



**APLIKASI *MONTESORI* PAD ASSISTEN UNTUK MEDIA  
*SHARING* METODE PEMBELAJARAN *MONTESORI***

**FADLAN FADIL**

**1820221080**

**PROGRAM STUDI TEKNIK ELEKTRO  
FAKULTAS TEKNIK  
UNIVERSITAS FAJAR  
2022**

**APLIKASI MONTESSORI PAD ASSISTEN UNTUK MEDIA  
SHARING METODE PEMBELAJARAN MONTESSORI**

**TUGAS AKHIR**

**Karya tulis sebagai salah satu syarat  
untuk memperoleh gelar Sarjana dari  
Universitas Fajar**

**Oleh**

**FADLAN FADIL**

**1820221080**



**PROGRAM STUDI TEKNIK ELEKTRO  
FAKULTAS TEKNIK  
UNIVERSITAS FAJAR  
2022**

**HALAMAN PENGESAHAN**

**Aplikasi *Montessori* Pad Assisten Untuk Media  
*Sharing* Metode Pembelajaran *Montessori***

oleh

**FADLAN FADIL**

**1820221080**

Menyetujui

Tim Pembimbing

Tanggal... *10 - Februari - 2023* .....

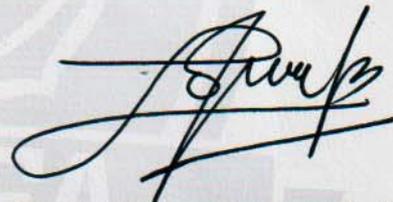
**Pembimbing I**



**Indah Purwitasari Ihsan, S.T., M.T.,**

**NIDN. 1221089001**

**Pembimbing II**



**Zagita Marna Putra, S.T., M.T.,**

**NIDN. 0922118603**

**Mengetahui,**

**Dekan Fakultas Teknik**


**Prof. Dr. Ir. Erniati, S.T., M.T.**

**NIDN. 0906107701**

**Ketua Program Studi**


**Safaruddin, S.SI., M.T.**

**NIDN. 0909106901**

## PERNYATAAN ORISINALITAS

Yang bertanda tangan di bawah ini :

Nama : Fadlan Fadil

Stambuk : 1820221080

Program Studi : Teknik Elektro

Menyatakan dengan sebenarnya bahwa tugas akhir ini yang berjudul “Aplikasi *Montessori* Pad Assisten Untuk Media *Sharing* Metode Pembelajaran *Montessori*” benar-benar merupakan hasil karya sendiri, bukan merupakan pengambil alihan tulisan atau pemikiran orang lain, Apabila dikemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan tugas akhir ini hasil karya orang lain, saya bersedia menerima sanksi atas perbuatan tersebut.

Makassar, 27-Oktober-2022

Yang menyatakan



FADLAN FADIL

## KATA PENGANTAR

Dengan Memanjatkan puji syukur kehadirat Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya, serta shalawat dan salam untuk junjungan kami, Nabi Muhammad SAW, penulis bersyukur dapat menyelesaikan penelitian yang berjudul “*Aplikasi Montessori Pad Assisten Untuk Media Sharing Metode Pembelajaran Montessori*” . Penyelesaian penelitian ini tidak lepas dari bantuan, doa, dan dukungan dari berbagai pihak.

Penulis menyadari bahwa pada penelitian ini masih memiliki banyak kekurangan, penulis berharap dapat menerima kritik maupun saran membangun dan bersifat terbuka yang bertujuan untuk menyempurnakan kekurangan dalam proposal penelitian ini, melalui kesempatan ini juga penulis mengucapkan terima kasih sebesar-besarnya kepada :

1. Ibu Prof. Dr. Ir. Erniati, S.T., M.T., selaku Dekan Fakultas Teknik Universitas Fajar.
2. Pak Safaruddin, S.Si., M.T. selaku Ketua Program Studi Teknik Elektro Universitas Fajar.
3. Ibu Indah Purwitasari Ihsan, S.T., M.T., sebagai Dosen Pembimbing I.
4. Bapak Zagita Marna Putra, S.T., M.T., selaku Dosen Pembimbing II.
5. Bapak Zagita Marna Putra, S.T., M.T., selaku Penasehat Akademik.
6. Orang tua yang telah memberikan dukungan, doa, motivasi dan pengorbanan materi dalam menyelesaikan proposal skripsi ini.
7. Dosen-dosen Program Studi Teknik Elektro Universitas Fajar.

Meskipun telah berusaha menyelesaikan penelitian ini dengan usaha maksimal, namun penulis menyadari bahwa pada penelitian ini masih terdapat kekurangan, oleh karena itu penulis mengharapkan kritik dan saran yang membangun dari para pembaca guna menyempurnakan segala kekurangan dalam penelitian ini. Akhir kata, penulis berharap semoga penelitian ini berguna bagi para pembaca dan pihak-pihak lain yang berkepentingan.

Makassar, 07 Februari 2022

## ABSTRAK

**Aplikasi *Montessori Pad Assisten Untuk Media Sharing Metode Pembelajaran Montessori, Fadalan Fadil.*** *Montessori* adalah suatu metode pembelajaran untuk menstimulasi anak untuk mengembangkan potensinya di segala bidang, dengan prinsip bahwa setiap individu harus mendidik dirinya sendiri, tugas guru atau orang tua adalah mendampingi dan memfasilitasi. Keterbatasan informasi karena masih jarang media *sharing* untuk saling bertukar informasi mengenai metode pembelajaran *Montessori* menyebabkan kesulitan dalam mencari ide-ide aktivitas *Montessori* untuk diterapkan. Pada penelitian ini dibuat sebuah aplikasi untuk menjadi media *sharing* seputar metode pembelajaran *Montessori* dan ide-ide kegiatan *Montessori* untuk memudahkan para orang tua dalam mencari ide kegiatan serta *sharing* pengalaman dalam menerapkan metode pembelajaran *Montessori* ini di rumah mereka masing – masing dengan menggunakan *smartphone android*. Metode pengembangan sistem yang digunakan adalah metode *SDLC (Sistem Development Life Cycle) Waterfall*. Bahasa pemrograman yang digunakan adalah Java. Berdasarkan hasil pengujian *black box* seluruh fungsionalitas sistem berfungsi dengan baik, berdasarkan hasil pengujian *white box* semua logika program telah benar, sedangkan berdasarkan pengujian kelayakan sistem dari 10 responden diperoleh hasil rata – rata 81.2% sistem telah layak untuk digunakan.

**Kata Kunci :** *Montessori, Pad Assisten, SDLC, Java, android*

## ABSTRACT

***Montessori Pad Assistant Application for Media Sharing Montessori Learning Method, Fadalan Fadil.*** Montessori is a learning method to stimulate children to develop their potential in all fields, with the principle that each individual must educate himself, the task of the teacher or Parent is to accompany and facilitate. Limited information due to the lack of sharing media to exchange information about the Montessori learning method makes it difficult to find ideas for Montessori activities to apply. In this study, an application was created to be a sharing medium about the Montessori learning method and ideas for Montessori activities to make it easier for Parents to find Activity ideas and share experiences in implementing this Montessori learning method in their respective homes using an Android smartphone. The system development method used is the SDLC (System Development Life Cycle) Waterfall method. The programming language used is Java. Based on the results of black box testing, all system functionality is functioning properly, based on the results of white box testing, all program logic is correct, while based on system feasibility testing of 10 respondents, the average result is 81.2% the system is feasible to use.

**Keywords :** *Montessori, Pad Assistant, SDLC, Java, android*

## DAFTAR ISI

	<b>Halaman</b>
HALAMAN SAMBUNG .....	i
KATA PENGANTAR .....	iii
HALAMAN PENGESAHAN.....	<b>Error! Bookmark not defined.</b>
PERNYATAAN ORISINALITAS .....	<b>Error! Bookmark not defined.</b>
ABSTRAK .....	v
DAFTAR ISI.....	vii
DAFTAR TABEL.....	ix
DAFTAR GAMBAR .....	xi
BAB I PENDAHULUAN.....	1
I.1 Latar Belakang.....	1
I.2 Rumusan Masalah.....	4
I.3 Tujuan Masalah.....	4
I.4 Batasan Masalah .....	4
BAB II TINJAUAN PUSTAKA.....	5
II.1 Tinjauan Teori .....	5
II.1.1 Aplikasi Mobile.....	5
II.1.2 Android.....	7
II.1.3 Media <i>Sharing</i> .....	8
II.1.4 Pembelajaran .....	8
II.1.5 Metode <i>Montessori</i> .....	9
II.1.6 Java.....	11
II.1.7 Basis Data ( <i>Database</i> ).....	12
II.1.8 <i>Firebase</i> .....	13
II.1.9 Flowchart.....	13
II.1.10 Unified Modeling Language (UML) .....	16
II.1.11 Pengujian Sistem .....	21
II.1.12 Pengembangan Sistem.....	22
II.2 Penelitian Terdahulu ( <i>State Of The Art</i> ).....	23
II.3 Kerangka Pemikiran .....	28

BAB III METODE PENELITIAN.....	29
III.1 Tahap Penelitian.....	29
III.2 Rancangan Penelitian/Sistem.....	30
III.2.1 Analisa Sistem Berjalan.....	30
III.2.2 Flowchart Sistem.....	31
III.2.3 Sistem yang Direncanakan.....	33
III.3 Waktu dan Lokasi Penelitian.....	54
III.4 Alat dan Bahan.....	54
III.5 Metode Pengumpulan Data.....	55
BAB IV HASIL DAN PEMBAHASAN.....	56
IV.1 Hasil.....	56
IV.1.1 Hasil Aplikasi.....	56
IV.2 Pembahasan.....	64
IV.2.1 Pengujian Black Box.....	64
IV.2.2 Pengujian <i>White Box</i> .....	72
IV.2.3 Pengujian Kelayakan Penggunaan Aplikasi.....	92
BAB V PENUTUP.....	98
V.1 Kesimpulan.....	98
V.2 Saran.....	99
DAFTAR PUSTAKA.....	100

## DAFTAR TABEL

	<b>Halaman</b>
Tabel II. 1 <i>Flow Direction Symbols</i> (Rosa, A. S., & Shalahudin, M. 2014).....	14
Tabel II. 2 <i>Processing Symbols</i> (Rosa, A. S., & Shalahudin, M. 2014). .....	14
Tabel II. 3 <i>Input / Output Symbols</i> (Rosa, A. S., & Shalahudin, M. 2014).....	15
Tabel II. 4 <i>Use Case Diagram</i> (Rosa, A. S., & Shalahudin, M. 2014).....	17
Tabel II. 5 <i>Class Diagram</i> (Rosa, A. S., & Shalahudin, M. 2014). .....	18
Tabel II. 6 <i>Sequence Diagram</i> (Rosa, A. S., & Shalahudin, M. 2014). .....	19
Tabel II. 7 <i>Activity Diagram</i> (Rosa, A. S., & Shalahudin, M. 2014). .....	20
Tabel II. 8 <i>State Of The Art</i> .....	23
Tabel III. 1 Deskripsi Use Case .....	33
Tabel III. 2 Kamus Data User .....	49
Tabel III. 3 Kamus Data AllPost.....	49
Tabel III. 4 Kamus Data AllCertificated.....	50
Tabel III. 5 Kamus Data Comment .....	50
Tabel III. 6 Kamus Data Like .....	51
Tabel III. 7 Kamus Data Learning .....	51
Tabel III. 8 Kamus Data Age .....	51
Tabel III. 9 Perangkat Keras .....	54
Tabel III. 10 Perangkat Lunak .....	55
Tabel III. 11 Metode Pengumpulan data.....	55
Tabel IV. 1 Pengujian Black Box .....	66
Tabel IV. 2 Rekapitulasi Hasil Pengujian Blackbox.....	72
Tabel IV. 3 Path Register .....	75
Tabel IV. 4 Jalur(path) Login.....	77
Tabel IV. 5 Jalur(path) Add Admin .....	80
Tabel IV. 6 Jalur(path) Add Category.....	82
Tabel IV. 7 Jalur(path) Approve Post .....	84
Tabel IV. 8 Jalur(path) Approve Certificated .....	86
Tabel IV. 9 Jalur(path) Add post.....	89
Tabel IV. 10 Rekapitulasi Hasil Pengujian White Box.....	90

Tabel IV. 11 Tabel Nilai Skor Tertinggi .....	92
Tabel IV. 12 Kriteria Skor .....	93
Tabel IV. 13 Hasil Kuesioner Pertanyaan 1 .....	93
Tabel IV. 14 Hasil Kuesioner Pertanyaan 2 .....	94
Tabel IV. 15 Hasil Kuesioner Pertanyaan 3 .....	94
Tabel IV. 16 Hasil Kuesioner Pertanyaan 4 .....	95
Tabel IV. 17 Hasil Kuesioner Pertanyaan 5 .....	96
Tabel IV. 18 Hasil Rata – Rata Persentase .....	97

## DAFTAR GAMBAR

	<b>Halaman</b>
Gambar II. 1 Data Penggunaan Mobile.....	6
Gambar II. 2 Logo Android .....	7
Gambar II. 3 Counting Socks.....	10
Gambar II. 4 Logo Java.....	11
Gambar II. 5 Logo Firebase .....	13
Gambar II. 6 SDLC Waterfall.....	22
Gambar II. 7 Kerangka Berfikir .....	28
Gambar III. 1 Tahapan Peneliatan .....	29
Gambar III. 2 Analisa Sistem Berjalan .....	30
Gambar III. 3 Flowchart Sistem Parent.....	31
Gambar III. 4 Flowchart Sistem Professional .....	32
Gambar III. 5 Use Case Diagram .....	33
Gambar III. 6 Activity Diagram Login Akun Admin .....	34
Gambar III. 7 Activity Diagram Admin Add Admin.....	35
Gambar III. 8 Activity Diagram Admin Add Category .....	36
Gambar III. 9 Activity Diagram Admin Approve Unggahan .....	37
Gambar III. 10 Activity Diagram Admin Approve Certificated.....	38
Gambar III. 11 Activity Diagram Register Akun Parent dan Professional.....	39
Gambar III. 12 Activity Diagram Login Akun Parent dan Professional.....	40
Gambar III. 13 <i>Activity</i> Diagram Mengunggah <i>Parent</i> .....	41
Gambar III. 14 Activity Diagram Mengunggah Professional .....	42
Gambar III. 15 Sequence Diagram Login Akun Admin .....	43
Gambar III. 16 <i>Sequence</i> Diagram Admin <i>Add Admin</i> .....	43
Gambar III. 17 Sequence Diagram Admin Add Category .....	44
Gambar III. 18 Sequence Diagram Admin Approve Post.....	44
Gambar III. 19 Sequence Diagram Admin Approve Certificated.....	45
Gambar III. 20 Sequence Diagram Register Akun User .....	45
Gambar III. 21 <i>Sequence</i> Diagram <i>Login</i> Akun <i>User</i> .....	46
Gambar III. 22 Sequence Diagram User Kelola Unggahan .....	47

Gambar III. 23 Class Diagram.....	47
Gambar III. 24 Tampilan Menu Start.....	52
Gambar III. 25 Tampilan Menu Register.....	52
Gambar III. 26 Tampilan Menu Login.....	53
Gambar III. 27 Tampilan Add Post.....	53
Gambar III. 28 Tampilan Menu Profile .....	54
Gambar IV. 1 Tampilan Awal Aplikasi .....	56
Gambar IV. 2 Tampilan Login.....	57
Gambar IV. 3 Tampilan Register .....	59
Gambar IV. 4 Tampilan Halaman Utama User.....	59
Gambar IV. 5 Tampilan Menu Add Post .....	60
Gambar IV. 6 Tampilan Menu Profile .....	60
Gambar IV. 7 Tampilan Halaman Detail Postingan .....	61
Gambar IV. 8 Tampilan Halaman Utama Admin .....	61
Gambar IV. 9 Tampilan List Approval Admin.....	62
Gambar IV. 10 Tampilan Halaman Approve Post Admin.....	62
Gambar IV. 11 Tampilan List Approval Certificated .....	63
Gambar IV. 12 Tampilan Halaman Approve Certificated.....	63
Gambar IV. 13 Tampilan Add Admin .....	64
Gambar IV. 14 Flow Graph Register .....	72
Gambar IV. 15 Flow Graph Login.....	76
Gambar IV. 16 Flow Graph Add Admin .....	78
Gambar IV. 17 Flow Graph Add Category .....	81
Gambar IV. 18 Flow Graph Approve .....	82
Gambar IV. 19 Flow Graph Add Post.....	87

# BAB I

## PENDAHULUAN

### I.1 Latar Belakang

Sejak lahir sampai usia enam tahun, anak-anak mengalami masa keemasannya. Ini adalah masa ketika anak-anak sensitif untuk memberi dan menerima rangsangan yang dapat mengembangkan keterampilan kognitif, bahasa, motorik dan sosial yang akan membantu mereka berkembang. (Sujiono, Y. N. 2009:2).

Usia ini merupakan masa keemasan perkembangan anak karena sel-sel otak tumbuh dengan pesat dan saling berhubungan dengan sangat cepat. Ada beberapa metode pembelajaran atau metode Montessori untuk memaksimalkan perkembangan anak. Metode pembelajaran Montessori merupakan metode pembelajaran anak usia dini untuk pengembangan keterampilan kognitif, motorik dan sosial.

Metode ini menekankan pentingnya menyesuaikan lingkungan belajar dengan tingkat perkembangan anak dan peran aktivitas fisik dalam mengasimilasi mata pelajaran sekolah dan keterampilan praktis. Pada dasarnya, metode pembelajaran *Montessori* mirip dengan sistem kurikulum tradisional dan selalu melibatkan peran siswa dan guru. Namun, di sekolah konvensional, semua kelas diajarkan berdasarkan kurikulum, sehingga tidak dapat dihindari bahwa siswa "dipaksa" untuk memahami segala sesuatu yang diajarkan.

*Montessori* adalah suatu metode pembelajaran bagi anak untuk merangsang dan mengembangkan potensinya di segala bidang berdasarkan prinsip bahwa setiap individu perlu mendidik dirinya sendiri dan peran guru atau orang tua adalah untuk mendukung dan menginformasikan. *Montessori* percaya bahwa kemampuan dasar pada ilmu pengetahuan bisa dipahami anak-anak dengan simpel, Jika mereka diperlihatkan alat-alat peraga yang konkret buat membantu mereka melakukan khayalan (Lillard, P. P. (1997:80). Metode *Montessori* sendiri merupakan suatu metode pembelajaran yang berfokus pada

anak usia dini. Metode tersebut dikembangkan oleh seorang tokoh bernama Maria *Montessori*. Metode *Montessori* telah banyak diterapkan di negara-negara lain. Hasilnya, anak-anak usia dini yang diajar dengan menggunakan metode *Montessori* lebih mandiri, kreatif, dan juga memiliki keahlian motorik dan kognitif yang lebih baik.

Permasalahan yang sering terjadi dalam mengimplementasikan metode tersebut adalah masih jarang media *sharing* untuk saling bertukar informasi mengenai bagaimana cara menerapkan metode pembelajaran *Montessori* di rumah serta kesulitan dalam mencari ide-ide aktivitas *Montessori* untuk diterapkan. Terkadang para ibu kebingungan mencari ide kegiatan *Montessori* dan memulainya karena keterbatasan informasi. Meskipun ada, namun hanya untuk sekolah-sekolah yang menerapkan metode tersebut ataupun untuk komunitas-komunitasnya.

Disisi lain, sarana informasi terus berkembang dan sangat diperlukan setiap saat karena melalui teknologi informasi manusia dapat menemukan informasi yang semakin berkembang, selain itu manusia juga dapat saling berinteraksi. Salah satu sarana perkembangan teknologi yang berkembang pesat di bidang teknologi dan informasi adalah aplikasi yang berpusat sebagai media informasi.

Berdasarkan uraian tersebut diatas, maka penelitian ini akan dibuat sebuah aplikasi yang bekerja pada *smartphone* yang berjalan pada sistem operasi *Android* untuk menjadi media *sharing* bagi pada orang tua dalam berbagi informasi seputar metode pembelajaran dan ide-ide kegiatan *Montessori*. Dengan adanya aplikasi media *sharing* dapat memudahkan para orang tua dalam mencari ide kegiatan serta *sharing* pengalaman dalam menerapkan metode pembelajaran *Montessori* ini di rumah mereka masing – masing. Berlandaskan hal tersebut, maka judul penelitian ini adalah “Aplikasi *Montessori* Pad Assisten Untuk Media *Sharing* Metode Pembelajaran *Montessori*”.



## **I.2 Rumusan Masalah**

Berdasarkan deskripsi yang dikemukakan, maka permasalahan yang akan dibahas ialah:

1. Bagaimana membuat media *sharing* metode pembelajaran *montessori* berbasis teknologi informasi?
2. Bagaimana kelayakan dari penggunaan aplikasi media *sharing* metode pembelajaran *montessori*?

## **I.3 Tujuan Masalah**

Penelitian ini bertujuan sebagai berikut :

1. Membuat aplikasi *montessori pad asisten* untuk media *sharing* metode pembelajaran *montessori* yang dapat dijalankan pada *smartphone* dengan sistem operasi android.
2. Mengetahui tingkat kelayakan dari penggunaan aplikasi *montessori pad asisten* sebagai media *sharing* seputar metode pembelajaran *montessori*.

## **I.4 Batasan Masalah**

Agar pembahasan tidak melenceng dari permasalahan, ruang lingkup penelitian ini akan dibatasi, yaitu:

1. Aplikasi ini hanya bekerja pada *smartphone* dengan sistem operasi android.
2. Sistem hanya difokuskan untuk *sharing* seputar metode pembelajaran *montessori*.
3. Fitur yang ada dalam aplikasi seperti komentar, *like*, *upload* teks dan gambar.

## BAB II

### TINJAUAN PUSTAKA

#### II.1 Tinjauan Teori

##### II.1.1 Aplikasi Mobile

Aplikasi *mobile* biasanya disebut dengan *mobile apps*, yaitu kata yang digunakan untuk mendeskripsikan aplikasi internet yang berjalan pada *smartphone* atau piranti *mobile* lainnya (Hingle, M., & Patrick, H. 2016). Aplikasi *mobile* biasanya membantu para penggunanya untuk terkoneksi dengan layanan internet. Aplikasi seluler adalah aplikasi yang dirancang untuk *platform* seluler (Ios, Android, Windows Mobile, dll.) (Pressman, R. S. 2010).

Dalam beberapa kasus, aplikasi seluler mencakup mekanisme korelasi unik yang disediakan oleh *platform* seluler, interoperabilitas menggunakan sumber daya berbasis *web* yang menyediakan akses ke berbagai informasi yang terkait dengan aplikasi, dan kemampuan untuk mengumpulkan, menganalisis, dan memproses informasi. antarmuka pengguna dengan kemampuan pemrosesan lokal untuk pemformatan. Sempurna untuk *platform* seluler. Selain itu, aplikasi seluler juga menawarkan kemungkinan penyimpanan persisten di *platform*.

Aplikasi *mobile* (*mobile apps*) merupakan aplikasi yang dibuat bagi perangkat-perangkat bergerak (*mobile*) seperti : *smartphone*, tablet, *smartwatch*, dan lainnya. *Software* aplikasi merupakan hasil dari pemrograman *mobile* yang dirancang menggunakan bahasa-bahasa pemrograman tertentu.

Aplikasi seluler adalah proses pengembangan aplikasi untuk perangkat seluler, telepon seluler, PDA, atau telepon pintar. Aplikasi yang sudah berjalan di ponsel, sering disebut aplikasi default. Pengguna juga dapat mengunduh aplikasi ini dari *app store* atau *play store* dan juga dari distribusi perangkat lunak *platform* seluler lainnya.



5. Jual Beli : aplikasi ini biasanya digunakan untuk bertransaksi, baik pembeli maupun penjual melalui *platform* yang biasanya dikenal dengan *e-commerce*.
6. Permainan : aplikasi ini biasanya digunakan agar pengguna tidak bosan dalam beraktivitas.
7. Kesehatan : aplikasi ini biasanya digunakan untuk konsultasi tanpa harus ke rumah sakit dulu.

## II.1.2 Android

Android adalah sistem operasi berbasis linux untuk perangkat seluler yang mencakup sistem operasi, *middleware*, dan aplikasi (Safaat, N. 2012). Android adalah sistem operasi berbasis linux untuk ponsel dan tablet layar sentuh (Dharma Kasman, A. (2015).



Gambar II. 2 Logo Android

(Sumber: <https://www.freepnglogos.com/images/android-logo-12379.html>)

Namun seiring perkembangan zaman, android menjadi *platform* inovasi yang cepat. Hal ini tidak terlepas dari developer utama di belakangnya, yaitu Google. Google yang membeli Android, kemudian mengembangkan *platform*.

Platform Android mencakup sistem operasi berbasis Linux, antarmuka pengguna grafis (GUI), browser web, dan aplikasi pengguna

akhir yang dapat diunduh, dan pengembang bebas untuk bekerja dan membuat aplikasi

*Platform* android mencakup sistem operasi berbasis linux, sebuah GUI (*Graphic User Interface*), *web browser* dan aplikasi *end-user* yang dapat yang dapat diunduh, dan pengembang bebas untuk bekerja dan membuat aplikasi sesuai dengan keinginan pengembang menggunakan bahasa pemrograman tertentu.

### II.1.3 Media *Sharing*

Kata media berasal dari bahasa latin *medius* yang berarti menengahi atau memperkenalkan. Kata media merupakan bentuk jamak dari kata *media* dan secara etimologis berarti mediasi atau pengantar. Media adalah perantara/komunikasi atau sarana komunikasi antara dua pihak seperti surat kabar, majalah, radio, televisi, film, poster, spanduk, dll (Save, M. D. 2006:634).

Semua media adalah bentuk yang digunakan orang untuk menyampaikan atau menyebarkan ide, gagasan, atau pendapat sehingga sampai kepada penerima yang dituju (Arsyad, A. 2002:4).

Dari beberapa pernyataan diatas dapat disimpulkan bahwa media sharing adalah alat untuk menyampaikan informasi kepada penerima dan dapat digunakan untuk menyampaikan pesan dari pengirim kepada penerima agar dapat diambil kembali. Untuk komunikasi yang efektif dan efisien.

### II.1.4 Pembelajaran

Pembelajaran adalah mengajar siswa untuk menggunakan prinsip-prinsip pendidikan dan teori belajar, yang merupakan penentu utama keberhasilan pendidikan. Belajar adalah komunikasi dua arah. Pengajaran dilakukan oleh guru sebagai pendidik, sedangkan belajar dilakukan oleh siswa (Sagala, S. 2010:61).

Belajar adalah usaha sadar oleh pendidik yang dapat mendorong peserta didik untuk berpartisipasi dalam kegiatan pembelajaran (Sudjana, N. 2010:28). Pembelajaran pada dasarnya adalah komunikasi

transaksional dua arah antara guru dan siswa dan antara siswa dengan siswa lainnya untuk mencapai tujuan yang telah ditetapkan. komunikasi yang dapat diterima, dipahami, dan disepakati oleh berbagai pihak yang terlibat dalam proses pembelajaran (Hernawan, A. H., Susilana, R., & Julaeha, S. 2014).

Berdasarkan deskripsi dari beberapa ahli di atas, dapat disimpulkan bahwa pembelajaran adalah suatu proses belajar yang melibatkan baik guru maupun peserta didik guna mencapai suatu tujuan yaitu keahlian belajar yang berpengaruh pada pengetahuan, sikap dan keterampilan peserta didik.

#### II.1.5 Metode *Montessori*

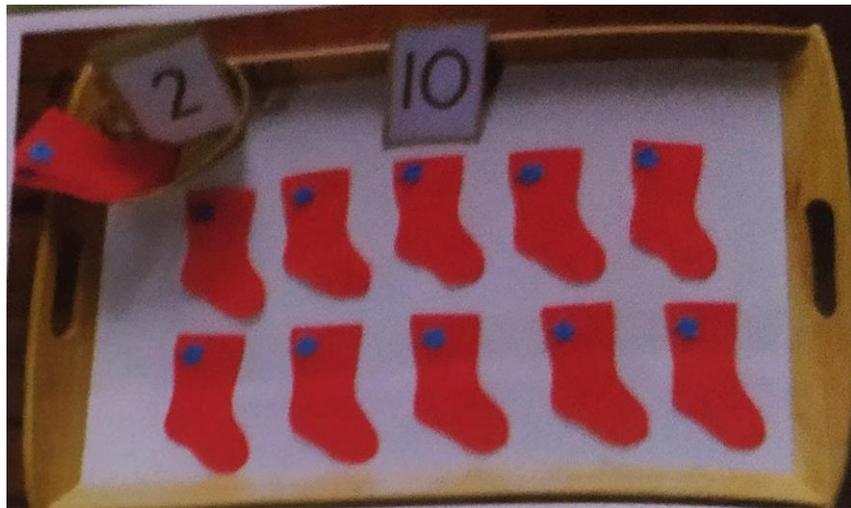
Montessori berpendapat bahwa anak adalah sebatang badan yang tumbuh dan sesosok jiwa yang berkembang, kedua bentuk fisik dan psikis memiliki sumber yang abadi yaitu hidup itu sendiri (Hughes, A. G., & Hughes, E. H. 2012). Pendapat Maria Montessori diatas telah jelas bahwa anak merupakan sebatang badan yang dapat tumbuh. Misalnya tinggi badannya bertambah, berat badannya bertambah dan pertumbuhan lainnya yang bisa dilihat dan diukur. Kemudian anak juga merupakan sesosok jiwa yang berkembang.

Hainstock menyatakan bahwa metode Montessori merupakan salah satu bentuk pembelajaran dengan menggunakan pendekatan individual, dimana anak mengatur belajarnya sendiri, memanfaatkan media pembelajaran yang dapat diawasi dan diperbaiki bila salah oleh mereka sendiri, pendidik hanya memantau kebiasaan dan gaya anak. Metode ini dirancang khusus untuk menumbuhkan minat belajar anak, mendalami segala potensi dan kemampuan anak baik fisik maupun psikisnya (Damayanti, E. 2019).

Menurut filosofi Dr. Montessori, anak-anak belajar paling baik di lingkungan berukuran sedang yang merangsang dan memfasilitasi penyerapan kemampuan kognitif (berpikir) anak ke dalam lingkungan. Dirancang untuk memungkinkan akses dan penggunaan bahan beracun.

Dalam lingkungan itu, anak-anak dapat memilih sendiri tugas dan kegiatan yang bermakna dan berorientasi pada tujuan. Misalnya untuk kegiatan teknologi untuk anak-anak, dalam kegiatan ini Montessori melakukan kegiatan seperti menggunting huruf besar dari ampas dan meminta anak untuk berlatih dengan jarinya kemudian dengan pensil atau kapur.

Contohnya :



Gambar II. 3 Counting Socks

(Sumber: Buku *Islamic Montessori Inspired Activity*)

*Title : Counting Socks*

*Category : Math, 3-6 tahun*

Alat dan bahan:

1. Alas kerja, nampan
2. kartu angka 1-10
3. kaus kaki dari kain flannel

Aplikasi Kegiatan

Tumpuk kartu angka 1-10 secara beruntun. Tunjuk angka 1 dan letakkan di sebelah kiri alas kerja, lanjutkan dengan angka lainnya. Tunjuk kembali angka 1, lalu tanyakan berapa kaus kaki yang kita butuhkan, letakkan satu kaus kaki di bawahnya. Lanjutkan hingga

hitungan 10 kaus kaki. Jelaskan kepada anak bahwa ia dapat mencobanya sendiri keesokan harinya.

*Montessori* membagi belajar menjadi tiga tahapan :

1. Tahapan 1 : Memperkenalkan identitas. Misalnya, buat koneksi antara objek yang ditampilkan dan nama objek.
2. Tahapan 2 : Memperkenalkan Perbandingan Langkah kedua adalah memastikan anak mengerti.
3. Tahapan 3 : Perbedaan: Benda Serupa Pada tahap ketiga ini, penting apakah anak benar-benar mengingat nama-nama benda tersebut.

Tujuan dari proses pembelajaran tiga langkah adalah untuk menanamkan konsep-konsep baru melalui pengulangan. Ini membantu anak-anak lebih memahami materi yang disajikan. Ini juga membantu guru melihat seberapa baik anak-anak memperoleh dan menyerap apa yang mereka pelajari.

#### II.1.6 Java

Java adalah nama seperangkat teknologi untuk membuat dan menjalankan perangkat lunak pada komputer *standalone* atau dalam lingkungan jaringan (Sukamto, R. A., & Shalahuddin, M. 2013:103). Bahasa pemrograman Java merupakan salah satu dari sekian banyak bahasa pemrograman yang dapat dijalankan di berbagai sistem operasi, termasuk ponsel (Nofriadi, M. 2015).



Gambar II. 4 Logo Java

(Sumber: <https://brandslogos.com/j/java-logo-1/>)

Bahasa pemrograman Java memiliki banyak fungsi yang membuatnya lebih populer di kalangan programmer. Berikut ini fungsi dari java antara lain:

1. Dapat digunakan pada banyak *platform* sistem operasi yang berbeda
2. Pemrograman berfokus/berbasis pada objek
3. Daftar *library* lengkap
4. Pengamanan java yang ketat

#### II.1.7 Basis Data (*Database*)

Basis data merupakan kumpulan data yang menjabarkan suatu aktivitas dari satu atau beberapa entitas yang berhubungan. Sebagai contoh basis data sebuah universitas memiliki entitas seperti mahasiswa, staf pengajar, mata kuliah, dan ruang kelas. Hubungan antara entitas tersebut seperti mahasiswa yang mendaftar untuk mengikuti mata kuliah, staf pengajar yang mengajarkan mata kuliah, dan penggunaan ruang kelas untuk suatu mata kuliah menurut (Ramakrishnan, R., Gehrke, J., & Gehrke, J. 2003:3). Dalam basis data atau *database* dikenal pula DBMS (*database management system*) merupakan paket perangkat lunak yang digunakan untuk memanipulasi *database*.

Basis data atau database adalah kumpulan informasi yang disimpan secara sistematis di dalam komputer sehingga dapat diverifikasi oleh program yang digunakan sebagai informasi tersebut (Sani, R. A., Manurung, S. R., & Suswanto, H. Sudiran. 2018). Basis data adalah aplikasi yang menyimpan banyak kumpulan data yang berbeda, yang masing-masing memiliki pekerjaan tertentu untuk dilakukan, mengelola, mengakses, dan mencari data yang terdapat dalam basis data (*database*). Dari sini kita dapat menyimpulkan bahwa *database* adalah aplikasi yang berisi kumpulan informasi yang disimpan secara sistematis di dalam komputer.

*Database* menyimpan data dalam sistem *database* terstruktur yang terdiri dari entitas dan objek secara detail. Basis data dengan menggunakan DBMS (*database management system*) adalah Perangkat lunak untuk pemrosesan basis data. DBMS yang umum digunakan saat ini adalah MySQL, Oracle, Microsoft SQL Server, Firebird dan masih banyak lagi.

#### II.1.8 *Firestore*

*Firestore* adalah API yang disediakan google untuk penyimpanan dan penyelarasan data ke dalam aplikasi Android, iOS, atau web (Payara, G. R., & Tanone, R. 2018). *Firestore* adalah salah satu layanan terkenal dari perusahaan Google mendukung pengembang perangkat lunak dalam mengembangkan aplikasi. Pengembangan aplikasi, terutama untuk aplikasi *mobile*. *Firestore* didirikan oleh Andrew Lee dan James Tamplin sebagai perusahaan *Involve* pada tahun 2011.



Gambar II. 5 Logo Firestore

Sumber : <https://firebase.google.com/brand-guidelines>

Firestore memiliki banyak fitur. Fitur-fitur tersebut sangat berguna dan memudahkan pengembang untuk mengembangkan aplikasi *mobile*. Saat mengembangkan aplikasi menggunakan Firestore, beberapa fitur yang dapat Anda gunakan antara lain autentikasi, database, penyimpanan, pengarsipan dan analitik.

#### II.1.9 Flowchart

*Flowchart* adalah suatu gambaran dari grafik atau bagian dari urutan prosedur yang ada didalam program dan memiliki hubungan antara proses beserta bentuknya (Dewi, I. R., & Malfiany, R. 2017:7). Diagram alir dokumen merupakan notasi standar yang digunakan oleh

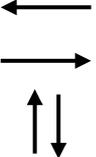
analisis sistem untuk menggambarkan diagram alir dokumen suatu sistem (Rosa, A. S., & Shalahudin, M. 2014).

Di bawah ini adalah beberapa notasi atau simbol yang digunakan, yang dapat dibagi menjadi tiga kelompok:

### 1. Simbol Penghubung/alur (*Flow Direction Symbols*)

Simbol digunakan untuk menghubungkan satu simbol dengan simbol lainnya. Berikut ini symbol-simbol yang terdapat pada *flow direction symbols*.

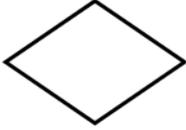
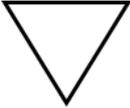
Tabel II. 1 *Flow Direction Symbols* (Rosa, A. S., & Shalahudin, M. 2014).

No	Symbol	Nama	Keterangan
1		Alur/Arus	Untuk menunjukkan arah
2		<i>Communication link</i>	Untuk menunjukkan adanya perubahan sebuah data
3		<i>Connector</i>	Untuk menunjukkan hubungan dalam hal/lembaran sama
4		<i>Offline Connector</i>	Untuk menunjukkan hubungan dalam halaman/lembaran berbeda

### 2. Simbol Proses (*Processing Symbols*)

Tabel II. 2 *Processing Symbols* (Rosa, A. S., & Shalahudin, M. 2014).

No.	Symbol	Nama	Keterangan
-----	--------	------	------------

No.	Symbol	Nama	Keterangan
1		Proses	Untuk menunjukkan suatu proses dari aplikasi
2		Simbol manual	Untuk menunjukkan suatu proses yang tidak dilakukan oleh komputer (manual)
3		<i>Decision / Logika</i>	Untuk menunjukkan suatu kondisi, dengan dua kemungkinan, <i>True / False</i>
4		<i>Predefined Process</i>	Untuk menunjukkan nilai awal sebuah variabel
5		Terminal	Untuk menunjukkan <i>start</i> atau <i>end</i> suatu program
6		<i>Offline Storage</i>	Untuk menunjukkan data disimpan dalam media tertentu
7		<i>Manual Input</i>	Untuk memasukkan data secara manual dengan menggunakan online keyword

### 3. Simbol Input – output (*Input / Output Symbols*)

Tabel II. 3 *Input / Output Symbols* (Rosa, A. S., & Shalahudin, M. 2014).

No	Symbol	Nama	Keterangan
1		<i>Input / output</i>	Untuk menunjukkan proses <i>input</i> dan <i>output</i> .

No	Symbol	Nama	Keterangan
2		<i>Disk Storage</i>	Untuk menunjukkan <i>input</i> berasal dari <i>disk</i> atau <i>output</i> disimpan ke <i>disk</i>
3		<i>Document</i>	Untuk menyetak dokumen

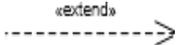
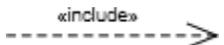
### II.1.10 Unified Modeling Language (UML)

*Unified modelling language* atau biasa UML adalah salah satu metodologi rekayasa perangkat lunak yang digunakan untuk menggambarkan aliran dan operasi sistem, fungsi, tujuan, dan mekanisme kontrol sistem (Abdillah, R., Kuncoro, A., & Kurniawan, I. 2019). Rekayasa perangkat lunak di bidang analisis dan desain sistem informasi, sebagai kombinasi dari konsep pemrograman berorientasi objek dan teknik produksi perangkat lunak, telah memungkinkan untuk melihat sistem sebagai objek terpisah yang sudah berisi data dan proses. atau dapat beroperasi secara mandiri dalam rangkaian sistem (paket) sistem. Ada empat model UML untuk pengembangan sistem informasi. Penggunaan yang paling efektif (Rosa, A. S., & Shalahudin, M. 2014) untuk menggambarkan desain sistem adalah:

#### 1. *Use Case Diagram*

*Use case* diagram digunakan untuk mengetahui interaksi pengguna (aktor) dan apa yang dapat dilakukan sistem. Sebuah *use case* dapat mewakili beberapa jalur interaksi manusia-sistem, setiap jalur disebut skenario.

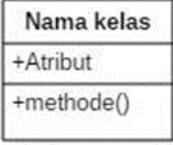
Tabel II. 4 *Use Case Diagram* (Rosa, A. S., & Shalahudin, M. 2014).

No.	Symbol	Nama	Keterangan
1		<i>Actor</i>	Untuk menunjukkan tokoh atau seseorang yang berinteraksi dengan sistem.
2		Use Case	Untuk menunjukkan fungsi dari system yang telah direncanakan.
3		Asosiasi/ <i>Association</i>	Untuk menunjukkan hubungan antara <i>use case</i> dan aktor.
4		Extensi/ <i>Extend</i>	Untuk menunjukkan arah panah secara putus-putus dari <i>use case</i> ke <i>base use case</i> .
5		<i>Include</i>	Untuk menunjukkan bahwa <i>use case</i> satu merupakan bagian dari <i>use case</i> lainnya.

## 2. *Class Diagram*

*Class diagram* adalah model statis yang membantu menampilkan data dan informasi di seluruh sistem. Penggunaan *class diagram* terkait dengan struktur *database* sistem atau dapat menggantikan ERD (*Entity Relational Diagram*) dalam proses diagram rekayasa perangkat lunak normal.

Tabel II. 5 *Class Diagram* (Rosa, A. S., & Shalahudin, M. 2014).

No.	Symbol	Nama	Keterangan
1		<i>Class</i>	Untuk menunjukkan 3 bagian utama <i>class</i> , yaitu nama kelas, atribut kelas dan metode kelas.
2		<i>Association</i>	Untuk menunjukkan hubungan statis antar kelas.
3		<i>Agregation</i>	Untuk menunjukkan hubungan antar kelas yang menjadi hubungan di kelas itu.
4		<i>Composition</i>	Untuk menunjukkan bentuk khusus dari <i>agregation</i> dimana kelas yang menjadi bagian diciptakan setelah kelas <i>whole</i> dibuat.
5		<i>Generalization</i>	Untuk menunjukkan relasi antar kelas
6		<i>Directed Association</i>	Untuk menunjukkan makna kelas digunakan oleh kelas lainnya.

### 3. *Sequence Diagram*

*Sequence* diagram urutan menunjukkan objek yang terlibat dalam *use case* dan menggambarkan aliran pesan antara objek dalam objek *use case*. Diagram urutan bersifat dinamis dan menunjukkan lebih banyak aktivitas objek berdasarkan urutan kronologis.

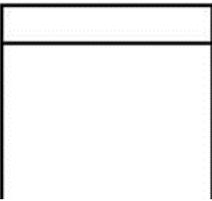
Tabel II. 6 *Sequence Diagram* (Rosa, A. S., & Shalahudin, M. 2014).

No.	Symbol	Nama	Keterangan
1		<i>Actor</i>	Untuk menunjukkan tokoh atau seseorang yang berinteraksi dengan sistem.
2		<i>Boundary</i>	Untuk menunjukkan hubungan kegiatan.
3		<i>Control</i>	Untuk menunjukkan penghubung antara <i>boundary</i> dengan tabel.
4		<i>Entity</i>	Untuk menunjukkan hubungan kegiatan.
5		<i>Message</i>	Untuk menunjukkan komunikasi antar objek.
6		<i>Life Line</i>	Untuk menunjukkan keberadaan suatu objek.

#### 4. Activity Diagram

Diagram aktivitas adalah desain aliran aktivitas atau alur kerja dalam sistem yang sedang berjalan. Diagram aktivitas juga digunakan untuk mendefinisikan atau mengelompokkan aliran dalam tampilan sistem. Diagram aktivitas memiliki elemen dengan bentuk tertentu yang dihubungkan oleh panah. Panah menunjukkan urutan kegiatan dari awal sampai akhir.

Tabel II. 7 Activity Diagram (Rosa, A. S., & Shalahudin, M. 2014).

No.	Symbol	Nama	Keterangan
1		<i>Initial</i>	Untuk menunjukkan titik awal aktivitas.
2		<i>Final</i>	Untuk menunjukkan titik akhir aktivitas.
3		<i>Activity</i>	Untuk menunjukkan adanya sebuah aktivitas
4		<i>Decision</i>	Untuk menunjukkan pilihan sebuah aktivitas
5		<i>Fork</i> atau <i>Join</i>	Untuk menunjukkan kegiatan aktivitas secara parallel.
6		<i>Flow</i> <i>Final</i>	Untuk menunjukkan akhir dari suatu aliran.
7		<i>Swimlane</i>	Untuk menunjukkan pengelompokan aktivitas berdasarkan aktor.

### II.1.11 Pengujian Sistem

Pengujian sistem yang sering digunakan untuk menguji dari kerja suatu sistem:

#### 1. *White Box*

*White box*, juga dikenal sebagai pengujian kotak kaca, adalah teknik desain kasus uji yang menggunakan interpretasi struktur kontrol untuk menghasilkan kasus uji sebagai bagian dari desain tingkat komponen (Pressman, R. S. 2010). Insinyur perangkat lunak menggunakan *white box* untuk menulis kasus uji yang memastikan bahwa setiap jalur independen dari modul telah dieksekusi setidaknya sekali, menguji semuanya, dan membuat keputusan logis tentang nilai benar dan salah. Ini dapat membuat keputusan dan menjalankan semua *loop* dalam batasnya dan batas operasi. Periksa struktur data internal untuk memastikan kebenaran. Contoh *white box* adalah pengujian detail desain aplikasi, seperti "tombol dalam aplikasi".

#### 2. *Black Box*

Pengujian black box adalah salah satu strategi pengujian yang juga dikenal sebagai pengujian berbasis data atau pengujian input/output (Brown, U. J., & Gaylor, K. 2002). Untuk menggunakan metode ini, program dipandang sebagai kotak hitam. Tujuannya adalah untuk mengabaikan perilaku dan struktur internal program. Agar penguji dapat fokus menemukan elemen yang hilang saat program tidak bekerja sesuai spesifikasinya. Dalam pendekatan ini, *instance* data dibuat hanya sesuai dengan spesifikasinya tanpa mengetahui struktur internal program.

Contoh pengujian black box adalah pengujian fungsi-fungsi yang ada pada aplikasi seperti penambahan komentar.

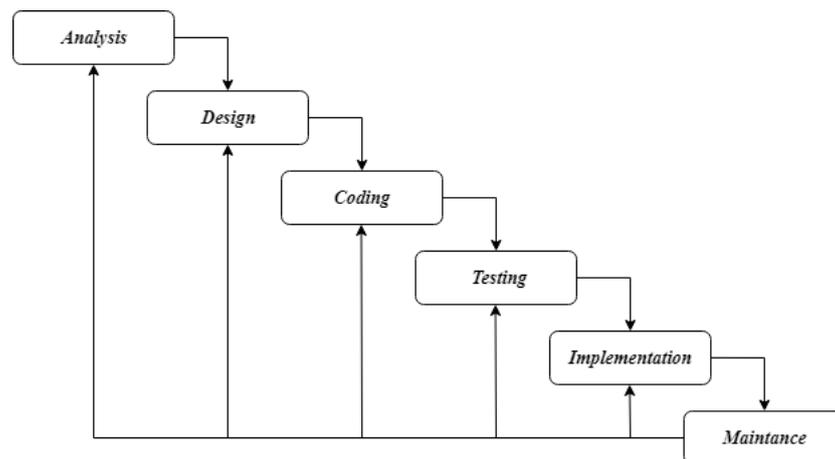
### II.1.12 Pengembangan Sistem

Metodologi pengembangan sistem adalah metode/langkah yang digunakan untuk membangun sistem sedemikian rupa sehingga terstruktur dengan cara yang sesuai dengan tujuan awal pembuatan sistem. Salah satu metode pengembangan sistem yang akan digunakan dalam penelitian ini adalah pendekatan *Waterfall Systems Development Life Cycle* (SDLC).

Tahap ini harus diselesaikan terlebih dahulu sebelum melanjutkan ke tahap berikutnya. Dengan demikian hasilnya akan berkaitan dengan setiap tahapan sehingga pekerjaan dilakukan secara optimal karena tidak ada pekerjaan yang berlangsung secara parallel.

Adapun beberap kelebihan menggunakan metode waterfall diantaranya:

1. Dengan menggunakan metode ini alur kerja menjadi lebih teratur langkah demi langkah.
2. Ini juga menguntungkan dari sudut pandang pengguna, karena semua data dan proses yang diperlukan dapat direncanakan dan disiapkan.



Gambar II. 6 SDLC Waterfall

(Sumber : <https://www.kibrispdr.org/pre-0/contoh-metode-waterfall.html> )

## II.2 Penelitian Terdahulu (*State Of The Art*)

Tabel II. 8 *State Of The Art*

No.	Nama Peneliti	Judul	Metode dan Tools	Hasil
1	Setiawan Assegaff  2017	Evaluasi Pemanfaatan Media Sosial sebagai Sarana Knowledge <i>Sharing</i>	Metode SEM ( <i>Structural Equation Modeling</i> ) dalam menganalisis data yang diperoleh dari hasil survei	Berdasarkan hasil survei, dapat disimpulkan bahwa anggota grup FB Indonesia saat ini puas dengan jejaring sosial (Facebook) untuk bertukar informasi dan pendapat. Anggota tim merasa puas setelah menggunakan jejaring sosial untuk pertama kalinya dan merasa bahwa jejaring sosial sebagai sarana berbagi pengetahuan telah membantu mereka memenuhi kebutuhan informasi mereka.

No.	Nama Peneliti	Judul	Metode dan Tools	Hasil
2	Andito Haryo Saputro 2017	Pembuatan aplikasi interaktif berhitung permulaan dengan metode <i>Montessori</i> untuk TK Pustaka Arridho	- Research and Development (R&D) - Metode RAD (Rapid Application Development) - Alat yang digunakan untuk membangun sistem adalah Adobe Flash	Hasil dari penelitian ini adalah aplikasi entry level Montessori berbasis Flash yang dioperasikan melalui PC. Peringkat kelayakan untuk aplikasi interaktif adalah 94,7% oleh profesional komunikasi dan 96,6% oleh profesional material. Dari data ini, kita dapat menyimpulkan bahwa menafsirkan skor sebagai aplikasi interaktif media pembelajaran adalah kriteria yang sangat praktis dan juga sistem telah layak untuk digunakan.

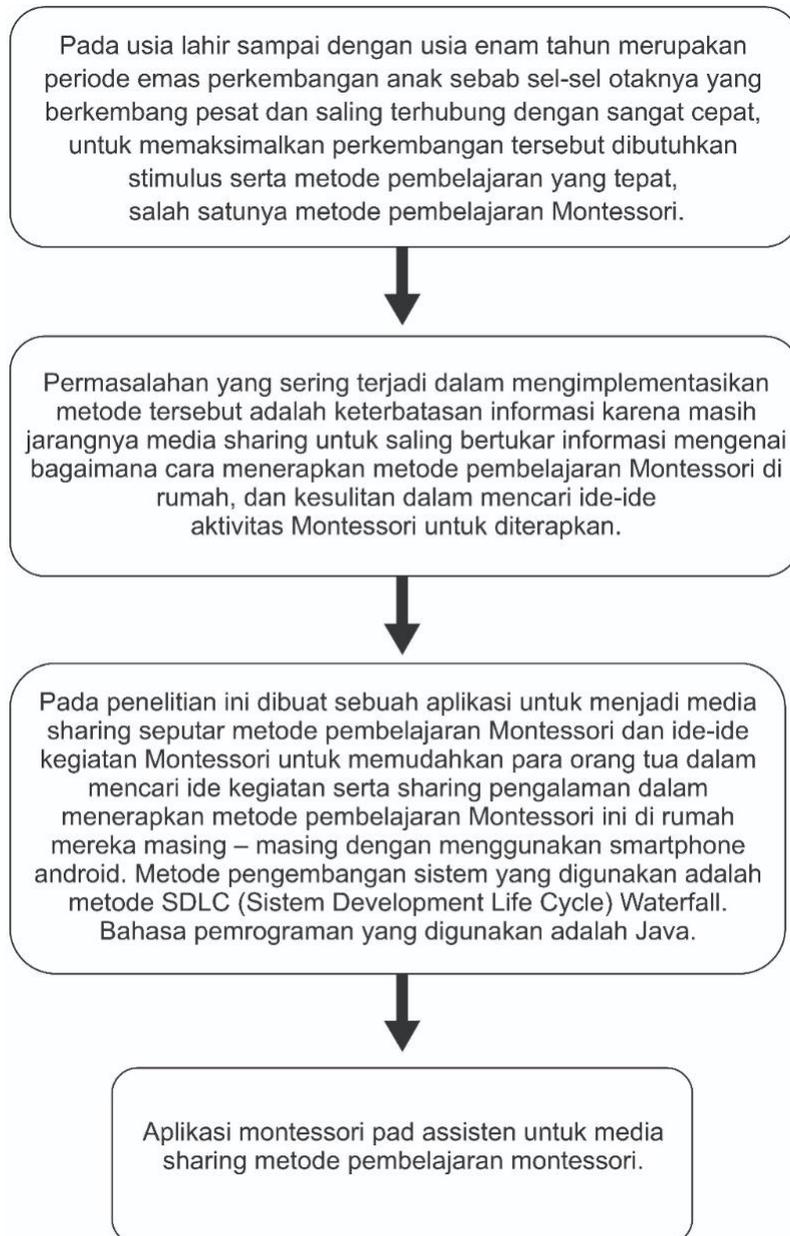
No.	Nama Peneliti	Judul	Metode dan Tools	Hasil
3	Dhiah Intan Permataputri, Amir Syamsudin  2021	Pembelajaran Nilai Agama dan Moral Anak Usia Dini melalui Metode <i>Montessori</i> selama Pandemi Covid-19	Kualitatif Deskriptif	Penerapan aspek pembelajaran nilai agama dan etika dengan metode Montessori memiliki perbedaan sebelum dan pada masa pandemi, mulai dari metode pelaksanaan, durasi pembelajaran, dukungan dalam melakukan pembelajaran, penilaian dan penugasan anak. Hasil pengembangan dilakukan secara online. Pengajaran mulai dari perencanaan hingga mengkomunikasikan hasil perkembangan anak memerlukan kondisi yang terkoordinasi bagi anak dan orang tua yang belajar di rumah.
4	Suwarto,	Pemanfaatan	Mempelajari,	Pada penelitian ini

No.	Nama Peneliti	Judul	Metode dan Tools	Hasil
	Ahmad Muzaki, Muhtarom 2021	Media YouTube sebagai Media Pembelajaran pada Siswa Kelas XII MIPA di SMA Negeri 1 Tawang Sari	menganalisis masalah, menggambarkan hasil masalah, memecahkan masalah dan menerapkan solusi	telah berhasil untuk memotivasi belajar siswa, meningkatkan aktivitas belajar siswa, meningkatkan kualitas diskusi, dan meningkatkan baik diskusi dengan siswa lain maupun diskusi dengan guru.
5	Syefriani Darnis 2018	Aplikasi <i>Montessori</i> Dalam Pembelajaran Membaca, Menulis Dan Berhitung Tingkat Permulaan Bagi Anak Usia Dini	Salah satu metode pembelajaran membaca, belajar menulis dan belajar berhitung pada usia prasekolah adalah metode Montessori.	Belajar membaca, menulis, dan berhitung di masa kanak-kanak harus dilakukan dengan pendekatan yang melibatkan anak-anak, tidak membebani mereka. Namun, lakukan saat bermain agar aktivitas utama anak usia ini, yaitu dunia bermainnya, tidak hilang.

No.	Nama Peneliti	Judul	Metode dan Tools	Hasil
6	Indah Purwitasari Ihsan, Muh. Sakir, Farida Yusuf 2021	Puzzle Elektronik <i>Montessori</i> Sand Board Letter(Sbl) Untuk Edukasi Suku Kata Bahasa Indonesia Pada Anak Usia Dini	Metode System Development Life Cycle(SDLC)	Berdasarkan uji fungsional dengan metode black box, bahwa keseluruhan fungsional sistem berfungsi sesuai dengan perancangan sistem. Hasil pengujian dan validasi oleh siswa Montessori bahwa sistem dirancang sesuai dengan aturan Montessori. Berdasarkan hasil observasi penggunaan, set puzzle elektronik Montessori Sand Board Letter (SBL) suku kata bahasa Indonesia selama 85 menit merangsang kemampuan anak untuk mengenal suku kata bahasa Indonesia.

### II.3 Kerangka Pemikiran

Pentingnya informasi sebagai alat untuk membantu masyarakat mendapatkan informasi, teknologi informasi dapat membantu memberikan informasi kepada masyarakat. Informasi dapat diperoleh dari hasil diskusi maupun *sharing* pengalaman melalui media/alat bantu yaitu teknologi.



Gambar II. 7 Kerangka Berfikir

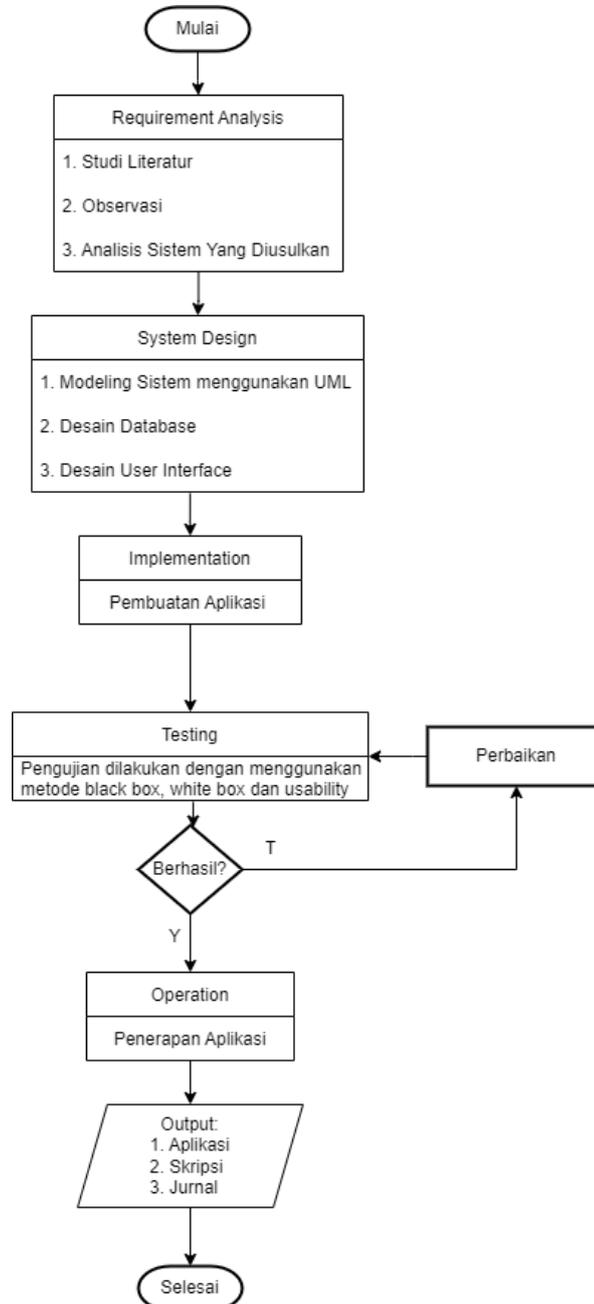
(Sumber : Pribadi)

# BAB III

## METODE PENELITIAN

### III.1 Tahap Penelitian

Tahapan penelitian yang dilakukan ditunjukkan pada gambar berikut:



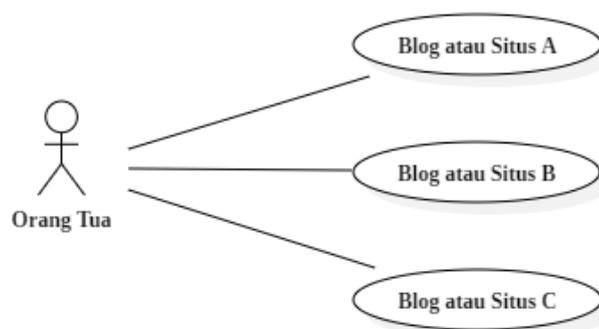
Gambar III. 1 Tahapan Penelitian

(Sumber : Pribadi)

Langkah penelitian dimulai dari dengan membuat analisis kebutuhan sistem diperoleh dari studi literatur, observasi dan membuat analisis sistem yang diusulkan, setelah itu lanjut ke proses mendesain sistem, pemodelan sistemnya menggunakan UML, menggambarkan pemodelan databasenya serta membuat desain rancangan *interface*, kemudian lanjut ke tahap implementasi yaitu membuat aplikasi sesuai dengan desain sistem yang dibuat sebelumnya, apabila aplikasi sudah dibuat maka pengujian *black box* akan dilakukan untuk memeriksa fungsionalitas sistem secara keseluruhan kemudian pengujian *white box* untuk memeriksa logika program dan pengujian efisiensi untuk memeriksa efisiensi aplikasi dengan kuesioner. Jika keseluruhan pengujian tersebut berhasil sesuai dengan tujuan pembuatan sistem maka aplikasi tersebut sudah bisa diterapkan, maka setelah itu *output* dari penelitian ini adalah aplikasi, skripsi dan jurnal.

### III.2 Rancangan Penelitian/Sistem

#### III.2.1 Analisa Sistem Berjalan



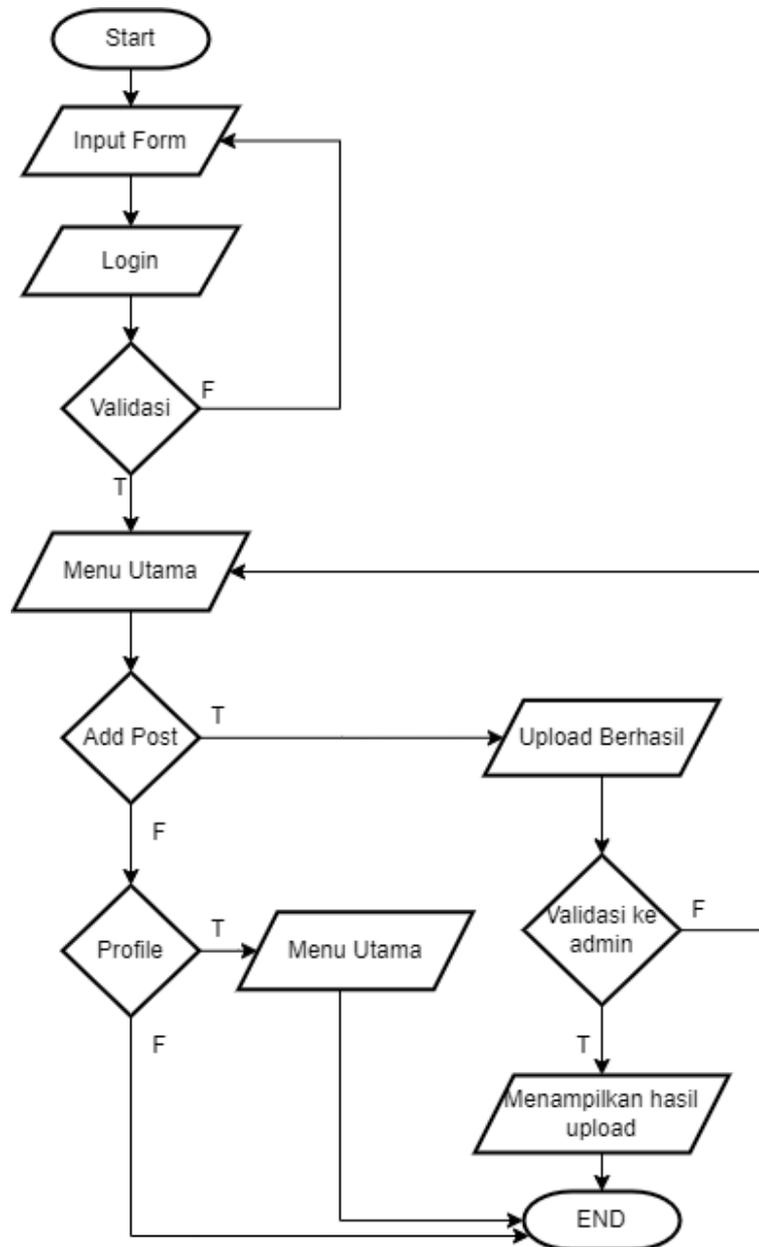
Gambar III. 2 Analisa Sistem Berjalan

(Sumber : Pribadi)

Orang tua masih mencari informasi metode pembelajaran *Montessori* dari blog 1 ke blog yang lain dan di blog tersebut belum menyediakan media *sharing*. Maka hal itu orang tua masih kesusahan untuk mengetahui metode pembelajaran *Montessori*.

### III.2.2 Flowchart Sistem

#### III.2.2.1 Flowchart Sistem Parent

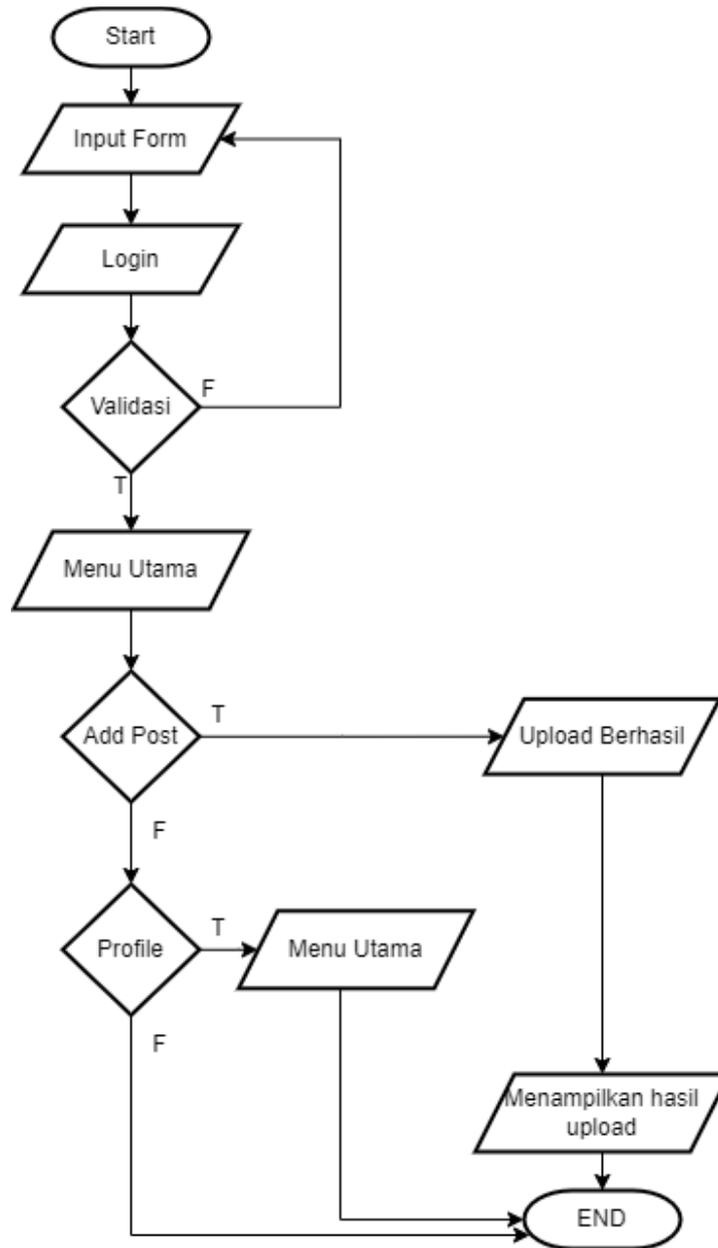


Gambar III. 3 *Flowchart Sistem Parent*

(Sumber : Pribadi)

*Flowchart* diatas menjelaskan alur kerja dari akun *parent*, akun *parent* memerlukan izin dari admin untuk dapat memposting hasil unggahannya.

### III.2.2.2 Flowchart Sistem Professional



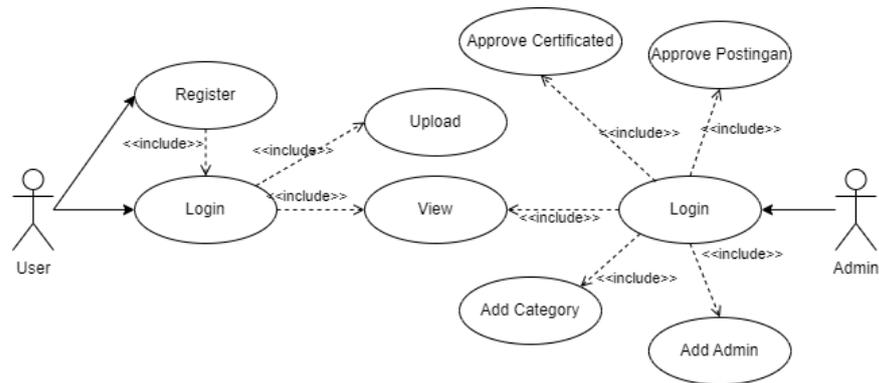
Gambar III. 4 Flowchart Sistem Professional

(Sumber : Pribadi)

Flowchart diatas menjelaskan alur kerja dari akun professional, akun professional tidak memerlukan izin dari admin untuk dapat memposting hasil unggahannya.

### III.2.3 Sistem yang Direncanakan

#### III.2.3.1 Use Case Diagram



Gambar III. 5 *Use Case Diagram*

(Sumber : Pribadi)

Terdapat 2 tipe akun *user* di aplikasi ini yaitu akun professional dan akun *Parent*. Definisi *use case* pada Gambar III.5 diatas dapat diuraikan sebagai berikut:

Tabel III. 1 Deskripsi *Use Case*

Use Case	Deskripsi
<i>Add Admin</i>	Tidak semua akun admin dapat melakukan <i>Add Admin</i> hanya akun super admin lah yang hanya dapat menambah admin.
<i>Add Category</i>	Semua akun admin dapat menambah <i>category</i> sesuai dengan yang diinginkan.
<i>Approve Cerificated</i>	Semua akun admin dapat menyetujui atau menolak sertifikat dari <i>user</i> .
<i>Approve Postingan</i>	Semua akun admin dapat menyetujui atau menolak postingan dari <i>user</i> .

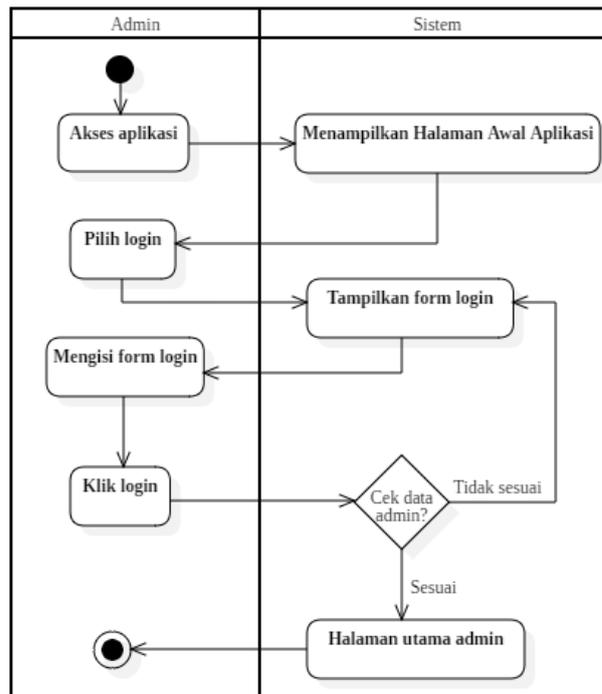
Use Case	Deskripsi
Detail Postingan	Akun Professional maupun akun <i>Parent</i> dan juga akun admin dapat melihat detail postingan, didalam detail postingan <i>user</i> dapat melakukan <i>like</i> dan <i>comment</i> .
<i>Upload</i>	<ul style="list-style-type: none"> <li>Professional dapat mengunggah informasi tanpa harus disetujui oleh admin.</li> <li><i>Parent</i> harus disetujui oleh admin agar hasil upload dapat ter posting.</li> </ul>

### III.2.3.2 Activity Diagram

Aktivitas digunakan untuk menunjukkan aliran pesan dari satu aktivitas ke aktivitas lainnya.

#### 1. Activity Diagram Admin

##### a. Activity Login Akun Admin

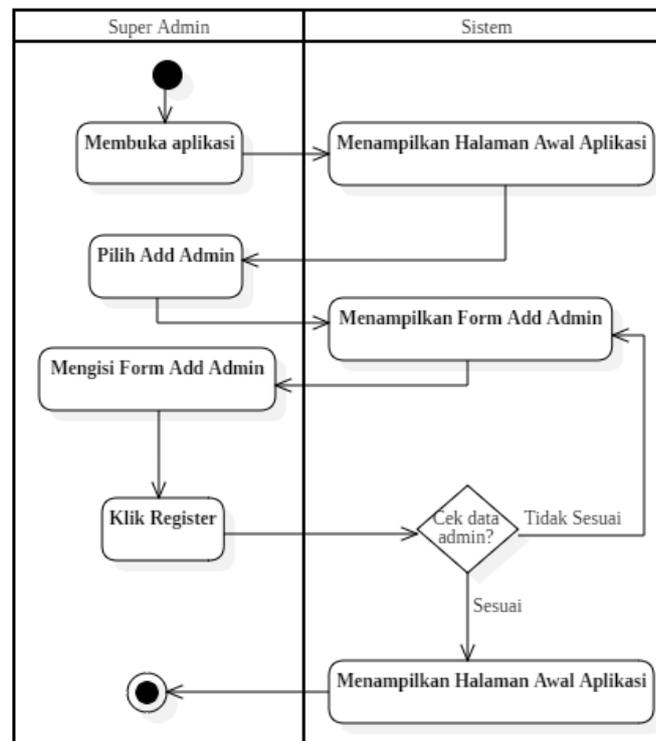


Gambar III. 6 Activity Diagram Login Akun Admin

(Sumber : Pribadi)

*Activity* diagram diatas dimulai dengan akun admin mengakses aplikasi lalu memilih menu *login*, akun admin dapat *login* dengan mengisi *form* menggunakan *email* dan *password* yang telah dibuat, setelah *login* admin diarahkan ke halaman utama dan admin dapat mengelola beberapa data.

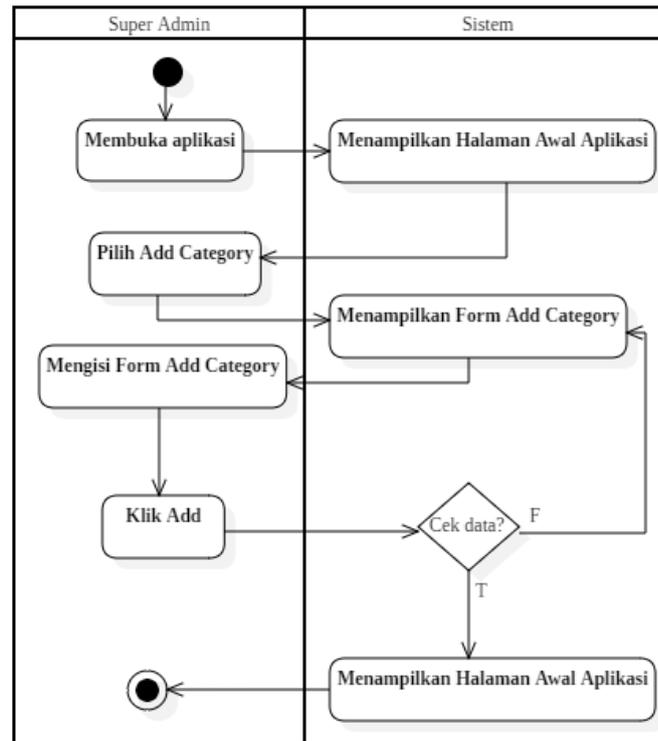
b. *Activity* Diagram Admin Add Admin



Gambar III. 7 *Activity* Diagram Admin Add Admin

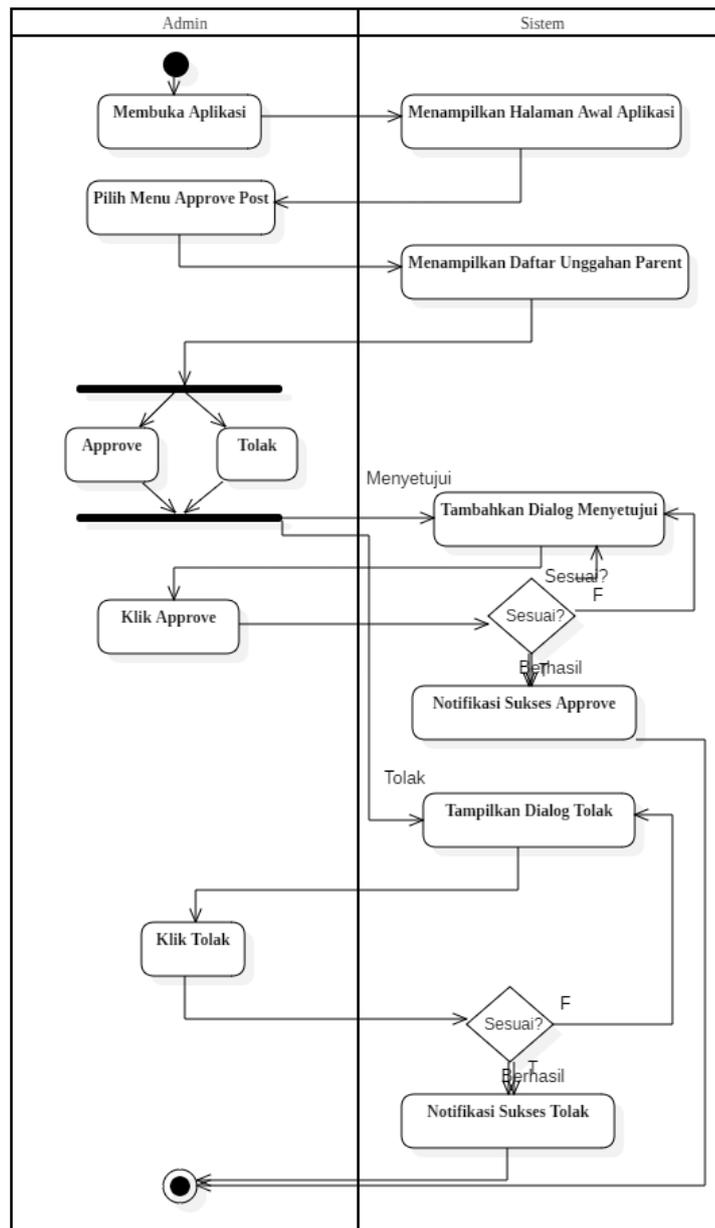
(Sumber : Pribadi)

*Activity* diagram diatas dimulai dengan akun admin mengakses aplikasi lalu memilih menu *Add* admin lalu admin mengisi *form* *add* admin setelah mengisi *form* tersebut admin akan dialihkan menuju halaman utama dari aplikasi. Menu *add* admin hanya dapat diakses oleh akun super admin.

c. *Activity Diagram Admin Add Category*Gambar III. 8 *Activity Diagram Admin Add Category*

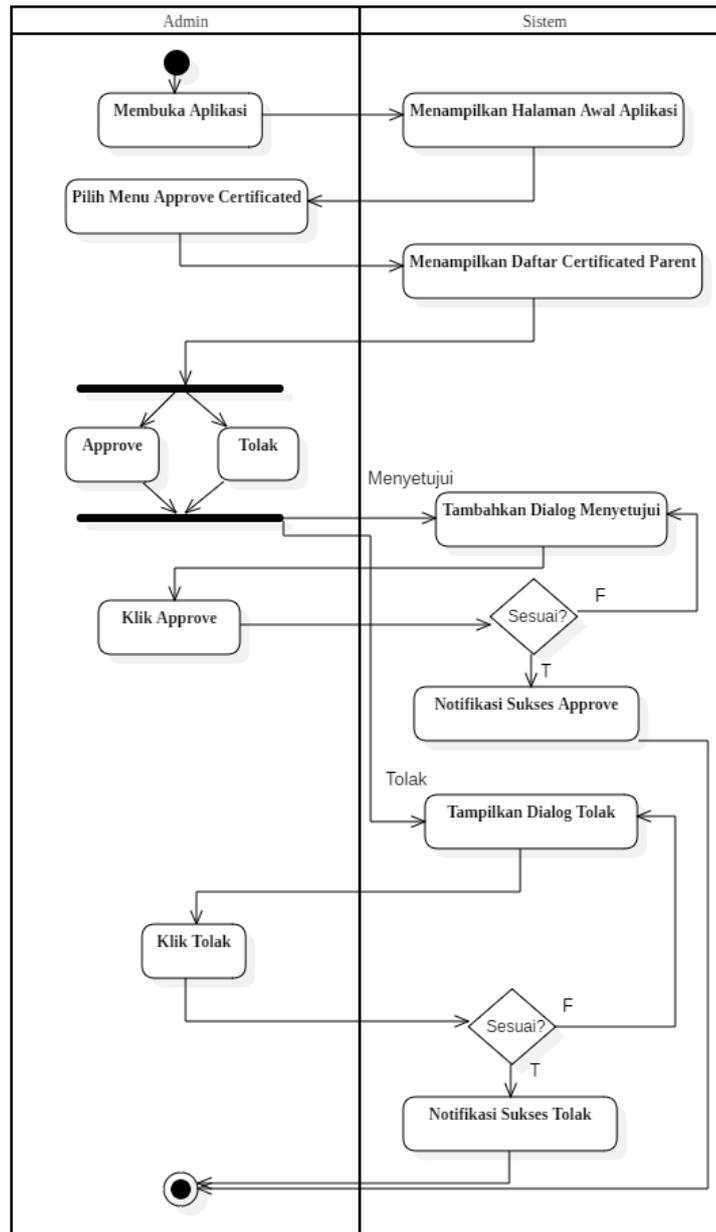
(Sumber : Pribadi)

*Activity* diagram diatas dimulai dengan akun admin mengakses aplikasi lalu memilih menu *add category* lalu admin mengisi *form add category* setelah mengisi *form* tersebut admin akan dialihkan menuju halaman utama dari aplikasi. Menu *add category* dapat menambahkan atau menghapus *category* baik dari *category learn* maupun *category age*. Menu *add category* ini dapat diakses oleh super admin dan juga sub admin.

d. *Activity Diagram Admin Approve Unggahan*Gambar III. 9 *Activity Diagram Admin Approve Unggahan*

(Sumber : Pribadi)

*Activity* diagram diatas dimulai dengan akun admin mengakses aplikasi lalu memilih menu *approve* unggahan lalu admin dapat memilih menolak ataupun menerima unggahan dari *user* agar dapat terposting.

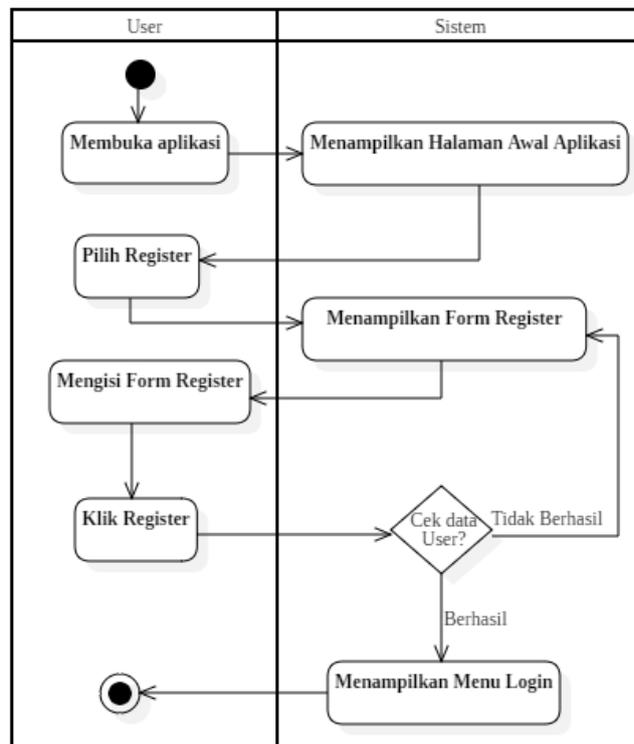
e. *Activity Diagram Admin Approve Certificated*Gambar III. 10 *Activity Diagram Admin Approve Certificated*

(Sumber : Pribadi)

*Activity* diagram diatas dimulai dengan akun admin mengakses aplikasi lalu memilih menu *approve certificated* lalu admin dapat memilih menolak ataupun menerima *certificated* dari *user*. Jika admin menerima *certificated* dari *user* maka *user* tersebut akan naik tingkat dari *user parent* ke *user professional*.

## 2. Activity Diagram *Parent* dan Professional

### a. Activity Register Akun *Parent* dan Professional

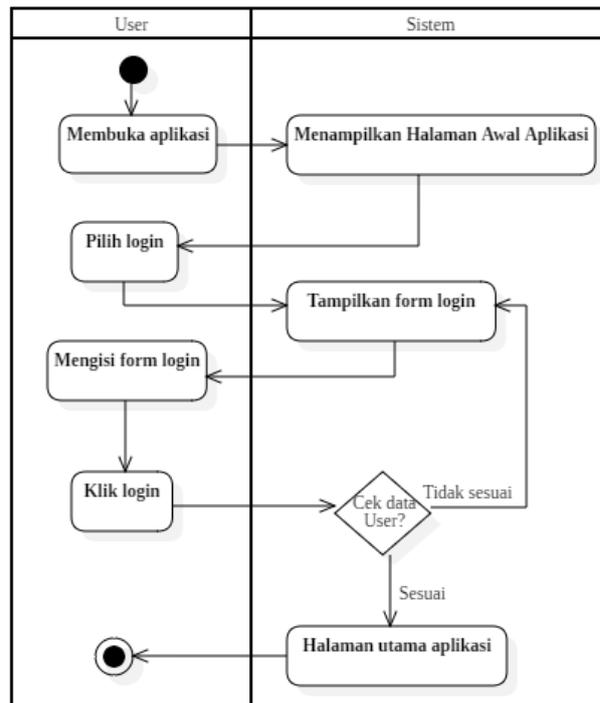


Gambar III. 11 Activity Diagram Register Akun *Parent* dan Professional

(Sumber : Pribadi)

*Activity* diagram diatas dimulai dengan akun *user* mengakses aplikasi lalu memilih menu register lalu *user* mengisi *form* register setelah mengisi *form* register data tersebut akan di cek terlebih dahulu apakah akun tersebut sudah terdaftar atau belum, jika belum terdaftar maka *user* akan dilahirkan menuju tampilan menu *login*.

b. *Activity Diagram Login Akun Parent dan Professional*

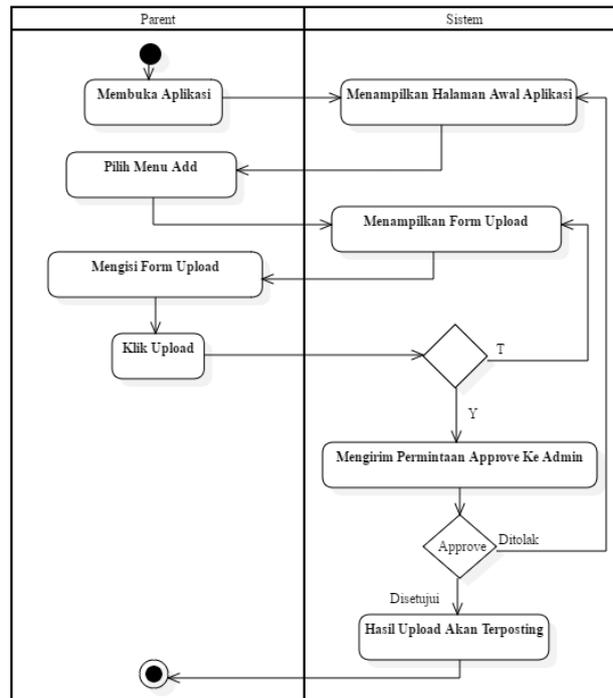


Gambar III. 12 *Activity Diagram Login Akun Parent dan Professional*

(Sumber : Pribadi)

*Activity* diagram diatas dimulai dengan akun *user* mengakses aplikasi lalu memilih menu login, akun *user* dapat *login* dengan mengisi *form* menggunakan *email* dan *password* yang telah dibuat, setelah *login user* diarahkan ke halaman utama aplikasi.

c. *Activity Diagram Mengunggah Parent*

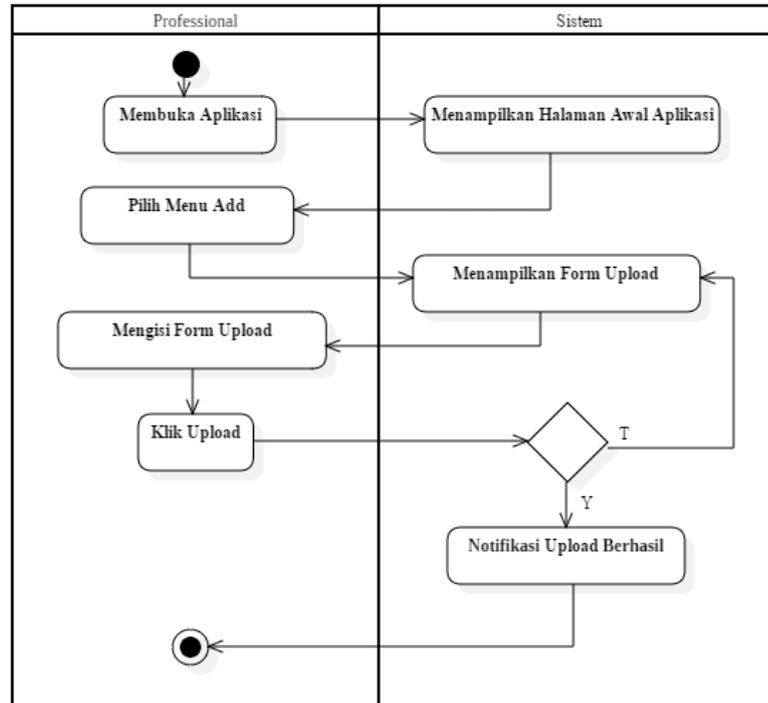


Gambar III. 13 *Activity Diagram Mengunggah Parent*

(Sumber : Pribadi)

*Activity diagram* diatas dimulai dengan akun *user parent* mengakses aplikasi lalu memilih menu *add*, di menu ini *user* dapat mengunggah informasi kegiatan *montessori* dengan mengisi *form add* setelah mengisi *user parent* harus memerlukan persetujuan dari admin agar hasil unggahan dapat terposting.

d. *Activity Diagram Mengunggah Professional*



Gambar III. 14 *Activity Diagram Mengunggah Professional*

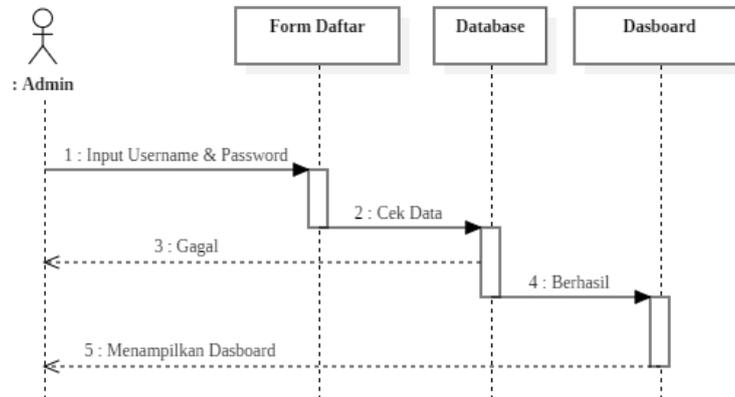
(Sumber : Pribadi)

*Activity diagram* diatas dimulai dengan akun *user* professional mengakses aplikasi lalu memilih menu *add*, di menu ini *user* dapat mengunggah informasi kegiatan *montessori* dengan mengisi *form add* setelah mengisi *user* tidak memerlukan persetujuan dari admin agar hasil unggahan dapat terposting.

### III.2.3.3 Sequence Diagram

#### 1. Sequence Diagram Akun Admin

##### a. Sequence Diagram Login Akun Admin

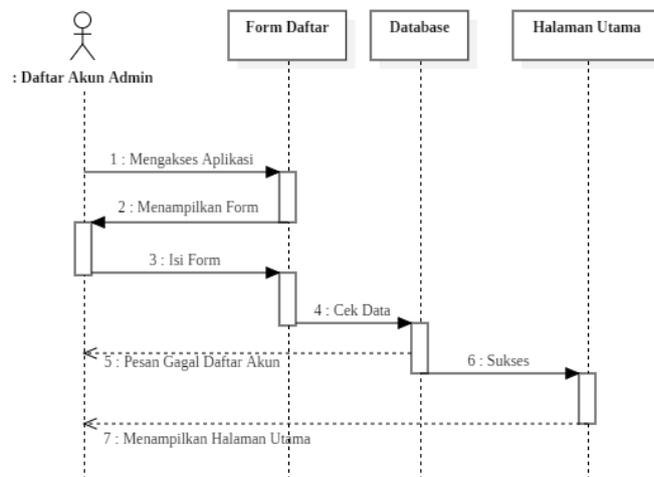


Gambar III. 15 Sequence Diagram Login Akun Admin

(Sumber : Pribadi)

Sequence diagram pada gambar 3.14 menjelaskan bahwa admin akan melakukan *login* dengan cara mengisi *form login* terlebih dahulu, setelah mengisi *form* dan juga menekan tombol *login* maka datanya akan dilakukan pemeriksaan jika data yang diinput sesuai maka akan dialihkan ke tampilan *dashboard* kepada admin jika tidak sesuai dengan data yang diinput maka akan muncul notifikasi gagal login.

##### b. Sequence Diagram Admin Add Admin

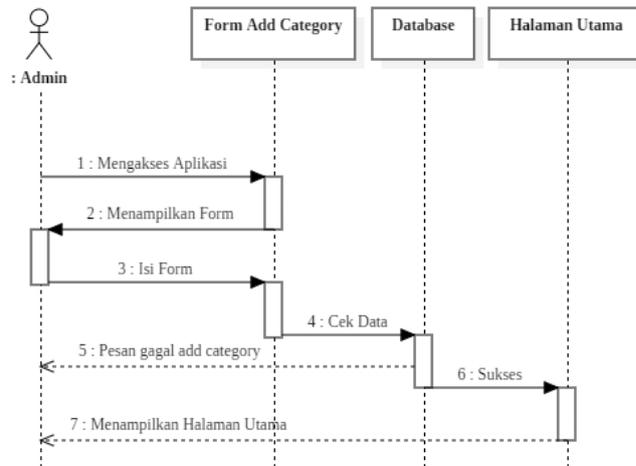


Gambar III. 16 Sequence Diagram Admin Add Admin

(Sumber : Pribadi)

*Sequence* diatas menjelaskan bahwa admin dapat melakukan penambahan admin dengan cara *Add Admin*.

c. *Sequence Diagram Admin Add Category*

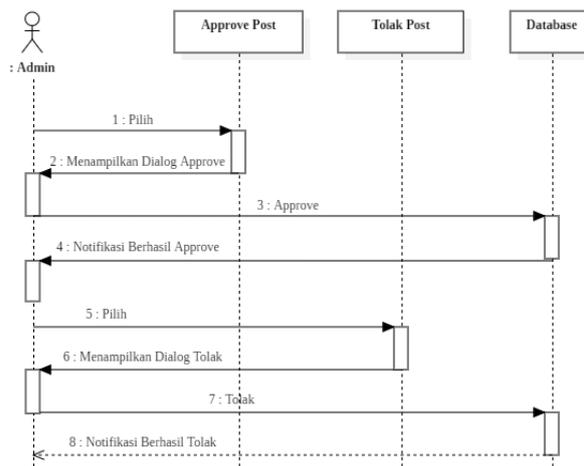


Gambar III. 17 *Sequence Diagram Admin Add Category*

(Sumber : Pribadi)

*Sequence* diatas menjelaskan bahwa admin dapat menambah *category* sesuai dengan kebutuhan.

d. *Sequence Diagram Admin Approve Post*



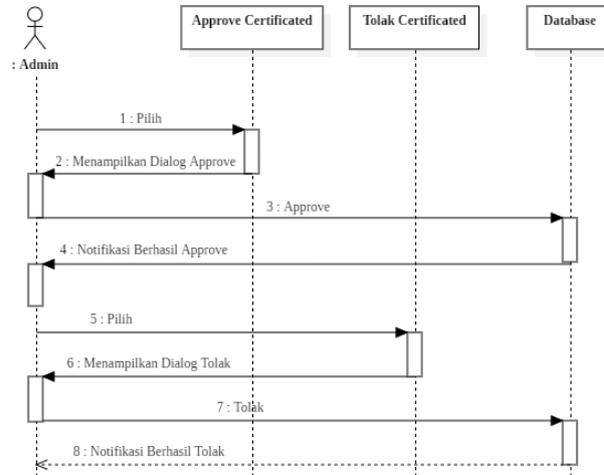
Gambar III. 18 *Sequence Diagram Admin Approve Post*

(Sumber : Pribadi)

*Sequence* diatas menjelaskan bahwa admin dapat menolak atau menyetujui postingan. Jika salah satu berhasil maka akan muncul

pesan sukses ke admin dan jika gagal akan menampilkan pesan gagal.

e. *Sequence Diagram Admin Approve Ceertificated*



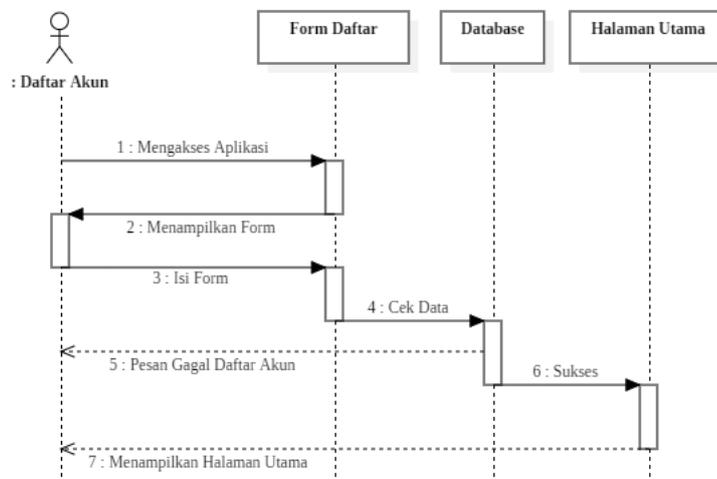
Gambar III. 19 *Sequence Diagram Admin Approve Ceertificated*

(Sumber : Pribadi)

*Sequence* diatas menjelaskan bahwa admin dapat menolak atau menyetujui postingan. Jika salah satu berhasil maka akan muncul pesan sukses ke admin dan jika gagal akan menampilkan pesan gagal.

2. *Sequence Diagram Akun Parent dan Professional*

a. *Sequence Diagram Register Akun User*

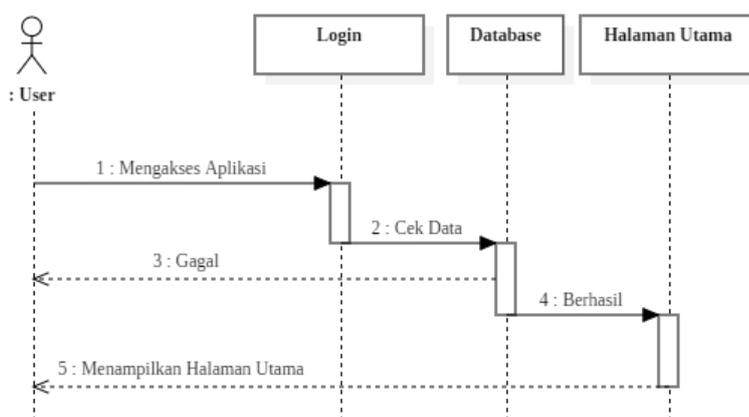


Gambar III. 20 *Sequence Diagram Register Akun User*

(Sumber : Pribadi)

*Sequence* diagram pada gambar 3.18 diatas dimulai dari *user* yang ingin mendaftar baik itu *Parent* maupun *professional* harus mengisi *form* pendaftaran terlebih dahulu setelah mengisi *form* pendaftaran tekan tombol daftar, setelah itu akan dilakukan pengecekan data, jika ada maka akan muncul notifikasi kesalahan dan jika data belum ada maka berhasil daftar dan akan menampilkan halaman utama.

b. *Sequence* Diagram Login Akun User

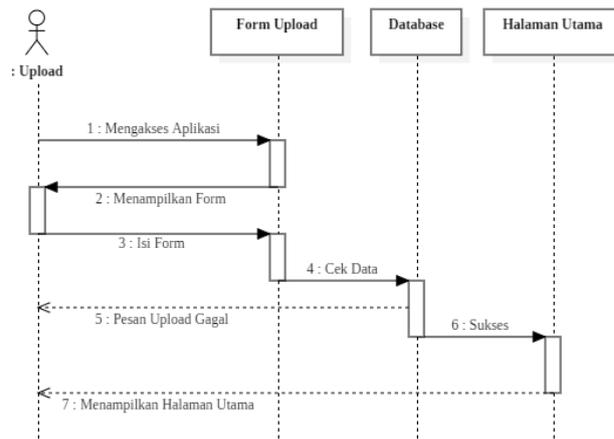


Gambar III. 21 *Sequence* Diagram Login Akun User

(Sumber : Pribadi)

*Sequence* diagram pada gambar 3.19 diatas menjelaskan bahwa user akan login dengan mengisi form login terlebih dahulu setelah mengisi form login dan menekan tombol login, maka data akan dicek terlebih dahulu, jika data benar maka akan ditampilkan halaman utama kepada user dan jika tidak maka akan muncul pesan error.

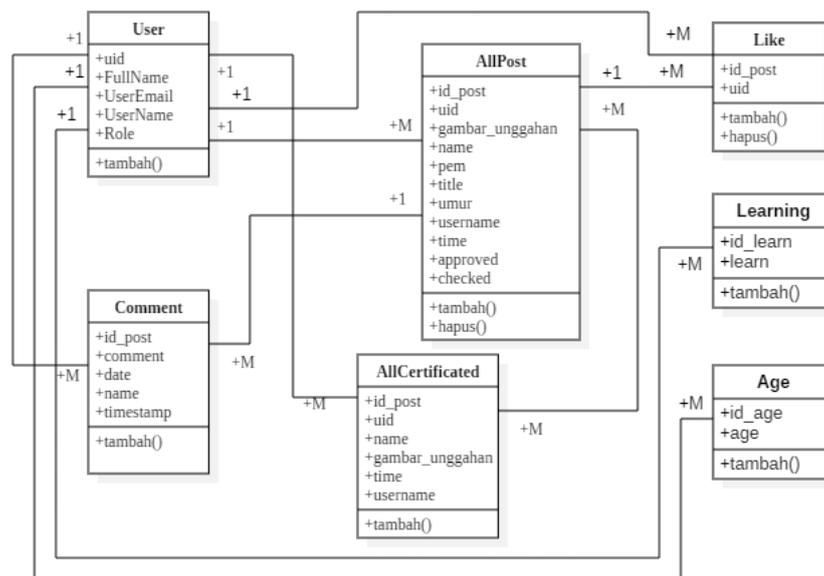
c. *Sequence* Diagram User Mengunggah



Gambar III. 22 *Sequence Diagram User Kelola Unggahan*  
(Sumber : Pribadi)

*Sequence* diatas menjelaskan bahwa pengguna mengupload. Jika salah satunya berhasil akan menampilkan pesan sukses kepada pengguna dan jika gagal akan menampilkan pesan gagal.

### III.2.3.4 *Class Diagram*



Gambar III. 23 *Class Diagram*  
(Sumber : Pribadi)

Diagram kelas atau *Class Diagram* adalah diagram yang digunakan untuk menunjukkan kelas-kelas yang ada dalam sistem

sebagai objek yang berevolusi dari satu kelas ke kelas lain dengan hubungan. *Class* diagram diatas menjelaskan bahwa terdapat 7 buah tabel yang saling berhubungan satu sama lain.

#### III.2.3.5 Kamus Data

Nama tabel : *User*

Fungsi : Untuk menyimpan data *user* dan admin

*Primary Key* : uid

*Foreign Key* : -

Tabel III. 2 Kamus Data *User*

No.	Nama Kolom	Tipe Data	Lebar	Keterangan	Key
1.	Uid	String	50	Id pengguna	PK
2.	FullName	String	50	Nama pengguna	
3.	<i>UserEmail</i>	String	50	Email pengguna	
4	<i>UserName</i>	String	50	<i>Username</i> pengguna	
5.	Role	String	50	Role Pengguna	

Nama tabel : *AllPost*

Fungsi : Untuk menyimpan data postingan

*Primary Key* : *id\_post*

*Foreign Key* : uid

Tabel III. 3 Kamus Data *AllPost*

No.	Nama Kolom	Tipe Data	Lebar	Keterangan	Key
1.	<i>id_post</i>	String	50	<i>Id post</i>	PK
2.	Uid	String	50	Id pengguna	FK
3.	<i>gambar_unggah</i>	String	50	Gambar	
4.	name	String	50	Nama	
5.	Pem	String	50	Pembelajaran	
6.	titile	String	50	Judul	
7.	Umur	String	50	umur	
8.	<i>username</i>	String	50	<i>username</i>	
9.	Time	String	50	waktu	

Nama tabel : *AllCertificated*

Fungsi : Untuk menyimpan data *certificated user*

*Primary Key* : *id\_certificated*

*Foreign Key* : *uid*

Tabel III. 4 Kamus Data AllCertificated

No.	Nama Kolom	Tipe Data	Lebar	Keterangan	Key
1.	<i>id_certificated</i>	String	50	Id sertifikat	PK
2.	<i>uid</i>	String	50	Id pengguna	FK
3.	<i>gambar_unggahan</i>	String	50	Gambar	
4.	<i>name</i>	String	50	Nama	
5.	<i>titile</i>	String	50	Judul	
6.	<i>username</i>	String	50	<i>username</i>	
7.	<i>time</i>	String	50	waktu	

Nama tabel : *Comment*

Fungsi : Untuk menyimpan data comment

*Primary Key* : *id\_post*

*Foreign Key* : *uid*

Tabel III. 5 Kamus Data *Comment*

No.	Nama Kolom	Tipe Data	Lebar	Keterangan	Key
1.	<i>id_post</i>	String	50	<i>Id post</i>	PK
2.	<i>Uid</i>	String	50	Id pengguna	FK
3.	<i>Nama</i>	Varchar	50	<i>Nama User</i>	
4.	<i>comment</i>	String	50	<i>Comment user</i>	
5.	<i>date</i>	String	50	waktu	
6.	<i>timestamp</i>	number	16	waktu	

Nama tabel : *Like*

Fungsi : Untuk menyimpan data comment

*Primary Key* : *id\_post*

*Foreign Key* : *uid*

Tabel III. 6 Kamus Data Like

No.	Nama Kolom	Tipe Data	Lebar	Keterangan	Key
1.	<i>id_post</i>	String	50	<i>Id post</i>	PK
2.	Uid	boolean		Id pengguna	FK

Nama tabel : *Learning*

Fungsi : Untuk menyimpan data *category learning*

*Primary Key* : *id\_learn*

*Foreign Key* : -

Tabel III. 7 Kamus Data *Learning*

No.	Nama Kolom	Tipe Data	Lebar	Keterangan	Key
1.	<i>id_learn</i>	String	50	Id learn	PK
2.	learn	String	50	Category learn	

Nama tabel : *Age*

Fungsi : Untuk menyimpan data *category Age*

*Primary Key* : *id\_age*

*Foreign Key* : -

Tabel III. 8 Kamus Data *Age*

No.	Nama Kolom	Tipe Data	Lebar	Keterangan	Key
1.	<i>id_age</i>	String	50	Id learn	PK
2.	age	String	50	<i>Category age</i>	

### III.2.3.6 Design Interface

#### a. Tampilan *Menu Start* Aplikasi



Gambar III. 24 Tampilan Menu Start  
(Sumber : Pribadi)

#### b. Tampilan *Menu Register* Aplikasi

Create an account

Register to get started with the app.

Your full name

Your email

Your username

Your password

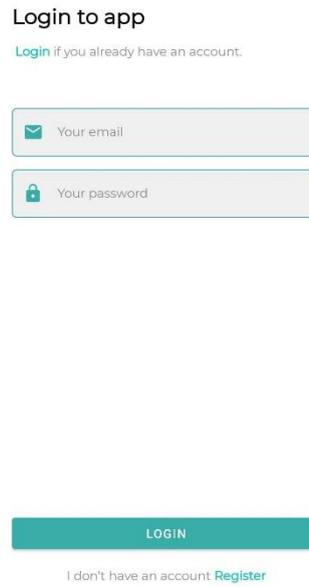
Confirm password

Register As Parents  Register As Professional

REGISTER

I already have an account [Login](#)

Gambar III. 25 Tampilan Menu Register  
(Sumber : Pribadi)

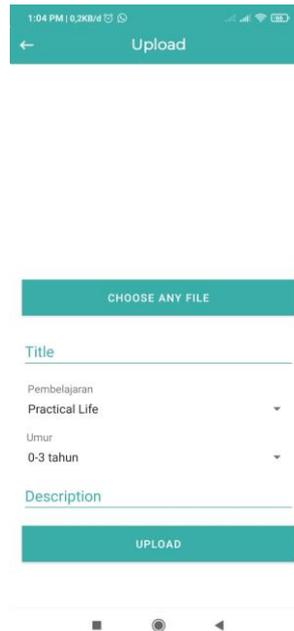
c. Tampilan *Menu Login* Aplikasi

The screenshot shows a login interface with the following elements:

- Title: **Login to app**
- Text: [Login](#) if you already have an account.
- Input field 1:  Your email (with an envelope icon)
- Input field 2:  Your password (with a lock icon)
- Button: **LOGIN**
- Text: I don't have an account [Register](#)

Gambar III. 26 Tampilan Menu *Login*  
(Sumber : Pribadi)

## d. Tampilan Menu Add Post

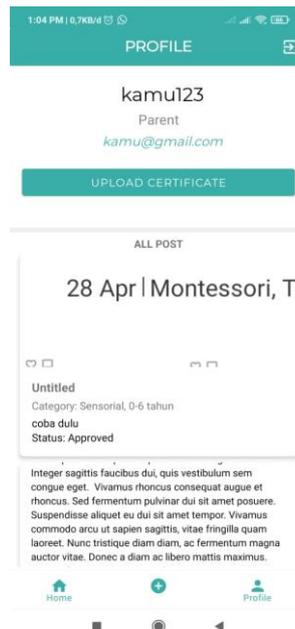


The screenshot shows an add post interface with the following elements:

- Header: **Upload** (with a back arrow)
- Text: **CHOOSE ANY FILE**
- Text: **Title**
- Text: Pembelajaran **Practical Life** (with a dropdown arrow)
- Text: Umur **0-3 tahun** (with a dropdown arrow)
- Text: **Description**
- Button: **UPLOAD**

Gambar III. 27 Tampilan Add Post  
(Sumber : Pribadi)

e. Tampilan *Menu Profile*



Gambar III. 28 Tampilan Menu Profile

(Sumber : Pribadi)

### III.3 Waktu dan Lokasi Penelitian

Waktu penelitian selama 11 bulan dari bulan Oktober sampai Agustus 2022. Penelitian ini dilakukan secara online..

### III.4 Alat dan Bahan

Persyaratan desain aplikasi meliputi perangkat keras dan perangkat lunak, antara lain:

Ada beberapa Perangkat Keras atau *Hardware* yang digunakan antara lain:

Tabel III. 9 Perangkat Keras

No.	Nama Perangkat	Unit	Spesifikasi
1.	Laptop Asus	1	- Intel Core i7 - Ram 8 Gb - SSD 256 Gb - VGA Nvidia Geforce 610M

No.	Nama Perangkat	Unit	Spesifikasi
2.	Hp Android	1	- Xiaomi Redmi Note 7 - 128 Gb - 4 Gb Ram
3.	Mouse	1	Logitech

Ada beberapa Perangkat lunak atau *Software* yang digunakan antara lain::

Tabel III. 10 Perangkat Lunak

No.	Nama Perangkat	Unit	Spesifikasi
1.	Visual Studio Code	1	- Intel Core i7 - Ram 8 Gb - SSD 256 Gb - VGA Nvidia Geforce 610M
2.	Database	1	Firestore
3.	Bahasa Pemrograman	1	Java Native

### III.5 Metode Pengumpulan Data

Metode pengumpulan data yang digunakan dalam penelitian ini adalah sebagai berikut:

Tabel III. 11 Metode Pengumpulan data

No.	Jenis	Keterangan
1.	Studi Literatur	Mengumpulkan data dengan cara mempelajari hal-hal yang berkaitan dengan penelitian.
2.	Studi Dokumentasi	Mempelajari penelitian yang dilakukan seperti dokumentasi pemrograman, seperti <i>source code</i> yang digunakan
3.	Observasi	Mengamati proses berjalannya aplikasi yang telah dibuat secara langsung.

## BAB IV

### HASIL DAN PEMBAHASAN

#### IV.1 Hasil

Pada bab ini akan disajikan hasil dari pengembangan aplikasi *Montessori* pad asisten untuk media *sharing* seputar metode pembelajaran *Montessori* yang berguna untuk membantu orang tua dalam mendapatkan informasi seputar metode pembelajara serta ide aktifitas kegiatan *Montessori*. Pada bagian hasil aplikasi akan menampilkan tangkapan layar dari aplikasi yang telah dibangun.

##### IV.1.1 Hasil Aplikasi

1. Aplikasi
  - a. Tampilan Awal Aplikasi



Gambar IV. 1 Tampilan Awal Aplikasi

(Sumber : Pribadi)

Tampilan awal aplikasi merupakan implementasi untuk fitur autentikasi secara otomatis di mana dilakukan pengecekan autentikasi pengguna. Setelah pengecekan selesai, pengguna akan diarahkan secara otomatis ke halaman *dashboard* apabila pengguna telah terautentikasi sebelumnya.

b. *Login*

The image shows a login interface with the following elements:

- Title: **Login to app**
- Text: [Login](#) if you already have an account.
- Input field: Your email (with an envelope icon)
- Input field: Your password (with a lock icon)
- Button: **LOGIN**
- Text: I don't have an account [Register](#)

Gambar IV. 2 Tampilan *Login*

(Sumber : Pribadi)

Tampilan *login* merupakan implementasi untuk fitur *login*. Pada halaman ini pengguna dapat melakukan autentikasi dengan memasukkan email serta kata sandi pengguna yang telah terdaftar pada kolom yang tersedia lalu kemudian menekan tombol *login* untuk melakukan proses *login*.

Dengan menekan tombol *login* maka sistem akan validasi apakah email yang dimasukan terdaftar atau tidak di dalam firebase. Jika email telah terdaftar selanjutnya sistem akan mengecek *password* yang dimasukan benar atau salah, jika email dan *password* yang dimasukan salah atau tidak terdaftar dalam firebase maka akan muncul notifikasi email / *password* salah



### c. Register

**Create an account**  
 Register to get started with the app.

Your full name

Your email

Your username

Your password

Confirm password

Register As Parents  Register As Professional

**REGISTER**

I already have an account [Login](#)

Gambar IV. 3 Tampilan Register

(Sumber : Pribadi)

Tampilan register merupakan implementasi untuk fitur daftar akun. Pada halaman ini pengguna dapat melakukan membuat akun, dengan menekan tombol register maka sistem akan validasi apakah email yang dimasukan telah terdaftar atau tidak di dalam firebase. Jika email belum terdaftar maka selanjutnya akan muncul notifikasi *account created*.

### d. Halaman Utama (*User*)



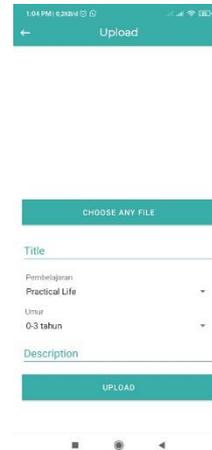
Gambar IV. 4 Tampilan Halaman Utama *User*

(Sumber : Pribadi)

Tampilan halaman utama *user* adalah tampilan yang ditemui oleh pengguna setelah tampilan *login*, tampilan halaman utama,

terdapat 3 menu utama yaitu menu *home* sebagai halaman utama, menu *add* sebagai tempat untuk mengupload postingan yang berupa *image* dan menu utama yang terakhir itu ialah menu *profile*.

e. Menu Add Post (*User*)

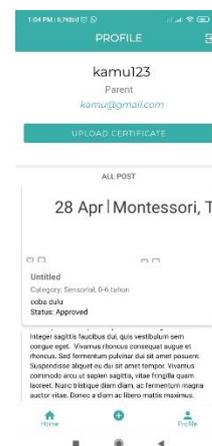


Gambar IV. 5 Tampilan Menu Add Post

(Sumber : Pribadi)

Tampilan halaman menu *add post* adalah tampilan untuk mengupload postingan yang berupa *image* baik *user Parent* maupun *user professional* dapat mengakses menu *add post*.

f. Menu Profile (*User*)



Gambar IV. 6 Tampilan Menu Profile

(Sumber : Pribadi)

Tampilan halaman menu *profile* adalah tampilan untuk dapat mengakses informasi *user* yang sedang *login*. Dan juga di menu ini *user* dapat melihat semua postingan yang telah diupload baik yang sudah di *approved* maupun belum di *approved*.

g. Halaman Detail Postingan



Gambar IV. 7 Tampilan Halaman Detail Postingan

(Sumber : Pribadi)

Tampilan halaman detail postingan adalah tampilan untuk mengakses seluruh informasi yang telah di upload oleh *user* tersebut.

h. Halaman Utama (Admin & Super Admin)



Gambar IV. 8 Tampilan Halaman Utama Admin

(Sumber : Pribadi)

Tampilan halaman utama admin adalah tampilan yang ditemui oleh pengguna admin setelah tampilan *login*, tampilan halaman utama, terdapat 4 menu utama yaitu menu *List Approval Post* (Admin), menu *List Approval Certificated* (Admin), menu *All Post* dan menu yang terakhir itu iyalah menu *Add Admin*.

i. List Approval Post (Admin)



Gambar IV. 9 Tampilan List Approval Admin

(Sumber : Pribadi)

Tampilan halaman *List Approval Post* (Admin) merupakan tampilan yang berisi list postingan yang belum ter*Approve* yang telah di kirimkan oleh *user* untuk admin.

j. Halaman *Approve* Post (Admin)

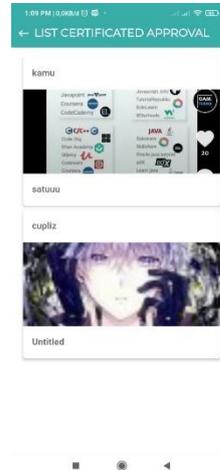


Gambar IV. 10 Tampilan Halaman *Approve* Post Admin

(Sumber : Pribadi)

Tampilan halaman Approval Post (Admin) merupakan tampilan yang berisi list postingan yang belum *approve* yang telah di kirimkan oleh *user* untuk admin.

k. List Approval Certificated (Admin)



Gambar IV. 11 Tampilan *List Approval Certificated*

(Sumber : Pribadi)

Tampilan halaman List Approval Post (Admin) merupakan tampilan yang berisi list postingan yang belum *terApprove* yang telah di kirimkan oleh *user* untuk admin.

l. Halaman *Approve Certificated* (Admin)



Gambar IV. 12 Tampilan Halaman *Approve Certificated*

(Sumber : Pribadi)

Tampilan halaman *List Approve Certificated* (Admin) merupakan tampilan yang berisi *certificated* yang belum terApprove.

m. *Add Admin* (Super Admin)

Gambar IV. 13 Tampilan *Add Admin*

(Sumber : Pribadi)

Tampilan halaman *Add Admin* (Super Admin) merupakan tampilan yang berisi list postingan yang belum terApprove.

## IV.2 Pembahasan

Pada bagian pembahasan akan terbagi menjadi 3 yaitu bagian pengujian *black box*, *white box* dan kelayakan. Pengujian fungsionalitas perangkat lunak dilakukan dengan menggunakan metode pengujian langsung berdasarkan teknik *black box testing* dengan menguji tujuan fungsi yang dibuat dengan kesesuaian pada aplikasi saat dijalankan. Sedangkan *white box testing* didasarkan pada pengecekan terhadap detail dari logika program yang telah dibuat. Bagian terakhir yaitu pengujian efektifitas menggunakan kuesioner yang dibagikan melalui Google *Form* kepada 10 penguji aplikasi.

### IV.2.1 Pengujian Black Box

Berikut ini hasil dari pengujian fungsional sistem, menggunakan metode *black box testing* untuk memastikan keseluruhan fungsi – fungsi dari sistem telah berjalan sebagaimana yang diharapkan. Hasil

pengujian fungsional ditampilkan dalam bentuk tabel pada tabel IV.1. berikut :

Tabel IV. 1 Pengujian *Black Box*

No.	Deskripsi Uji Coba	Sampel Data	Hasil yang diinginkan	Hasil yang sebenarnya
1.1	Pengguna menekan tombol “ <i>Login</i> ” dengan menginput email yang sudah diverifikasi & kata sandi yang benar.	Email:kamu@gmial.com <i>Password:</i> kacamata11	Mengarahkan pengguna ke tampilan <i>dashboard</i> setelah berhasil <i>login</i> .	Sesuai dengan yang diharapkan
	Pengguna menekan tombol “Masuk” dengan input e-mail yang tidak <i>valid</i> .	E-mail: asdfghjkl <i>Password:</i> asdfghjkl	Menampilkan pesan kesalahan “The email <i>address</i> is badly <i>formatted</i> ” pada kolom e- mail.	Sesuai dengan yang diharapkan
	Pengguna menekan tombol “ <i>Login</i> ” dengan menginput email yang <i>valid</i> dan kata sandi yang salah.	Email:akukamu@gmail.com <i>Password:</i> 123456	Menampilkan pesan kesalahan “The <i>password</i> is invalid or the <i>user</i> does not have a <i>password</i> ” pada kolom kata sandi.	Sesuai dengan yang diharapkan

No.	Deskripsi Uji Coba	Sampel Data	Hasil yang diinginkan	Hasil yang sebenarnya
	Pegguna menekan tombol “Login” dengan menginput email yang belum terdaftar.	Email: akukamuaku@gmail.com Password: 1234567	Menampilkan pesan kesalahan “There is no user record corresponding to the identifier. The user may have been deleted.” pada kolom e-mail.	Sesuai dengan yang diharapkan
2.1	Pegguna menekan tombol “Register” dengan menginput fullname, email, username, password, confirm password	Fullname : Data Skripsi Email: <a href="mailto:dataskripsi2022@gmail.com">dataskripsi2022@gmail.com</a> Username : dataskripsi22 Password : 1234567 Confirm password : 1234567	Menampilkan pesan “Account created” setelah itu pengguna akan dialihkan ke tampilan login	Sesuai dengan yang diharapkan
2.2	Pegguna menekan tombol “Register” dengan kolom fullname, email, username, password, confirm password yang masih kosong	Fullname : Email: Username : Password : Confirm password :	Menampilkan pesan “Please fill in all data!”	Sesuai dengan yang diharapkan
3.1	Pegguna terautentifikasi pada aplikasi	--	Mengarahkan pengguna ke halaman dashboard setelah	Sesuai dengan yang

No.	Deskripsi Uji Coba	Sampel Data	Hasil yang diinginkan	Hasil yang sebenarnya
			proses autentifikasi	diharapkan
3.2	Pengguna tidak terautentifikasi pada aplikasi	--	Mengarahkan pengguna ke halaman start setelah proses autentifikasi	Sesuai dengan yang diharapkan
4.1	Pengguna berhasil melakukan logout	--	Mengarahkan pengguna ke halaman <i>login</i>	Sesuai dengan yang diharapkan
5.1	Pengguna akun menambahkan postingan dengan mengisi semua kolom pada halaman upload	--	Menampilkan pesan "Image Uploaded"	Sesuai dengan yang diharapkan
5.2	Pengguna menekan tombol "Upload" dengan tanpa mengisi semua kolom pada halaman upload		Menampilkan pesan "Please fill all field"	Sesuai dengan yang diharapkan
6.1	Pengguna akun " <i>Parent</i> "	--	Menampilkan pesan	Sesuai dengan

No.	Deskripsi Uji Coba	Sampel Data	Hasil yang diinginkan	Hasil yang sebenarnya
	menambahkan foto sertifikat		“Certificated Uploaded”	yang diharapkan
7.1	Pengguna memilih salah satu postingan yang ingin dilihat detailnya	--	Mengarahkan pengguna ke detail postingan	Sesuai dengan yang diharapkan
8.1	Pengguna memilih salah satu <i>category</i> yang ingin dilihat detailnya	--	Mengarahkan pengguna ke detail category	Sesuai dengan yang diharapkan
9.1	Pengguna berusaha menghapus postingan yang bukan miliknya	--	Icon hapus akan hilang pada detail <i>post</i>	Sesuai dengan yang diharapkan
9.2	Pengguna berhasil menghapus <i>post</i> yang telah diunggahnya	--	Icon hapus akan terlihat pada detail <i>post</i>	Sesuai dengan yang diharapkan
10.1	Pengguna yang memiliki akun dengan role “Professional”	--	Tombol Upload Sertifikat akan hilang pada menu <i>profile</i>	Sesuai dengan yang

No.	Deskripsi Uji Coba	Sampel Data	Hasil yang diinginkan	Hasil yang sebenarnya
				diharapkan
10.2	Pengguna yang memiliki akun role “ <i>Parent</i> ”	--	Tombol Upload Sertifikat akan terlihat pada menu <i>profile</i>	Sesuai dengan yang diharapkan
11.1	Admin <i>Approve</i> postingan <i>user</i>	--	Menampilkan pesan “ <i>namaUser</i> + <i>post Approved</i> ”	Sesuai dengan yang diharapkan
11.2	Admin <i>rejected</i> postingan <i>user</i>	--	Menampilkan pesan “ <i>namaUser</i> + <i>post rejected</i> ”	Sesuai dengan yang diharapkan
12.1	Admin <i>Approve</i> certificated <i>user</i>	--	Menampilkan pesan “ <i>namaUser</i> + <i>certificated Approved</i> ”	Sesuai dengan yang diharapkan
12.2	Admin <i>rejected</i> certificated <i>user</i>	--	Menampilkan pesan “ <i>namaUser</i> + <i>certificated rejected</i> ”	Sesuai dengan yang diharapkan

No.	Deskripsi Uji Coba	Sampel Data	Hasil yang diinginkan	Hasil yang sebenarnya
13.1	Pegguna menekan tombol "Register" dengan menginput <i>username</i> , <i>email</i> , <i>password</i>	Email: <i>cipto@gmail.com</i> <i>Username</i> : <i>cipto</i> <i>Password</i> : <i>1234567</i>	Menampilkan pesan " <i>Account created</i> " setelah itu pengguna akan dialihkan ke tampilan halaman awal admin	Sesuai dengan yang diharapkan
13.2	Pegguna menekan tombol "Register" dengan kolom <i>email</i> , <i>username</i> , <i>password</i> yang masih kosong	Email: <i>Username</i> : <i>Password</i> :	Menampilkan pesan "Please fill in all data!"	Sesuai dengan yang diharapkan

Tabel IV. 2 Rekapitulasi Hasil Pengujian *Blackbox*

No	Modular	Berhasil	Tidak berhasil
1.	Pengujian <i>login</i>	✓	-
2.	Pengujian <i>register</i>	✓	-
3.	Pengujian <i>autentifikasi</i> otomatis	✓	-
4.	Pengujian <i>logout</i>	✓	-
5.	Pengujian <i>add post</i>	✓	-
6.	Pengujian <i>add certificated</i>	✓	-
7.	Pengujian mengakses <i>detail post</i>	✓	-
8.	Pengujian mengakses <i>detail category</i>	✓	-
9.	Pengujian menghapus <i>post</i>	✓	-
10.	Pengujian tombol <i>upload certificated</i>	✓	-
11.	Pengujian <i>Approve post</i>	✓	-
12.	Pengujian <i>Approve certificated</i>	✓	-
13.	Pengujian <i>Add Admin</i>	✓	-

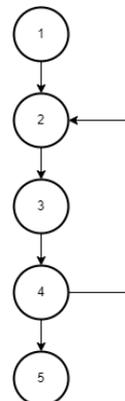
Berdasarkan Tabel IV.2 didapatkan hasil rekapitulasi maka dapat disimpulkan dari semua pengujian fungsional menggunakan metode *black box*, menyatakan bahwa semua fungsi dari sistem bekerja sesuai dengan fungsinya.

#### IV.2.2 Pengujian *White Box*

Pengujian *white box* dilakukan untuk menguji kebenaran logika program yang telah dibuat.

##### a. Pengujian Register

Menu registrasi digunakan untuk *user* yang berfungsi untuk mendaftarkan identitas pengguna sebelum memasuki aplikasi tersebut.

Gambar IV. 14 *Flow Graph* Register

(Sumber : Pribadi)

Dari *flowgraph registrasi* diatas dapat diketahui terdapat 5 *edge* dan 5 *node*, berikut ini adalah potongan-potongan *script* dari *node* diatas yaitu:

*Script Node 1 : Start*

```
setContentView(R.layout.Activity_register);
```

*Script Node 2 : Input form register*

```
checkField(fullName);
    checkField(email);
    checkField(password);
    checkField(username);
checkField(confirmPs);
public boolean checkField(EditText textField) {
    if (textField.getText().toString().isEmpty()) {
        textField.setError("Error");
        Toast.makeText(this, "Please fill in all data!",
Toast.LENGTH_SHORT).show();
        valid = false;
    } else {
        valid = true;
    }
    return valid;
}
```

*Script Node 3 : Cek data apakah data sudah ada atau belum*

```
if (valid) {
fAuth.createUserWithEmailAndPassword(email.getText().toString
(), password.getText().toString()).addOnSuccessListener(new
OnSuccessListener<AuthResult>() {
}).addOnFailureListener(new OnFailureListener() {
@Override
```

```

        public void onFailure(@NonNull Exception
e) {
            Toast.makeText(RegisterActivity.this,
"Failed to Create Account",
Toast.LENGTH_SHORT).show();
        }
    });
}

```

*Script Node 4 : Menyimpan data*

```

FirebaseUser user = FirebaseAuth.getCurrentUser();
startActivity(new
Intent(getApplicationContext(), LoginActivity.class));
Toast.makeText(RegisterActivity.this,
"Account created", Toast.LENGTH_SHORT).show();
DocumentReference df =
fStore.collection("User").document(user.getUid());
Map<String, Object> userInfo = new
HashMap<>();
userInfo.put("FullName",
fullName.getText().toString());
userInfo.put("UserEmail",
email.getText().toString());
userInfo.put("UserName",
username.getText().toString());
userInfo.put("Password",
password.getText().toString());
userInfo.put("role", Constants.ROLE_PARENT);

```

*Script Node 5 : END*

```

}

```

Bagan alur registrasi standar untuk kompleksitas siklomatik dihitung dalam tiga cara:

1. Grafik alur memiliki 2 area(*region*)
2.  $V(G) = 5 \text{ tepian}(\textit{edge}) - 5 \text{ simpul}(\textit{node}) + 2 = 2$
3.  $V(G) = 1 \text{ simpul yang diperkirakan} + 1 = 2$

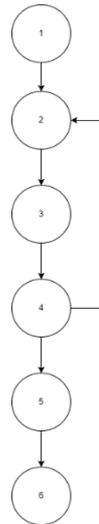
Oleh karena itu, diagram alur yang ditunjukkan pada gambar memiliki kompleksitas siklomatik 2. Jalur independen adalah :

Jalur(*path*) 1: 1-2-3-4-5-6

Jalur(*path*) 2: 1-2-3-2-3-4-5

Tabel IV. 3 Path Register

Jalur( <i>path</i> )	1
Jalur( <i>path</i> )	1-2-3-4-5
Skenario	Mulai Isi <i>form</i> Registrasi Cek <i>email</i> (tidak ada) Menyimpan data Selesai
Hasil Pengujian	Berhasil
Jalur( <i>path</i> )	2
Jalur( <i>path</i> )	1-2-3-2-3-4-5
Skenario	Mulai Isi <i>form</i> Registrasi Cek <i>email</i> (ada) Isi <i>form</i> Registrasi Cek <i>email</i> (tidak ada) Menyimpan data Selesai
Hasil Pengujian	Berhasil

b. Pengujian *Login*Gambar IV. 15 *Flow Graph Login*

(Sumber : Pribadi)

Dari *flow graph login* diatas dapat diketahui terdapat 6 *edge* dan 6 *node*, berikut ini adalah potongan-potongan *script* dari *node* diatas yaitu:

*Script Node 1 : Start*

```
setContentView(R.layout.Activity_login);
```

*Script Node 2 : Input form login*

```
if (isFieldNotValid(tvEmail) || isFieldNotValid(tvPassword)) {
    return;
}
```

*Script Node 3 : Tekan button login*

```
btnLogin.setOnClickListener(view ->
```

*Script Node 4 cek data user*

```
if (authResult.getUser() != null) {
    Toast.makeText(LoginActivity.this, "Login Success",
    Toast.LENGTH_SHORT).show();
```

```

        checkUserAccessLevel(authResult.getUser().getUid());
    } else {
        Toast.makeText(LoginActivity.this, "No User Found",
Toast.LENGTH_SHORT).show();
    }

```

*Script Node 5* : menampilkan halaman utama

```

Helper.openAdminDashboard(this, uid);
Helper.openUserDashboard(this, uid);

```

*Script Node 6* : END

```

}

```

Bagan alur pendaftaran standar untuk kompleksitas siklomatik dihitung dalam tiga cara:

1. Grafik alir memiliki dua area(region)
2.  $V(G) = 6 \text{ tepian(edge)} - 6 \text{ simpul(node)} + 2 = 2$
3.  $V(G) = 1 \text{ simpul yang diperkirakan} + 1 = 2$

Oleh karena itu, diagram alur yang ditunjukkan pada gambar memiliki kompleksitas siklomatik 3. Jalur independen adalah:

Jalur(*path*) 1: 1-2-3-4-5-6

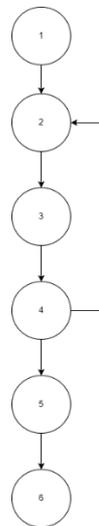
Jalur(*path*) 2: 1-2-3-4-2-3-4-5-6

Tabel IV. 4 Jalur(*path*) Login

Jalur( <i>path</i> )	1
Jalur( <i>path</i> )	1-2-3-4-5-6
Skenario	Mulai Isi <i>form Login</i> Tekan <i>button login</i> Cek <i>user</i> (Jika ada) Menampilkan halaman utama Selesai
Hasil Pengujian	Sukses

Jalur( <i>path</i> )	2
Jalur( <i>path</i> )	1-2-3-4-2-3-4-5-6
Skenario	Mulai Isi <i>form Login</i> Tekan <i>button login</i> Cek <i>user</i> (Jika tidak ada) Isi <i>form Login</i> Tekan <i>button login</i> Cek <i>user</i> (Jika ada) Menampilkan halaman utama Selesai
Hasil Pengujian	Sukses

c. Pengujian *Add Admin*



Gambar IV. 16 *Flow Graph Add Admin*

(Sumber : Pribadi)

Dari *flow graph add admin* diatas dapat diketahui terdapat 6 *edge* dan 6 *node*, berikut ini adalah potongan-potongan *script* dari *node* diatas yaitu:

*Script Node 1 : Start*

```
setContentView(R.layout.activity_add_admin);
```

*Script Node 2 : Input form add admin*

```
userInfo.put("UserEmail", email.getText().toString());
userInfo.put("UserName", username.getText().toString());
userInfo.put("Password", password.getText().toString());
userInfo.put("role", Constants.ROLE_ADMIN);
```

*Script Node 3 : Tekan button Register*

```
BtnRegister.setOnClickListener(new View.OnClickListener()
```

*Script Node 4 cek data admin*

```
fAuth.createUserWithEmailAndPassword(email.getText().toString(
), password.getText().toString()).addOnSuccessListener(new
OnSuccessListener<AuthResult>()
```

*Script Node 5 : menampilkan notifikasi sukses*

```
Toast.makeText(AddAdminActivity.this, "Admin Account Has
Been Created", Toast.LENGTH_SHORT).show();
```

*Script Node 6 : END*

```
}
```

Bagan alur pendaftaran standar untuk kompleksitas siklomatik dihitung dalam tiga cara:

1. Grafik alir memiliki dua area(region)
2.  $V(G) = 6 \text{ tepian(edge)} - 6 \text{ simpul(node)} + 2 = 2$
3.  $V(G) = 1 \text{ simpul yang diperkirakan} + 1 = 2$

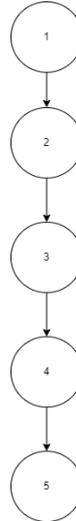
Oleh karena itu, diagram alur yang ditunjukkan pada gambar memiliki kompleksitas siklomatik 3. Jalur independen adalah:

Jalur(*path*) 1: 1-2-3-4-5-6

Jalur(*path*) 2: 1-2-3-4-2-3-4-5-6

Tabel IV. 5 Jalur(*path*) Add Admin

Jalur( <i>path</i> )	1
Jalur( <i>path</i> )	1-2-3-4-5-6
Skenario	Mulai Isi <i>form add admin</i> Tekan <i>button register</i> Cek <i>user</i> (Jika tidak ada) Menampilkan notifikasi Selesai
Hasil Pengujian	Sukses
Jalur( <i>path</i> )	2
Jalur( <i>path</i> )	1-2-3-4-2-3-4-5-6
Skenario	Mulai Isi <i>form add admin</i> Tekan <i>button register</i> Cek <i>user</i> (Jika ada) Isi <i>form add admin</i> Tekan <i>button register</i> Cek <i>user</i> (Jika tidak ada) Menampilkan notifikasi Selesai
Hasil Pengujian	Sukses

d. Pengujian *Add Category*Gambar IV. 17 *Flow Graph Add Category*

(Sumber : Pribadi)

Dari *flow graph add category* diatas dapat diketahui terdapat 4 *edge* dan 5 *node*, berikut ini adalah potongan-potongan *script* dari *node* diatas yaitu:

*Script Node 1 : Start*

```
setContentView(R.layout.activity_add_category);
```

*Script Node 2 : Input form add category*

```
savePem(etPem.getText().toString())
saveAge(etAge.getText().toString())
```

*Script Node 3 : Tekan button daftar*

```
btn_pem.setOnClickListener(new View.OnClickListener()
btn_age.setOnClickListener(new View.OnClickListener()
```

*Script Node 4 : menampilkan notifikasi sukses*

```
Toast.makeText(getApplicationContext(), "Succeed",
Toast.LENGTH_SHORT).show();
```

*Script Node 5 : END*

```
}
```

Bagan alur pendaftaran standar untuk kompleksitas siklomatik dihitung dalam tiga cara:

1. Grafik alir memiliki satu area(region)
2.  $V(G) = 4 \text{ tepian(edge)} - 5 \text{ simpul(node)} + 2 = 1$
3.  $V(G) = 0 \text{ simpul yang diperkirakan} + 1 = 1$

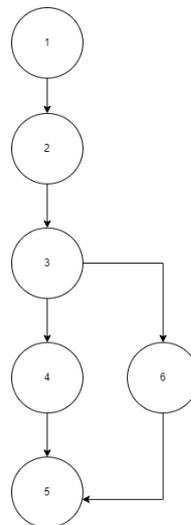
Oleh karena itu, diagram alur yang ditunjukkan pada gambar memiliki kompleksitas siklomatik 3. Jalur independen adalah:

Jalur(path) 1: 1-2-3-4-5

Tabel IV. 6 Jalur(path) *Add Category*

Jalur(path)	1
Jalur(path)	1-2-3-4-5
Skenario	Mulai Isi <i>form add category</i> Tekan <i>button register</i> Menampilkan notifikasi Selesai
Hasil Pengujian	Sukses

e. Pengujian *Approve Post*



Gambar IV. 18 *Flow Graph Approve Post*

(Sumber : Pribadi)

Dari *flow graph approve post* diatas dapat diketahui terdapat 6 *edge* dan 6 *node*, berikut ini adalah potongan-potongan *script* dari *node* diatas yaitu:

*Script Node 1 : Start*

```
setContentView(R.layout.activity_admin_approval);
```

*Script Node 2 : Memilih Postingan*

```
holder.itemView.setOnClickListener(view -> {
    Intent intent = new
    Intent(context,AdminApprovalActivity.class);
    intent.putExtra(IntentNameExtra.POST_DATA, data);
    context.startActivity(intent);
});
```

*Script Node 3 : Tekan button approve*

```
btnApprove.setOnClickListener(view ->
updateApprovalData(postData, true))
```

*Script Node 4* menampilkan notifikasi sukses

```
message = String.format(new Locale("id", "ID"),
getString(R.string.post_approved_message), data.getFullName());
```

*Script Node 5 : END*

```
}
```

*Script Node 6 :* menampilkan notifikasi gagal

```
message = String.format(new Locale("id", "ID"),
getString(R.string.post_rejected_message), data.getFullName());
```

Bagan alur *approve post* standar untuk kompleksitas siklomatik dihitung dalam tiga cara:

1. Grafik alir memiliki dua area(region)

$$2. V(G) = 6 \text{ tepian(edge)} - 6 \text{ simpul(node)} + 2 = 2$$

$$3. V(G) = 1 \text{ simpul yang diperkirakan} + 1 = 2$$

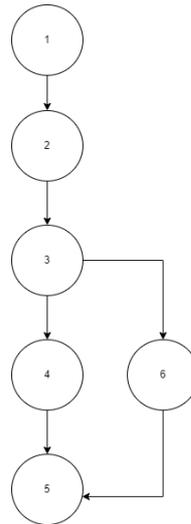
Oleh karena itu, diagram alur yang ditunjukkan pada gambar memiliki kompleksitas siklomatik 2. Jalur independen adalah:

Jalur(*path*) 1: 1-2-3-4-5

Jalur(*path*) 2: 1-2-3-6-5

Tabel IV. 7 Jalur(*path*) Approve Post

Jalur( <i>path</i> )	1
Jalur( <i>path</i> )	1-2-3-4-5
Skenario	Mulai Memilih Postingan Tekan <i>button approve</i> Menampilkan notifikasis sukses Selesai
Hasil Pengujian	Sukses
Jalur( <i>path</i> )	2
Jalur( <i>path</i> )	1-2-3-6-5
Skenario	Mulai Memilih Postingan Tekan <i>button approve</i> Menampilkan notifikasi gagal Selesai
Hasil Pengujian	Sukses

f. Pengujian *Approve Certificated*Gambar IV. 19 *Flow Graph Approve Certificated*

(Sumber : Pribadi)

Dari *flow graph approve certificated* diatas dapat diketahui terdapat 6 *edge* dan 6 *node*, berikut ini adalah potongan-potongan *script* dari *node* diatas yaitu:

*Script Node 1 : Start*

```
setContentView(R.layout.activity_admin_approval_certificated);
```

*Script Node 2 : Memilih Postingan*

```
holder.itemView.setOnClickListener(view -> {
    Intent intent = new Intent(context,
    AdminApprovalCertificatedActivity.class);
    intent.putExtra(IntentNameExtra.POST_DATA, data);
    context.startActivity(intent);
});
```

*Script Node 3 : Tekan button approve*

```
btnApprove.setOnClickListener(view ->
updateApprovalData(postData, true))
```

*Script Node 4* menampilkan notifikasi sukses

```
message = String.format(new Locale("id", "ID"),
getString(R.string.post_approved_message), data.getFullName());
```

*Script Node 5* : END

```
}
```

*Script Node 6* : menampilkan notifikasi gagal

```
message = String.format(new Locale("id", "ID"),
getString(R.string.post_rejected_message), data.getFullName());
```

Bagan alur *approve certificated* standar untuk kompleksitas siklomatik dihitung dalam tiga cara:

1. Grafik alir memiliki dua area(region)
2.  $V(G) = 6 \text{ tepian(edge)} - 6 \text{ simpul(node)} + 2 = 2$
3.  $V(G) = 1 \text{ simpul yang diperkirakan} + 1 = 2$

Oleh karena itu, diagram alur yang ditunjukkan pada gambar memiliki kompleksitas siklomatik 2. Jalur independen adalah:

Jalur(*path*) 1: 1-2-3-4-5

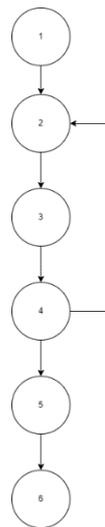
Jalur(*path*) 2: 1-2-3-6-5

Tabel IV. 8 Jalur(*path*) *Approve Certificated*

Jalur( <i>path</i> )	1
Jalur( <i>path</i> )	1-2-3-4-5
Skenario	Mulai Memilih <i>Certificated</i> Tekan <i>button approve</i> Menampilkan notifikasis sukses Selesai
Hasil Pengujian	Sukses
Jalur( <i>path</i> )	2
Jalur( <i>path</i> )	1-2-3-6-5

Skenario	Mulai Memilih <i>Certificated</i> Tekan <i>button approve</i> Menampilkan notifikasi gagal Selesai
Hasil Pengujian	Sukses

g. Pengujian *Add Post*



Gambar IV. 20 *Flow Graph Add Post*

(Sumber : Pribadi)

Dari *flow graph add post* diatas dapat diketahui terdapat 6 *edge* dan 6 *node*, berikut ini adalah potongan-potongan *script* dari *node* diatas yaitu:

*Script Node 1 : Start*

```
setContentView(R.layout.activity_post);
```

*Script Node 2 : Input form add post*

```
String categoryPem = spinnerPem.getSelectedItem().toString();
String categoryAge = spinnerUmur.getSelectedItem().toString();
title = etTitle.getText().toString();
String desc = etDesc.getText().toString();
```

*Script Node 3 : Tekan button upload*

```
btnUpload.setOnClickListener(view ->
```

*Script Node 4 cek data*

```
try {
    uploadPost();
} catch (Exception e) {
    Toast.makeText(this, "Error occurred when uploading.",
        Toast.LENGTH_SHORT).show();
    Log.e("PostActivity", "Error occurred: " +
        e.getLocalizedMessage());
}
```

*Script Node 5 : menampilkan notifikasi sukses*

```
Toast.makeText(PostActivity.this, "Image Uploaded",
    Toast.LENGTH_SHORT).show();
```

*Script Node 6 : END*

```
}
```

Bagan alur pendaftaran standar untuk kompleksitas siklomatik dihitung dalam tiga cara:

1. Grafik alir memiliki dua area(region)
2.  $V(G) = 6 \text{ tepian(edge)} - 6 \text{ simpul(node)} + 2 = 2$
3.  $V(G) = 1 \text{ simpul yang diperkirakan} + 1 = 2$

Oleh karena itu, diagram alur yang ditunjukkan pada gambar memiliki kompleksitas siklomatik 3. Jalur independen adalah:

Jalur(*path*) 1: 1-2-3-4-5-6

Jalur(*path*) 2: 1-2-3-4-2-3-4-5-6

Tabel IV. 9 Jalur(*path*) *Add post*

Jalur( <i>path</i> )	1
Jalur( <i>path</i> )	1-2-3-4-5-6
Skenario	Mulai Isi <i>form add post</i> Tekan <i>button upload</i> Cek data (Jika tidak ada) Menampilkan notifikasi sukses Selesai
Hasil Pengujian	Sukses
Jalur( <i>path</i> )	2
Jalur( <i>path</i> )	1-2-3-4-2-3-4-5-6
Skenario	Mulai Isi <i>form add post</i> Tekan <i>button upload</i> Cek data (Jika ada) Isi <i>form add post</i> Tekan <i>button upload</i> Cek data (Jika tidak ada) Menampilkan notifikasi sukses Selesai
Hasil Pengujian	Sukses

h. Rekapitulasi Hasil Pengujian *White Box*

Tabel IV. 10 Rekapitulasi Hasil Pengujian White Box

<b>Kesimpulan dan Hasil Uji</b>			
<b>Halaman yang diuji</b>	<b>Skenario Test Case</b>	<b>Pengamatan</b>	<b>Hasil</b>
Pengujian <i>Login</i>	Jalur 1: 1-2-3-4-5-6 Jalur 2: 1-2-3-2-3-4-5	Pengujian yang dilakukan berdasarkan scenario test case aplikasi dapat berjalan dengan sesuai	Berhasil
Pengujian <i>Register</i>	Jalur 1: 1-2-3-4-7 Jalur 2: 1-2-3-5-7 Jalur 3: 1-2-6-7	Pengujian yang dilakukan berdasarkan scenario test case aplikasi dapat berjalan dengan sesuai	Berhasil
Pengujian <i>Add Admin</i>	Jalur 1: 1-2-3-4-5-6 Jalur 2: 1-2-3-4-2-3-4-5-6	Pengujian yang dilakukan berdasarkan scenario test case aplikasi dapat berjalan dengan sesuai	Berhasil
Pengujian <i>Add Category</i>	Jalur1: 1-2-3-4-5	Pengujian yang dilakukan berdasarkan scenario test case aplikasi dapat berjalan dengan sesuai	Berhasil

<b>Kesimpulan dan Hasil Uji</b>			
<b>Halaman yang diuji</b>	<b>Skenario Test Case</b>	<b>Pengamatan</b>	<b>Hasil</b>
Pengujian <i>Approve Post</i>	Jalur 1: 1-2-3-4-5 Jalur 2: 1-2-3-6-5	Pengujian yang dilakukan berdasarkan scenario test case aplikasi dapat berjalan dengan sesuai	Berhasil
Pengujian <i>Approve Certificated</i>	Jalur 1: 1-2-3-4-5 Jalur 2: 1-2-3-6-5	Pengujian yang dilakukan berdasarkan scenario test case aplikasi dapat berjalan dengan sesuai	Berhasil
Pengujian <i>Add Post</i>	Jalur1: 1-2-3-4-5-6 Jalur 2: 1-2-3-4-2-3-4-5-6	Pengujian yang dilakukan berdasarkan scenario test case aplikasi dapat berjalan dengan sesuai	Berhasil

Berdasarkan Tabel IV.9 didapatkan hasil rekapitulasi maka dapat disimpulkan dari semua pengujian sistem menggunakan metode *white box*, menyatakan bahwa semua logika program sesuai dan benar.

#### IV.2.3 Pengujian Kelayakan Penggunaan Aplikasi

Pengujian kelayakan penggunaan aplikasi dilakukan dengan menggunakan kuesioner yang terdiri dari 5 pertanyaan yang dibagikan kepada 10 responden, tanggapan responden diklasifikasikan pada skala Likert dari 1 sampai 5. Berdasarkan data yang dihasilkan dari kuesioner, perhitungannya dibuat dengan menggunakan skala likert, salah satu metode perhitungan yang digunakan untuk mengetahui apakah responden setuju atau tidak setuju dengan pernyataan tersebut. Saat menghitung skor maksimum untuk setiap jawaban, kalikan skor dengan jumlah total responden, yaitu 10 responden.

Tabel IV. 11 Tabel Nilai Skor Tertinggi

Jawaban	Skor	Skor tertinggi (Skor * Jumlah responden)
Sangat setuju	5	50
Setuju	4	40
Cukup setuju	3	30
Kurang setuju	2	20
Tidak setuju	1	10

Kemudian persentase setiap jawaban dapat dicari menggunakan rumus:

$$Y = \frac{TS}{Skor\ Ideal} \times 100\%$$

Y = Nilai persentase

TS = Total skor responden =  $\Sigma$  skor \* responden

Skor ideal = skor \* jumlah responden = 5 x 10 = 50

Tabel IV. 12 Kriteria Skor

Kategori	Keterangan
0% - 20%	Tidak setuju
21% - 40%	Kurang setuju
41% - 60%	Cukup setuju
61% - 80%	Setuju
81% - 100%	Sangat Setuju

Di bawah ini adalah hasil persentase untuk setiap jawaban yang dijawab. Survei ini diujikan pada 10 responden (hasil survei terlampir).

#### 1. Pertanyaan 1

Saya merasa informasi yang terdapat pada aplikasi ini mudah untuk dipahami? Hasil angket pertanyaan kesatu tersebut dapat dilihat pada tabel berikut ini:

Tabel IV. 13 Hasil Kuesioner Pertanyaan 1

Pertanyaan	Jawaban	Skor	Responden	Jumlah Skor	Nilai Persentase
1	Sangat setuju	5	4	20	(42:50) * 100
	Setuju	4	4	16	
	Cukup setuju	3	2	6	
	Kurang setuju	2	0	0	
	Tidak setuju	1	0	0	
Jumlah			10	42	84%

Berdasarkan nilai penyajian pertanyaan pertama dapat disimpulkan bahwa 84% responden menilai informasi yang terdapat pada aplikasi mudah untuk dipahami.

## 2. Pertanyaan 2

Saya merasa ide-ide kegiatan pada aplikasi ini sudah bagus? Hasil angket pertanyaan kedua tersebut dapat dilihat pada tabel berikut ini:

Tabel IV. 14 Hasil Kuesioner Pertanyaan 2

Pertanyaan	Jawaban	Skor	Responden	Jumlah Skor	Nilai Persentase
2	Sangat setuju	5	2	10	(41:50) * 100
	Setuju	4	7	28	
	Cukup setuju	3	1	3	
	Kurang setuju	2	0	0	
	Tidak setuju	1	0	0	
Jumlah			10	41	82%

Berdasarkan nilai penyajian pertanyaan kedua dapat disimpulkan bahwa 82% responden menilai ide-ide kegiatan pada aplikasi ini sudah bagus.

## 3. Pertanyaan 3

Saya merasa informasi pada aplikasi ini penting untuk diterapkan pada anak-anak? Hasil angket pertanyaan ketiga tersebut dapat dilihat pada tabel berikut ini:

Tabel IV. 15 Hasil Kuesioner Pertanyaan 3

Pertanyaan	Jawaban	Skor	Responden	Jumlah Skor	Nilai Persentase
3	Sangat setuju	5	1	5	
	Setuju	4	8	32	

Pertanyaan	Jawaban	Skor	Responden	Jumlah Skor	Nilai Persentase
	Cukup setuju	3	1	3	(40:50) * 100
	Kurang setuju	2	0	0	
	Tidak setuju	1	0	0	
Jumlah			10	40	80%

Berdasarkan nilai penyajian pertanyaan ketiga dapat disimpulkan bahwa 80% responden menilai informasi pada aplikasi ini penting untuk diterapkan pada anak-anak.

#### 4. Pertanyaan 4

Saya merasa informasi pada aplikasi ini beraneka ragam? Hasil angket pertanyaan keempat tersebut dapat dilihat pada tabel berikut ini:

Tabel IV. 16 Hasil Kuesioner Pertanyaan 4

Pertanyaan	Jawaban	Skor	Responden	Jumlah Skor	Nilai Persentase
4	Sangat setuju	5	2	10	(40:50) * 100
	Setuju	4	6	24	
	Cukup setuju	3	2	6	
	Kurang setuju	2	0	0	
	Tidak setuju	1	0	0	
Jumlah			10	40	80%

Berdasarkan nilai penyajian pertanyaan keempat dapat disimpulkan bahwa 84% responden menilai informasi pada aplikasi ini beraneka ragam.

#### 5. Pertanyaan 5

Saya merasa aplikasi ini sudah menyediakan sarana untuk sharing seputar ide-ide kegiatan pada anak? Hasil angket pertanyaan kelima tersebut dapat dilihat pada tabel berikut ini:

Tabel IV. 17 Hasil Kuesioner Pertanyaan 5

Pertanyaan	Jawaban	Skor	Responden	Jumlah Skor	Nilai Persentase
5	Sangat setuju	5	1	5	(40:50) * 100
	Setuju	4	8	32	
	Cukup setuju	3	1	3	
	Kurang setuju	2	0	0	
	Tidak setuju	1	0	0	
Jumlah			10	40	80%

Berdasarkan nilai penyajian pertanyaan ke-5, dapat disimpulkan bahwa 80% responden menilai aplikasi ini sudah menyediakan sarana untuk sharing seputar ide-ide kegiatan pada anak.

## 6. Hasil Rata – Rata Persentase

Tabel IV. 18 Hasil Rata – Rata Persentase

Pertanyaan	Nilai Persentase	Keterangan
1	84%	Sangat setuju
2	82%	Sangat setuju
3	80%	Sangat setuju
4	80%	Sangat setuju
5	80%	Sangat setuju
Total Persentase	$84\% + 82\% + 80\% + 80\% + 80\% = 406$	Sangat setuju
Rata - Rata	$406 / 5 = 81.2 \%$	

Hasil setiap pertanyaan dalam angket dihitung rata-rata persentase pertanyaan secara keseluruhan. Kemudian membandingkan untuk menarik kesimpulan. Dari tabel di atas, dapat disimpulkan bahwa 81.2% responden sangat setuju dengan kualitas aplikasi, oleh karena itu aplikasi telah dibuat dapat digunakan.

## BAB V

### PENUTUP

#### V.1 Kesimpulan

Berdasarkan penelitian yang dilakukan, dapat ditarik kesimpulan sebagai berikut:

1. Aplikasi *montessori pad* asisten untuk media *sharing* metode pembelajaran *Montessori* dikembangkan menggunakan bahasa pemrograman java dan firebase sebagai sistem manajemen database. Fitur-fitur yang terdapat pada aplikasi tersebut untuk *user* adalah *upload* aktifitas dimana pada fitur ini orang tua dan professional dapat memasukkan cerita dan ide aktifitas seputar kegiatan metode pembelajaran *montessori*, fitur yang lain adalah *like* dan *comment*. Sedangkan fitur untuk admin adalah *Add Admin* untuk menambahkan admin dan *Approve postingan* untuk menyeleksi dan menyetujui postingan mana yang layak diterbitkan. Berdasarkan hasil pengujian fungsional menggunakan metode *black box* diketahui bahwa semua fungsi dari sistem bekerja sesuai dengan fungsinya, sedangkan berdasarkan hasil pengujian sistem menggunakan metode *white box* bahwa semua logika program sesuai dan benar.
2. Berdasarkan hasil pengujian kelayakan penggunaan aplikasi yang telah dilakukan oleh 10 responden diperoleh tingkat kelayakan dengan presentase rata-rata 81.2% yaitu sangat setuju dengan kualitas yang ada pada aplikasi sehingga kami dapat menyimpulkan bahwa aplikasi ini layak untuk digunakan.

## V.2 Saran

Penelitian ini tentunya belum lengkap dan masih banyak kekurangannya. Untuk itu perlu dilakukan pengembangan atau penyempurnaan aplikasi agar menjadi lebih baik. Saran untuk membuat aplikasi ini bekerja lebih baik dan lebih menarik adalah sebagai berikut:

1. Aplikasi ini memiliki tampilan yang masih terbilang biasa saja, diharapkan kedepannya peneliti dapat mengembangkan tampilannya agar membuat lebih interaktif dan lebih menarik.
2. Aplikasi ini masih hanya dapat bekerja pada satu *platform*, yaitu Android. Kelemahan ini menjadi acuan untuk dikembangkan kembali agar bisa *cross-platform*.
3. Aplikasi ini hanya dapat mengupload gambar, kelemahannya inilah yang dapat dikembangkan lagi agar kedepannya aplikasi ini dapat mengupload video
4. Diharapkan peneliti kedepannya agar dapat menambahkan fitur-fitur seperti *serach* dan *chat*

## DAFTAR PUSTAKA

- Abdillah, R., Kuncoro, A., & Kurniawan, I. (2019). Analisis Aplikasi Pembelajaran Matematika Berbasis Android dan Desain Sistem Menggunakan UML 2.0. *Jurnal Theorems*, 4(1), 301752.
- Arsyad, A. (2002). *Media Pembelajaran*, edisi 1 jakarta: PT. Raja grafindo persada.
- Assegaff, S. (2017). Evaluasi pemanfaatan media sosial sebagai sarana knowledge sharing. *Jurnal Manajemen Teknologi*, 16(3), 271-293.
- Brown, U. J., & Gaylor, K. (2002). Organizational Commitment in Higher. *Education. Jackson State University: Mississippi*.
- Damayanti, E. (2019). Meningkatkan Kemandirian Anak melalui Pembelajaran Metode Montessori. *Jurnal Obsesi: Jurnal Pendidikan Anak Usia Dini*, 4(1), 463-470.
- Darnis, S. (2018). Aplikasi Montessori Dalam Pembelajaran Membaca, Menulis Dan Berhitung Tingkat Permulaan Bagi Anak Usia Dini. *Jurnal Caksana: Pendidikan Anak Usia Dini*, 1(01).
- Dewi, I. R., & Malfiany, R. (2017). Perancangan Sistem Informasi Administrasi Pembayaran Pada Sdit Lampu Iman Karawang Berbasis Visual Basic 6.0. *Jurnal Interkom: Jurnal Publikasi Ilmiah Bidang Teknologi Informasi Dan Komunikasi*, 12(2), 4-12.
- Dharma Kasman, A. (2015). *Trik Kolaborasi ANDROID dengan PHP dan MySQL*. Yogyakarta: Lokomedia.
- Hernawan, A. H., Susilana, R., & Julaeha, S. (2014). Pengembangan Kurikulum dan pembelajaran di SD. *Universitas Terbuka*, 1-40.
- Hingle, M., & Patrick, H. (2016). There are thousands of apps for that: navigating mobile technology for nutrition education and behavior. *Journal of nutrition education and behavior*, 48(3), 213-218.

- Hughes, A. G., & Hughes, E. H. (2012). *Learning & teaching pengantar psikologi pembelajaran modern*. Penerjemah SPA teamwork.
- Ihsan, I. P., Sakir, M., & Yusuf, F. (2021). PUZZLE ELEKTRONIK MONTESSORI SAND BOARD LETTER (SBL) UNTUK EDUKASI SUKU KATA BAHASA INDONESIA PADA ANAK USIA DINI.
- Lillard, P. P. (1997). *Montessori in the classroom: A teacher's account of how children really learn*. Schocken.
- Nofriadi, M. (2015). *Java Fundamental dengan Netbeans 8.0. 2*. Deepublish.
- Payara, G. R., & Tanone, R. (2018). Penerapan Firebase Realtime Database Pada Prototype Aplikasi Pemesanan Makanan Berbasis Android. *Jurnal Teknik Informatika dan Sistem Informasi*, 4(3), 397-406.
- Permataputri, D. I., & Syamsudin, A. (2021). Pembelajaran Nilai Agama dan Moral Anak Usia Dini melalui Metode Montessori selama Pandemi Covid-19. *Jurnal Obsesi: Jurnal Pendidikan Anak Usia Dini*, 6(2), 693-703.
- Pressman, R. S. (2010). A practitioner's approach. *Software Engineering*, 2, 41-42.
- Pressman, R. S. (2010). *Software Engineering: A practitioners Ap-proach*. McGrawHill. *New York*, 68.
- Ramakrishnan, R., Gehrke, J., & Gehrke, J. (2003). *Database management systems* (Vol. 3). New York: McGraw-Hill.
- Rosa, A. S., & Shalahudin, M. (2014). *Rekayasa Perangkat Lunak Informatika*. Bandung.
- Safaat, N. (2012). *Android; Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*.
- Sagala, S. (2010). *Supervisi Pendidikan dalam Profesi Pendidikan*. Bandung. Penerbit Alfabeta.

- Sani, R. A., Manurung, S. R., & Suswanto, H. Sudiran (2018). Penelitian Pendidikan. *Tangerang: Tira Smart*.
- Saputro, A. H. (2017). Pembuatan aplikasi interaktif berhitung permulaan dengan metode montessori untuk TK Pustaka Arridho.
- Save, M. D. (2006). Kamus Besar Ilmu Pengetahuan, cet ke-5. *Jakarta: Lembaga Pengkajian Kebudayaan Nusantara*.
- Sudjana, N. (2010). Penilaian hasil proses belajar mengajar.
- Sujiono, Y. N. (2009). Konsep dasar pendidikan anak usia dini. *Jakarta: PT indeks*.
- Sukamto, R. A., & Shalahuddin, M. (2013). Analisa dan Desain Sistem Informasi. *Yogyakarta: Andi Offset*.
- Suwarto, S., Muzaki, A., & Muhtarom, M. (2021). Pemanfaatan media youtube sebagai media pembelajaran pada siswa kelas XII MIPA di SMA Negeri 1 Tawang Sari. *Media Penelitian Pendidikan: Jurnal Penelitian dalam Bidang Pendidikan dan Pengajaran*, 15(1), 26-30.

**L**

**A**

**M**

**P**

**I**

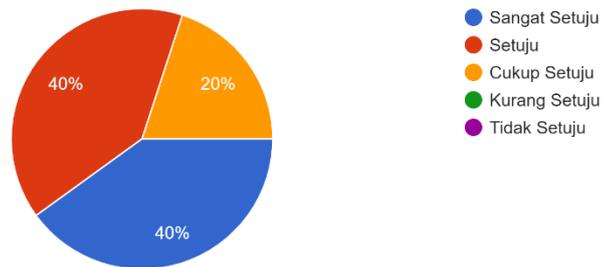
**R**

**A**

**N**

Saya merasa informasi yang terdapat pada aplikasi ini mudah untuk dipahami?

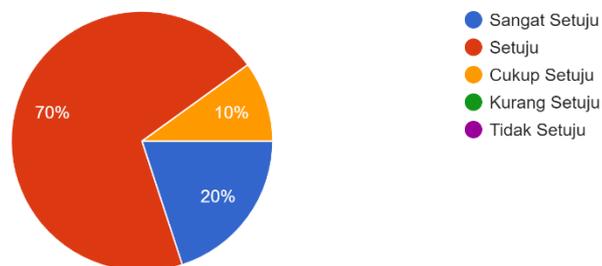
10 jawaban



### Lampiran 1 Kuesioner Pertanyaan Nomor 1

Saya merasa ide-ide kegiatan pada aplikasi ini sudah bagus?

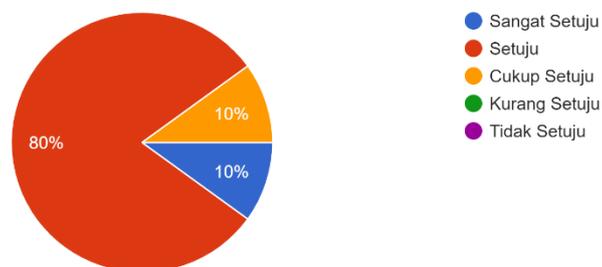
10 jawaban



### Lampiran 2 Kuesioner Pertanyaan Nomor 2

Saya merasa informasi pada aplikasi ini penting untuk diterapkan pada anak-anak?

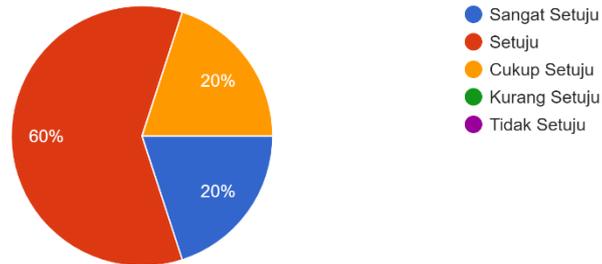
10 jawaban



### Lampiran 3 Kuesioner Pertanyaan Nomor 3

Saya merasa informasi pada aplikasi ini beraneka ragam?

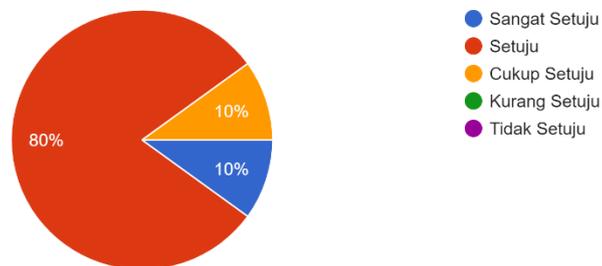
10 jawaban



#### Lampiran 4 Kuesioner Pertanyaan Nomor 4

Saya merasa aplikasi ini sudah menyediakan sarana untuk sharing seputar ide-ide kegiatan pada anak?

10 jawaban



#### Lampiran 5 Kuesioner Pertanyaan Nomor 5