BAB 2
TINJAUAN PUSTAKA

2.1 Peribahasa Indonesia

Peribahasa, menurut sebagian besar pakar dan sastra, adalah kalimat-kalimat pendek berkias yang tetap bentuknya. Ungkapan-ungkapan ini oleh masyarakat dimanfaatkan untuk berbagai tujuan, seperti menyindir, menyerang lawan bicara, menasehati, atau sebagai pegangan dan tuntutan dalam menghadapi berbagai persoalan hidup sehari-hari.

Peribahasa adalah kelompok kata atau kalimat yang tetap susunannya dan biasanya mengisahkan maksud tertentu. Dalam peribahasa termasuk juga bidal, ungkapan, perumpamaan, atau kalimat-kalimat ringkas, padat yang berisi perbandingan, nasehat, perinsip hidup, aturan tingkah laku (*KBBI*, 1990: 671). Peribahasa mengandung nilai luhur berisi nasehat, penggambaran kehidupan, dan petuah yang bermanfaat bagi hidup dan kehidupan[3].

Peribahasa mempunyai nilai kesastraan karena cara pengungkapannya yang istimewa, berbentuk singkat tapi padat, rumusannya konkret tetapi mengandung makna umum, kata-kata yang digunakan mengandung pengertian tertentu yang bernilai rasa. Peribahasa merupakan rumusan dari kebijaksanaan masyarakat yang menunjukan adanya sikap *waspada lan eling* yang berkaitan erat dengan moral dan kebajikan hidup yang sangat berguna untuk menghadapi hidup dan kehidupan, selain itu juga menjadi milik rakyat yang bisa menjadi sarana pengungkap batin dan penanda nilai luhur budaya suatu masyarakat [8].

2.1.1 Sejarah Peribahasa Indonesia

Peribahasa Indonesia termasuk juga ke dalam peribahasa Melayu. Peribahasa bermula sebelum orang-orang Melayu pandai menulis dan membaca. Peribahasa dikatakan berasal dari orang Minangkabau dan ada juga yang mengatakan peribahasa berasal dari orang Riau. Sebelum orang-orang Melayu pandai membaca dan menulis mereka telah dididik melalui peribahasa [19].

Menurut Mohd Gani Ahmad, ribuan peribahasa yang terdapat pada saat ini merupakan hasil cipta oleh ahli fikir, cendekiawan, para pemuka bangsa atau pujangga sebuah masyarakat. Kemungkinan peribahasa di cipta oleh masyarakat biasa sangat tipis, jikapun ada mungkin secara kebetulan saja. Seseorang yang mencipta peribahasa mestilah seorang yang dihormati karena kedudukannya dalam masyarakat atau juga karena usianya [19].

Pada mulanya, percakapan hanya pertuturan kata biasa yang kurang indah, kemudian berubah dari masa ke masa yang akhirnya tercipta peribahasa yang dikenal sekarang ini. Melalui perkembangan sejarah sejak zaman dulu bahasa Melayu mulai berkembang dari bahasa yang rendah nilainya menjadi bahasa yang dapat memperkatakan soal-soal yang berhubungan dengan falsafah. Peribahasa yang merupakan aspek bahasa ikut juga berkembang [19].

2.1.2 Jenis Peribahasa

Peribahasa mempunyai berbagai jenis berdasarka sifat dan tujuannya tertentu diantaranya sebagai berikut:[3]

1. Pepatah, merupakan peribahasa yang mengandung nasihat atau ajaran dari orang tua. Peribahasa pepatah biasanya dipakai untuk atau diucapkan untuk mematahkan lawan bicara.
2. Perumpamaan, merupakan peribahasa yang berisi perbandingan, yang ditandai dengan penggunaan kata-kata yang bermakna membandingkan diantaranya bak, seperti, bagaikan, laksana, seumpama. Peribahasa perumpamaan biasanya dipakai untuk menggambarkan sesuatu yang abstrak dengan sesuatu yang konkret dalam kehidupan sehari-hari, seperti penggunaan kata konkret minyak, air, gajah, harimau, pahat, pohon, rumput sebagai perbandingan untuk hal-hal yang abstrak, seperti persahabatan, kehidupan, permasalahan, dan sejenisnya.
3. Pemeo, merupakan peribahasa yang dijadikan semboyan biasanya berfungsi untuk mengobarkan semangat dan suasan hidup.
4. Bidal, merupakan peribahasa yang mengandung sindiran maupun peringatan.
5. Tamsil, merupakan peribahasa yang seringkali memakai kata ibarat untuk membuat perbandingan antara suatu hal.

2.2 *Game*

Game adalah sekumpulan permainan yang melibatkan satu atau beberapa pemain yang memiliki sasaran, kekangan dan konsekuensi. Pada dasarnya *game* merupakan media hiburan untuk berbagai kalangan. *Game* berbeda dengan perangkat lunak lainnya. Pada *game* tujuan yang ingin diperoleh adalah untuk bersenag-senang dan menikmati permainan tersebut. *Game* berkembang dengan sangat pesat mulai dari *game* yang hanya dimainkan seorang diri hingga *game* yang dapat dimainkan oleh banyak orang diseluruh dunia secara bersamaan dengan menggunakan koneksi internet sebagai penghubung. Berkat kemajuan teknologi yang pesat kini *game* dapat dimainkan di *smartphone* dengan grafik yang sudah 3D. Belakangan ini *game mobile* banyak diminati, sehingga mendorong para pengembang *game* dalam menciptakan *game* yang bagus [11].

2.2.1 Sejarah *Game*

Game pertama kali dibuat pada tahun 1966 oleh Ralph Baer bersma timnya yang berjumlah 500 orang bersama dengan insinyur dan teknisi. *Game* yang pertamakali dibuat hanya bisa dimainkan pada komputer seharga US $40.000. Proses pembuatan dan pengembangan *game* tersebut di biayai oleh Pentagon.

Tahun 1972, muncul nama baru dalam dunia *game* yaitu Nolan Bushnel. Pada tanggal 27 juni 1992 dia mendirikan perusahaan Atari, dan membuat *game arcade* Pong yang cukup populer diwaktu itu. Setelah itu pada tahun 1988 *game* semakin terkenal dan banyak perusahhan yang mulai menggeluti dunia *game*. Nintendo, yang awalnya hanya memproduksi mesin *photocopy*, beralih memproduksi *game*. Produsen pembuat *game* di dominisasi oleh prusahan Jepang. Saat ini *game* sudah sangat berbeda dengan semakin majunya teknologi. Tampilan *game* sudah 3D dan terlihat nyata dan didukung oleh efek animasi serta audio yang semakin bagus.

2.2.2 Jenis *Game*

Game mempunyai beberapa jenis sesuai dengan *platform* yang digunakannya, diantarnya sebagai berikut:[12]

1. *Arcade Game,* yaitu *game* dengan mesin khusus yang di desain sesuai jenis *game*-nya serta memiliki hardware yang mendukung seperti pistol, kursi khusus, stir mobil, dan lain sebagainya.
2. *PC Game,* yaitu *game* yang dimainkan di PC (*Personal Computer*) yang memiliki antar muka yang baik dan dengan penggunaan hardware yang semakin tinggi spesifikasinya maka *game* pun akan semakin bagus.
3. *Console Game,* yaitu *game* yang dimainkan menggunakan mesin khusus tertentu dan tampilan visualnya menggunakan TV atau monitor. Console *game* diantrannya seperti, Playstation 3, Playstation 4, XBOX One, Nintendo Switch, dan lain sebagainya.
4. *Online Game,* yaitu *game* yang dimainkan dengan menggunakan koneksi internet atau LAN sehingga semua pemain dapat bermain bersama-sama.
5. *Mobile Game,* yaitu *game* yang dimainkan pada perangkat *mobile phone* atau *smartphone.*

2.2.3 *Genre Game*

Dalam perkembangannya, g*ame* dibagi menjadi beberapa genre sesuai dengan cara bermainnya, diantaranya: [12]

1. *Role Playing Game* (RPG), yaitu *game* yang memerankan karakter perwakilan dari pemain, yang biasanya adalah tokoh utama cerita *game* tersebut. Karakter tersebut akan semakin berkembang dan berubah dengan parameter yang biasanya ditentukan dengan naiknya level, naiknya status, bertambahnya kekuatan karakter, dan lain sebagainya. Berikut ini adalah contoh *game* RPG dapat dilihat pada gambar 2.1.



Gambar 2. 1 Contoh *Game* RPG

2. *Real Time Strategy* (RTS), yaitu *game* yang biasanya membuat pemainnya tidak hanya menggerakan satu karakter tetapi banyak karakter bahkan membangun bangunan. *Game* strategi kebanyakan bertema perang. Berikut ini adalah contoh *game* RTS dapat dilihat pada gambar 2.2.



Gambar 2. 2 Contoh *Game* RTS

3. *Frist Person Shooter* (FPS), yaitu *game* yang mensimulasikan tembak-menembak, saling pukul, dan apapun itu yang sesuai dengan cerita dan tokohnya, bisanya *game* FPS bertemakan militer. Berikut ini adalah contoh *game* FPS dapat dilihat pada gambar 2.3.



Gambar 2. 3 Contoh *Game* FPS

4. *Sports Games* (*Game* Olahraga), sesuai namanya *game* jenis ini bertemakan olahraga dimana karakter yang ada didalamnya dibuat semirip mungkin dengan yang aslinya. *Game* berupa kompetisi antara dua pemain atau lebih. Saat ini sepak bola merupakan *game* olahraga yang paling terkenal. Berikut ini adalah contoh *game* olahraga dapat dilihat pada gambar 2.4.



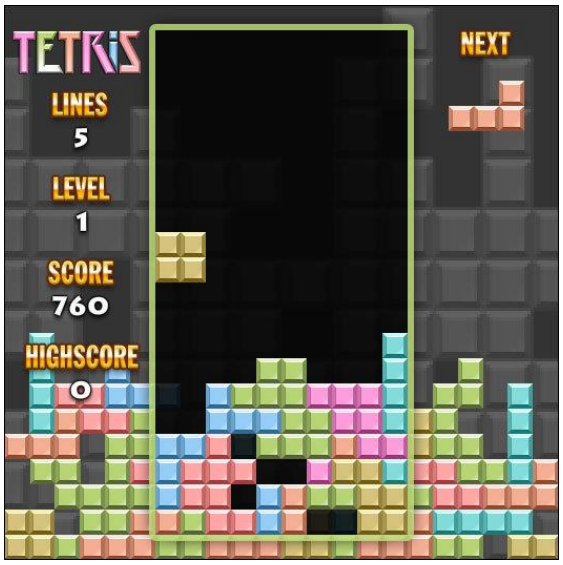
Gambar 2. 4 Contoh *Game* Olahraga

5. *Simulation Game* (*Game* Simulasi), yaitu *game* yang menggambarkan dunia didalamnya semirip mungkin dengan dunia nyata serta memperhatikan dengan detail berbagai faktor. Dari mencari jodoh, pekerjaan, membangun rumah, gedung hingga kota dan lain sebagainya. Berikut ini adalah contoh *game* simulasi dapat dilihat pada gambar 2.5.



Gambar 2. 5 Contoh *Game* Simulasi

6. *Puzzle Games* (*Game* Puzzle), yaitu *game* yang memecahkan suatu masalah, teka-teki baik itu perhitungan, menyamakan warna, menyusun balok dan lain sebagainya. Berikut ini adalah contoh *game* puzzle dapat dilihat pada gambar 2.6.



Gambar 2. 6 Contoh *Game* Puzzle

7. *Fighting Games* (*Game* Pertarungan), *game* jenis ini memerlukan kecepatan refleks mata dan tangan serta inti dari *game* ini adalah penguasaan *skill* karakter yang dimainkan. Kemampuan pemain akan teruji jika bermain dengan pemain

lain. Berikut ini adalah contoh *game* pertarungan dapat dilihat pada gambar 2.7.



Gambar 2. 7 Contoh *Game* Petarungan

8. *Advanture Game* (*Game* petualangan), yaitu *game* yang membuat pemainnya memainkan sebuah tokoh dalam cerita *game* tersebut. Lebih menekankan pada alur cerita, memecahkan suatu permasalahan serta bercakap dengan karakter lain. Berikut ini adalah contoh *game* petualangan dapat dilihat pada gambar 2.8.



Gambar 2. 8 Contoh *Game* Petualangan

9. *Racing Game* (*Game* balapan), yaitu *game* balapan dimana pemain mengendalikan sebuah mobil dan berkompetisi dengan pemain lain atupun *bot*

untuk menjadi pemenang balapan. Berikut ini adalah contoh *game* balapan dapat dilihat pada gambar 2.9.



Gambar 2. 9 Contoh *Game* Balapan

10. *Rhythim Game* (*Game* musik), yaitu *game* yang memainkan suatu alat musik tertentu dengan lagu yang sudah tersedia didalamnya. Berikut ini adalah contoh *game* musik dapat dilihat pada gambar 2.10.



Gambar 2. 10 Contoh *Game* Musik

2.2.4 NPC (*Non-Player Character*)

NPC adalah karakter pada *game* yang sepenuhnya dikendalikan oleh komputer dan tidak bisa dimainkan oleh pemian. NPC memiliki peranan yang cukup penting dalam sebuah *game* untuk mendukung alur cerita misalnya sebagai penjual barang atau item kepada pemain, sebagai pemberi misi, dan sebagai penunjuk arah.

2.3 *Game* Edukasi

Game edukasi adalah *game* digital yang dirancang untuk pengayaan pendidikan (mendukung pengajaran dan pembelajaran), menggunakan teknologi multimedia interaktif. Perancangan yang baik haruslah memenuhi kriteria dari *educaion game* itu sendiri. Berikut ini adalah beberapa kriteria dari sebuah *education game* menurut Hurd dan Jenuings, yaitu : [13]

1. Nilai Keseluruhan (*Overall Value*)

   Nilai keseluruhan dari suatu game terpusat pada desain dan panjang durasi *game*. Aplikasi yang dibangun haruslah menarik dan interaktif, salah satunya dengan desain yang baik. Untuk penentuan panjang durasi dapat menggunakan fitur *timer*.

2. Dapat Digunakan (*Usability*)

   Mudah digunakan dan diakses adalah poin penting bagi pembuat *game*. Aplikasi ini merancang sistem dengan *interface* yang *user friendly* sehingga *user* dapat dengan mudah mengakses aplikasi.

3. Keakuratan (*Accuracy*)

   Keakuratan diartikan sebagai bagaimana kesuksesan model/gambaran sebuah *game* dapat dituangkan ke dalam percobaan atau perancangannya. Perancangan aplikasi harus sesuai dengan model *game* pada tahap perencanaan.

4. Kesesuaian (*Appropriateness*)

   Kesesuaian dapat diartikan bagaimana isi dan desain *game* dapat diadaptasikan terhadap keperluan *user* dengan baik. Dengan menyediakan menu dan fitur yang diperlukan oleh *user* untuk membantu pemahaman *user* dalam menggunakan aplikasi.

5. Relevan (*Relevance*)

   Relevan artinya dapat mengaplikasikan isi *game* ke targer *user*. Agar dapat relevan terhadap *user*, sistem harus membimbing mereka dalam pencapaian tujuan pembelajaran. Desain yang digunakan harus sesuai dengan tujuan *user*.

6. Objektivitas (*Objectives*)

Objektivitas menentukan tujuan *user* dan kriteria dari kesuksesan atau kegagalan. Dalam aplikasi ini objektivitas adalah usaha untuk mempelajari hasil dari permainan.

7. Umpan Balik (*Feedback*)

Untuk membantu pemahaman *user* bahwa permainan (*performance*) mereka sesuai dengan objek *game* atau tidak, *feedback* harus disediakan.

Adapun contoh dari *game* edukasi dapat dilihat pada gambar 2.11.



Gambar 2. 11 Contoh *Game* Edukasi

2.4 Android

Android adalah sistem operasi telepon seluler atau yang sekarang terkenal disebut dengan *smartphone*. Android menyediakan *flatform* terbuka atau *open source* bagi para pengembang untuk membuat dan berkreasi dalam pembuatan aplikasi mereka sendiri. [9].Pengguna Android pada bulan september 2017 sudah mencapai 73,52% di seluruh dunia dari data yang dirilis oleh *StatCounter* [5].

2.4.1 Sejarah Android

Google Inc. membeli Android Inc., perusahaan yang membuat perangkat lunak untuk telepon seluler. Untuk mengembangkan Android, dibentuk Open Handset Alliance, konsorsium dari perusahaan *hardware, software*, dan telekomunikasi. Pada saat perilisan pertama Android, 5 November 2007, Android bersama Open Handset Alliance menyatakan mendukung pengembangan standar terbuka pada

perangkat seluler. Terdapat dua jenis distributor sistem operasi Android. Pertama yang mendapat dukungan penuh dari Google atau *Google Mail Services* (GMS) dan kedua adalah yang benar-benar bebas distribusinya tanpa dukungan langsung Google yang dikenal sebagai *Open Handset Distribution* (OHD) [9].

2.4.2 Arsitektur Android

Arsitektur Android dapat dijelaskan dan digambarkan sebagai berikut [9]:

1. *Application* dan *Widgets*

   Application dan *Widgets* merupakan *layer* dimana kita berhubungan dengan aplikasi saja, termasuk *download* aplikasi kemudian proses instalasi dan aplikasi dapat dijalankan. Semua aplikasi Android menggunakan bahasa pemrograman Java.

2. *Application Frameworks*

   Android memberi kebebasan kepada pengembang untuk membangun aplikasi yang bagus dan inovatif serta memiliki akses penuh dalam mengakses API *Frameworks. Application Frameworks* ini merupakan *layer* dimana para pembuat aplikasi melakukan pengembangan atau pembuatan aplikasi, pada *layer* inilah aplikasi dapat dirancang dan dibuat. Terapat beberapa komponen yang termasuk di dalam *Application Frameworks* diantaranya *Views, Content Provider, Resource Manager, Notification Manager,* dan *Activity Manager.*

3. *Libraries*

   Libraries ini adalah *layer* dimana fitur-fitur Android berada, biasanya para pengembang aplikasi mengakses *libraries* untuk menjalankannya. Berjalan diatas karnel, *layer* ini meliputi berbagai *library* C/C++ inti seperti Libe dan SSL.

4. *Android Run Time*

   Layer yang membuat aplikasi Android dapat dijalankan dimana dalam prosesnya menggunakan Implementasi Linux. *Dalvik Virtual Machine* (DVM) merupakan mesin yang membentuk dasar kerangka aplikasi Android. *Android Run Time* dibagi menjadi dua bagian yaitu *Core Libraries* dan *Dalvik Virtual Machine.*

5. Linux *Kernel*

   Linux *kernel* merupakan *layer* inti dari sistem operasi Android itu sendiri. Berisi *file-file* sistem yang mengatur sistem *processing, memory, resource, drivers,* dan sistem-sistem operasi android lainnya.

2.4.3 Versi Android

Adapun versi Android yang pernah dirilis samapi saat ini adalah sebagai berikut [9]:

1. Android versi 1.1

   Google merilis Android versi 1.1 pada 9 maret 2009. Versi ini delengkapi dengan pembaruan estetis pada aplikasi, jam alarm, pencarian suara, pengiriman pesan dengan Gmail, dan pemberitahuan email.

2. Android versi 1.5 (*Cupcake*)

   Pada 30 april 2009 google merilis Android versi 1.5, menggunakan kernel Linux 2.6.27. versi ini merupakan versi pertama yang menggunakan nama kode berdasarkan nama makanan pencuci mulut “Cupcake”, nama yang kemudian digunakan untuk semua versi selanjutnya. Terdapat beberapa pembaharuan dan penambahan fitur diantaranya adalah kemampuan merekam dan menonton video dengan modus kamera, mengunggah video ke Youtube dan gambar ke Picasa langsung dari perangkat, dukungan *Bluetooth* A2DP, kemampuan terhubung secara otomatis ke *headset Bluetooth*, animasi layar, dukungan widget, *keyboard virtual* dengan prediksi text dan kamus pengguna, dan perubaha UI.

3. Android versi 1.6 (*Donut*)

   Versi 1.6 dirilis pada 15 september 2009 dengan nama kode Donut. Terdapat beberapa pembaharuan dan penambahan fitur pada versi 1.6 ini diantaranya yaitu pencarian text dan suara diperluas, galeri, kamera, dan perekam video yang lebih terintegrasi serta akses kamera yang lebih cepat. Pembaruan dukungan teknologi CDMA/EVDO, 802. 1x, VPN, dan mesin pengucap text.

4. Android versi 2.0, 2.1 (*Eclair*)

   Pada 26 oktober 2009 Android 2.0 dirilis dengan kode nama Eclair. Perubahan yang dilakukan adalah pengoptimalan *hardware.* Peningkatan Google *Maps* 3.1.2, perubahan UI dengan Browser baru serta dukungan HTML5, daftar kontak baru, dukungan flash untuk kamera 3,2 MP, sinkronisasi akun diperluas, dukungan *Bluetooth* 2.1, peningkatan kecepatan mengetik pada *keyboard virtual*.

5. Android versi 2.2 (*Froyo: Frozent Yoghurt*)

   Pada 20 mei 2010 Android versi 2.2 dengan nama kode Froyo dirilis oleh Google. Perubahan-perubahan umumnya terhadap versi sebelumnya antara lain dukungan Adobe Flash 10.1, kecepatan kinerja dan aplikasi dua sampai lima kali lebih cepat, integrasi V8 JavaScript engine yang dipakai Google Chrome yang mempercepat kemampuan *rendering* pada *browser,* pemasangan aplikasi dalam SD *Card*, meningkatkan fungsionalitas USB *tethering* dan Hotspot WiFi dan kemampuan *auto update* pada Android Market.

6. Android versi 2.3 (*Gingerbread*)

   Google merilis Android versi 2.3 pada 6 desember 2010 dengan nama kode Gingerbread. Terdapat perubahan dan penambahan fitur pada versi ini diantarnya yaitu memperbaharui desain antarmuka dengan meningkatkan kecepatan, dukungan resolusi dan ukuran layar besar, peningkatan fungsi *copy paste*, dukungan format video VP8 dan WebM, dukungan *Near Field Communication* (NFC), *download manager* baru, dukungan sensor yang lebih banyak, peningkatan manajemen daya dengan peran lebih aktif dalam mengelola aplikasi yang beroperasi terlalu lama, dan dukungan *multi* kamera pada perangkat.

7. Android versi 3.0 (*Honeycomb*)

   Android versi 3.0 dirilis pada 22 februari 2011 dengan nama kode Honeycomb. Versi Android pertama yang memiliki dukungan untuk perangkat tablet. Fitur-fitur yang diperbaharui antaralain adalah pengoptimalan dukungan tablet dengan antarmuka baru, dukungan obrolan

video dengan mengunakan Google Talk, dukungan procesor multi-core, peningkatan kinerja WiFi, dukungan format *audio* FLAC, akselerasi perangkat keras, *widget* dapat diganti ukurannya, penambahan *system bar* dan *action bar.*

8. Android versi 4.0 (*Ice Cream Sandwich*)

   Pada 19 oktober 2011 Android versi 4.0 dirilis dengan nama kode Ice Cream Sandwich membawa fitur Android versi 3.0 dan menambahkan fitur baru termasuk membuka kunci dengan pengenalan wajah, jaringan data pemantauan data dan kontrol terpadu kontak jaringan sosial, perangkat tambahan fotografi, mencari email secara *ofline*, dukungan perekaman video 1080p pada perangkat tertentu, dan berbagi informasi dengan menggunakan NFC.

9. Android versi 4.1, 4.2, 4.3 (*Jelly Bean*)

   Google mengumumkan Android versi 4.1 dengan nama kode Jelly Bean dalam konferensi Google I/O pada 27 juni 2012. Terdapat perbaharuan dan penambahan fitur penting pada versi ini diantaranya yaitu peningkatan antarmuka yang lebih halus, peningkatan *keyboard virtual* yang bisa dimodifikasi oleh pengguna, pencarian melalui suara lebih cepat, peningkatan *aksesbilitas* dan lain sebagainya.

10. Android versi 4.4 (*Kitkat*)

    Pada 31 Oktober 2013 Google merilis Android versi 4.4 yang diberi nama kode Kitkat. Terdapat pembaharuan dan penambahan fitur diantaranya yaitu pembaruan antarmuka dengan bar status dan navigasi transparan, optimasi kinerja pada perangkat yang memiliki spesifikasi rendah, kerangka kerja baru untuk transisi antarmuka, perekam aktifitas layar yang terintegrasi, peningkatan suara *audio,* dukungan MAP.

11. Android versi 5.0 (*Lollipop*)

    Android versi 5.0 pertama kali diperkenalkan dengan nama kode Lollipop pada 25 juni 2014 dikonferensi Google I/O. Terdapat dukungan dari sejumlah platform Android yang berorientasi dan teknologi baru diantaranya Android TV, Android Wear, dan *platform* kesehatan Google *Fit*. Perubahan yang

menonjol pada versi ini adalah tampilan antarmuka yang semakin bagus dan dinamai material desain dengan peningkatan penggunaan *layout* berbasis *grid*, animasi dan transisi responsif, *padding*, dan efek kedalaman seperti pencahayaan dan bayangan. Penambahan fitur *Project Volta* dimana daya hidup baterai 30 persen lebih lama.

12. Android versi 6.0 (*Marshmallow*)

    Android versi 6.0 yang memiliki nama kode Marshmallow terdapat perubahan dan penambahan fitur dari versi sebelumnya diantarnya yaitu pembaharuan manajemen daya baterai sehingga tahan lama, penyempurnaan stilus Bluetooth, dukungan MIDI dan lain sebagainya.

13. Android versi 7.0 (*Nougat*)

    Pada 22 agustus 2016 Google merilis Android 7.0 dengan nama kode Nougat. Seperti versi-versi sebelumnya pada Android 7.0 terdapat perbaikan dan penambahan fitur seperti mendukungnya platform *virtual reality*, peningkatan fungsionalitas Doze untuk memperpanjang masa pakai baterai, penambahan emoji baru, dan lain sebagainya.

14. Android versi 8.0 (*Oreo*)

    Pada 21 agustus 2017 google merilis versi Android terbaru yang diberi nama kode Oreo. Terdapat beberapa fitur baru diantaranya *Picture in Picture* dimana pengguna dapat menonton video sambil menjalankan aplikasi lainnya, dukungan codec Sony LDAC, layar kunci di desain ulang dan lain sebagainya.

2.5 Multimedia

Setiap orang dapat mengartikan multimedia berbeda-beda sesuai sudut pandang orang tersebut. Secara umum, multimedia adalah gabungan satu atau lebih media yang berbeda untuk menyajikan suatu informasi. Rekaman musik hanya menggunakan media suara maka tidak disebut dengan multimedia tetapi *music video* adalah bentuk dari multimedia karena gabungan dari media suara dan *video* [10].

Dalam dunia teknologi informasi dan komputer, multimedia berhubungan erat dengan perangkat lunak yang digunakan untuk menyampaikan informasi kepada pengguna. Multimedia dalam bidang komputer nyatanya bukan merukan hal baru, karena gabungan antara text dan gambar merupakan media utama pada perangkat lunak aplikasi yang sudah lama dilakukan [10].

Pada saat ini, multimedia tidak hanya gabungan dan integrasi dari text dan grafik sederhana saja, tetapi sudah dilengkapi dengan suara, video, dan animasi sehingga informasi yang disampaikan kepada pengguna menjadi lebih interaktif. Multimedia terdiri dari beberapa komponen, yaitu:[10]

1. Teks

Teks merupakan dasar dari pengolahan kata dan informasi multimedia. Dengan gabungan dari media lain sehingga dapat menyajikan informasi kepada pengguna aplikasi dengan cepat, sehingga teks yang ditampilkan tidak harus rinci dan teliti karena sudah dibantu oleh media lain seperti gambar maupun suara.

2. Images

Secara umum *images* berarti gambar raster (*halftone drawing*), seperti foto. Pada saat ini foto maupun gambar ukurannya semakin besar dan kualitasnya pun semakin bagus. Hal ini yang menyebabkan ukuran aplikasi multimedia semakin besar dan memerlukan media penyimpanan yang cukup besar.

3. Animasi

Animasi menggambarkan objek yang bergerak agar kelihan hidup. Untuk membuat suatu animasi memerlukan gambar yang cukup banyak dan dari kumpulan gambar tersebut membentuk suatu gerakan seperti kartun. Animasi mulai dikenal sejak media televisi mulai menyajikan gambar-gambar bergerak yang berasal dari rekaman kamera atau karya animator. Teknik pengerjaan animasi terus berkembang denagan seiring dengan perkembangan teknologi komputer sehingga pengerjaannya jadi lebih cepat dan garik dan efek yang ditampilkan semakin bagus.

4. Suara

Suara dapat lebih memperjelas pengertian yang ditampilkan dengan cara lain. Contohnya, narasi merupakan kelengkapan dari penjelasan yang dilihat dari video. Ada banyak format suara sehinnga dapat diputar diperangkat komputer misalnya

Advance Audio Coding (.m4a), *Waveform Audio* (.wav), Mpeg Audio Layer 3 (.mp3), MIDI (*Music Instrument digital interface*), dan lain sebagainya.

5. *Interactive link*

Multimedia haruslah interaktif dalam penyampaiannya, dimana pengguna dapat menekan suatu objek pada screen seperti button sehingga menyebabkan aplikasi melakukan perintah tertentu. Interactive link diperlukan untuk menghubungkan beberapa elemen multimedia sehingga menjadi suatu informasi yang terpadu.

2.6 OOP (*Object Oriented Programming*)

Object oriented programming atau pemrograman berorientasi objek merupakan cara memandang persoalan menggunakan model-model yang di organisasikan seputar konsep objek yang mengkombinasikan struktur data dan perilaku suatu entitas. Pada pendekatan ini, organisasi perangkat lunak adalah sebagai kumpulan objek diskrit yang saling berkerjasama, berkomunikasim, dan berinteraksi menuju sasaran tertentu.

Pada pendekatan berorientasi objek, blok pembangun utama sistem perangkat lunak adalah objek dan atau kelas. Berdasarkan Abbot, kemudian diikuti oleh Rumbaught menyatakan bahwa kandidat-kandidat kelas umumnya dapat merupakan sesuatu yang dapat diperoleh dari kosakata (*vocabulary*) di ruang persoalan dan atau ruang solusi. Kelas adalah deskripsi himpunan objek yang serupa. Kelas merupakan cetak biru dari objek. Setiap objek memiliki identitas (berbeda dengan objek lain), *state* (data yang diasosiasikan dengannya), dan prilaku (kita dapat melakukan sesuatu pada objek dan objek dapat melakukan sesuatu pada objek lain).

Pendekatan berorientasi objek menawarkan pemakaian bersama di beberapa tingkat berbeda. Pewarisan struktur data dan prilaku memungkinkan struktur dan prilaku dipakai bersama diantara beberapa subklas serupa tanpa harus terjadi redudansi. Pemakaian bersama kode diseluruh hirarki pewarisan adalah salah satu keunggulan utama pendekatan orientasi objek yang diterapkan pada bahasa berorientasi objek. Pengembangan berorientasi objek tidak hanya memungkinkan

informasi dipakai bersama di dalam satu aplikasi, tapi juga penggunaan ulang (*rause*) rancangan dan kode pada proyek-proyek berikutnya [18].

2.7 UML (*Unifled Modeling Language*)

UML (*Unifled Modeling Language*) merupakan bahasa grafis untuk mendokumentasi, menspesifikasikan, dan membangun sistem perangkat lunak. UML berorientasi objek, menerapkan banyak level abstraksi, tidak bergantung proses pengembangan, tidak bergantung bahasa dan teknologi, pemaduan beberapa notasi di beragam metodologi, usaha bersama dari banyak pihak, didukung oleh kakas-kakas yang diintegrasikan lewat XML (XMI).

UML adalah bahasa pemodelan untuk menspesifikasikan, memvisualisasikan, membangun dan mendokumentasikan artifak-artifak dari sistem. UML bukanlah bahasa pemrograman visual, tetapi bahasa pemodelan visual. Ada perbedaan anatara metode dan bahasa pemodelan. Perbedaan utama antara metode dan bahasa pemodelan adalah bahasa pemodelan tidak menyatakan proses atau instruksi untuk melakukan apa, bagaimana melakukannya, kapan melakukan, dan kenapa dilakukan.

Uml menyediakan sejumlah diagram untuk mengekspresikan pemodelan berorientasi objek yang dilakukan. Kegunaan diagram tersebut untuk formalisasi ekspresi model secara koheren, presisi dan mudah dirumuskan. Diagram UML yang digunakan pada penelitian ini diantarnya *Use case Diagram*, *Activity Diagram*, *Class Diagram,* dan *Sequence Diagram* [14].

2.7.1 *Use Case Diagram*

Diagram *use case* digunakan untuk mendeskripsikan apa yang seharusnya dilakukan oleh sistem. Setiap diagram *use case* menunjukan sekumpulan *use case,* aktor, dan hubungannya masing-masing. *Use case* penting untuk memvisualisasikan, menspesifikasikan, dan mendokumentasikan kebutuhan prilaku sistem. Diagram *use case* melibatkan sistem yang akan dibangun, aktor sebagai pengguna sistem, *use case* sebagai fungsionalitas yang dipersepsi oleh aktor, dan relasi anatar aktor dan *use case*.

Tujuan utama pemodelan *use case* adalah untuk memutuskan dan mendeskripsikan kebutuhan-kebutuhan fungsional sistem. Memberikan deskripsi jelas dan konsisten dari apa yang seharusnya dilakukan oleh sistem serta menyediakan basis untuk melakukan pengujian sistem yang memverfikasi sistem. Ada beberapa elemen pada diagram *use case* diantaranya adalah:[14]

1. Aktor

Aktor adalah pengguna sistem, dapat berupa manusia atau sistem terotomatisasi lain. Sebagai pengguna aktor dapat berinteraksi dengan sistem dimana aktor dapat mengirim atau menerima pesan ke atau dari sistem, atau bertukar informasi dengan sistem. Aktor merepresentasikan peran bukan pemakai individu dari sistem.

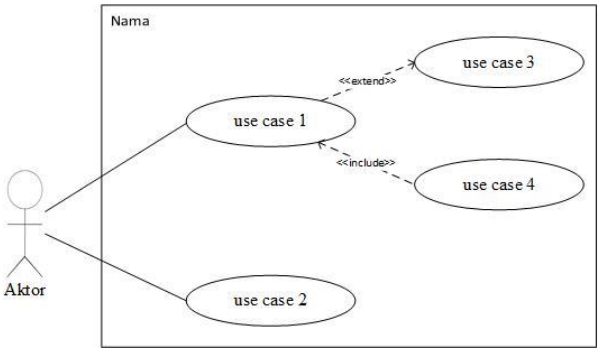
2. *Use case*

Use case menspesifikasikan perilaku sistem atau bagian sistem dan merupakan deskripsi sekumpulan sekuen aksi termasuk varian-varian yang dilakukkan sistem untuk memberikan hasil hasil atau nilai ke aktor. *Use case* merupakan fungsionalitas yang terdapat pada sistem dan saling berinteraksi antar *use case* dan aktor. Kegunaan *use case* untuk menangkap kebutuhan sistem serta berkomunikasi dengan pemakai atau aktor dan juga dapat digunakan untuk pengujian sistem.

3. Hubungan antar *use case*

Ada beberapa jenis hubungan atau relasi antar *use case* dengan *use case* lainnya, diantaranya:

1. *Assosiation*, menggambarkan hubungan antara *use case* dengan aktor.
2. *Generalization*, menggambarkan hubungan antara dua aktor atau dua *use case* untuk membuatnya lebih spesifik.
3. *Dependency*, menggambarkan hubungan ketergantungan antara *use case* yang satu dengan yang lain.



Gambar 2. 12 Contoh *Use Case*

2.7.2 *Activity Diagram*

Activity diagram atau diagram aktivitas berguna untuk memperlihatkan aliran satu aktivitas ke aktivitas lain yang dideskripsikan oleh *use case* dan berguna untuk memodelkan aspek dinamis sistem. Diagram aktivitas mendeskripsikan aksi-aksi dan hasilnya. Diagram aktivitas berfokus pada aktivitas-aktivitas, potongan-potongan dari proses yang boleh jadi (mungkin) berkorespondensi dengan metode-metode atau fungsi-fungsi anggota dan pengurutan dari aktivitas-aktivitas ini [14].

Diagram aktivitas memberikan representasi grafis dari aliran-aliran interaksi didalam suatu skenario yang sifatnya spesifik. *Use case-use case*, di dalam diagram aktivitas bersifat berorientasi pada prosedur. Mereka menggambarkan bagaimana berbagai aktor memanggil fungsi-fungsi tertentu yang sifatnya spesifik (atau tahapan-tahapn prosedural lainnya) sehingga sesuai dengan kebutuhan-kebutuhan sistem atau perangkat lunak[15] .

2.7.3 *Class Diagram*

Class diagram atau diagram kelas adalah pemodelan paling utama di pendekatan berorientasi objek. Pemodelan kelas menunjukan kelas-kelas yang ada pada sistemdan hubungan antar kelas-kelas itu, atribut-atribut dan opersi-operasi di kelas. Diagram kelas digunakan untuk menunjukan peran yang terdapat pada entitas-entitas pada sistem yang memberikan fungsionalitas sistem.

Diagram kelas menunjukan aspek statik sistem terutama untuk mendukung kebutuhan fungsional sistem. Kebutuhan fungsional berarti layanan-layanan yang

harus disediakan sistem ke pengguna. Meskipun diagram kelas serupa dengan model data, namun kelas-kelas tidak hanya menunjukan struktur informasi tapi juga mendeskripsikan perilaku.

Kelas merupakan elemen terpenting dalam sistem berorientasi objek. Kelas mendeskripsikan satu blok pembangun sistem dimana didalam kelas terdapat nama kelas, atribut kelas yang mendeskripsikan nilai yang dipunyai instan kelas, method atau operasi untukmendeskripsikan fungsi yang terdapat pada sistem.

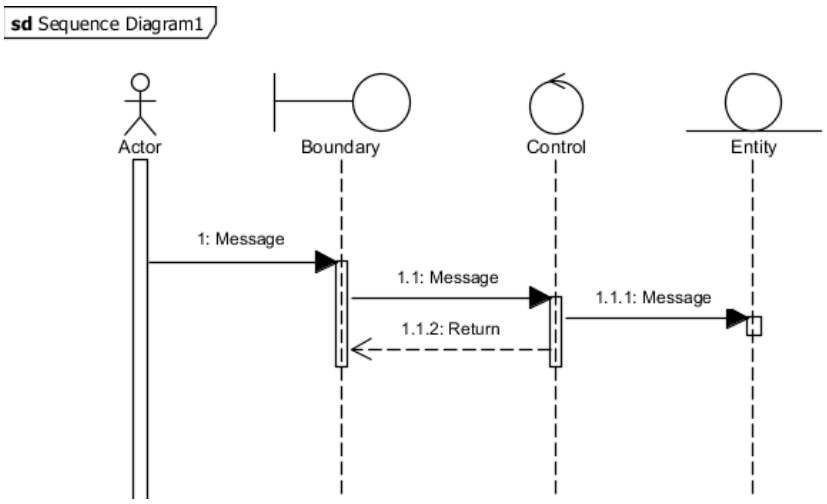
Ada beberapa hubungan atau relasi antar kelas pada diagram kelas diantaranya[14]:

1. Asosiasi, hubungan statis antar kelas dimana suatu kelas memerlukan atribut dari kelas lain.
2. Generalisasi, gabungan dua atau lebih class yang memiliki sifat sama yang membentuk kelas baru.
3. Dependensi, hubungan antar kelas dimana suatu kelas mengacu kepada kelas lainnya.
4. Aggregasi, meunjukan bahwa sebuah kelas merupakan bagian dari kelas lain tetapi bersifat tidak wajib.
5. Komposisi, menunjukan bahwa suatu kelas merupakan bagian dari kelas lain.
6. Realisasi. Menunjukan peran suatu *interface* terhadap kelas.

2.7.4 *Sequence Diagram*

Sequence diagram untuk memodelkan skenario penggunaan dengan mendeskripsikan waktu hidup objek dan *message* selama satu eksekusi sistem. *Sequence diagram* mendeskripsikan komunikasi diantara objek-objek, meliputi pesan-pesan yang ada dan urutan pesan tersebut muncul.

Sequence diagram diambil dari *use case* yang di detailkan kembali. Objek ditunjukan sebagai garis *vertikal* dan setiap *message* digambarkan dengan garis *horizontal* dari objek pengirim ke objek penerima. Waktu digambarkan dengan *bar* yang berada pada garis objek dan menunjukan berapa lama waktu eksekusi sebuah proses [14].

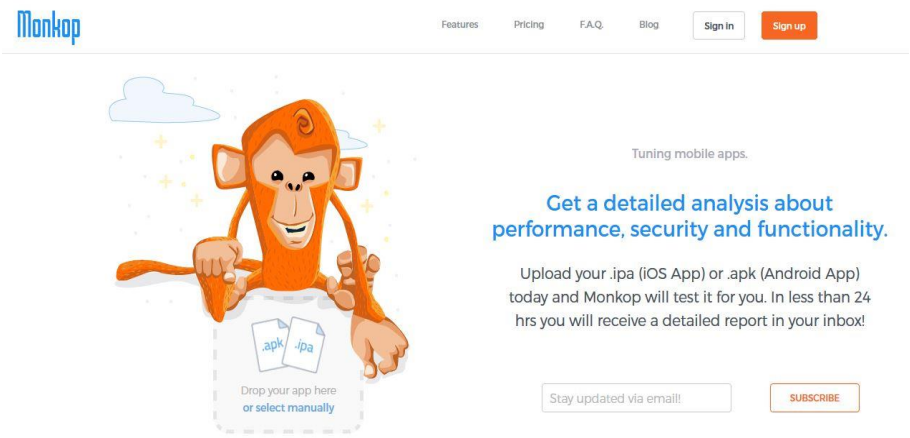


Gambar 2. 13 Contoh Sequence Diagram

2.8 Pengujian *Performance*

Pengujian *performance* digunakan untuk mengukur parameter sistem atau aplikasi dalam hal respon, beban aplikasi, kinerja dan penggunaan sumber daya. Tujuan dari pengujian *performance* yaitu untuk menghilangkan hambatan-hambatan yang dapat menyebabkan aplikasi lambat untuk diakses, *load* data lama dan terlalu menguras sumber daya pada perangkat keras *smartphone.*

Pengujian *performance* yang dilakukan pada *game* edukasi peribahasa Indonesia menggunakan aplikasi *website* yaitu Monkop.com. Monkop adalah Android *Tuning Advisor* yang menyediakan data unik tentang kinerja dan keamanan dengan laboratorium perangkat yang mendukung berbagai jenis merek *smartphone* dan tablet, versi sistem operasi, ukuran layar dan konfigurasi. Berikut ini tampilan dari *website* Monkop dapat dilihat pada gambar 2.14.



Gambar 2. 14 Tampilan Website Monkop

Monkop bekerja dengan cara menginstal aplikasi secara bersamaan pada berbagai jenis perangkat ponsel dan tablet yang mempertimbangkan tingkat komtabilitas. Berikutnya menjalankan aplikasi dan menganalisis setiap layar dan menjalankan tindakan yang berbeda dengan data sintetis atau nyata (jika diberikan sebelum pengujian). Sepanjang peroses, Monkop akan mengukur setiap indikator untuk mendapatkan informasi yang akurat tentang kinerja, kerentanan dan perilaku yang akan dianalisis oleh mesin tuning Monkop [17].

Hasil dari pengujian aplikasi webiste Monkop berupa laporan rinci tentang aplikasi pada setiap perangkat termasuk video eksekusi, layar, grafik rinci tentang CPU, RAM, Threads, GCollection, koneksi jaringan, render, *database*, tingkat konsumsi daya dan lain sebagainya [17].

2.9 Pengujian *Blackbox*

Pengujian *blackbox*, juga disebut pengujian prilaku, untuk menguji fungsionalitas perangkat lunak. *Blackbox* dapat memungkinkan untuk membuat beberapa kumpulan kondisi masukan yang sepenuhnya akan melakukan kebutuhan fungsional untuk program. Pengujian *blackbox* dimaksudkan untuk menemukan kesalahan, diantarnya kesalahan fungsi yang salah atau hilang, kesalahan *interface* atau antarmuka, kesalahan dalam struktur data atau akses basis data eksternal, kesalahan prilaku atau kinerja, dan kesalahan inisialisasi dan penghentian [15].

2.10 Skala *Likert*

Skala *Likert* digunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang tentang fenomena sosial. Dalam penelitian, fenomena sosial ini telah di tetapkan secara spesifik oleh peneliti, yang selanjutnya disebut dengan variabel penelitian.

Dengan skala *Likert,* maka variabel yang akan diukur dijabarkan menjadi indikator variabel. Kemudian indikator tersebut dijadikansebagai titik tolak untuk menyusun item-item instrumen yang dapat berupa pernyataan atau pertanyaan.

Jawaban setiap item instrumen yang menggunakan skala *Likert* mempunyai gradasi dari sangat positif sampai sangat negatif dan setiap jawaban tersebut diberi skor untuk keperluan analisis kuantitatif, misalnya dapat dilihat pada tabel 2.1 [21].

Tabel 2. 1 Skor Skala Likert

Keterangan	Skor
Sangat Setuju	5
Setuju	4
Ragu-ragu	3
Tidak Setuju	2
Sangat Tidak Setuju	1

2.11 S*oftware* Yang Digunakan

Adapun beberapa *tools* atau perangkat lunak yang digunakan dalam pembangunan *game edukasi* peribahasa Indonesia diantaranya adalah Unity, Blender, Mixamo.

2.11.1 Unity

Unity merupakan sebuah *game engine*, yaitu *software* pengolah gambar, grafik, *audio*, *input*, dan lain sebagainya yang ditujukan untuk membuat suatu *game*. Kelebihan dari unity yaitu *game engine multiplatform*, dimana *game* tersebut dapat di-*publish* secara *standalone* (.exe), berbasis web, android, IoS, XBOX, nintendo maupun playstation, dengan catatan harus ada lisensinya terlebihdahulu. Unity dibagi menjadi dua versi, yaitu versi berbayar dan versi gratis. Versi gratis tentunya -tidak semua fitur ada misalnya tidak dapat melakukan konversi game ke console

seperti XBOX, playstation maupun nintendo. Banyak fitur yang dimiliki unity diantaranya yaitu:[16]

1. *Rendering*

   Unity telah mendukung penggunaan *graphic engin*, seperti Direct3D (Windows, Xbox), OpenGL (Mac, Windows, Linux, Playstation), OpenGL ES (Android, iOS), dan APIs (Nintendo). Selain itu Unity juga mendukungpenggunaan *bump mapping, reflection mapping, parallax mapping, screen space ambient occlusion* (SSAO), *dynamic shadows* menggunkan *shadow maps,* render to texture dan full screen post processing effect.

2. *Scripting*

   Dalam pembuatan suatu *game* tentunya memerlukan bahasa pemrograman akan tetapi *game engine* berbeda dengan pembuatan *game* menggunakan bahasa pemrograman langsung, dengan *game engine* tidak perlu membuat *game* dari nol, dikarenakan *game engine* sudah menyediakan bahan-bahan dasar dalam pembuatan *game,* seperti karakter, peta, tleset dan lain sebagainya. *Scripting* yang digunakan pada Unity dibangun menggunakan Mono 2.6. Mono 2.6 merupakan implementasi *open source* dari .NET Framework. Bahasa pemrograman yang didukung oleh Unity diantaranya JavaScript, C#, dan Boo (menggunakan sintaks Phyton).

3. *Asset Store*

   Asset merupakan aspek dari permainan yang akan direfrensikan oleh beberapa komponen, asset itu sendiri, atau kelengkapan penunjang pembuatan *game. Asset store* merupakan tempat untuk mendapatkan asset yang digunakan untuk menunjang dalam pembuatan *game*. Ada ribuan asset yang tersedia pada *asset store* baik yang gratis maupun berbayar. Asset yang ada pada Unity dibagi menjadi dua, yaitu eksternal dan internal. Asset eksternal merupakan asset yang ditambahkan dari sumber di luar Unity, seperti 3D model, Texture, Sound Effect dan sebagainya. Asset internal merupakan asset yang sudah ada pada Unity, seperti Materials, Shaders, Cube Maps, Physics Materials, dan Prefabs.

2.11.2 Mixamo

Mixamo adalah sebuah *web aplication online* yang memiliki berbagai fitur untuk membantu dalam pembuatan animasi atau gerakan pada objek 3D. Fitur pertama adalah *auto rigging* atau pembuatan kerangka tulang pada model 3D secara otomatis. Mixamo juga menyediakan berbagai jenis animasi yang dapat digunakan misalnya animasi *idle,* loncat, lari, jalan dan lain sebagainya. Model yang sudah di *rigging* dan di animasikan dapat langsung di *export* ke *file* .fbx, .fbx khusus Unity dan .obj.