

NO: 01022079/INF/2021

***FINITE STATE MACHINE* UNTUK MENGHASILKAN KALIMAT SEBAGAI ALAT  
BANTU KOMUNIKASI BAGI ANAK PENYANDANG AUTISME**

**SKRIPSI**

Digunakan untuk memenuhi persyaratan penyelesaian program S-1  
Program Studi Informatika Fakultas Teknologi Industri  
Universitas Kristen Petra

Oleh :  
Steven Hans Gunadi Chua  
NRP : C14170030

**PROGRAM STUDI INFORMATIKA**



**FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS KRISTEN PETRA  
SURABAYA  
2021**

**LEMBAR PENGESAHAN**

**SKRIPSI**

***FINITE STATE MACHINE* UNTUK MENGHASILKAN KALIMAT SEBAGAI ALAT  
BANTU KOMUNIKASI BAGI ANAK PENYANDANG AUTISME**

Oleh :

Steven Hans Gunadi Chua

NRP : C14170030

Diterima Oleh :

Program Studi Informatika  
Fakultas Teknologi Industri  
Universitas Kristen Petra

Surabaya, 23 Juni 2021

Pembimbing I

Pembimbing II

LILIANA, S.T., M.Eng., Ph.D.  
NIP: 03-024

LILY EKA SARI, S.S., M.A.  
NIP: 16-011

Ketua Tim Penguji

Prof. Dr.(H.C.) Ir. ROLLY INTAN, M.A.Sc., Dr.Eng.  
NIP: 92-008

Ketua Program Studi

HENRY NOVIANUS PALIT, Ph.D.  
NIP: 14001

**LEMBAR PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN  
AKADEMIS**

Sebagai mahasiswa Universitas Kristen Petra, yang bertanda tangan dibawah ini,  
saya:

Nama : Steven Hans Gunadi Chua

NRP : C14170030

Demi mengembangkan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Kristen Petra Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalty-Free Right*) atas karya ilmiah saya yang berjudul: *Finite State Machine* untuk Menghasilkan Kalimat Sebagai Alat Bantu Komunikasi Bagi Anak Penyandang *Autism*. Dengan Hak Bebas Royalti Non-Eksklusif ini Universitas Kristen Petra berhak menyimpan, mengalih-media format-kan, mengelolanya dalam bentuk pangkalan data (*database*), mendistribusikannya dan menampilkan/ mempublikasikannya di internet atau media lain untuk kepentingan akademis tanpa perlu meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta.

Saya bersedia untuk menanggung secara pribadi, tanpa melibatkan pihak Universitas Kristen Petra, segala bentuk tuntutan hukum yang timbul atas pelanggaran Hak Cipta dalam karya ilmiah saya ini.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Surabaya, 11 Juni 2021

Yang menyatakan,

(Steven Hans Gunadi Chua)

## KATA PENGANTAR

Puji syukur kepada Tuhan Yang Maha Esa atas berkat dan pimpinan-Nya yang telah diberikan selama pengerjaan skripsi ini, sehingga penulis dapat menyelesaikan skripsi ini dengan baik.

Pada kesempatan ini penulis menyampaikan ucapan terima kasih yang sebesar-besarnya kepada orang-orang yang telah berperan sehingga skripsi ini dapat terselesaikan, antara lain:

1. Henry Novianus Palit, Ph.D., selaku Ketua Program Studi Teknik Informatika dan Sistem Informasi Bisnis Universitas Kristen Petra.
2. Liliana, S.T., M.Eng., Ph.D., selaku dosen pembimbing I, yang telah memberikan arahan, motivasi serta meluangkan waktu selama proses pembuatan skripsi berlangsung.
3. Lily Eka Sari, S.S., M.A., selaku dosen pembimbing II, yang telah memberikan arahan, motivasi serta meluangkan waktu selama proses pembuatan skripsi berlangsung.
4. Silvia Rostianingsih, M.MT., selaku Koordinator Skripsi Program Teknik Informatika dan Sistem Informasi Bisnis Universitas Kristen Petra.
5. Segenap dosen dan staf pengajar di Program Studi Teknik Informatika Universitas Kristen Petra.
6. Keluarga yang telah banyak memberikan dukungan doa dan motivasi hingga penulis mampu menyelesaikan skripsi ini.
7. Semua teman-teman yang telah banyak memberikan dukungan dan bantuan kepada penulis sehingga penulis dapat menyelesaikan skripsi ini.
8. Pihak Pusat Terapi Autis Surabaya yang sudah meluangkan waktu untuk memberikan kesempatan uji coba dan memberikan informasi untuk keperluan skripsi ini.
9. Pihak-pihak lain yang telah memberikan bantuan secara langsung maupun tidak langsung dalam pembuatan skripsi ini yang tidak dapat disebutkan satu per satu.

Peneliti menyadari bahwa penulisan skripsi ini masih jauh dari sempurna. Oleh karena itu, penulis mengharapkan segala petunjuk, kritik, dan saran yang membangun dari pembaca agar dapat menunjang pengembangan dan perbaikan penulisan.

Akhir kata, penulis mohon maaf apabila ada kekurangan dalam penulisan tugas akhir ini dan penulis dengan senang hati menerima saran dan kritik yang membangun dari pembaca.

Surabaya, 11 Juni 2021

Penulis

## ABSTRAK

Steven Hans Gunadi Chua:

Skripsi

*Finite State Machine* untuk Menghasilkan Kalimat Sebagai Alat Bantu Komunikasi  
Bagi Anak Penyandang *Autism*

Salah satu hambatan bagi anak penyandang autisme adalah komunikasi. Anak penyandang autisme non-verbal kesulitan untuk melakukan interaksi dengan orang lain atau hanya sekedar menyampaikan keinginannya. Pada saat keinginan anak tidak terpenuhi, anak penyandang autisme akan menjadi tantrum, yang tentu saja akan membahayakan karena bisa menyakiti anak tersebut. Saat ini sudah ada intervensi dengan menggunakan metode PECS yang dapat membantu anak penyandang autisme belajar berkomunikasi menggunakan media kartu bergambar. Akan tetapi kekurangan dari metode PECS ini adalah banyaknya kartu bergambar yang harus disediakan untuk anak penyandang autisme.

Pada penelitian ini dibuat sebuah aplikasi Android untuk membantu anak penyandang autisme belajar berkomunikasi. Aplikasi yang dibuat mengikuti tahapan penggunaan PECS dan juga memanfaatkan *finite state* untuk merangkai kalimat.

Berdasarkan hasil uji coba yang telah dilakukan, aplikasi yang dibuat dapat membantu anak penyandang autisme non-verbal untuk melakukan komunikasi, meskipun beberapa anak memerlukan waktu lebih untuk memahami penggunaan aplikasi.

Kata kunci:

Autisme, PECS, *Finite state machine*, Android

## ABSTRACT

Steven Hans Gunadi Chua:

Undergraduate Thesis

Finite State Machine to Produce Sentences as Communication Aid for Children with Autism

One of the obstacles for children with autism is communication. Children with autism, who are nonverbal, have difficulty interacting with other people or just expressing their needs. When the child's wishes are not fulfilled, the child with autism will tend to throw a tantrum, which may be harmful because it can hurt the child. Currently, there is intervention by using the PECS method that can help children with autism learn to communicate by using picture cards. However, the drawback of this PECS method is the large number of picture cards that must be provided for children with autism.

In this study, an Android application created to help children with autism learn to communicate. The application follows the stages of using PECS and also utilizes the finite state machine to compose sentences.

Based on the results of this study, the application was able to help children with autism, who are nonverbal to communicate, although some children need more time to understand how to use of the application.

Keyword:

*Autism, PECS, Finite state machine, Android*

## DAFTAR ISI

HALAMAN JUDUL .....	i
LEMBAR PENGESAHAN.....	ii
LEMBAR PERNYATAAN .....	iii
KATA PENGANTAR.....	iv
ABSTRAK .....	vi
ABSTRACT .....	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	xi
DAFTAR TABEL.....	xii
DAFTAR SEGMENT.....	xiii
1. PENDAHULUAN .....	1
1.1. Latar Belakang .....	1
1.2. Perumusan masalah.....	3
1.3. Tujuan Skripsi .....	3
1.4. Ruang Lingkup.....	3
1.5. Metodologi Penelitian .....	4
1.6. Sistematika penulisan .....	4
2. LANDASAN TEORI .....	6
2.1. Tinjauan Pustaka.....	6
2.1.1. Speech Generating.....	6
2.1.2. Finite state machine.....	6
2.1.3. PECS .....	7
2.1.4. Autism Spectrum Disorder.....	9
2.2. Tinjauan Studi10	
2.2.1. Game Edukasi Berbasis Kinect untuk Anak Berkebutuhan Khusus (Autis) dengan Metode Finite State Machine (Anugraha et al., 2018) .....	10
2.2.2. Penerapan metode finite state machine pada game “The Relationship” (Rahadian et al., 2016).....	11
3. ANALISA DAN DESAIN SISTEM .....	12
3.1. Analisa Sistem .....	12



3.1.1.	Finite State Machine .....	12
3.2.	Flowchart .....	14
3.2.1.	Flowchart menu favorit .....	14
3.2.2.	Flowchart menu beda gambar .....	15
3.2.3.	Flowchart menu kalimat .....	16
3.3.	Desain aplikasi .....	17
3.3.1.	Halaman utama.....	17
3.3.2.	Menu favorit .....	18
3.3.3.	Menu beda gambar.....	22
3.3.4.	Menu kalimat .....	23
3.3.5.	Halaman pengaturan .....	27
3.3.6.	Halaman gambar yang ditambah .....	28
3.3.7.	Halaman gambar favorit .....	29
3.3.8.	Halaman Tambah gambar .....	30
4.	IMPLEMENTASI SISTEM .....	32
4.1.	Implementasi menu favorit .....	32
4.1.1.	Implementasi memilih kategori .....	32
4.1.2.	Menampilkan data gambar sesuai kategori.....	33
4.1.3.	Menyimpan data .....	35
4.1.4.	Menampilkan data favorit .....	36
4.2.	Implementasi menu beda gambar.....	37
4.3.	Implementasi menu kalimat .....	42
4.4.	Implementasi tambah gambar .....	49
4.4.1.	Fitur pilih gambar .....	49
4.4.2.	Fitur tambah gambar .....	50
4.5.	Daftar gambar yang ditambah .....	50
4.6.	Implementasi text-to-speech.....	51
5.	PENGUJIAN SISTEM .....	53
5.1.	Efektivitas aplikasi.....	53
5.1.1.	Kemampuan anak mengoperasikan aplikasi.....	53
5.2.	Finite state machine .....	56
5.3.	Survei kepada terapis atau orang tua .....	56
6.	KESIMPULAN DAN SARAN .....	57

6.1. Kesimpulan .....	57
6.2. Saran .....	57
7. DAFTAR REFERENSI.....	58
LAMPIRAN .....	60

## DAFTAR GAMBAR

2.1 Contoh diagram state sederhana.....	7
2.2 Ilustrasi untuk menggambarkan PECS.....	8
2.3 Ilustrasi untuk menggambarkan PECS.....	8
3.1 Finite state machine.....	12
3.2 Flowchart menu favorit.....	14
3.3 Flowchart menu beda gambar.....	15
3.4 Flowchart menu komunikasi.....	16
3.5 Tampilan halaman utama.....	18
3.6 Tampilan memilih kategori .....	19
3.7 Tampilan memilih gambar favorit.....	20
3.8 Tampilan menyimpan gambar favorit.....	21
3.9 Tampilan gambar favorit.....	22
3.10 Tampilan halaman beda gambar.....	23
3.11 Tampilan awal halaman komunikasi .....	24
3.12 Tampilan setelah menekan gambar "saya mau" .....	25
3.13 Tampilan setelah menekan gambar predikat .....	26
3.14 Tampilan setelah menekan gambar objek.....	27
3.15 Tampilan halaman pengaturan .....	28
3.16 Tampilan halaman yang ditambah.....	29
3.17 Tampilan halaman gambar favorit .....	30
3.18 Tampilan halaman tambah gambar .....	31

## DAFTAR TABEL

5.1 Pengujian responden AB .....	55
5.2 Pengujian responden RH .....	55
5.3 Pengujian responden HB .....	55
5.4 Pengujian responden JB .....	56

## DAFTAR SEGMENT

4.1 Memilih kategori .....	32
4.2 Menampilkan data sesuai kategori .....	33
4.3 Menyiapkan data.....	35
4.4 Menampilkan data favorit.....	36
4.5 Menu beda gambar .....	37
4.6 Menu kalimat .....	43
4.7 Pilih gambar.....	49
4.8 Tambah gambar .....	50
4.9 Daftar gambar yang telah ditambah .....	50
4.10 Text-to-speech .....	51

## 1. PENDAHULUAN

### 1.1. Latar Belakang

Manusia merupakan makhluk sosial, oleh karena itu manusia pasti memerlukan orang lain. Agar setiap manusia dapat saling mengerti maka diperlukan komunikasi. Akan tetapi tidak semua manusia terlahir dengan kemampuan berkomunikasi yang cukup baik dengan orang lain. Contohnya adalah anak penyandang autis. Menurut Center of Disease Control and Prevention (n.d), anak penyandang autisme memiliki hambatan dengan keterampilan sosial, emosional dan komunikasi. Hal ini dapat diatasi antara lain dengan memanfaatkan kemajuan teknologi. Beberapa aplikasi di saat ini yang menggunakan *finite state machine* untuk membantu anak penyandang autisme belajar sambil bermain.

Menurut Bors (2018), *Finite state machine* adalah model komputasi yang didasarkan pada mesin hipotesis yang terbuat dari satu keadaan atau lebih. Hanya satu status mesin yang dapat aktif pada saat yang bersamaan. Artinya mesin harus beralih dari satu kondisi ke kondisi lainnya untuk melakukan tindakan yang berbeda. Kondisi akan berubah berdasarkan input, dan memberikan output yang dihasilkan untuk perubahan yang diterapkan. Pengguna dapat berpindah ke *state* selanjutnya apabila telah memenuhi kondisi di *state* sebelumnya. Hal ini dapat dimanfaatkan dalam pembuatan aplikasi untuk anak penyandang autisme.

*Autism Spectrum Disorder* (ASD) adalah gangguan perkembangan saraf yang ditandai dengan defisit perilaku sosial dan interaksi non verbal seperti berkurangnya kontak mata, ekspresi wajah, dan gerak tubuh dalam 3 tahun pertama kehidupan (Park et al., 2016). Menurut Autism Resource Center (n.d), Autisme adalah kelainan perkembangan seumur hidup yang mempengaruhi kemampuan seseorang untuk memahami dunia dan berhubungan dengan orang lain. Anak penyandang autis cenderung mengalami kesulitan dalam berkomunikasi dengan orang lain atau berhubungan sosial. Menurut Centers for Disease Control and Prevention (n.d), Anak penyandang autis memiliki masalah dalam menyampaikan keinginan atau kebutuhan mereka. Anak penyandang autis tidak dapat disembuhkan total. Akan tetapi dapat dilakukan terapi yang dapat mengurangi gejala dari autisme tersebut dan untuk meningkatkan kemampuan anak penyandang autisme. Komunikasi pada anak berkebutuhan khusus merupakan elemen yang penting. Dengan komunikasi dapat membantu anak berkebutuhan khusus untuk membentuk keterampilan sosial, perilaku dan meningkatkan prestasi akademik

dari anak tersebut. Ketika keterampilan sosial dan perilaku dari anak terganggu maka dapat menyebabkan perkembangan anak akan menjadi lambat.

Saat ini, terdapat beberapa metode terapi bagi anak penyandang autisme, di antaranya adalah terapi komunikasi dan perilaku yang menggunakan PECS (*Picture Exchange Communication System*). PECS merupakan suatu program analisis terapan yang dimodifikasi dan dirancang untuk pelatihan komunikasi simbolik non verbal awal (Vicker, n.d). Pada dasarnya metode PECS menggunakan media kartu dengan gambar untuk mengajarkan suatu kata atau objek kepada penyandang autis. Contoh penggunaan PECS adalah anak penyandang autisme akan diberikan beberapa gambar objek yang mereka sukai seperti makanan atau mainan. Kemudian ketika anak ingin memakan sesuatu yang ada pada gambar tersebut maka anak akan mengambil gambar makanan tersebut untuk diserahkan ke pendamping atau orang tua mereka. Penggunaan media kartu tersebutlah yang menjadi kekurangan pada metode PECS ini. Hal ini disebabkan karena banyaknya kartu yang perlu disediakan oleh pengajar. Selain itu, pengajar juga menjadi kesulitan ketika berpindah tempat karena harus membawa banyak kartu bergambar tersebut. Oleh karena itu penelitian ini menggunakan aplikasi berbasis Android untuk memudahkan komunikasi kepada anak penyandang autis.

Anugraha et al (2018) melakukan penelitian dengan cara menciptakan game edukasi untuk anak penyandang autisme menggunakan *finite state machine* berbasis Kinect. Pada game tersebut, *finite state machine* dimanfaatkan untuk berpindah dari pertanyaan-pertanyaan yang dimunculkan. Contohnya adalah ketika pertanyaan pertama dijawab dengan benar maka akan dilanjutkan ke pertanyaan kedua dan apabila pada pertanyaan kedua salah menjawab maka akan kembali ke pertanyaan pertama. Sebaliknya, apabila pertanyaan kedua dijawab dengan benar maka akan lanjut ke pertanyaan ketiga, begitu seterusnya hingga permainan selesai.

Aplikasi yang dibuat dalam penelitian ini adalah aplikasi yang dapat membantu penyandang autisme dalam berkomunikasi dengan memanfaatkan *finite state machine*. Perbedaan aplikasi yang dibuat dengan penelitian sebelumnya adalah pada penelitian sebelumnya, *finite state machine* dimanfaatkan dalam game edukasi untuk anak penyandang autisme sedangkan pada penelitian ini berfokus untuk membantu komunikasi anak penyandang autisme. *Finite state machine* dapat digunakan untuk menampilkan gambar sehingga setiap gambar yang muncul dapat dipilih dan dirangkai menjadi kalimat. Implementasi sederhananya adalah apabila gambar pertama dipilih oleh pengguna maka akan lanjut ke *state* berikutnya. apabila pada *state* kedua gambar yang ditampilkan tidak sesuai dengan apa yang pengguna inginkan maka akan di *generate* gambar lainnya dan *state* tetap berada pada *state* kedua

tersebut. Fitur *touch screen* digunakan untuk membuat aplikasi menjadi lebih interaktif bagi anak Penyandang autisme dan fitur *text-to-speech* yang ada pada Android untuk mendukung komunikasi kepada anak penyandang autisme. Fitur tersebut dapat membantu seseorang yang tidak memiliki kemampuan untuk berkomunikasi dengan baik, untuk melakukan komunikasi dengan orang lain dengan bantuan aplikasi. Fitur pada aplikasi yang dibuat adalah menghasilkan suara untuk membantu anak penyandang autisme yang non verbal atau yang sulit berbicara untuk menyampaikan maksud kepada orang lain.

### 1.2. Perumusan masalah

Berdasarkan masalah yang terdapat pada latar belakang, maka rumusan masalah pada penelitian ini adalah :

1. Bagaimana efektivitas aplikasi yang dibuat terhadap anak penyandang autisme?
2. Bagaimana aplikasi yang dibuat dapat membantu proses komunikasi anak penyandang autisme?
3. Seberapa akurat aplikasi yang dibuat untuk menjadi media komunikasi bagi anak berkebutuhan khusus dengan orang lain?
4. Bagaimana memanfaatkan *finite state machine* agar dapat membuat kalimat yang sesuai dengan *grammar* Bahasa Indonesia yang baik dan benar?

### 1.3. Tujuan Skripsi

Tujuan skripsi ini adalah merancang alat bantu komunikasi untuk anak penyandang autisme dengan memanfaatkan metode *finite state machine*.

### 1.4. Ruang Lingkup

Ruang lingkup dibatasi pada :

1. Fitur *finite state machine* dimanfaatkan untuk menghasilkan kalimat berbahasa Indonesia yang sesuai dengan gambar yang dipilih oleh user
2. Gambar yang dimunculkan berupa hal-hal yang disukai oleh anak penyandang autisme
3. Ketika salah satu gambar dipilih, maka akan di *generate* gambar lain untuk mendukung keterangan gambar yang dipilih sebelumnya sehingga akan merangkai suatu kalimat.
4. Fitur *touch screen* dimanfaatkan untuk melakukan pemilihan gambar.



5. Penyuaaran kalimat yg dihasilkan menggunakan *text-to-speech*.
6. Gambar yang digunakan dalam aplikasi, digambar oleh Felicia Evelyn
7. Fitur untuk menambahkan gambar yang disimpan lokal menggunakan database ROOM.
8. Bahasa yang digunakan pada aplikasi adalah Bahasa Indonesia.
9. Hasil pengujian efektivitas didapatkan dengan cara mencatat berapa kali anak menggunakan aplikasi secara mandiri untuk menyampaikan keinginannya.
10. Aplikasi dapat digolongkan membantu anak penyandang autis jika anak tersebut sudah mampu mengoperasikan aplikasi untuk menyampaikan keinginannya
11. Pembuatan aplikasi dilakukan pada *platform* Android.
12. Aplikasi dibuat menggunakan Android Studio dan menggunakan bahasa pemrograman Kotlin.

### 1.5. Metodologi Penelitian

Langkah-langkah dalam pembuatan skripsi:

1. Studi literatur
  - 1.1. *Speech Generating*.
  - 1.2. *Finite State Machine*.
  - 1.3. PECS.
  - 1.4. *Autism Spectrum Disorder*.
2. Perancangan aplikasi
  - 2.1. Membuat desain *user interface*.
3. Pembuatan aplikasi
  - 3.1. Membuat aplikasi menggunakan Android Studio.
4. Pengujian dan analisis
  - 4.1. Melakukan pengujian kepada anak penyandang autisme.
5. Pembuatan laporan

### 1.6. Sistematika penulisan

Penulisan laporan skripsi ini dibagi menjadi beberapa bab, yaitu:

Bab 1 : Pendahuluan

Bab ini berisikan judul, latar belakang, perumusan masalah, ruang lingkup, tujuan skripsi, metodologi penelitian, dan sistematika penulisan yang akan digunakan.

## Bab 2 : Landasan Teori

Bab ini berisikan teori-teori serta metode-metode yang digunakan dalam pembuatan skripsi.

## Bab 3 : Analisis dan Desain Sistem

Bab ini berisikan analisis dan desain sistem yang dibuat meliputi perencanaan sistem yang akan dibuat, desain database yang digunakan, dan desain tampilan *user interface* yang akan dibuat.

## Bab 4: Implementasi Sistem

Bab ini berisikan tentang implementasi sistem berdasarkan desain sistem pada aplikasi atau program.

## Bab 5: Pengujian Sistem

Bab ini berisikan pengujian sistem yang telah dibuat pada Bab 4.

## BAB 6: Kesimpulan dan Saran

Bab ini berisikan kesimpulan yang dapat diambil terhadap hasil yang dicapai, dan saran-saran yang berguna bagi pengembangan selanjutnya.

## 2. LANDASAN TEORI

Pada bab ini akan dijelaskan mengenai teori-teori yang menunjang sebagai bahan studi literatur dalam pembuatan skripsi. Teori tersebut meliputi *speech generating, finite state machine, PECS, autism spectrum disorder*.

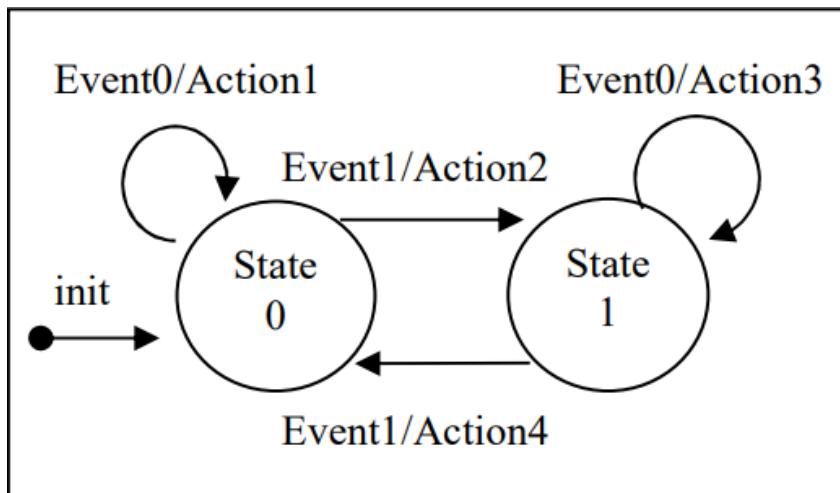
### 2.1. Tinjauan Pustaka

#### 2.1.1. *Speech Generating*

*Speech generating device* adalah sistem elektronik yang digunakan untuk melengkapi atau menggantikan ucapan atau tulisan untuk individu dengan gangguan bicara yang parah, memungkinkan mereka untuk berkomunikasi secara verbal. Perangkat penghasil ucapan adalah perangkat khusus yang hanya digunakan untuk komunikasi, diantaranya adalah tablet, laptop, atau perangkat umum lainnya dengan aplikasi atau fungsi penghasil ucapan (*Speech-generating devices*, n.d.). Tampilan pada perangkat dapat berbeda-beda jumlah ikon yang dapat ditampilkan. Hal ini bergantung pada kemampuan anak penyandang autis dalam menggunakan perangkat tersebut. *Speech generating* dapat digunakan sebagai alat terapi bagi anak penyandang autis karena membantu anak penyandang autis dalam berkomunikasi, karena dengan fitur *speech generating* dapat mengeluarkan kata ataupun kalimat. Contohnya adalah ketika menekan tombol dengan gambar apel, maka perangkat dapat mengeluarkan suara “saya ingin makan apel”. Hal ini tentu saja memudahkan anak penyandang autis karena keterbatasan dalam melakukan komunikasi dengan orang lain.

#### 2.1.2. *Finite state machine*

*Finite state machine* adalah sebuah metodologi perancangan sistem kontrol yang menggambarkan tingkah laku atau prinsip kerja sistem dengan menggunakan tiga hal berikut: *state* (Keadaan), *event* (kejadian) dan *action* (aksi) (Nedya, et.al, 2015). Pada satu periode waktu yang cukup signifikan, sistem akan berada dalam satu state yang aktif. Sistem dapat bertransisi menuju state lain jika mendapatkan masukan atau event tertentu, baik yang berasal dari sistem atau perangkat luar.

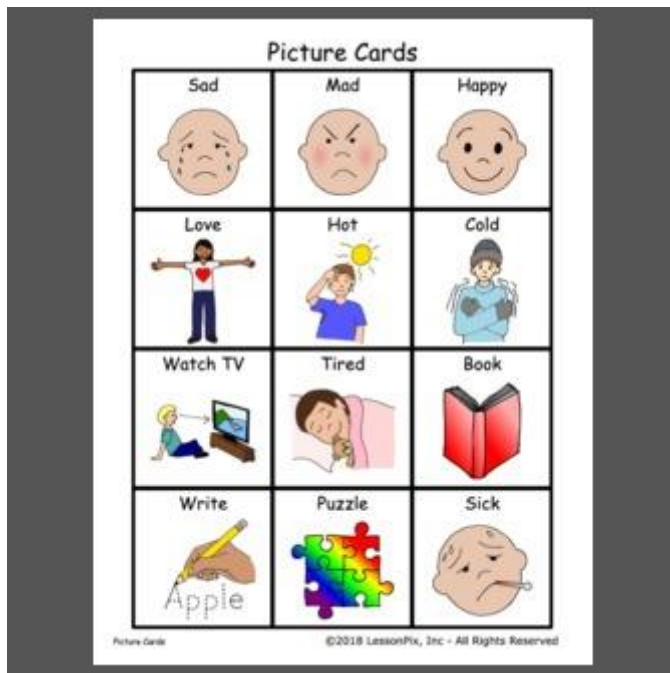


Gambar 2.1 Contoh diagram *state* sederhana

Diagram diatas merupakan contoh sederhana dari *finite state machine* dengan menggunakan 2 buah *state*, 2 buah *input* dan 4 buah *output*. Ketika pertama kali sistem dijalankan maka akan menuju "*State 0*", pada state ini sistem akan menghasilkan "*Action 1*" jika terjadi masukan "*Event 0*". Sedangkan apabila terjadi "*Event 1*" maka akan dilakukan "*Action 2*" dan bertransisi menuju "*State 1*" dan begitu seterusnya. Pada umumnya FSM banyak dimanfaatkan pada game untuk memberikan alur cerita yang berbeda-beda kepada pemain.

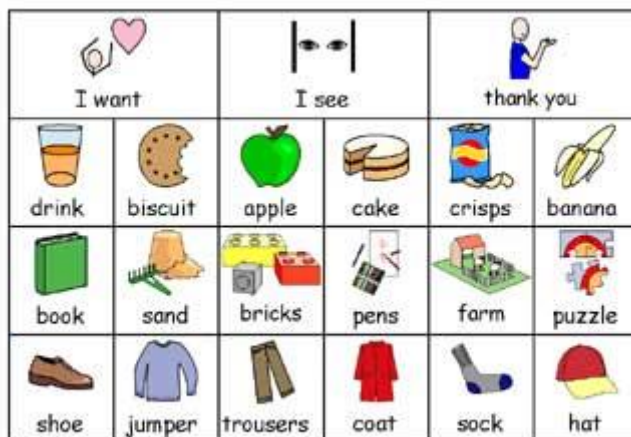
### 2.1.3. PECS

PECS merupakan singkatan dari *Picture Exchange Communication System*, sebuah metode yang dikembangkan oleh Andy Bondy dan Lori Frost pada tahun 1994 untuk mengajarkan kemampuan komunikasi fungsional pada anak autis (Vistasari & Patria, 2019). Metode PECS secara tradisional menggunakan kartu dengan gambar dan tulisan untuk mendeskripsikan gambar yang ada. Dengan menggunakan simbol PECS, anak penyandang autis menjadi lebih bebas untuk berkomunikasi dengan orang lain karena simbol atau gambar yang digunakan akan dengan mudah dipahami oleh orang lain.



Gambar 2.2 Ilustrasi untuk menggambarkan PECS

Sumber : Hanley, (n.d). *PECS Picture Cards*.



Gambar 2.3 Ilustrasi untuk menggambarkan PECS

Sumber : Holy Angels, (n.d). *P.E.C.S. (Picture Exchange Communication System)*.

Gambar yang ada pada metode PECS umumnya merupakan gambar dari kehidupan sehari-hari atau benda yang sering digunakan. Seperti pada yang ditunjukkan pada gambar 2.2 dan gambar 2.3. Menurut Pyramid (n.d.) dan National Autism Resource (n.d.), dalam metode PECS terdapat 6 tahapan yang dapat digunakan untuk mengajarkan komunikasi kepada anak. Tahapan-tahapan tersebut, bisa diuraikan sebagai berikut :

- 1) Tahap 1 – Cara berkomunikasi : Pada tahap ini anak penyandang autisme belajar untuk menukar gambar tunggal dengan barang atau aktivitas yang benar-benar mereka inginkan.
- 2) Tahap 2 – Jarak dan kegigihan : Pada tahap ini anak penyandang autisme masih menggunakan gambar tunggal, anak penyandang autisme belajar untuk menggeneralisasi keterampilan baru ini dengan menggunakannya ditempat yang berbeda, dengan lawan komunikasi yang berbeda.
- 3) Tahap 3 – Diskriminasi gambar : Pada tahap ini anak penyandang autisme belajar untuk memilih dua atau lebih gambar untuk menanyakan hal-hal favorit mereka. Gambar-gambar ini ditempatkan di buku komunikasi menggunakan velco strips sehingga gambar mudah dilepas.
- 4) Tahap 4 – Struktur kalimat : Pada tahap ini anak penyandang autisme belajar untuk menyusun kalimat sederhana pada strip kalimat yang dapat dilepas menggunakan gambar “saya ingin” diikuti dengan gambar barang yang diminta.
- 5) Tahap 5 – Menjawab pertanyaan : Pada tahap ini anak penyandang autisme belajar untuk menjawab pertanyaan “apa yang kamu inginkan?”
- 6) Tahap 6 – Mengomentari : Pada tahap ini anak penyandang autisme belajar untuk memberikan komentar untuk menanggapi pertanyaan “apa yang anda lihat?”, “apa yang anda dengar?”, “apa ini?”. Anak penyandang autisme belajar untuk membuat kalimat dengan “saya melihat”, “saya mendengar”, “ini adalah”.

#### **2.1.4. Autism Spectrum Disorder**

*Autism Spectrum Disorder* dapat dideteksi sejak dini. Gejala autis bisa mulai muncul pada waktu yang berbeda-beda pada setiap anak. Gejala bisa muncul pada saat anak berusia 1 tahun atau pada anak yang awalnya bertumbuh secara normal, kemudian kehilangan kemampuan berkomunikasi dan perilaku dari anak menjadi lebih agresif. Gejala pada anak penyandang autis dapat dengan mudah dikenali yaitu: kurangnya komunikasi atau interaksi sosial dengan orang lain dan juga perilaku anak yang cenderung berulang setiap harinya. Selain itu, pada saat berkomunikasi dengan orang lain, anak penyandang autis sulit melakukan kontak mata dengan lawan bicaranya bahkan cenderung menghindari tatapan mata dengan orang lain. Anak penyandang autis juga sulit untuk mengerti emosi atau perasaan orang lain. Pada umumnya penyandang autis tidak dapat berkomunikasi ataupun jika bisa, nada bicara

penyandang autisme akan seperti robot dan cenderung mengulangi kata atau kalimat dari orang lain serta tidak memahami arti dari kata tersebut (Tan, 2018).

## **2.2. Tinjauan Studi**

### **2.2.1. Game Edukasi Berbasis Kinect untuk Anak Berkebutuhan Khusus (Autis) dengan Metode Finite State Machine (Anugraha et al., 2018)**

#### **a. Permasalahan :**

Pada penelitian ini, peneliti ingin membuat sebuah game edukasi untuk membantu anak penyandang autisme. Anak penyandang autisme kesulitan untuk mengenal nama benda-benda yang ada di rumah. Oleh karena itu game yang dibuat oleh peneliti bertujuan untuk mengenalkan nama benda yang ada di rumah sehingga anak penyandang autisme dapat mengenal benda-benda di sekitar mereka. Peneliti memanfaatkan *finite state machine* untuk menentukan alur pengguna dalam menjawab suatu pertanyaan yang tampil dalam game. Selain itu, peneliti juga memanfaatkan kinect sebagai sistem inputan.

#### **b. Metode :**

Pada game yang dibuat, terdapat 3 *state* yang dituangkan dalam pilihan *level*, yaitu *level* mudah, sedang dan sulit. Contohnya, pengguna memilih *level* mudah, maka akan masuk ke tampilan memilih objek. Selain itu, di dalam game juga terdapat *timer* untuk membatasi waktu pengguna dalam menjawab pertanyaan yang dimunculkan. Ketika pengguna tidak berhasil menjawab pertanyaan hingga waktu habis maka akan kembali ke pertanyaan sebelumnya. Setiap *level* yang dibuat mempunyai pertanyaan yang berbeda-beda. Untuk *level* mudah, pengguna memilih objek yang dilihat. Untuk *level* sedang, pengguna melengkapi kata dan untuk *level* sulit, pengguna diminta untuk berhitung. Game dimulai dengan memilih *level* terlebih dahulu, jika pengguna menjawab semua pertanyaan dengan benar maka akan lanjut ke level berikutnya. Sedangkan jika salah menjawab maka akan kembali ke pertanyaan sebelumnya.

#### **c. Hasil :**

*Game* ini dapat berjalan maksimal dengan spesifikasi minimal RAM 4GB dan kecepatan processor lebih dari 2 Ghz. Kinect digunakan untuk menangkap sensor gerakan tangan kanan yang digunakan untuk kursor dalam menjalankan game. Dalam pengujian FPS dilakukan dalam jarak ideal antara kinect dengan pengguna sebesar 2,2 meter dan hasil rata-rata FPS terbaik dan terburuk yang didapat sebesar 113,2 fps dan 82,5 fps.

### 2.2.2. Penerapan metode finite state machine pada game “The Relationship” (Rahadian et al., 2016)

#### a. Permasalahan :

Industri game yang sudah maju menyuguhkan sebuah dunia maya yang baru yang terus berkembang untuk mencoba mendekati dunia nyata. Game dibuat untuk sarana menghibur diri saat penat. Penerapan *finite state machine* pada game berguna untuk menentukan berbagai macam respon NPC berdasarkan interaksi yang dilakukan oleh pemain. Hal ini disebabkan karena *finite state machine* dapat digunakan untuk mendesain dan menentukan respon perilaku yang dilakukan terhadap perubahan kondisi.

#### b. Metode :

*Finite state machine* yang diterapkan pada NPC berfungsi untuk memberikan respon yang berbeda saat menerima masukan dari pemain. Pada saat game dimulai, NPC akan bertransisi ke state *idle*. NPC akan bertransisi menuju state “berjalan” ketika NPC diatur untuk berjalan menuju arah tertentu hingga NPC berada pada posisi tujuan. Ketika NPC menyentuh pemain akan bertransisi menuju state “berbicara” yang akan menyesuaikan tingkat kedekatan NPC dengan pemain. Jika tingkat kedekatan masih “0” maka NPC akan bertransisi ke state “perkenalan”. Tingkat kedekatan antara “0” hingga “20” akan membuat NPC bertransisi menuju state “teman” dan berlanjut ke state “pertanyaan 1” dan “pertanyaan 2”. Jika tingkat kedekatan NPC berada pada 20 hingga 50 maka akan bertransisi menuju state “teman baik” dan berlanjut ke state “pertanyaan 3” dan “pertanyaan 4”. Sedangkan jika tingkat kedekatan diatas 50 maka akan bertransisi menuju state “teman dekat” dan berlanjut ke state “pertanyaan 5” dan “pertanyaan 6”. Ketika permainan sudah mencapai 30 hari maka NPC akan bertransisi menuju state “perasaan akhir”. Pada state ini pemain diharuskan untuk menyatakan perasaan kepada NPC untuk menentukan hasil akhir permainan.

#### c. Hasil :

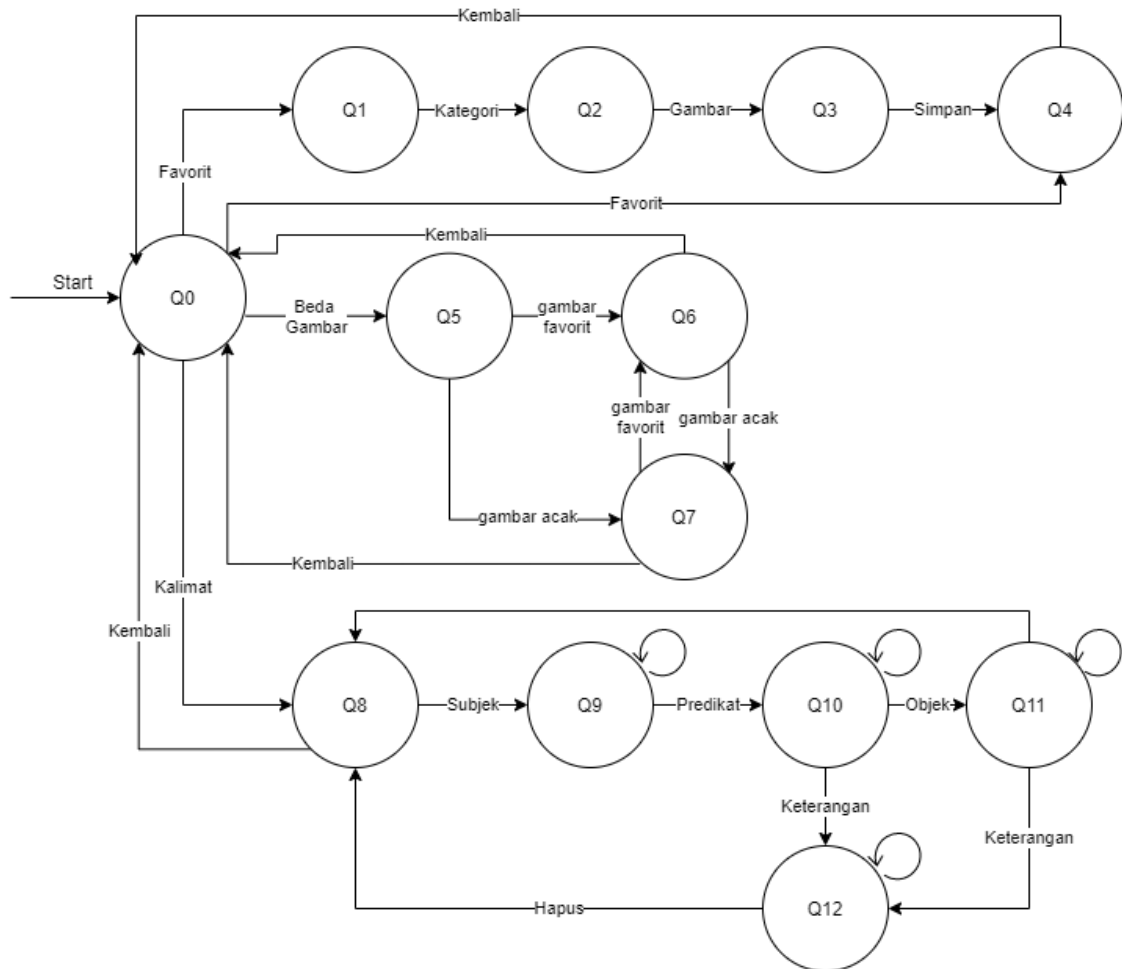
Hasil dari penelitian ini adalah *finite state machine* yang diterapkan dapat menghasilkan akhir permainan yang berbeda-beda menyesuaikan permainan yang dilakukan oleh pemain. Karena penerapan *finite state machine* pada permainan sehingga akhir cerita dari setiap pemain bisa berbeda-beda.



### 3. ANALISA DAN DESAIN SISTEM

#### 3.1. Analisa Sistem

##### 3.1.1. Finite State Machine



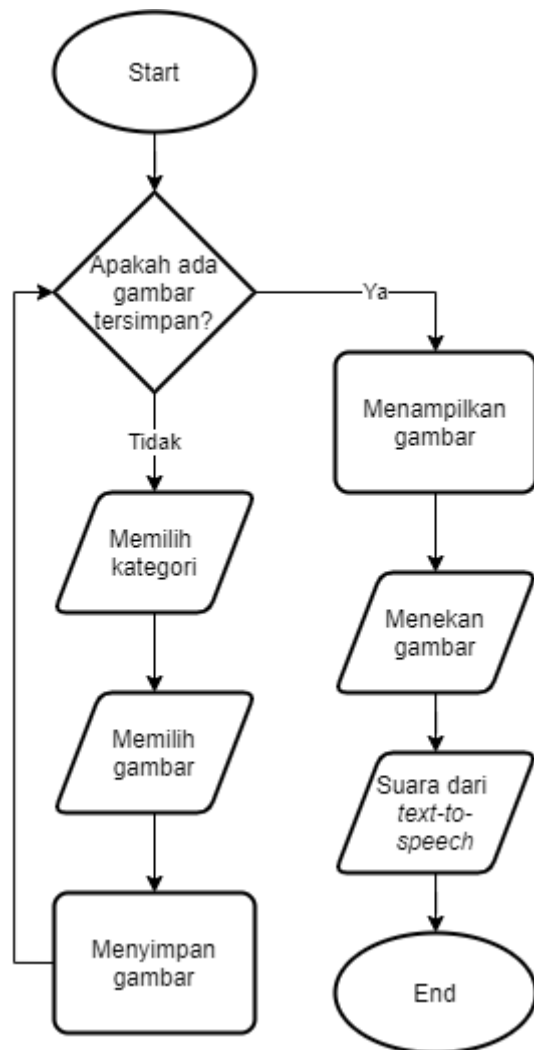
Gambar 3.1 Finite state machine

Penjelasan dari *finite state* diagram pada gambar 3.1 adalah sebagai berikut, Alur *finite state* dimulai dengan *state* Q0 terlebih dahulu. Apabila pengguna memilih menu favorit maka *state* akan berpindah ke *state* Q1. Pada *state* ini akan ditampilkan kategori-kategori yang dapat dipilih oleh pengguna. Setelah pengguna memilih kategori, selanjutnya *state* akan berpindah ke *state* Q2. Pada *state* ini akan ditampilkan gambar apa saja yang sesuai dengan kategori yang telah dipilih pada *state* sebelumnya. Setelah pengguna memilih gambar yang akan menjadi gambar favorit, *state* akan berpindah ke *state* Q3. Pada *state* Q3 ini, apabila pengguna menekan tombol simpan data maka akan menuju *state* Q4. Jika gambar telah disimpan maka setiap saat

pengguna menggunakan aplikasi dan memilih menu favorit, *state* akan otomatis berpindah ke *state* Q4. Apabila pengguna memilih menu beda gambar, maka *state* akan berpindah ke *state* Q5. Pada *state* ini, akan ditampilkan 2 gambar. 1 gambar berupa gambar favorit yang telah dipilih pada menu favorit sedangkan gambar lainnya merupakan gambar acak yang sesuai dengan kategori gambar favorit. *State* akan berpindah ke *state* Q6 apabila pengguna menekan gambar favorit dan akan mengeluarkan *output* dengan memanfaatkan *text to speech* untuk memberikan pernyataan jika gambar yang dipilih benar, sedangkan bila pengguna menekan gambar acak maka akan berpindah ke *state* Q7 dan akan mengeluarkan *output* memanfaatkan *text to speech* untuk memperingatkan pengguna apabila gambar yang dipilih salah. Jika pengguna memilih menu kalimat, maka *state* akan berpindah dari *state* Q0 menuju *state* Q8. Pada *state* ini, ditampilkan sebuah gambar yaitu gambar “saya mau”. Apabila pengguna menekan gambar tersebut maka *state* akan berpindah menuju *state* Q9. Pada *state* Q9 ini, akan ditampilkan gambar acak dari predikat. Pengguna dapat terus memilih gambar hingga menemukan gambar yang diinginkan. Apabila gambar predikat sudah dipilih maka akan berpindah menuju *state* Q10. Pada *state* ini akan dicek apakah predikat yang dipilih memerlukan objek atau tidak. Jika memerlukan objek maka akan ditampilkan gambar pilihan objek. Sedangkan jika gambar predikat yang dipilih tidak memerlukan objek maka akan ditampilkan gambar keterangan. Jika gambar predikat yang dipilih memerlukan objek, pengguna dapat terus memilih gambar hingga menemukan gambar objek yang diinginkan. Ketika gambar objek telah dipilih maka akan berpindah menuju *state* Q11. Pada *state* Q11 akan di munculkan gambar keterangan yang dapat dipilih oleh pengguna. Pengguna dapat memilih untuk mengakhiri kalimat atau menambahkan gambar keterangan sebagai pendukung untuk kalimat yang sedang dirangkai. Jika pengguna memilih untuk mengakhiri kalimat, maka *state* akan kembali ke *state* Q8, sedangkan jika pengguna memilih gambar keterangan, maka akan berpindah ke *state* Q12. Jika pengguna telah memilih gambar keterangan maka akan kembali ke *state* Q8. Akan tetapi jika predikat yang dipilih tidak memerlukan objek maka pada *state* Q10 akan langsung ditampilkan gambar keterangan. Jika gambar keterangan telah dipilih maka *state* akan berpindah menuju *state* Q12. Jika pengguna menekan icon “hapus” maka *state* akan kembali ke *state* Q8 dan akan kembali menampilkan gambar “saya mau” .

### 3.2. Flowchart

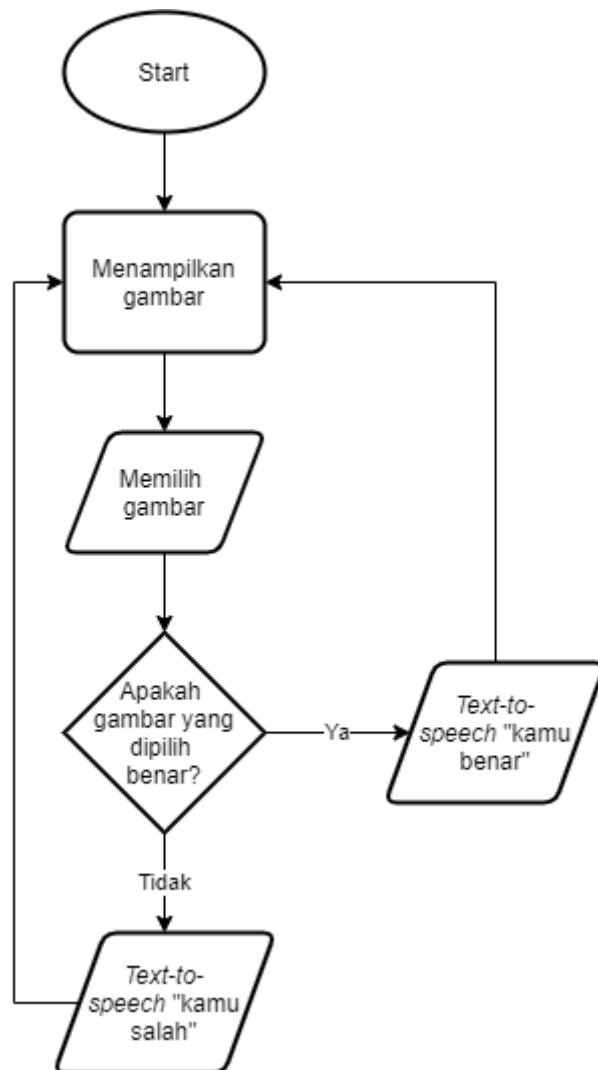
#### 3.2.1. Flowchart menu favorit



Gambar 3.2 Flowchart menu favorit

Penjelasan dari *flowchart* menu *favorit* yang ada pada gambar 3.2 adalah sebagai berikut, ketika pengguna menekan menu *favorit* pada halaman utama, sistem akan mengecek di dalam *database* apakah ada data gambar yang tersimpan atau tidak. Jika tidak ada gambar yang tersimpan maka pengguna dapat memilih kategori dari gambar yang akan dipilih. Setelah itu, pengguna dapat memilih gambar yang disukai dan menyimpan gambar tersebut kedalam *database*. Apabila gambar sudah tersimpan didalam *database*, maka setiap saat pengguna menekan menu *favorit* di halaman utama, akan langsung ditampilkan gambar yang telah dipilih sebelumnya.

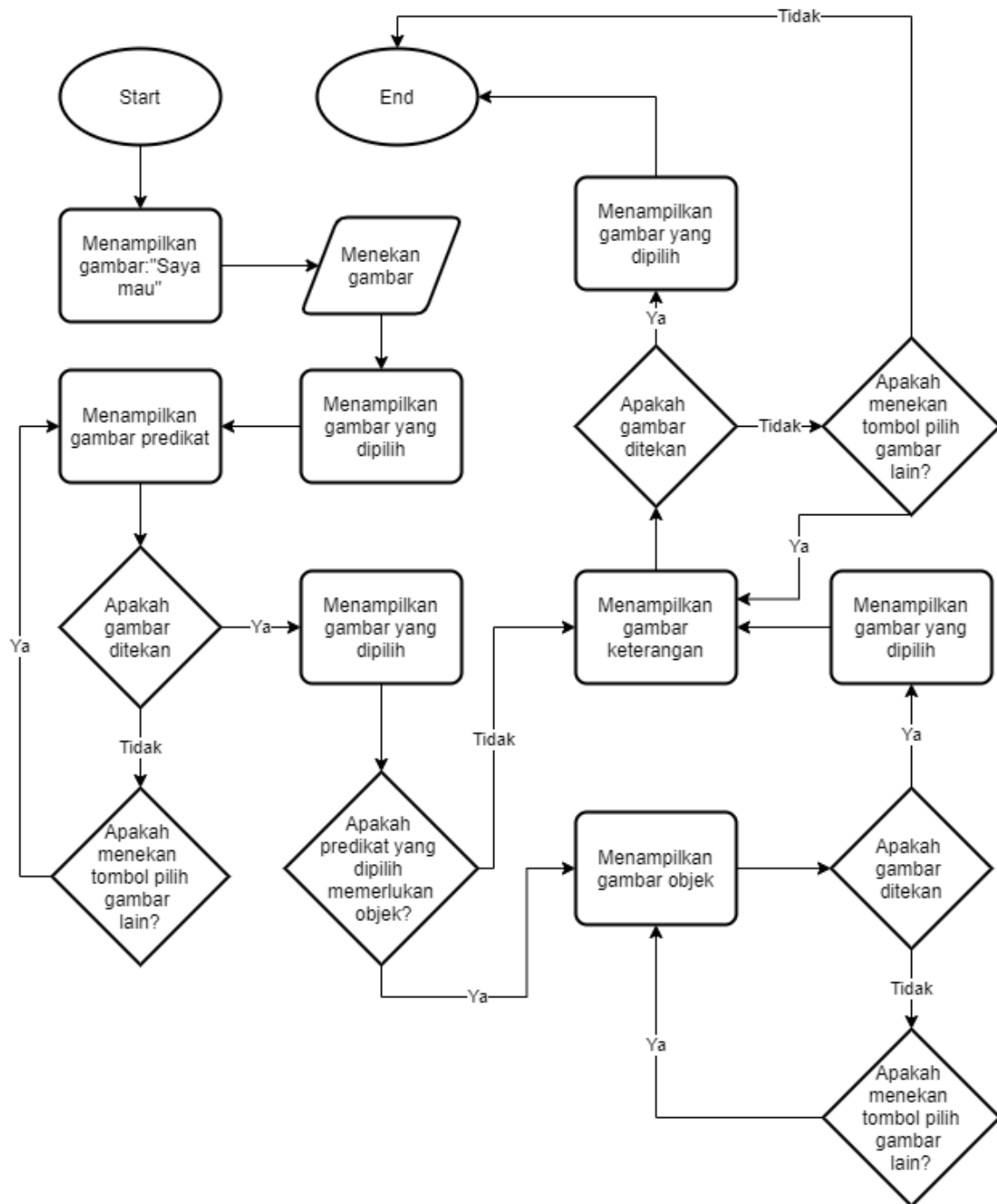
### 3.2.2. Flowchart menu beda gambar



Gambar 3.3 Flowchart menu beda gambar

Penjelasan dari flowchart menu beda gambar yang ada pada gambar 3.3 adalah sebagai berikut, saat pengguna menekan tombol “beda gambar” pada halaman utama maka akan di generate gambar berupa 1 gambar random dan 1 gambar *favorit* yang telah dipilih user sebelumnya. Apabila gambar yang dipilih bukan gambar *favorit* maka aplikasi akan mengeluarkan suara “kamu salah”. Akan tetapi apabila gambar yang dipilih merupakan gambar *favorit* maka aplikasi akan mengeluarkan suara “kamu benar”. Setiap gambar yang ditekan oleh pengguna, maka akan otomatis men-*generate* gambar random untuk ditampilkan.

### 3.2.3. Flowchart menu kalimat



Gambar 3.4 Flowchart menu komunikasi

Penjelasan dari *flowchart* menu komunikasi yang ada pada gambar 3.4 adalah sebagai berikut, pada saat pengguna menekan menu “komunikasi” pada halaman menu maka selanjutnya akan ditampilkan gambar “saya ingin” di bagian atas halaman tersebut. Ketika pengguna menekan gambar “saya ingin” selanjutnya gambar tersebut akan ditampilkan. Pada saat bersamaan, akan di *generate* gambar dengan kategori “predikat”. Apabila pengguna tidak

memilih gambar tersebut, pengguna dapat mengganti dengan gambar lain dengan cara menekan tombol “pilih gambar lain”. Setelah gambar kategori predikat dipilih maka selanjutnya gambar tersebut akan ditampilkan di bagian atas halaman tersebut. Pada saat gambar predikat telah dipilih maka akan di cek apakah predikat tersebut memerlukan objek atau tidak. Jika predikat yang dipilih memerlukan objek maka akan di *generate* gambar dengan kategori “objek”. Apabila gambar yang di *generate* dipilih oleh pengguna, maka gambar tersebut akan ditampilkan di bagian atas halaman. Akan tetapi jika pengguna ingin memilih gambar objek lainnya, pengguna dapat menekan tombol “pilih gambar lain”. Pada saat gambar yang dipilih ditampilkan, maka akan di *generate* gambar selanjutnya yaitu gambar dengan kategori keterangan untuk mendukung gambar-gambar yang telah dipilih sebelumnya. Pengguna dapat memilih apakah ingin menggunakan gambar dengan kategori keterangan atau tidak. Apabila pengguna ingin menggunakan gambar berkategori keterangan maka pengguna dapat memilih gambar yang diinginkan, sedangkan apabila pengguna tidak ingin memilih gambar dengan kategori keterangan maka tidak perlu menekan gambar keterangan.

### **3.3. Desain aplikasi**

Pada bagian ini akan dijelaskan mengenai desain yang dibuat pada aplikasi. Pada aplikasi terdapat beberapa halaman yaitu halaman menu utama, halaman *favorit*, halaman beda gambar, halaman komunikasi, halaman tambah gambar, halaman pengaturan, halaman gambar yang ditambahkan dan halaman untuk melihat gambar favorit.

#### **3.3.1. Halaman utama**

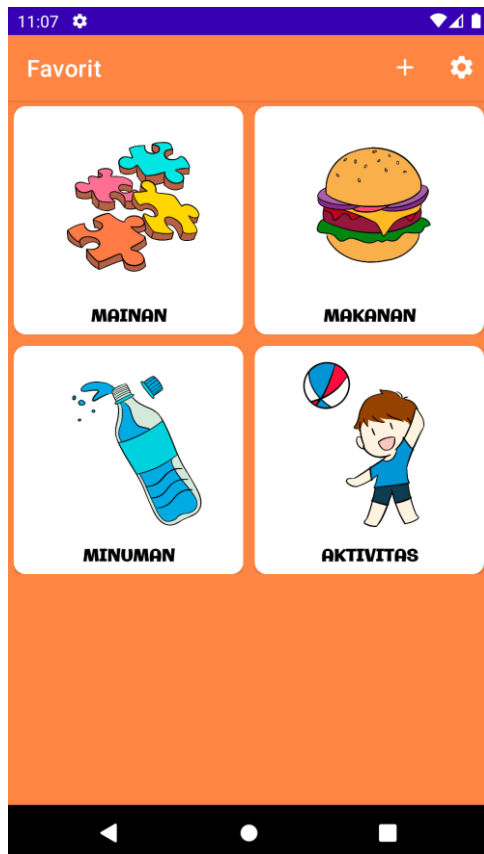
Gambar 3.5 merupakan tampilan awal dari aplikasi. Pada halaman ini terdapat 4 tombol yang digunakan untuk berpindah ke halaman lainnya. Untuk tombol tahap 1 digunakan untuk berpindah ke halaman tahap 1, sedangkan tombol tahap 2 digunakan untuk berpindah ke halaman tahap 2. Tombol tahap 3 digunakan untuk berpindah ke halaman tahap 3. Tombol pengaturan digunakan untuk berpindah ke halaman pengaturan.



Gambar 3.5 Tampilan halaman utama

### 3.3.2. Menu favorit

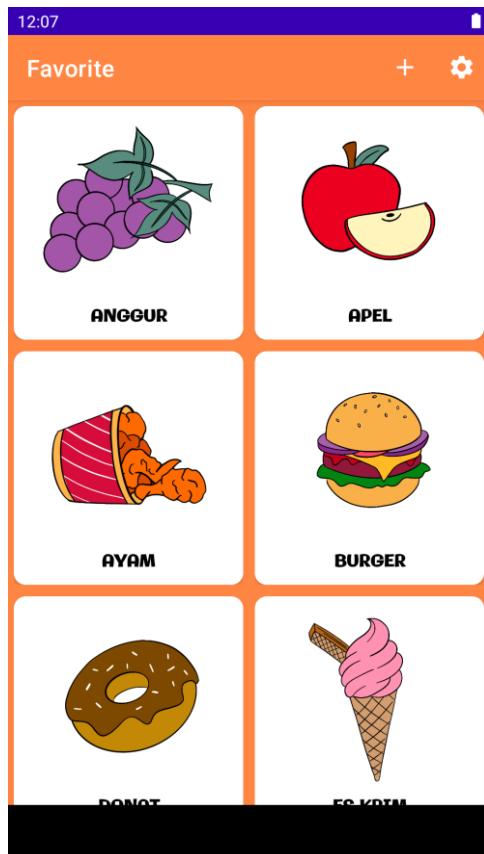
Gambar 3.6 merupakan tampilan halaman menu favorit. Halaman ini akan tampil setelah pengguna menekan tombol favorit pada halaman menu. Pada halaman ini pertama kali akan ditampilkan semua daftar kategori dari gambar yang ada. Pada bagian toolbar di atas terdapat 2 *icon* yang berfungsi untuk berpindah ke halaman lain. Untuk *icon* tambah digunakan untuk berpindah ke halaman tambah gambar. Sedangkan *icon* pengaturan menuju halaman pengaturan.



Gambar 3.6 Tampilan memilih kategori

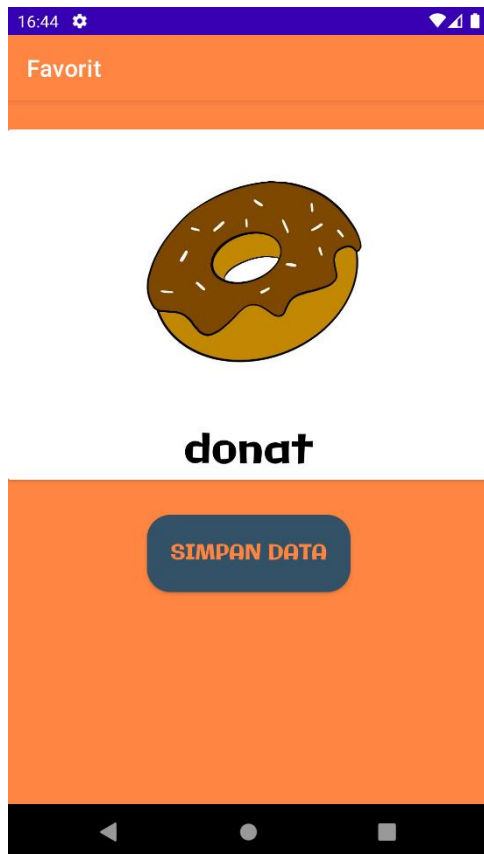
Setelah pengguna memilih kategori, maka daftar gambar akan berubah sesuai dengan kategori yang dipilih. Misalkan pengguna memilih kategori "makanan", maka tampilan selanjutnya adalah daftar gambar dari makanan. Seperti yang ditampilkan di gambar 3.7





Gambar 3.7 Tampilan memilih gambar favorit

Ketika pengguna telah memilih gambar, maka tampilan akan menampilkan gambar yang telah dipilih sebelumnya dan terdapat tombol untuk menyimpan gambar tersebut. Seperti yang ditampilkan di gambar 3.8



Gambar 3.8 Tampilan menyimpan gambar favorit

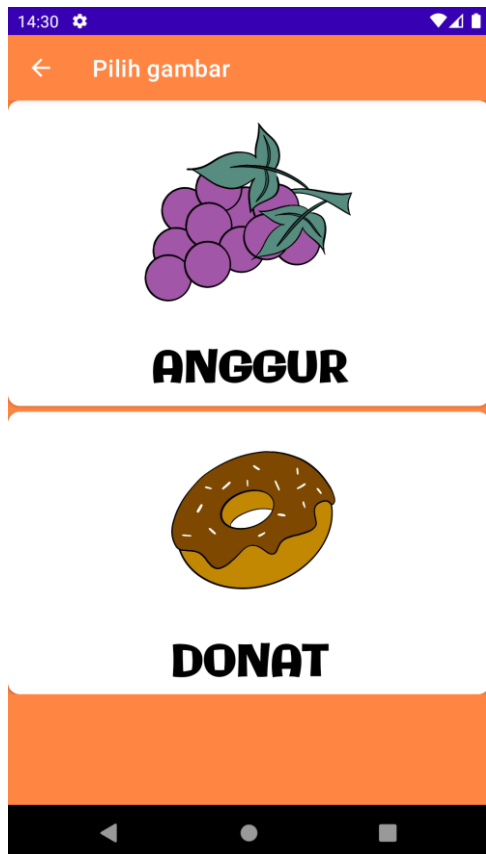
Apabila data gambar *favorit* telah tersimpan maka setiap saat user menekan tombol menu *favorit* maka akan langsung ditampilkan gambar *favorit* yang telah dipilih sebelumnya. Apabila anak penyandang autisme menekan gambar tersebut maka akan menghasilkan suara yang sesuai dengan text yang ada di gambar. Tampilan menu *favorit* bisa dilihat pada gambar 3.9.



Gambar 3.9 Tampilan gambar favorit

### 3.3.3. Menu beda gambar

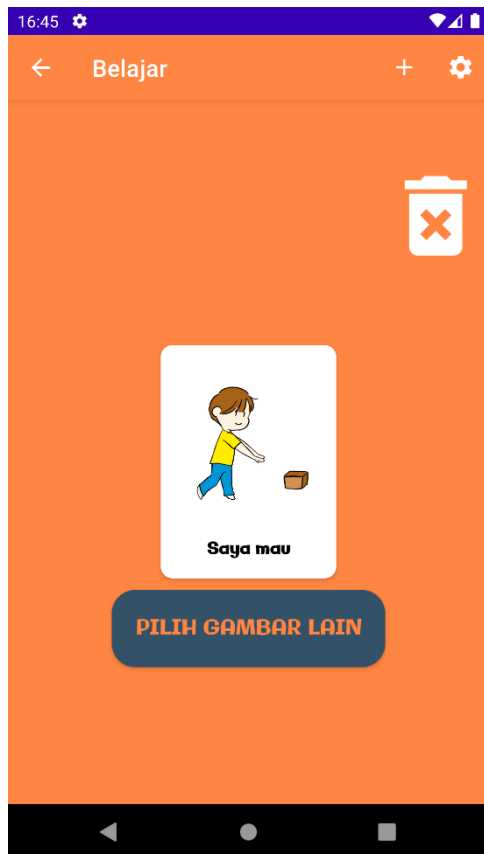
Menu beda gambar berfungsi untuk melatih anak penyandang autisme dalam memilih gambar yang diinginkan. Oleh karena itu di menu beda gambar disediakan 2 gambar, dimana salah satu gambar merupakan gambar favorit yang telah dipilih sebelumnya. Sedangkan gambar lainnya merupakan gambar yang ditampilkan secara acak. Apabila anak penyandang autisme memilih gambar yang salah maka aplikasi akan mengeluarkan suara bahwa gambar yang dipilih salah. Tampilan halaman beda gambar dapat dilihat pada gambar 3.10.



Gambar 3.10 Tampilan halaman beda gambar

#### 3.3.4. Menu kalimat

Tampilan awal menu komunikasi dapat dilihat pada gambar 3.11. Gambar yang ditampilkan pada halaman ini adalah gambar “saya ingin”. Selain itu, pada halaman ini juga terdapat *icon delete* dan tombol pilih gambar lain. *Icon delete* berfungsi untuk menghapus gambar yang telah dipilih, sedangkan tombol pilih gambar lain berfungsi untuk mengganti pilihan gambar yang ditampilkan. Jika pengguna telah menekan gambar saya ingin, maka gambar akan ditampilkan diatas seperti yang ditunjukkan oleh gambar 3.12. Selanjutnya akan di *generate* gambar predikat. Jika gambar predikat telah dipilih maka akan di *generate* gambar objek seperti yang dapat dilihat pada gambar 3.13. Setelah gambar objek telah dipilih maka akan ditampilkan seperti yang dapat dilihat pada gambar 3.14. Setelah gambar objek dipilih, pengguna dapat mengakhiri kalimat yang telah dirangkai dengan menekan *icon delete* atau menambahkan gambar keterangan dengan memilih gambar keterangan. Jika pengguna menekan *icon delete* maka gambar akan dihapus dan menampilkan kembali gambar saya ingin.



Gambar 3.11 Tampilan awal halaman komunikasi



Gambar 3.12 Tampilan setelah menekan gambar "saya mau"



Gambar 3.13 Tampilan setelah menekan gambar predikat

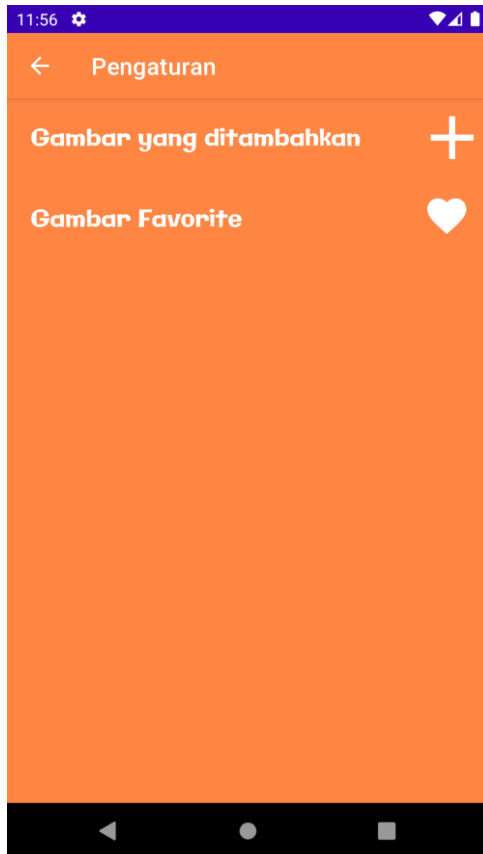


Gambar 3.14 Tampilan setelah menekan gambar objek

### 3.3.5. Halaman pengaturan

Tampilan halaman pengaturan dapat dilihat pada gambar 3.15. Pada halaman ini terdapat 2 menu yang digunakan untuk melihat gambar apa saja yang sudah ditambah oleh pengguna kedalam aplikasi dan gambar favorit yang telah ditambahkan melalui tahap 1.

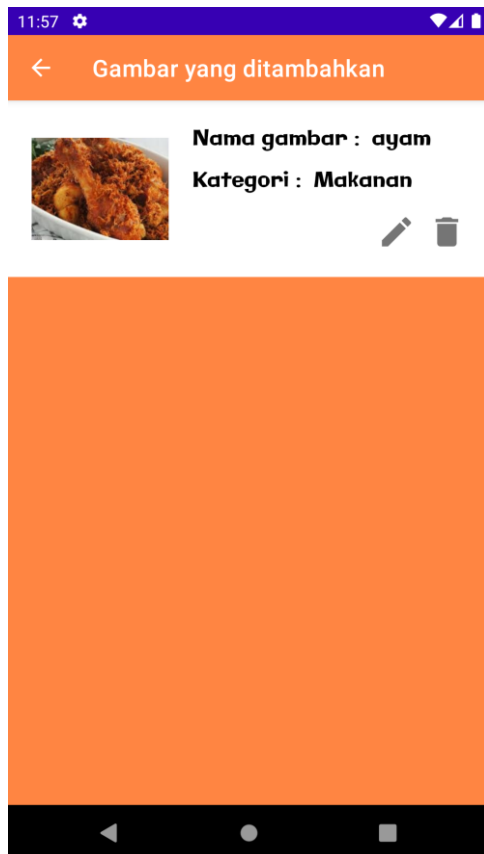




Gambar 3.15 Tampilan halaman pengaturan

### 3.3.6. Halaman gambar yang ditambah

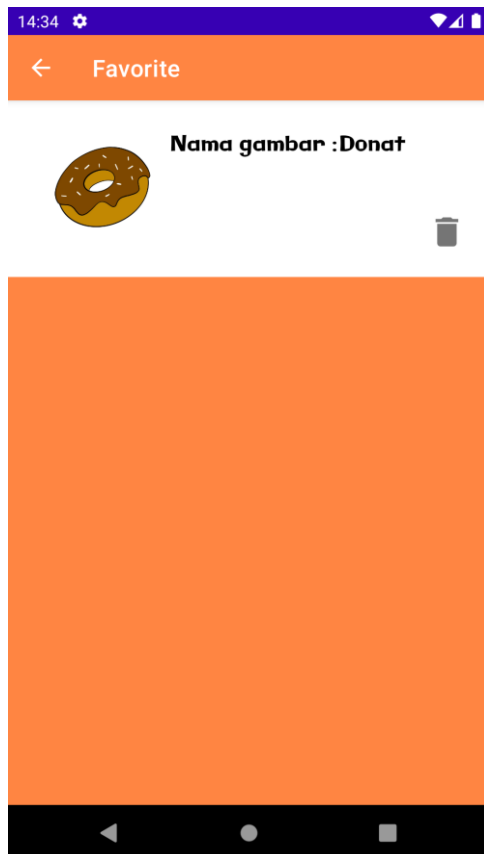
Tampilan halaman gambar yang telah ditambah dapat dilihat pada gambar 3.16. Halaman ini berfungsi untuk menampilkan gambar apa saja yang telah ditambahkan oleh pengguna ke dalam aplikasi. Selain itu pengguna juga dapat merubah nama, gambar, atau kategori dari gambar tersebut. Gambar juga dapat dihapus dengan menekan *icon* tempat sampah



Gambar 3.16 Tampilan halaman yang ditambah

### 3.3.7. Halaman gambar favorit

Tampilan halaman gambar favorit dapat dilihat pada gambar 3.17. Halaman ini hanya digunakan untuk menghapus gambar favorit yang telah ditambahkan melalui tahap 1.



Gambar 3.17 Tampilan halaman gambar favorit

### 3.3.8. Halaman Tambah gambar

Halaman tambah gambar digunakan untuk menambahkan gambar secara lokal yang dilakukan oleh pengguna. Pengguna dapat memilih gambar yang ada di galeri *handphone* untuk ditambahkan ke dalam aplikasi dengan cara menekan tombol pilih gambar. Setelah menekan tombol tersebut, maka otomatis pengguna akan berpindah ke galeri *handphone* dan memilih gambar yang ingin diinginkan. Setelah memilih gambar, pengguna dapat memberikan nama dari gambar tersebut. Pengguna dapat menambahkan jenis kategori untuk gambar yang dipilih dengan menekan *spinner* yang ada di atas tombol simpan. Ketika pengguna menekan tombol simpan maka otomatis gambar akan tersimpan di database lokal. Tampilan halaman tambah gambar dapat dilihat pada gambar 3.17.



Gambar 3.18 Tampilan halaman tambah gambar

## 4. IMPLEMENTASI SISTEM

Pada bab ini akan dibahas implementasi sistem yang sesuai dengan analisa dan desain sistem yang telah dibuat pada bab sebelumnya. Implementasi berupa pemrograman aplikasi Android.

### 4.1. Implementasi menu favorit

#### 4.1.1. Implementasi memilih kategori

Untuk dapat memilih gambar favorit, pengguna harus memilih kategori terlebih dahulu. Untuk *code* memilih kategori dapat dilihat pada segmen 4.1.

Segmen 4.1 Memilih kategori

```
private fun SiapkanDataawal(){
    dtkategori = resources.getStringArray(R.array.kategorithp1)
    dtkategorigambar = resources.obtainTypedArray(R.array.kategorigbrthp1)
}
private fun tambahdataawal() {
    for (position in dtkategori.indices){
        val data = tahapawalpertamaa(
            dtkategorigambar.getResourceId(position, -1),
            dtkategori[position]
        )
        arData.add(data)
    }
}
private fun tampildataawal(){
    recviewatahapawal.layoutManager = GridLayoutManager(applicationContext, 2,
    LinearLayoutManager.VERTICAL, false)
    adaptertahapawalpertama = adaptertahapawalpertama(arData)
    recviewatahapawal.adapter = adaptertahapawalpertama
}
```

#### 4.1.2. Menampilkan data gambar sesuai kategori

Untuk dapat menampilkan data gambar yang sesuai dengan kategori, dilakukan pengiriman data kategori pada saat pengguna menekan gambar kategori yang akan dipilih. Setelah data diterima, maka akan di cek terlebih dahulu data tersebut sesuai dengan kategori apa. Lalu gambar akan ditampilkan sesuai dengan kategori tersebut. Untuk *code* dapat dilihat pada segmen 4.2.

Segmen 4.2 Menampilkan data sesuai kategori

```
dataTerima = intent.getStringExtra(extraText)
if (dataTerima.toString().equals("Mainan")){
    kategori = "Mainan"
    dttahapkedua = resources.getStringArray(R.array.Mainan_text)
    dtgambartahapkedua = resources.obtainTypedArray(R.array.Mainan)
    for (position in dttahapkedua.indices){
        val data = gambar(
            dtgambartahapkedua.getResourceId(position, -1),
            dttahapkedua[position],
        )
        arData.add(data)
    }
    recviewtahapawalkedua.layoutManager = GridLayoutManager(applicationContext, 2,
    LinearLayoutManager.VERTICAL, false)
    adaptertahapawalkedua = adaptertahapawalkedua(arData)
    roomadapter = roomadapter(arrayListOf())
    val concatAdapter = ConcatAdapter(adaptertahapawalkedua, roomadapter)
    recviewtahapawalkedua.adapter = concatAdapter
}else if (dataTerima.toString().equals("Makanan")){
    kategori = "Makanan"
    dttahapkedua = resources.getStringArray(R.array.Makanan_text)
    dtgambartahapkedua = resources.obtainTypedArray(R.array.Makanan)
    for (position in dttahapkedua.indices){
```

```

        val data = gambar(
            dtgambartahapkedua.getResourceId(position, -1),
            dttahapkedua[position],
        )
        arData.add(data)
    }
    recviewtahapawalkedua.layoutManager = GridLayoutManager(applicationContext, 2,
    LinearLayoutManager.VERTICAL, false)
    adaptertahapawalkedua = adaptertahapawalkedua(arData)
    roomadapter = roomadapter(arrayListOf())

    val concatAdapter = ConcatAdapter(adaptertahapawalkedua, roomadapter)
    recviewtahapawalkedua.adapter = concatAdapter
} else if (dataTerima.toString().equals("Minuman")){
    kategoris = "Minuman"
    dttahapkedua = resources.getStringArray(R.array.Minuman_text)
    dtgambartahapkedua = resources.obtainTypedArray(R.array.Minuman)
    for (position in dttahapkedua.indices){
        val data = gambar(
            dtgambartahapkedua.getResourceId(position, -1),
            dttahapkedua[position],
        )
        arData.add(data)
    }
    recviewtahapawalkedua.layoutManager = GridLayoutManager(applicationContext, 2,
    LinearLayoutManager.VERTICAL, false)
    adaptertahapawalkedua = adaptertahapawalkedua(arData)
    roomadapter = roomadapter(arrayListOf())

    val concatAdapter = ConcatAdapter(adaptertahapawalkedua, roomadapter)
    recviewtahapawalkedua.adapter = concatAdapter
}

```

```

else if (dataTerima.toString().equals("Aktivitas")){
    kategoris = "Aktivitas"
    dttahapkedua = resources.getStringArray(R.array.KataKerja_text)
    dtgambartahapkedua = resources.obtainTypedArray(R.array.KataKerja)
    for (position in dttahapkedua.indices){
        val data = gambar(
            dtgambartahapkedua.getResourceId(position, -1),
            dttahapkedua[position],
        )
        arData.add(data)
    }
    recviewtahapawalkedua.layoutManager = GridLayoutManager(applicationContext, 2,
    LinearLayoutManager.VERTICAL, false)
    adaptertahapawalkedua = adaptertahapawalkedua(arData)
    roomadapter = roomadapter(arrayListOf())

    val concatAdapter = ConcatAdapter(adaptertahapawalkedua, roomadapter)
    recviewtahapawalkedua.adapter = concatAdapter

```

#### 4.1.3. Menyimpan data

Pada tahap ini, gambar yang telah dipilih akan ditampilkan dan disimpan ke dalam *room database* dan memanfaatkan *shared preference* sehingga pengguna tidak perlu memilih gambar favorit setiap menggunakan aplikasi. Untuk *code* menyimpan data dapat dilihat pada segmen 4.3.

##### Segmen 4.3 Menyiapkan data

```

val image = (gbrtahaptigarec.getDrawable() as BitmapDrawable).bitmap
val stream = ByteArrayOutputStream()
image.compress(Bitmap.CompressFormat.JPEG, 100, stream)
val imageInByte: ByteArray = stream.toByteArray()

```



```

val data = gambar(
    dtgambartahapketiga.getResourceId(idgambar!!, -1),
    tulistahaptigarec.text.toString()
)
arTemp.add(data)

sharedPref = getSharedPreferences(sharedPrefName, Context.MODE_PRIVATE)
val editor: SharedPreferences.Editor = sharedPref.edit()
editor.putString(keyData, arTemp.toString())
editor.apply()

CoroutineScope(Dispatchers.IO).launch {
    DB.DataDao().tambahdata1(
        DataKesukaan(
            1,
            tulistahaptigarec.text.toString(),
            imageInByte,
            datakategori.toString()
        )
    )
}

```

#### 4.1.4. Menampilkan data favorit

Data favorit yang ditampilkan didapatkan dari data yang telah tersimpan di *room database*. Untuk *code* menampilkan data favorit bisa dilihat pada segmen 4.4.

Segmen 4.4 Menampilkan data favorit

```

CoroutineScope(Dispatchers.IO).launch {
    val user = DB.DataDao().getAllData1()
    Log.d("abc", user.toString())
    withContext(Dispatchers.Main){

```

```

        adaptertahapempat.isiData(user)
    }
}
adaptertahapempat = adaptertahapempat(arrayListOf(),mTTS)
recviewtahap4.layoutManager = LinearLayoutManager(this)
recviewtahap4.adapter = adaptertahapempat

```

#### 4.2. Implementasi menu beda gambar

Pada menu beda gambar, akan ditampilkan 2 gambar yang berbeda. Salah satu gambar merupakan gambar favorit yang telah dipilih pada menu favorit, sedangkan satu gambar lainnya merupakan gambar acak yang sesuai dengan kategori dari gambar favorit. Untuk *code* beda gambar dapat dilihat pada segmen 4.5.

Segmen 4.5 Menu beda gambar

```

fun tampildata(){
    getdata()
    if (kgr.equals("Mainan")){
        ktgr = "Mainan"
        val randomInt = random.nextInt(dtmainangambar.length())
        val drawableID: Int = dtmainangambar.getResourceId(randomInt, -1)
        gambarmain.setImageResource(drawableID)
        tulisanmain.setText(dtmainan[randomInt])
        cardviewmain.setOnClickListener{
            ktgr = "Mainan"
            if (tulisanmain.text.toString() == tulisanmain2.text.toString()){
                Toast.makeText(this, "Benar", Toast.LENGTH_SHORT).show()
                mTTS.speak("Kamu Benar", TextToSpeech.QUEUE_FLUSH, null)
                val randomInt = random.nextInt(dtmainangambar.length())
                val drawableID: Int = dtmainangambar.getResourceId(randomInt, -1)
                gambarmain.setImageResource(drawableID)
                tulisanmain.setText(dtmainan[randomInt])
            }
        }
    }
}

```

```

    }else{
        Toast.makeText(this, "Salah", Toast.LENGTH_SHORT).show()
        mTTS.speak("Kamu salah pilih", TextToSpeech.QUEUE_FLUSH, null)
        val randomInt = random.nextInt(dtmainangambar.length())
        val drawableID: Int = dtmainangambar.getResourceId(randomInt, -1)
        gambarmain.setImageResource(drawableID)
        tulisanmain.setText(dtmainan[randomInt])
    }

}

}else if (kgr.equals("Makanan")){
    ktgr = "Makanan"
    val randomInt = random.nextInt(dtmakanangambar.length())
    val drawableID: Int = dtmakanangambar.getResourceId(randomInt, -1)
    gambarmain.setImageResource(drawableID)
    tulisanmain.setText(dtmakanan[randomInt])
    cardviewmain.setOnClickListener{
        ktgr = "Makanan"
        if (tulisanmain.text.toString() == tulisanmain2.text.toString()){
            Toast.makeText(this, "Benar", Toast.LENGTH_SHORT).show()
            mTTS.speak("Kamu Benar", TextToSpeech.QUEUE_FLUSH, null)
            val randomInt = random.nextInt(dtmakanangambar.length())
            val drawableID: Int = dtmakanangambar.getResourceId(randomInt, -1)
            gambarmain.setImageResource(drawableID)
            tulisanmain.setText(dtmakanan[randomInt])
        }else{
            Toast.makeText(this, "Salah", Toast.LENGTH_SHORT).show()
            mTTS.speak("Kamu salah pilih", TextToSpeech.QUEUE_FLUSH, null)
            val randomInt = random.nextInt(dtmakanangambar.length())
            val drawableID: Int = dtmakanangambar.getResourceId(randomInt, -1)
            gambarmain.setImageResource(drawableID)
            tulisanmain.setText(dtmakanan[randomInt])
        }
    }
}

```

```

    }

    }

}

else if (kgr.equals("Aktivitas")){
    ktgr = "Aktivitas"

    val randomInt = random.nextInt(dtaktifitasgambar.length())
    val drawableID: Int = dtaktifitasgambar.getResourceId(randomInt, -1)
    gambarmain.setImageResource(drawableID)
    tulisanmain.setText(dtaktifitas[randomInt])
    cardviewmain.setOnClickListener{
        ktgr = "Aktivitas"
        if (tulisanmain.text.toString() == tulisanmain2.text.toString()){
            Toast.makeText(this, "Benar", Toast.LENGTH_SHORT).show()
            mTTS.speak("Kamu Benar", TextToSpeech.QUEUE_FLUSH, null)
            val randomInt = random.nextInt(dtaktifitasgambar.length())
            val drawableID: Int = dtaktifitasgambar.getResourceId(randomInt, -1)
            gambarmain.setImageResource(drawableID)
            tulisanmain.setText(dtaktifitas[randomInt])
        }else{
            Toast.makeText(this, "Salah", Toast.LENGTH_SHORT).show()
            mTTS.speak("Kamu salah pilih", TextToSpeech.QUEUE_FLUSH, null)
            val randomInt = random.nextInt(dtaktifitasgambar.length())
            val drawableID: Int = dtaktifitasgambar.getResourceId(randomInt, -1)
            gambarmain.setImageResource(drawableID)
            tulisanmain.setText(dtaktifitas[randomInt])
        }
    }

}

}

else if (kgr.equals("Minuman")){
    ktgr = "Minuman"

```

```

val randomInt = random.nextInt(dtminumgambar.length())
val drawableID: Int = dtminumgambar.getResourceId(randomInt, -1)
gambarmain.setImageResource(drawableID)
tulisanmain.setText(dtminum[randomInt])
cardviewmain.setOnClickListener{
    ktgr = "Minuman"
    if (tulisanmain.text.toString() == tulisanmain2.text.toString()){
        Toast.makeText(this, "Benar", Toast.LENGTH_SHORT).show()
        mTTS.speak("Kamu Benar", TextToSpeech.QUEUE_FLUSH, null)
        val randomInt = random.nextInt(dtminumgambar.length())
        val drawableID: Int = dtminumgambar.getResourceId(randomInt, -1)
        gambarmain.setImageResource(drawableID)
        tulisanmain.setText(dtminum[randomInt])
    }else{
        Toast.makeText(this, "Salah", Toast.LENGTH_SHORT).show()
        mTTS.speak("Kamu salah pilih", TextToSpeech.QUEUE_FLUSH, null)
        val randomInt = random.nextInt(dtminumgambar.length())
        val drawableID: Int = dtminumgambar.getResourceId(randomInt, -1)
        gambarmain.setImageResource(drawableID)
        tulisanmain.setText(dtminum[randomInt])
    }
}
}else{
    val randomInt = random.nextInt(dtgambar.length())
    val drawableID: Int = dtgambar.getResourceId(randomInt, -1)
    gambarmain.setImageResource(drawableID)
    tulisanmain.setText(dtnama[randomInt])
    cardviewmain.setOnClickListener{
        if(tulisanmain.text.toString() == tulisanmain2.text.toString()){
            Toast.makeText(this, "Benar", Toast.LENGTH_SHORT).show()
            mTTS.speak("Kamu Benar", TextToSpeech.QUEUE_FLUSH, null)
            val randomInt = random.nextInt(dtgambar.length())

```

```

        val drawableID: Int = dtgambar.getResourceId(randomInt, -1)
        gambarmain.setImageResource(drawableID)
        tulisanmain.setText(dtnama[randomInt])
    }else{
        Toast.makeText(this, "Salah", Toast.LENGTH_SHORT).show()
        mTTS.speak("Kamu salah pilih", TextToSpeech.QUEUE_FLUSH, null)
        val randomInt = random.nextInt(dtgambar.length())
        val drawableID: Int = dtgambar.getResourceId(randomInt, -1)
        gambarmain.setImageResource(drawableID)
        tulisanmain.setText(dtnama[randomInt])
    }
}
}
}
fun getdata(){

    arrfavorite.forEach{
        val imgViewer: ImageView = findViewById<View>(R.id.gambarmain2) as ImageView
        val bitmap = BitmapFactory.decodeByteArray(it.Gambar, 0, it.Gambar.size)
        imgViewer.setImageBitmap(bitmap)
        tulisanmain2.setText(it>Nama)
        kgr = it.kategori
    }

    carviewmain2.setOnClickListener {
        Toast.makeText(this, "Benar", Toast.LENGTH_SHORT).show()
        mTTS.speak("Kamu Benar", TextToSpeech.QUEUE_FLUSH, null)
        if (kgr.equals("Makanan")){
            ktgr = "Makanan"
            val randomInt = random.nextInt(dtmakanangambar.length())
            val drawableID: Int = dtmakanangambar.getResourceId(randomInt, -1)
            gambarmain.setImageResource(drawableID)

```

```

        tulisanmain.setText(dtmakanan[randomInt])
    }else if (kgr.equals("Minuman")){
        ktgr = "Minuman"
        val randomInt = random.nextInt(dtminumgambar.length())
        val drawableID: Int = dtminumgambar.getResourceId(randomInt, -1)
        gambarmain.setImageResource(drawableID)
        tulisanmain.setText(dtminum[randomInt])
    }else if(kgr.equals("Aktivitas")){
        ktgr = "Aktivitas"
        val randomInt = random.nextInt(dtaktifitasgambar.length())
        val drawableID: Int = dtaktifitasgambar.getResourceId(randomInt, -1)
        gambarmain.setImageResource(drawableID)
        tulisanmain.setText(dtaktifitas[randomInt])
    }else if (kgr.equals("Mainan")){
        ktgr = "Mainan"
        val randomInt = random.nextInt(dtmainangambar.length())
        val drawableID: Int = dtmainangambar.getResourceId(randomInt, -1)
        gambarmain.setImageResource(drawableID)
        tulisanmain.setText(dtmainan[randomInt])
    }
}
}

```

#### 4.3. Implementasi menu kalimat

Pada tahap ini, gambar yang akan ditampilkan pertama adalah gambar “saya mau”. Ketika pengguna telah menekan gambar “saya mau” maka selanjutnya akan ditampilkan gambar acak dari gambar predikat. Ketika pengguna sudah memilih gambar predikat maka selanjutnya akan dicek apakah gambar predikat yang telah dipilih tersebut memerlukan objek atau tidak. Jika memerlukan objek maka selanjutnya akan di tampilkan gambar acak dari objek, sedangkan apabila predikat yang dipilih tidak memerlukan objek maka akan ditampilkan gambar acak keterangan. Untuk *code* menu kalimat dapat dilihat pada segmen 4.6.

#### Segmen 4.6 Menu kalimat

```
fun tampildata() {
    var randomInt: Int = 0
    gambarmain.setImageResource(R.drawable.saya_mau)
    tulisanmain.setText("Saya mau")
    cardviewmain.setOnClickListener {
        mTTS.speak(tulisanmain.text.toString(),
TextToSpeech.QUEUE_FLUSH, null)
        if (ktgr.equals("")) {
            ktgr = "Sayaingin"
            Log.d("kategori", ktgr)
            arTemp.add(main(R.drawable.saya_mau,
gbruser,tulisanmain.text.toString(), ktgr, false))
            randomInt =
randomGenerator.nextInt(dtpredikatgambar.length() + dataaaa.size)
            if (randomInt < dtpredikatgambar.length()){
                drawableID =
dtpredikatgambar.getResourceId(randomInt, -1)
                gambarmain.setImageResource(drawableID)
                tulisanmain.setText(dtpredikat[randomInt])
                isData = false
            }else{
                randomdata(ktgr)
                isData = true
            }
        } else if (ktgr.equals("Sayaingin")) {
            ktgr = "predikat"
            tulisan = tulisanmain.text.toString()
            if (isData == true){
//                aruser.add(Data(userid, tulisan, gbruser,
ktgr))
                arTemp.add(main(0,gbruser, tulisan, ktgr, true))
            }else{
                arTemp.add(main(drawableID, gbruser ,tulisan,
ktgr, false))
            }
            if (tulisan.equals("makan")){
                Log.v("stevencheck", "Masuk makan")
                ktgr = "predikat"
```



```

        ktkrj = "Makan"
        randomInt =
randomGenerator.nextInt(dtmakanangambar.length() +
arraymakanan.size)
        if (randomInt < dtmakanangambar.length()){
            drawableID =
dtmakanangambar.getResourceId(randomInt, -1)
            gambarmain.setImageResource(drawableID)
            tulisanmain.setText(dtmakanan[randomInt])
            isData = false
            ktgr = "predikat"
        }else{
            randompredikat(ktkrj)
            isData = true
            ktgr = "predikat"
        }
    }else if (tulisan.equals("minum")){
        Log.v("stevencheck", "Masuk Minum")
        ktgr = "predikat"
        ktkrj = "Minum"
        randomInt =
randomGenerator.nextInt(dtminumangambar.length() +
arrayminuman.size)
        if (randomInt < dtminumangambar.length()){
            drawableID =
dtminumangambar.getResourceId(randomInt, -1)
            gambarmain.setImageResource(drawableID)
            tulisanmain.setText(dtminum[randomInt])
            isData = false
            ktgr = "predikat"
        }else{
            randompredikat(ktkrj)
            isData = true
            ktgr = "predikat"
        }
    } else if (tulisan.equals("bermain")){
        Log.v("stevencheck", "Masuk Bermain")
        ktgr = "predikat"
        ktkrj = "Bermain"
    }
}

```

```

        randomInt =
randomGenerator.nextInt(dtmainangambar.length() + arraymainan.size)
        if (randomInt < dtmainangambar.length()){
            drawableID =
dtmainangambar.getResourceId(randomInt, -1)
            gambarmain.setImageResource(drawableID)
            tulisanmain.setText(dtmainan[randomInt])
            isData = false
            ktgr = "predikat"
        }else{
            randompredikat(ktkrj)
            isData = true
            ktgr = "predikat"
        }
    } else if (tulisan.equals("menonton") ||
tulisan.equals("tidur") || tulisan.equals("berdoa")
|| tulisan.equals("duduk") || tulisan.equals("jalan") || tulisan.equals("
masak") || tulisan.equals("menari") || tulisan.equals("menggambar")) {
        Log.v("stevencheck", "Masuk Bebas")
        ktgr = "objek"
        randomInt =
randomGenerator.nextInt(dtketerangangambar.length())
        if (randomInt < dtketerangangambar.length()){
            drawableID =
dtketerangangambar.getResourceId(randomInt, -1)
            gambarmain.setImageResource(drawableID)
            tulisanmain.setText(dtketerangan[randomInt])
            isData = false
            ktgr = "objek"
        }
    }
} else{
    ktgr = "predikat"
    Log.v("stevencheck", "Masuk kosong")
    randomInt =
randomGenerator.nextInt(dtobjekgambar.length() + arrayobjek.size)
    if (randomInt < dtobjekgambar.length()){
        drawableID =
dtobjekgambar.getResourceId(randomInt, -1)

```

```

        gambarmain.setImageResource(drawableID)
        tulisanmain.setText(dtobjek[randomInt])
        isData = false
    }else{
        randomdata(ktgr)
        isData = true
    }
}
} else if (ktgr.equals("predikat")) {
    ktgr = "objek"
    tulisan = tulisanmain.text.toString()
    if (isData == true){
        arTemp.add(main(0,gbruser, tulisan, ktgr, true))
    }else{
        arTemp.add(main(drawableID, gbruser, tulisan,
ktgr, false))
    }
    randomInt =
randomGenerator.nextInt(dtketerangangambar.length())
    randomInt < dtketerangangambar.length()
    drawableID =
dtketerangangambar.getResourceId(randomInt, -1)
    gambarmain.setImageResource(drawableID)
    tulisanmain.setText(dtketerangan[randomInt])
}
else if (ktgr.equals("objek")){
    ktgr = "Keterangan"
    tulisan = tulisanmain.text.toString()
    arTemp.add(main(drawableID,gbruser, tulisan,
ktgr, false))
}
adapterkomunikasi = adapterkomunikasi(arTemp, mTTS)
Log.d("masuk", arTemp.toString())
rvtahapmain.layoutManager = LinearLayoutManager(this,
LinearLayoutManager.HORIZONTAL, false)
rvtahapmain.adapter = adapterkomunikasi
}
nextbtn.setOnClickListener {
    if (ktgr.equals("Sayaingin")){

```

```

        randomInt=
randomGenerator.nextInt(dtpredikatgambar.length() + dataaaa.size)
        Log.d("random1",
getcountbykategori("Aktivitas").toString())
        Log.d("random", dtpredikatgambar.length().toString())
        if (randomInt < dtpredikatgambar.length()){
            drawableID =
dtpredikatgambar.getResourceId(randomInt, -1)
            gambarmain.setImageResource(drawableID)
            tulisanmain.setText(dtpredikat[randomInt])
            isData = false
        }else{
            randomdata("Aktivitas")
            isData = true
        }
    } else if (ktgr.equals("predikat")){
        if (tulisan.equals("makan")){
            Log.v("stevencheck", "Masuk next Makan")
            ktkrj = "Makan"
            randomInt =
randomGenerator.nextInt(dtmakanangambar.length() +
arraymakanan.size)
            if (randomInt < dtmakanangambar.length()){
                drawableID =
dtmakanangambar.getResourceId(randomInt, -1)
                gambarmain.setImageResource(drawableID)
                tulisanmain.setText(dtmakanan[randomInt])
                isData = false
            }else{
                randompredikat(ktkrj)
                isData = true
                ktkrj = "Makan"
            }
        }else if (tulisan.equals("minum")){
            Log.v("stevencheck", "Masuk next Minum")
            ktkrj = "Minum"
            randomInt =
randomGenerator.nextInt(dtminumangambar.length() +
arrayminuman.size)

```

```

        if (randomInt < dtminumangambar.length()){
            drawableID =
dtminumangambar.getResourceId(randomInt, -1)
            gambarmain.setImageResource(drawableID)
            tulisanmain.setText(dtminum[randomInt])
            isData = false
        }else{
            randompredikat(ktkrj)
            isData = true
            ktkrj = "Minum"
        }
    } else if (tulisan.equals("bermain")){
        Log.v("stevencheck", "Masuk next Bermain")
        ktkrj = "Bermain"
        randomInt =
randomGenerator.nextInt(dtmainangambar.length() + arraymainan.size)
        if (randomInt < dtmainangambar.length()){
            drawableID =
dtmainangambar.getResourceId(randomInt, -1)
            gambarmain.setImageResource(drawableID)
            tulisanmain.setText(dtmainan[randomInt])
            isData = false
        }else{
            randompredikat(ktkrj)
            isData = true
            ktkrj = "Bermain"
        }
    }
}
else{
    Log.v("stevencheck", "Masuk next Kosong")
    randomInt=
randomGenerator.nextInt(dtobjekgambar.length() + arrayobjek.size)
    if (randomInt < dtobjekgambar.length()){
        drawableID =
dtobjekgambar.getResourceId(randomInt, -1)
        gambarmain.setImageResource(drawableID)
        tulisanmain.setText(dtobjek[randomInt])
        isData = false
    }else{

```

```

        randomdata(ktgr)
        isData = true
    }
}
}else if (ktgr.equals("objek")){
    randomInt =
randomGenerator.nextInt(dtketerangangambar.length())
    randomInt < dtketerangangambar.length()
    drawableID =
dtketerangangambar.getResourceId(randomInt, -1)
    gambarmain.setImageResource(drawableID)
    tulisanmain.setText(dtketerangan[randomInt])
}
}

```

#### 4.4. Implementasi tambah gambar

##### 4.4.1. Fitur pilih gambar

Fitur ini digunakan untuk memilih gambar yang ada pada galeri *smartphone* pengguna. Untuk *code* pilih gambar dapat dilihat pada segmen 4.7

Segmen 4.7 Pilih gambar

```

plhgbr.setOnClickListener {
    val intent = Intent()
    intent.type = "image/*"
    intent.action = Intent.ACTION_GET_CONTENT
    startActivityForResult(Intent.createChooser(intent, "Select Picture"),
PICK_IMAGE_REQUEST)
}

```

#### 4.4.2. Fitur tambah gambar

Fitur tambah gambar digunakan untuk memasukkan gambar yang telah dipilih oleh pengguna dan kemudian dimasukkan ke dalam *room database*. Untuk *code* tambah gambar dapat dilihat pada segmen 4.8.

Segmen 4.8 Tambah gambar

```
tbhgbr.setOnClickListener {
    val image = (imageView2.getDrawable() as BitmapDrawable).bitmap
    val stream = ByteArrayOutputStream()
    image.compress(Bitmap.CompressFormat.JPEG, 50, stream)
    val imageInByte: ByteArray = stream.toByteArray()

    CoroutineScope(Dispatchers.IO).launch {
        DB.DataDao().tambahData(
            Data(0, inputnama.text.toString(), imageInByte, kategori.selectedItem.toString())
        )
    }
    startActivity(Intent(this, menuutama::class.java))
    Log.d("masuk", "berhasil ditambahkan")
    Toast.makeText(this, "Data Berhasil Ditambahkan", Toast.LENGTH_LONG).show()
}
```

#### 4.5. Daftar gambar yang ditambah

Fitur ini digunakan untuk melihat gambar apa saja yang telah ditambahkan oleh pengguna ke dalam aplikasi. Pada fitur ini juga terdapat menu untuk menghapus gambar yang telah ditambahkan. Untuk *code* dapat dilihat pada segmen 4.9.

Segmen 4.9 Daftar gambar yang telah ditambah

```
CoroutineScope(Dispatchers.IO).launch {
    val user = DB.DataDao().getAllData()
```

```

Log.d("abc", user.toString())
withContext(Dispatchers.Main){
    adaptergambarlokal.isiData(user)
}
}
adaptergambarlokal = adaptergambarlokal(arrayListOf())
recviewlokal.layoutManager = LinearLayoutManager(this)
recviewlokal.adapter = adaptergambarlokal
adaptergambarlokal.listener = this
CoroutineScope(Dispatchers.IO).launch {
    DB.DataDao().deleteData(dataData)
    Log.d("hapus", "Berhasil dihapus")
}

```

#### 4.6. Implementasi *text-to-speech*

*Text-to-speech* dimanfaatkan untuk menghasilkan suara ketika gambar ditekan oleh pengguna. Untuk *code* dapat dilihat pada segmen 4.10.

##### Segmen 4.10 *Text-to-speech*

```

mTTS = TextToSpeech(applicationContext, TextToSpeech.OnInitListener { status ->
    if (status != TextToSpeech.ERROR) {
        var result = mTTS.setLanguage(Locale("id", "ID"))
        if (result == TextToSpeech.LANG_MISSING_DATA || result ==
TextToSpeech.LANG_NOT_SUPPORTED) {
            Log.e("TTS", "This Language is not supported")
            val installIntent = Intent()
            installIntent.action = TextToSpeech.Engine.ACTION_INSTALL_TTS_DATA
            startActivity(installIntent)
        }
    } else {
        Log.d("masuk3", "Local")
    }
}

```



```
}  
})  
mTTS.speak(tulisanmain.text.toString(), TextToSpeech.QUEUE_FLUSH, null)
```

## **5. PENGUJIAN SISTEM**

Pada bab ini akan dijelaskan mengenai pengujian terhadap aplikasi yang telah dibuat. Pengujian aplikasi dilakukan agar dapat memastikan aplikasi yang dibuat dapat membantu anak penyandang autisme berkomunikasi. Proses pengujian dilakukan di Pusat Terapis Autis ACT Surabaya. Pengujian dilakukan selama 5 hari dan diikuti oleh 4 orang anak penyandang autisme. Dari ke 4 anak ini, salah satu diantaranya merupakan anak penyandang autisme non-verbal dengan kemampuan pendengaran yang kurang. Sedangkan 3 anak lainnya merupakan anak penyandang autisme non-verbal.

### **5.1. Efektivitas aplikasi**

Aplikasi dapat dikatakan efektif apabila anak penyandang autisme dapat mengoperasikan aplikasi secara mandiri dan menyampaikan keinginannya terhadap orang tua atau lawan komunikasinya dengan baik.

#### **5.1.1. Kemampuan anak mengoperasikan aplikasi**

Pengujian hari pertama diikuti oleh 2 orang anak penyandang autisme, anak pertama berinisial AB yang berusia 4,5 tahun dan anak kedua berinisial RH yang berusia 5,5 tahun. Pada saat melakukan pengujian di hari pertama, dilakukan tahap pengenalan aplikasi terhadap anak penyandang autisme. Hal ini bertujuan agar anak dapat mengoperasikan aplikasi dengan baik dan benar. Oleh karena itu, kemampuan anak untuk mengoperasikan aplikasi pada pengujian di hari pertama masih sangat minim. Anak cenderung menekan tombol secara acak dan memerlukan bimbingan dari pembimbing. Untuk menu favorit, anak hanya tertarik untuk melihat gambar yang ditampilkan. Pada saat gambar sudah ditampilkan, anak akan menekan tombol kembali dan memilih gambar lainnya lagi untuk dilihat. Oleh karena itu untuk menu favorit, tidak ada gambar yang disimpan sebagai gambar favorit oleh anak. Pada menu beda gambar, anak hanya senang mendengarkan suara yang dihasilkan pada saat gambar ditekan. Sedangkan untuk menu kalimat, 2 anak tersebut hanya asal menekan gambar tanpa melihat gambar apa yang ditekan. Kedua anak tersebut juga mengabaikan bantuan verbal maupun fisik yang diberikan oleh pembimbing sehingga pengujian hari pertama tidak mendapatkan hasil.

Pengujian hari kedua diikuti oleh anak yang berbeda karena jadwal masuk terapi yang berbeda setiap anaknya. Di hari kedua ini pengujian diikuti oleh 2 orang anak penyandang

autisme yaitu, HB yang berusia 7 tahun dan AB yang juga telah mengikuti uji coba pada hari pertama. Pada uji coba kedua ini, AB sudah menunjukkan perkembangan dalam menggunakan aplikasi tersebut. AB telah dapat menyimpan gambar favorit, dari yang sebelumnya hanya melihat-lihat gambar yang ada di aplikasi. Meskipun masih membutuhkan bantuan verbal dari pembimbingnya sebanyak 4x. Akan tetapi untuk menu beda gambar masih belum ada perkembangan karena AB belum dapat membedakan gambar yang telah dipilih pada menu favorit dan gambar lainnya yang ditampilkan. AB hanya senang mendengarkan suara yang dihasilkan ketika menekan gambar-gambar tersebut. Sedangkan untuk menu kalimat, AB belum dapat memilih gambar lainnya dan juga menghapus gambar yang telah dipilih sehingga AB terus menekan gambar yang dimunculkan tanpa menekan tombol pilih gambar lainnya. HB memiliki kemampuan mengoperasikan aplikasi lebih baik dibandingkan dengan RH dan juga AB. Pada saat HB melakukan uji coba pertama, HB sudah dapat menyimpan gambar favorit yang benar-benar HB inginkan. Untuk menu kalimat, HB sudah dapat menyusun kalimat sesuai dengan apa yang HB inginkan. Ketika uji coba dilakukan, HB terus-menerus menyusun kalimat “Saya mau bermain”. Lalu ketika pembimbing HB bertanya “HB ingin bermain apa?”, HB dapat mencari gambar objek permainan yang dia inginkan. Lalu HB menekan gambar objek “boneka”. Setelah itu pembimbing dari HB mengizinkan HB untuk mengambil boneka yang berada di ruangan lainnya. Sedangkan untuk menu beda gambar, kemampuan HB masih sama dengan AB dan RH. Pada pengujian ini HB mampu mengoperasikan aplikasi sebanyak 3x dan membutuhkan bantuan dari pembimbing sebanyak 4x. Ketiga anak ini cenderung hanya senang mendengarkan suara yang dihasilkan ketika mereka menekan gambar-gambar tersebut.

Pengujian hari ke 3 rencananya diikuti oleh 4 anak, akan tetapi karena RH berhalangan untuk mengikuti uji coba akhirnya hanya diikuti oleh AB, HB dan JB. Untuk AB pada uji coba ke 3 ini, belum ada perkembangan dalam menggunakan aplikasi. AB masih cenderung asal menekan gambar tanpa memperhatikan gambar-gambar tersebut. Untuk menu favorit AB sudah dapat menyimpan gambar seperti pada pengujian sebelumnya. AB masih senang mendengarkan suara ketika menekan gambar. Pada pengujian ini AB masih memerlukan bantuan dari pembimbingnya sebanyak 5x Sedangkan untuk HB, sudah bisa menggunakan aplikasi secara mandiri meskipun terkadang masih membutuhkan bantuan verbal dari pembimbingnya. Pada saat melakukan uji coba HB dapat mengoperasikan aplikasi secara mandiri sebanyak 17x dan mengoperasikan dengan bantuan verbal sebanyak 9x. Pada uji coba ini merupakan uji coba pertama bagi JB yang berusia 14 tahun dengan kemampuan pendengaran

yang kurang. Pada saat melakukan uji coba, JB sudah mampu mengoperasikan aplikasi secara mandiri sebanyak 2x dan mengoperasikan aplikasi dengan bantuan verbal sebanyak 4x.

Pengujian hari ke 4 diikuti oleh AB dan HB. Pada uji coba ke 4 ini, AB sudah menunjukkan perkembangan karena mampu mengoperasikan aplikasi secara mandiri sebanyak 3x untuk menu kalimat dan dengan bantuan verbal sebanyak 5x. Pada uji coba sebelumnya, AB belum mampu untuk menghapus gambar yang telah dipilih sedangkan pada uji coba ke 4 ini sudah bisa menghapus gambar-gambar yang telah dipilih. Untuk HB, pada uji coba ke 4 ini sudah tidak menggunakan bantuan verbal untuk mengoperasikan aplikasi. HB mengoperasikan aplikasi secara mandiri sebanyak 7x.

Pengujian hari ke 5 diikuti oleh AB dan JB. Pada pengujian ini, AB sudah bisa mengoperasikan aplikasi secara mandiri sebanyak 8x dan dengan bantuan verbal sebanyak 3x. Akan tetapi untuk menu beda gambar tidak ada perkembangan karena AB hanya senang mendengar suara yang dihasilkan. Sedangkan untuk JB, pada pengujian ini menggunakan aplikasi secara mandiri sebanyak 4x dan dengan bantuan verbal sebanyak 2x.

Inisial	Usia	Pengujian ke	Mandiri	Bantuan
AB	4.5 Tahun	1	0	0
		2	0	4
		3	0	5
		4	3	5
		5	8	3

Tabel 5.1 Pengujian responden AB

Inisial	Usia	Pengujian ke	Mandiri	Bantuan
RH	5.5 Tahun	1	0	0
		2		
		3		
		4		
		5		

Tabel 5.2 Pengujian responden RH

Inisial	Usia	Pengujian ke	Mandiri	Bantuan
HB	7 Tahun	1	3	4
		2	17	9
		3	7	0
		4		
		5		

Tabel 5.3 Pengujian responden HB

Initial	Usia	Pengujian ke	Mandiri	Bantuan
JB	14 Tahun	1	2	4
		2	4	2
		3	4	2
		4		
		5		

Tabel 5.4 Pengujian responden JB

## 5.2. Finite state machine

Untuk pengujian *finite state machine* dilakukan dengan melihat gambar yang ditekan oleh anak, apakah memenuhi S-P-O-K atau tidak. Pada pengujian hari pertama, *finite state* terpenuhi karena anak menekan gambar hingga ke gambar keterangan. Pada pengujian hari kedua, *finite state* juga masih terpenuhi karena 4 gambar yang dipilih oleh anak. 4 gambar tersebut terdiri dari gambar subjek, predikat, objek dan keterangan. Pada saat melakukan uji coba kepada Habibie di hari ketiga, *finite state* tidak berjalan maksimal karena HB terkadang hanya menekan gambar subjek, predikat dan objek, bahkan terkadang hanya menekan gambar subjek dan predikat. Sedangkan untuk AB masih terpenuhi hingga ke *state* Q12. Pada saat pengujian dilakukan kepada JB, *finite state* tidak selalu berjalan hingga ke *state* Q12 karena JB hanya memenuhi kondisi dari *state* Q9 dan *state* Q10. Pada saat berada di *state* Q10, gambar yang telah dipilih sebelumnya dihapus sehingga *state* kembali kepada *state* Q8.

## 5.3. Survei kepada terapis atau orang tua

Pada penelitian ini juga dilakukan survei kepada orang tua atau terapis yang mendampingi anak di ACT Surabaya. Survei dilakukan dengan bertanya kepada orang tua atau terapis apakah aplikasi yang dibuat membantu atau memudahkan mereka untuk berkomunikasi dengan anak penyandang autisme. Dari hasil survei yang dilakukan, didapatkan hasil bahwa aplikasi yang dibuat membantu atau lebih memudahkan orang tua atau terapis untuk mengerti keinginan anak. Dengan anak menekan gambar yang sudah disediakan berdasarkan aktifitas harian mereka, orang tua atau terapis sudah mengetahui keinginan anak. Hal ini juga cukup menekan stress anak karena apa yang mereka inginkan dapat dengan mudah dimengerti oleh orang tua atau terapis. Meskipun terkadang anak masih memerlukan bantuan verbal untuk memberikan jawaban terhadap apa yang mereka inginkan.

## 6. KESIMPULAN DAN SARAN

Bab ini berisikan tentang kesimpulan dan saran dari hasil uji coba aplikasi terhadap anak penyandang autisme dengan memanfaatkan *finite state machine*.

### 6.1. Kesimpulan

Dari hasil uji coba yang telah dilakukan kepada beberapa anak penyandang autisme, dapat ditarik kesimpulan, antara lain:

- Pengenalan aplikasi kepada anak penyandang autisme membutuhkan waktu yang berbeda-beda tergantung dari kemampuan retorika dan motorik setiap anak.
- *Finite state* yang digunakan didalam aplikasi sudah mampu untuk membantu anak dalam merangkai kalimat.
- Aplikasi yang dibuat sudah dapat digolongkan membantu anak penyandang autisme non-verbal untuk menyampaikan keinginannya.

### 6.2. Saran

Saran yang dapat diberikan untuk mengembangkan aplikasi ini lebih lanjut adalah :

- Menambahkan lebih banyak fitur yang dapat membantu anak penyandang autisme.
- Kalimat yang dapat dirangkai tidak hanya sebatas saya mau.

## 7. DAFTAR REFERENSI

- Anugraha, D. M., Agustina, I., & Fauziah. (2018, January). *Game Edukasi Berbasis Kinect untuk Anak Berkebutuhan Khusus (Autis) dengan Metode Finite State Machine*. Vol. 3, No. 1. 10.31328/jointecs.v3i1.501
- Autism Resource Center (n.d). *What is Autism?* <https://www.autism.org.sg/living-with-autism/what-is-autism>
- Bors, M. L. (2018, March 11). *What is a Finite State Machine?* <https://medium.com/@mlbors/what-is-a-finite-state-machine-6d8dec727e2c>
- Center of Disease Control and Prevention (n.d). *What is Autism Spectrum Disorder?* <https://www.cdc.gov/ncbddd/autism/facts.html>
- Hanley T (n.d). *PECS Picture Cards. LessonPix custom Learning Material*. <https://lessonpix.com/materials/5420632/PECS+Picture+Cards>
- Holy Angels (n.d). *P.E.C.S. (Picture Exchange Communication System)*. <https://www.laholyangels.org/resident-life/additional-habilitation-services/p-e-c-s/>
- Mobile Operating System Market Share Worldwide Oct 2019 - Oct 2020*. (2020, October). <https://gs.statcounter.com/os-market-share/mobile/worldwide>
- National Autism Resources (n.d). *The Picture Exchange Communication System (PECS)*. <https://nationalautismresources.com/the-picture-exchange-communication-system-pecs/>
- Nendya M. B., Gunanto S. G., dan Santosa R. g., (2015, October 2). *Pemetaan Perilaku Non-Playable Character Pada Permainan Berbasis Role Playing Game Menggunakan Metode Finite State Machine*. <https://doi.org/10.24821/jags.v1i2.1304>
- Park, H. R., et. al, (2016, February 29). *A Short Review on the Current Understanding of Autism Spectrum Disorders*. <https://doi.org/10.5607/en.2016.25.1.1>
- Pyramid Educational Consultant (n.d). *Picture Communication Exchange System*. <https://pecsusa.com/pecs/>
- Rahadian, M. F., Suyatno, A., & Maharani, S. (2016, February 1). *Penerapan Metode Finite State Machine Pada Game "The Relationship"*, Vol. 11 No. 1, pp. 14-22. 10.30872/jim.v11i1.198
- Setiawan, I. 2006. *Perancangan Software Embedded System Berbasis FSM*. Semarang: Universitas Diponegoro

- Speech-generating devices.* (n.d.). <https://raisingchildren.net.au/autism/therapies-guide/speech-generating-devices>
- Tan, L. (2018, November 26). *Gangguan Spektrum Autisme.* <https://www.sehatq.com/penyakit/gangguan-spektrum-autisme>
- Vicker, B. (n.d), *What is the Picture Exchange Communication System or PECS?* <https://www.iidc.indiana.edu/irca/articles/what-is-the-picture-exchange-communication-system-or-pecs.html>
- Vistasari, R., & Patria, B. (2019, May 5). *Program PECS (Picture Exchange Communication System) untuk Meningkatkan Kemampuan Berbicara Terstruktur pada Anak Autis, Volume 5, No 1*, pp. 94-107. 10.22146/gamajpp.48590



## **LAMPIRAN**

### **Lampiran 1 kuisisioner penelitian**

#### **Pertanyaan kuisisioner :**

Nama anak :

Usia :

Nama pendamping :

Kuisisioner

1. Berapa lama hari yang dibutuhkan oleh anak untuk mempelajari cara menggunakan aplikasi?
2. Berapa kali anak menggunakan aplikasi dengan menggunakan bantuan verbal dan fisik? Berapa kali anak menggunakan aplikasi dengan mandiri?
3. Apakah aplikasi yang dibuat memudahkan bapak/ibu untuk mengerti keinginan anak?
4. Menurut bapak/ibu, apakah aplikasi yang dibuat membantu proses komunikasi dengan anak?