berdasarkan nama sebuah makanan pencuci mulut, seperti cupcake, donut, dan sebagainya. Berikut beberapa versi dari Android, yaitu :

1. Android Versi 1.0 diberi nama *Apple Pie*
2. Android Versi 1.1 diberi nama *Banana Bread*.
3. Android Versi 1.5 diberi nama *Cupcake*.
4. Android Versi 1.6 diberi nama *Donut*.
5. Android Versi 2.0/ 2.1 diberi nama *Eclair*.
6. Android Versi 2.2 diberi nama *Froyo*.
7. Android Versi 2.3 diberi nama *Gingerbread*.
8. Android Versi 3.0 diberi nama *Honeycomb*
9. Android Versi 4.0 diberi nama *Ice Cream Sandwich*
10. Android Versi 4.1 diberi nama *Jelly Bean*.

11. Android Versi 4.4 diberi nama *Kitkat*.
12. Android Versi 5.0 diberi nama *Lollipop*.
13. Android Versi 6.0 diberi nama *Marshmallow*.
14. Android Versi 7.0 diberi nama *Nougat*.
15. Android Versi 8.0 diberi nama *Oreo*.
16. Android Versi 9.0 diberi nama *Pie*
17. Android Versi 10
18. Android Versi 11
19. Android Versi 12

2.6.2. Android Studio

Android Studio merupakan *Integrated Development Environment* (IDE) resmi untuk pengembangan aplikasi berbasis Android. Jika sebelumnya banyak pengembang yang mempergunakan IDE seperti *Eclipse* yang dilengkapi dengan *Android Development Tools* (ADT), namun IDE yang mendukung update terbaru dari teknologi Android adalah Android Studio. Sama seperti *Eclipse* ADT, Android Studio merupakan aplikasi *open source* yang dapat diunduh dan dipergunakan secara bebas untuk mengembangkan aplikasi berbasis android.



Gambar 2. 1. Layout Android Studio

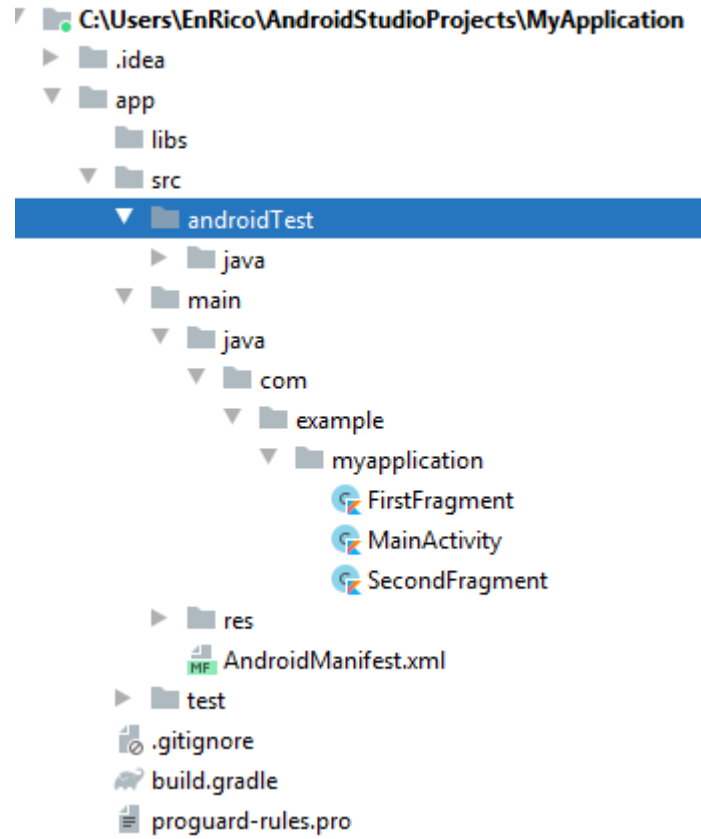
Berikut ini adalah beberapa fitur yang perlu diketahui dalam mengembangkan aplikasi android menggunakan android studio:

a. Layout

Fitur yang dimiliki Android Studio hampir sama dengan *Eclipse* ADT, di mana terdapat 2 *layout*, yaitu *text* dan *design*. Yang membedakan adalah pada Android Studio terdapat fasilitas *live preview*, sehingga tidak perlu bergantian membuka *mode design* dan *text* untuk melihat hasil desain yang dibuat. Pada gambar 2.4 ditampilkan *layout* Android Studio dan dapat dilihat bahwa terdapat *live preview* yang akan menampilkan secara langsung kode desain dalam mode *text*.

b. Struktur Folder

Terdapat folder *.idea*, *app*, *gradle* dan file – file yang ada di *root folder*. Folder *app/* adalah folder utama pada aplikasi android, seperti dapat dilihat pada gambar 2.2.

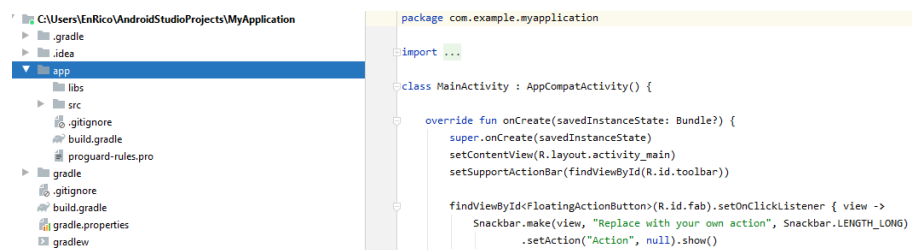


Gambar 2. 2. Struktur folder *project android*

Pada *folder build* seperti *gen/* di *eclipse*, isinya file-file hasil *generate* dari IDE. Kemudian *folder libs/* sama seperti di *eclipse*. Jika akan menggunakan *library* dalam bentuk *.jar* maka tempatkan di *folder libs*. Selanjutnya *folder src/* adalah *folder* dimana tersimpan *source java* dan *layout* dalam bentuk XML, secara *default* android studio akan *generate* dua *folder* yaitu, *androidtest* dan *main*. *Folder androidtest* adalah folder khusus *UnitTest*.

c. *Graddle*

Fitur di Android Studio adalah *graddle*, sebuah *featured build automation*. File *Graddle* berisi *library* yang digunakan, versi aplikasi, *signed key properties*, lokasi *repository* dan lain sebagainya. File yang akan sering diubah adalah file *build.graddle* yang berada dalam *folder app/* seperti dapat dilihat pada gambar 2.3



Gambar 2. 3. File *Build Graddle*

2.7. *Text to Speech* (TTS)

2.7.1. *Pengertian Text to Speech*

Text-to-speech (TTS) adalah jenis teknologi bantu yang membaca teks digital dengan keras. Dengan mengklik tombol atau sentuhan jari, TTS dapat mengambil kata-kata di komputer atau perangkat digital lainnya dan mengubahnya menjadi audio. TTS sangat membantu untuk anak-anak dan orang dewasa yang kesulitan membaca. Tetapi juga dapat membantu dengan menulis dan mengedit, dan bahkan dengan fokus.

TTS bekerja dengan hampir semua perangkat digital pribadi, termasuk komputer, ponsel cerdas, dan tablet.. Bahkan halaman web online dapat dibaca dengan keras.

Berdasarkan definisi di atas maka dapat disimpulkan bahwa TTS dapat didefinisikan sebagai “Sistem yang dapat mengubah suatu teks menjadi ucapan secara otomatis dengan cara fonetisasi (penyusunan fonem-fonem untuk membentuk ucapan)”. Sebuah sistem TTS dapat mengucapkan kata apapun, dan kosa katanya tidak terbatas.

2.7.2. Teknologi *Text to Speech* (TTS)

Pada prinsipnya, TTS terdiri dari dua sub sistem, yaitu :

- a) Bagian Konverter Teks ke Fonem (*Text to Phoneme*).
- b) Bagian Konverter Fonem to Ucapan (*Phoneme to Speech*)

Bagian Konverter Teks ke Fonem berfungsi untuk mengubah kalimat masukan dalam suatu bahasa tertentu yang berbentuk teks menjadi rangkaian kode-kode bunyi yang biasanya direpresentasikan dengan kode fonem, durasi serta pitchnya. Bagian ini bersifat sangat *language dependant*. Untuk suatu bahasa baru, bagian ini harus dikembangkan secara lengkap khusus untuk bahasa tersebut

Bagian Konverter Fonem ke Ucapan akan menerima masukan berupa kode-kode fonem serta pitch dan durasi yang dihasilkan oleh bagian sebelumnya. Berdasarkan kode-kode tersebut, bagian Konverter Fonem ke ucapan akan menghasilkan bunyi atau sinyal ucapan yang sesuai dengan kalimat yang ingin diucapkan. Ada beberapa alternatif teknik yang dapat digunakan untuk implementasi bagian ini. Dua teknik yang banyak digunakan adalah *formant synthesizer*, serta *diphone concatenation*.

Formant synthesizer bekerja berdasarkan suatu model matematis yang akan melakukan komputasi untuk menghasilkan sinyal ucapan yang

diinginkan. *Synthesizer* jenis ini telah lama digunakan pada berbagai aplikasi. Walaupun dapat menghasilkan ucapan dengan tingkat kemudahan interpretasi yang baik, *synthesizer* ini tidak dapat menghasilkan ucapan dengan tingkat kealamian yang tinggi.

Synthesizer yang menggunakan teknik *diphone concatenation* bekerja dengan cara menggabung-gabungkan segmen-segmen bunyi yang telah direkam sebelumnya. Setiap segmen berupa diphone (gabungan dua buah fonem). *Synthesizer* jenis ini dapat menghasilkan bunyi ucapan dengan tingkat kealamian (*naturalness*) yang tinggi.

Struktur sistem seperti di atas pada prinsipnya merupakan konfigurasi tipikal yang digunakan pada berbagai sistem *Text to Speech* berbagai bahasa. Namun demikian, pada setiap sub-sistem terdapat sifat-sifat serta proses-proses yang sangat spesifik dan sangat tergantung dari bahasanya.

Konversi dari teks ke fonem sangat dipengaruhi oleh aturan-aturan yang berlaku dalam suatu bahasa. Pada prinsipnya proses ini melakukan konversi dari simbol-simbol tekstual menjadi simbol-simbol fonetik yang merepresentasikan unit bunyi terkecil dalam suatu bahasa. Setiap bahasa memiliki aturan cara pembacaan dan cara pengucapan teks yang sangat spesifik. Hal ini menyebabkan implementasi unit konverter teks ke fonem menjadi sangat spesifik terhadap suatu bahasa.

Untuk mendapatkan ucapan yang lebih alami, ucapan yang dihasilkan harus memiliki prosodi. Secara kuantisasi, prosodi adalah perubahan nilai pitch selama pengucapan kalimat dilakukan atau pitch sebagai fungsi waktu. Pada prakteknya, informasi pembentuk prosodi berupa data-data pitch serta

durasi pengucapannya untuk setiap fonem yang dibangkitkan. Nilai-nilai yang dihasilkan diperoleh dari suatu model prosodi. Prosodi bersifat sangat spesifik untuk setiap bahasa, sehingga model yang diperlukan untuk membangkitkan data-data prosodi menjadi sangat spesifik juga untuk suatu bahasa. Beberapa model umum prosodi pernah dikembangkan, tetapi untuk digunakan pada suatu bahasa masih perlu banyak penyesuaian yang harus dilakukan.

Konverter fonem ke ucapan berfungsi untuk membangkitkan sinyal ucapan berdasarkan kode-kode fonem yang dihasilkan dari proses sebelumnya.

Sub sistem ini harus memiliki pustaka setiap unit ucapan dari suatu bahasa. Pada sistem yang menggunakan teknik diphone concatenation, sistem harus didukung oleh suatu diphone database yang berisi rekaman segmen-segmen ucapan yang berupa diphone. Ucapan dalam suatu bahasa dibentuk dari satu set bunyi yang mungkin berbeda untuk setiap bahasa, oleh karena itu setiap bahasa harus dilengkapi dengan diphone database yang berbeda. (Voice processing, Gordon E Pelton)

2.7.3. Text to Speech pada Android

Teknologi Text To Speech (TTS) pada android telah ada sejak android versi 1.6 (API 4) yang disebut juga "speech synthesis". TTS memungkinkan perangkat android untuk "berbicara" dari teks bahasa yang berbeda. *Engine* TTS pada platform android akan mengubah kata menjadi suara dan suara yang dihasilkan dari *Engine* TTS dapat diatur dengan aksesn suara dari sejumlah bahasa, yaitu Inggris, Jerman, Itala dan Spanyol. Jika akses suara

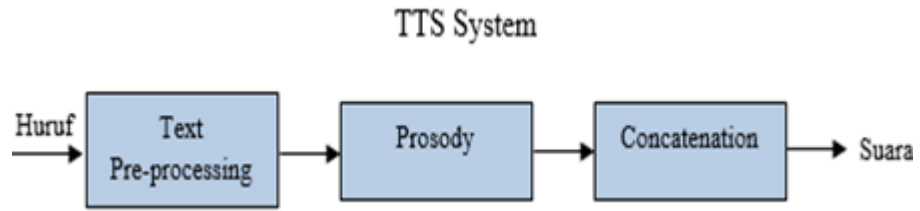
yang dipilih dari bahasa Inggris (US) maka pengucapan kata-kata tersebut diucapkan dengan aksentuasi bahasa Inggris (US). *Engine TTS* perlu untuk mengetahui terlebih dahulu bahasa apa yang dipergunakan. Sehingga misalkan terdapat kata 'Paris', akan berbeda ketika diucapkan dengan aksentuasi Inggris dan aksentuasi Perancis (Developer, 2019).

Saat ini teknologi TTS pada platform android telah mendukung banyak bahasa, seperti Bengali (Bangladesh), Kanton (Hong Kong), Cina (Cina), Cina (Taiwan) Denmark, Belanda, Inggris (Australia), Inggris (India), Bahasa Inggris (United Kingdom), Inggris (Amerika Serikat), Finlandia, Perancis, Jerman, Hindi, Hungaria, Indonesia, Italia, Jepang, Korea, Norwegia, Polandia, Portugis (Brasil), Rusia, Spanyol (Mexico), Spanyol (Spanyol), Spanyol (Amerika Serikat), Swedia, Thai, Turki, dan Vietnam.

Meskipun semua perangkat android telah mendukung fungsi Engine TTS, namun beberapa perangkat memiliki penyimpanan terbatas, sehingga mungkin tidak memiliki seluruh file sumber daya untuk dukungan seluruh bahasa. Jika pengguna ingin menambahkan kemampuan fitur bahasa tertentu, maka pengguna dapat mendownload file API yang spesifik sehingga dapat mengembangkan fitur TTS untuk bahasa tersebut.

2.7.4. Prinsip Kerja Sistem *Text to Speech* (TTS)

Text to Speech synthesis system terdiri dari 3 bagian, yaitu *text pre-processing*, pembangkitan *prosody* dan *concatenation*. Pada gambar 2.4 ditunjukkan diagram blok *text to speech synthesis system*.

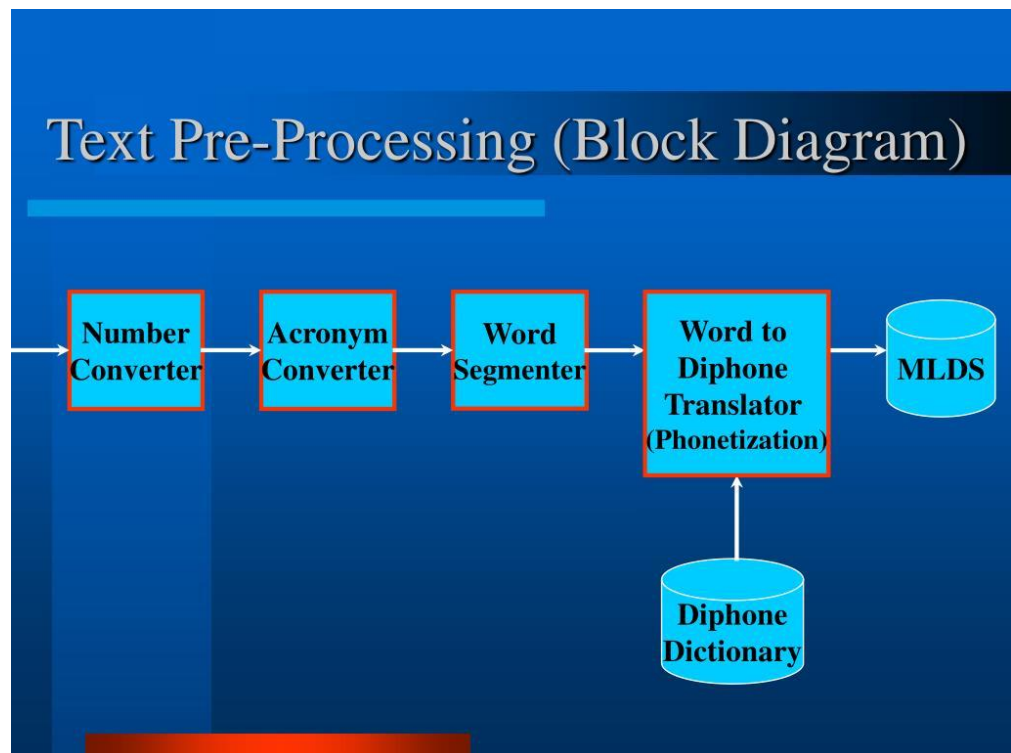


Gambar 2. 4. Diagram blok text to speech synthesis system

Berikut ini adalah penjelasan untuk masing-masing bagian di atas :

a. Text Pre-Processing

Yaitu pengkonversian dari input yang berupa teks menjadi diphone (gabungan dua buah fonem). Ketika input yang berupa teks, akronim (singkatan) ataupun angka maka bagian ini akan mengkonversikan menjadi diphone yang telah tersedia di databasediphone. Pada gambar 2.2 ditunjukkan diagram blok *text pre-processing*



Gambar 2. 5. Diagram blok *text pre-processing*

Dari blok diagram sistem dapat dijelaskan cara kerja sistem yaitu :

1) *Number Converter*

Jika input pada sistem berupa angka, maka system mengkonversikan angka ke dalam representasi diphone (gabungan dua buah fonem), misal 0.25 menjadi *point two five*.

2) *Acronym Converter*

Jika input pada sistem berupa kata singkatan dalam bahasa Inggris, maka sistem mengkonversika singkatan ke dalam representasi diphone (gabungan dua buah fonem), misalnya *Mr.* menjadi Mister.

3) *Word Segmenter*

Jika input pada sistem berupa kata atau kalimat maka sistem mengkonversikan kata atau kalimat ke dalam representasi diphone (gabungan dua buah fonem)

4) *Diphone Dictionary*

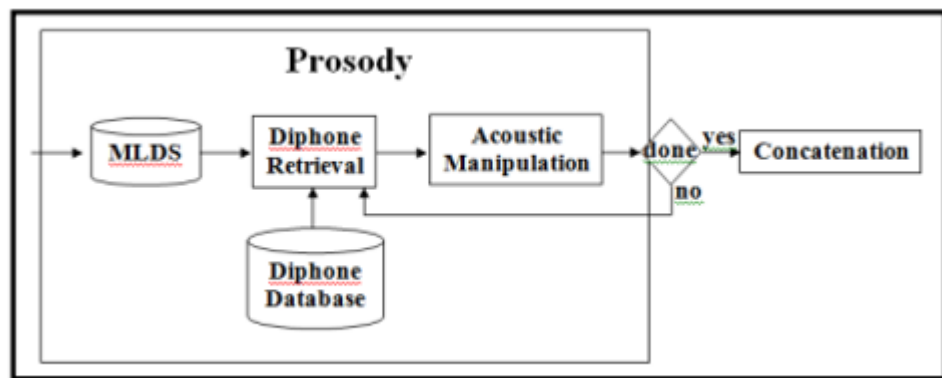
Merupakan database yang berupa kumpulan dari diphone – diphone. Pembuatan diphone dilakukan dengan melakukan pelabelan pada sinyal wicara.

5) *MLDS (Multi Level Data Structure)*

Terdiri dari semua data yang diperlukan untuk sub sistem berikutnya, dalam hal ini adalah proses prosody. MLDS terdiri dari representasi diphone-diphone hasil pengkonversian inputan.

b. *Prosody*

Yaitu untuk mendapatkan ucapan yang lebih alami, ucapan yang dihasilkan harus memiliki intonasi (*prosody*). Secara kuantisasi, prosodi adalah perubahan nilai pitch (frekuensi dasar) selama pengucapan kalimat dilakukan atau pitch sebagai fungsi waktu. Prosodi bersifat sangat spesifik untuk setiap bahasa, sehingga model yang diperlukan untuk membangkitkan data-data prosodi menjadi sangat spesifik juga untuk suatu bahasa. Pada gambar 2.6 ditunjukkan diagram blok *prosody*.



Gambar 2. 6. Diagram Block Prosody

Berikut adalah penjelasan untuk proses yang terjadi:

1) *Diphone Retrieval*

Didalamnya terdapat tiga tahapan yang terjadi, yaitu database hasil perekaman diphone, menyimpan bentuk gelombang diphone dan Prosodic parameter dalam variable

2) *Accoustic Manipulation*

Di dalamnya terdapat proses pengenalan file-file gelombang . WAV(*load, play, write*), *vast array* dari peralatan *signal processing, builtin function, ease debugging*, dan *GUI-capable*.

3) *Concatenation*

Yaitu penggabungan segmen-segmen bunyi yang telah direkam sebelumnya. Setiap segmen berupa diphone (gabungan dua buah fonem). Pada perekaman suara dilakukan beberapa kali agar mendapatkan hasil yang akurat.


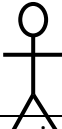

2.8. Perancangan Sistem UML (*Unified Modeling Language*)


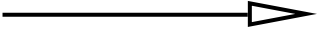

UML (*Unified Modeling Language*) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. Rosa A.S dan M. Shalahuddin (2018)

1. *Use Case Diagram*

Use case diagram atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat.

Table 2. 1. Simbol *Use Case Diagram*

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 frase nama <i>use case</i> .
Aktor/ <i>actor</i> 	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang.
Asosiasi/ <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




Simbol	Deskripsi
Ekstensi/extend <<extend>> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek biasanya <i>use case</i> tambahan memiliki nama depan
Generalisasi/generalization 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
Menggunakan/Include/uses <<include>> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini


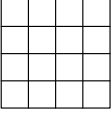


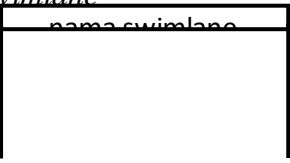
(Sumber: Rosa dan Shalahuddin, 2018)

2. Activity Diagram

Activity diagram atau Diagram aktivitas menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Table 2. 2. Simbol Activity Diagram

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan/decision 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu

Simbol	Deskripsi
Penggabungan/ <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
Tabel 	Suatu file komputer dari mana data bisa dibaca atau direkam selama kejadian bisnis
Dokumen 	Menunjukkan dokumen sumber atau laporan
Status akhir 	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

Sumber: (Rosa & Shalahudin, 2018)

2.9. Pengujian *Black Box*

Pendekatan pengujian *Black-Box* adalah metode pengujian di mana data tes berasal dari persyaratan fungsional yang ditentukan tanpa memperhatikan struktur program akhir. Karena hanya fungsi dari modul perangkat lunak yang menjadi perhatian, pengujian *Black-Box* juga mengacu pada uji fungsional, metode pengujian menekankan pada menjalankan fungsi dan pemeriksaan inputan dan data output (Howden, 2017).

Pengujian *black-box* berusaha menemukan kesalahan dalam kategori sebagai berikut :

- a. Fungsi – fungsi yang tidak benar atau hilang,

- b. Kesalahan interface
- c. Kesalahan dalam struktur data atau akses eksternal
- d. Kesalahan kinerja
- e. Inisialisasi dan kesalahan terminasi

Pada *black box testing* terdapat jenis teknik design tes yang dapat dipilih berdasarkan pada tipe testing yang akan digunakan, diantaranya sebagai berikut:

1. *Equivalence Class Partitioning*
2. *Boundary Value Analysis*
3. *State Transitions Testing*
4. *Cause-Effect Graphing*

2.10. Usability

Usability, sejauh mana suatu produk atau sistem dapat digunakan oleh pengguna tertentu untuk mencapai tujuan secara efektif, efisiensi, dan kepuasan tertentu dalam konteks penggunaan. Ciri ini terbagi menjadi beberapa ciri yaitu.

- a. Ketepatan pengenalan, sejauh mana pengguna dapat mengetahui apakah sistem atau produk memenuhi kebutuhan mereka.
- b. Learnability, sejauh mana suatu produk atau sistem dapat digunakan oleh pengguna untuk mencapai tujuan tertentu dengan belajar menggunakan sistem atau produk tersebut secara efisien, efektif, bebas dari risiko dan kepuasan dalam konteks tertentu.
- c. Operabilitas, sejauh mana produk atau sistem mudah dioperasikan dan dikendalikan.

- d. Perlindungan kesalahan pengguna, sejauh mana produk atau sistem melindungi pengguna dari membuat kesalahan.
- e. Estetika antarmuka pengguna, sejauh mana antarmuka pengguna suatu produk atau sistem memungkinkan interaksi yang ramah pengguna dan memuaskan.
- f. Aksesibilitas, sejauh mana suatu produk atau sistem dapat digunakan oleh semua kelompok untuk mencapai tujuan tertentu sesuai dengan konteks penggunaannya.

2.11. Skala Likert

Skala likert yang digunakan adalah skala Likert, skala yang didasarkan pada penjumlahan sikap responden dalam merespon pernyataan berkaitan indikator-indikator suatu konsep atau variable yang sedang diukur (Sanusi, 2012). Skala Likert umumnya menggunakan lima titik dengan label netral pada posisi tengah (ketiga). Skala Likert dapat dilihat pada Tabel 2.3:

Table 2. 3. Skala Likert

Jawaban	Skor
Sangat Setuju	5
Setuju	4
Netral	3
Tidak Setuju	2
Sangat Tidak Setuju	1

Hasil penilaian responden akan dihitung *persentase* kelayakannya dengan menggunakan perhitungan, dapat dilihat dibawah ini:

$$\text{Persentase} = \frac{\text{Skor Aktual (f)}}{\text{Skor Ideal (n)}} \times 100\%$$

Persentase kelayakan yang diperoleh kemudian dibandingkan dengan Tabel konversi yang berpedoman pada acuan konversi nilai, dapat dilihat pada Tabel 2.4.

Table 2. 4. Skala Konversi Nilai

<i>Persentase Pencapaian (%)</i>	<i>Interpretasi</i>
$90 \leq x$	<i>Sangat Baik</i>
$80 \leq x < 90$	Baik
$70 \leq x < 80$	Cukup
$60 \leq x < 70$	Kurang
$X < 60$	Sangat Kurang

Keterangan:

x = persentase hasil pengujian.