

**PEMBANGUNAN APLIKASI MONITORING
DAN PENGENDALIAN KONTEN NEGATIF
PADA PERANGKAT PINTAR BERBASIS ANDROID
MENGUNAKAN API CLARIFAI**

SKRIPSI

Diajukan untuk Menempuh Ujian Akhir Sarjana

RUDIANTO

10112084



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK DAN ILMU KOMPUTER
UNIVERSITAS KOMPUTER INDONESIA**

2017

DAFTAR ISI

Daftar isi.....	i
Daftar Gambar	iv
Daftar Tabel.....	vii
Daftar Simbol	ix
BAB 1	1
PENDAHULUAN.....	1
1.1 Latar Belakang Masalah	1
1.2 Identifikasi Masalah	3
1.3 Maksud dan Tujuan Penelitian	4
1.4 Batasan Masalah.....	4
1.5 Metodologi Penelitian	5
1.5.1 Identifikasi Masalah dan Perumusan masalah	6
1.5.2 Pengumpulan Data	7
1.5.3 Analisa dan Perancangan Sistem	7
1.5.4 Perancangan Sistem	8
1.5.5 Pembangunan Sistem	9
1.5.6 Pengujian Sistem.....	9
1.5.7 Penarikan Kesimpulan	9
1.6 Sistematika Penulisan.....	9
BAB 2	11
Landasan teori.....	11
2.1 Aplikasi	11
2.2 Monitoring.....	11

2.3	Konten Negatif	12
2.4	Perangkat Pintar	12
2.5	Smartphone.....	13
2.6	Android.....	13
2.7	Arsitektur Android	15
2.8	Komponen Aplikasi Android	18
2.9	Android SDK(Software Development Kit)	19
2.10	Android Studio.....	20
2.11	Bahasa Pemrograman Java	22
2.12	Clarifai	23
2.12.1	Model Clarifai	24
2.12.2	Authentication	24
2.12.3	Predict	25
2.12.4	Train	25
2.13	Web Service	26
2.14	Representational State Transfer (REST).....	26
2.15	JavaScript Object Notation (JSON).....	28
2.16	Location Base Service (LBS)	32
2.17	Mysql	34
2.18	Structured Query Language (SQL).....	35
2.19	Personal Home Page (PHP)	39
2.20	CodeIgniter (CI).....	42
2.21	Unified Modeling language (UML).....	47
2.21.1	Use Case Diagram.....	47
2.21.2	Class Diagram	47

2.21.3	Sequence Diagram	48
2.21.4	Activity Diagram.....	48
BAB 3	49
Analisa dan perancangan sistem	49
3.1	Analisis Sistem	49
3.1.1	Analisis Masalah	49
3.1.2	Analisis Arsitektur Sistem	50
3.1.3	Analisis Teknologi	51
3.1.4	Analisis Kebutuhan Spesifikasi Perangkat Lunak	57
3.1.5	Analisis Kebutuhan Non Fungsional	59
3.1.6	Analisis Kebutuhan Fungsional	61
3.2	Perancangan Sistem.....	99
3.2.1	Perancangan Skema Relasi	100
3.2.2	Perancangan Struktur Tabel	100
3.2.3	Perancangan Struktur Menu	102
3.2.4	Perancangan Antar Muka	103
3.2.5	Perancangan Pesan	111
3.2.6	Perancangan Jaringan Semantik.....	111
3.2.7	Perancangan Prosedural	112
Daftar Pustaka	128

DAFTAR GAMBAR

Gambar 1 Kerangka Kerja Penelitian	6
Gambar 2 Arsitektur Android	15
Gambar 3 File Proyek di Tampilan Android	21
Gambar 4 Konsep kerja Clarifai	23
Gambar 5 Train	26
Gambar 6 JSON Object.....	30
Gambar 7 JSON Array	30
Gambar 8 Value JSON.....	30
Gambar 9 String JSON	31
Gambar 10 JSON Angka.....	32
Gambar 11 LBS Sebagai simpang tiga teknologi	33
Gambar 12 Komponen dasar LBS	34
Gambar 13 Logo CodeIgniter	43
Gambar 14 Konsep Model View Controller(MVC)	45
Gambar 15 model sistem.....	50
Gambar 16 Membuat Proyek Baru di Situs Clarifai	52
Gambar 17 Mendapatkan Autentikasi Key	52
Gambar 18 Gambar menu Explorer di Clarifai.....	53
Gambar 19 Tampilan Proyek yang telah dibuat di menu explorer	53
Gambar 20 Membuat Konsep baru di Clarifai	54
Gambar 21 Menambahkan Citra pada Concept yang telah dibuat.....	54
Gambar 22 Membuat Model Baru Calarifai	55
Gambar 23 Training Pada Model.....	55
Gambar 24 Model id Clarifai	56
Gambar 25 Pengujian API Clarifai dengan Library Java.....	56
Gambar 26 Response dari Library Api Clarifai dari Java.....	57
Gambar 27 Use Case Diagram	62
Gambar 28 Activity Diagram Daftar.....	80
Gambar 29 Activity Diagram Lupa password	81
Gambar 30 Activity Diagram Login	82

Gambar 31 Activity Diagram Log Out	83
Gambar 32 Activity Diagram Profil.....	84
Gambar 33 Activity Diagram Ubah nama	85
Gambar 34 Activity Diagram Ubah Password.....	86
Gambar 35 Activity Diagram Monitoring Konten Negatif.....	87
Gambar 36 Activity Diagram Penghapusan Berkas Gambar.....	88
Gambar 37 Activity Diagram Monitoring History Browser	89
Gambar 38 Activity Diagram Monitoring Lokasi Anak	90
Gambar 39 Activity Diagram Backup Data kontak	91
Gambar 40 Activity Diagram Notifikasi.....	92
Gambar 41 Activity Diagram Notifikasi Monitoring Konten Negatif.....	93
Gambar 42 Activity Diagram Notifikasi Monitoring History Browser	94
Gambar 43 Activity Diagram Notifikasi Lokasi Anak	95
Gambar 44 Aplikasi Monitoring Dan Pengendalian Konten Negatif Pada Perangkat Pintar Berbasis Android Menggunakan API Clarifai	95
Gambar 45 Sequence Diagram Daftar	96
Gambar 46 Sequence Diagram Lupa Password	96
Gambar 47 Sequence Diagram Login	96
Gambar 48 Sequence Diagram Log Out	96
Gambar 49 Sequence Diagram Profil	97
Gambar 50 Sequence Diagram Ubah Nama	97
Gambar 51 Sequence Diagram Ubah Password.....	97
Gambar 52 Sequence Diagram Monitoring Konten Negatif.....	97
Gambar 53 Sequence Diagram Penghapusan Berkas Gambar	97
Gambar 54 Sequence Diagram Monitoring Lokasi Anak.....	98
Gambar 55 Sequence Diagram Monitoring History Browser.....	98
Gambar 56 Sequence Diagram Backup Data Kontak	98
Gambar 57 Sequence Diagram Notifikasi.....	98
Gambar 58 Sequence Diagram Notifikasi Monitoring Konten Negatif.....	99
Gambar 59 Sequence Diagram Notifikasi Monitoring History Browser.....	99

Gambar 60 Sequence Diagram Notifikasi Lokasi Anak	99
Gambar 61 Skema Relasi	100
Gambar 62 Struktur Menu	103
Gambar 63 Perancangan Antar Muka Login	104
Gambar 64 Perancangan Antarmuka Lupa password	104
Gambar 65 Perancangan Antarmuka Daftar	105
Gambar 66 Perancangan Antarmuka Monitoring Konten Negatif	106
Gambar 67 Perancangan Antarmuka Penghapusan berkas gambar	106
Gambar 68 Perancangan Antarmuka Monitoring History Browser	107
Gambar 69 Perancangan Antarmuka Monitoring Lokasi	108
Gambar 70 Perancangan Antar Muka Backup Kontak	108
Gambar 71 Perancangan Antarmuka Profil	109
Gambar 72 Perancangan Antar Muka Ubah Nama	110
Gambar 73 Perancangan Antarmuka Ubah Password	110
Gambar 74 Jaringan Semantik	112
Gambar 75 Prosedural Daftar	113
Gambar 76 Prosedural Lupa Password	114
Gambar 77 Prosedural Login	115
Gambar 78 Prosedural Log Out	116
Gambar 79 Prosedural Profil	117
Gambar 80 Prosedural Ubah Nama	118
Gambar 81 Prosedural Ubah Password	119
Gambar 82 Prosedural Monitoring Konten Negatif	120
Gambar 83 Prosedural Penghapusan Berkas Gambar	121
Gambar 84 Prosedural Monitoring Lokasi Anak	122
Gambar 85 Prosedural Monitoring History Browser	123
Gambar 86 Prosedural Backup Data Kontak	124
Gambar 87 Prosedural Notifikasi	125
Gambar 88 Prosedural Notifikasi Monitoring Konten Negatif	126
Gambar 89 Prosedural Notifikasi Monitoring History Browser	126
Gambar 90 Prosedural Notifikasi Lokasi Anak	127




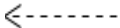

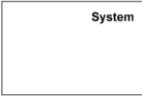
DAFTAR TABEL

Tabel 2.1 Rilis Versi Android	14
Tabel 3.1 Tabel Spesifikasi Kebutuhan Perangkat Lunak Fungsional	58
Tabel 3.2 Tabel Spesifikasi Kebutuhan Perangkat Lunak Non-Fungsional	58
Tabel 3.3 Spesifikasi Kebutuhan Perangkat Keras	59
Tabel 3.4 Spesifikasi Kebutuhan Perangkat Keras	59
Tabel 3.5 Spesifikasi Kebutuhan Perangkat Lunak	60
Tabel 3.6 Tabel Spesifikasi Kebutuhan Perangkat Lunak	60
Tabel 3.7 Karakteristik Pengguna	61
Tabel 3.8 Definisi Aktor	63
Tabel 3.9 Definisi Use Case	63
Tabel 3.10 Skenario Use Case Daftar	64
Tabel 3.11 Skenario Use Case Lupa Password	65
Tabel 3.12 Skenario Use Case Login	66
Tabel 3.13 Skenario Use Case Log Out	67
Tabel 3.14 Skenario Use Case Profil	68
Tabel 3.15 Skenario Use Case Ubah Nama	69
Tabel 3.16 Skenario Use Case Ubah Password	70
Tabel 3.17 Skenario Use Case Monitoring Konten Negatif	71
Tabel 3.18 Skenario Use Case Penghapusan Berkas Gambar	72
Tabel 3.19 Skenario Use Case Monitoring Lokasi Anak	73
Tabel 3.20 Monitoring History Browser	73
Tabel 3.21 Backup data Kontak	74
Tabel 3.22 Notifikasi	75
Tabel 3.23 Notifikasi Monitoring Konten Negatif	76
Tabel 3.24 Case Notifikasi monitoring History Browser	77
Tabel 3.25 Notifikasi Monitoring Lokasi	78
Tabel 3.26 Struktur Tabel t_user	101
Tabel 3.27 Struktur Tabel t_konten	101




Tabel 3.28 Struktur Tabel t_history	101
Tabel 3.29 Struktur Tabel t_lokasi.....	102
Tabel 3.30 Struktur Tabel t_kontak	102
Tabel 3.31 Perancangan Pesan.....	111


DAFTAR SIMBOL

Simbol *Use Case Diagram*

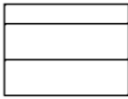


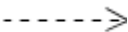
NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
3		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
4		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
5		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
6		<i>System Boundary</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.

Simbol *Activity Diagram*

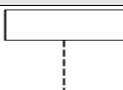



NO	GAMBAR	NAMA	KETERANGAN
1		<i>Action</i>	<i>State</i> dari sistem yang mencerminkan eksekusi dari suatu aksi
2		<i>Initial Node</i>	Menunjukkan bagaimana <i>activity</i> diawali.
3		<i>Activity Final Node</i>	Menunjukkan bagaimana <i>activity</i> berakhir



4		<i>Node</i>	Menunjukkan arah/arus dari satu <i>action</i> ke <i>action</i> selanjutnya
---	---	-------------	--

Simbol *Class Diagram*



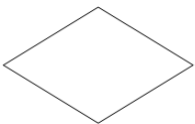
NO	GAMBAR	NAMA	KETERANGAN
1		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
2		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya
3		<i>Composition</i>	Sebuah kelas tidak bisa berdiri sendiri dan harus merupakan bagian dari <i>class</i> yang lain, maka <i>class</i> tersebut memiliki relasi <i>composition</i>
4		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri

Simbol *Sequence Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Participants</i>	Menggambarkan bagian dari sistem yang saling berinteraksi dengan bagian lainnya
		<i>Lifeline</i>	Menunjukkan kapan suatu bagian terlibat didalam urutan rantai sekuensial
2		<i>Activation Bar</i>	Untuk menunjukkan sebuah partisipan sedang aktif menerima atau mengirim pesan
3		<i>Synchronous Message</i>	Menunjukkan komunikasi antar partisipan yang bersifat sinkronus

4		<i>Asynchronous Message</i>	Menunjukkan komunikasi antar partisipan yang bersifat asinkronus
5		<i>Return Message</i>	Menunjukkan pesan balasan komunikasi antar partisipan

Simbol *Flowchart*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Terminator</i>	Digunakan untuk memulai dan mengakhiri sebuah proses
2		<i>Process</i>	Digunakan untuk menunjukkan sebuah proses kegiatan
3		<i>Decision</i>	Menentukan arah proses berikutnya berdasarkan kondisi yang ada

BAB 1

PENDAHULUAN

1.1 Latar Belakang Masalah

Teknologi informasi saat ini semakin maju dan berkembang pesat, terlebih setelah kehadiran perangkat *mobile*, dimana kita semakin dimudahkan untuk mengakses informasi apapun yang diinginkan, menurut survey yang dilakukan Asosiasi Penyelenggara Jasa Internet Indonesia (APJII) pengguna internet di Indonesia tahun 2016 sebanyak 132,7 juta jiwa atau sekitar 51,5% dari jumlah penduduk di Indonesia sebesar 256,2 juta jiwa, dimana paling banyak menggunakan perangkat *mobile smartphone* untuk mengakses internet sebesar 63,1 juta jiwa atau 47,6% [1]. Pengguna internet di Indonesia didominasi oleh anak-anak dan remaja yang menggunakan *smartphone* mereka untuk mengakses internet baik itu untuk mencari referensi belajar seperti untuk tugas-tugas, ataupun bersosial media, forum diskusi dan lain sebagainya [2] namun, belum tentu konten yang disajikan aman terutama untuk anak-anak, seperti menurut survey yang dilakukan APJII bahwa sebanyak 76,4% berpendapat terhadap keamanan berinternet bagi anak tidak aman [1].

Menurut hasil dari pengolahan data kuisioner mengenai tanggapan orang tua terhadap pola perilaku anak dalam menggunakan *smartphone* kepada 20 responden orangtua, diketahui bahwa orang tua tidak dapat mengetahui ataupun memantau sepenuhnya aktivitas yang dilakukan oleh sang anak dengan *smartphone* yang dimilikinya untuk melakukan pengawasan terhadap situs mana sajakah yang anak kunjungi. Padahal, pengawasan orang tua sangatlah diperlukan kepada anak untuk menjaga dari resiko yang tentunya tidak diharapkan oleh setiap orang tua [3] dan tercapai program internet sehat [4].

Untuk mendukung penelitian, dilakukan wawancara kepada ibu Yanti seorang ibu rumah tangga yang memiliki dua orang anak laki-laki kelas 6 dan 3 Sekolah Dasar, dari hasil wawancara tersebut dapat diketahui bahwa orang tua merasa kesulitan dalam memeriksa *smartphone* yang dimiliki oleh anaknya

dikarenakan orang tua hanya dapat menggunakan *smartphone* sebatas pada penggunaan komunikasi seperti menelpon, mengirim pesan singkat, dan sosial media saja sehingga orang tua tidak dapat memeriksa berkas apa saja yang tersimpan di *smartphone* anaknya seperti berkas gambar yang bisa saja mengandung konten negatif.

Masalah lain juga dapat timbul pada saat jam sekolah dan jam pulang sekolah, dimana orang tua tidak dapat mengetahui dengan pasti dimana anak berada, apakah berada di sekolah pada saat jam sekolah dan kemana anak pergi pada saat jam pulang sekolah, dikarenakan terkadang anak tidak meminta izin kepada orang tua untuk pergi ke suatu tempat setelah pulang sekolah.

Orang tua kesulitan untuk mengetahui siapa saja yang ada di kontak *smartphone* anak, dimana untuk mengetahui siapa sajakah teman dari sang anak, untuk dapat mengetahui dengan siapa saja anak bergaul dan berguna untuk orang tua agar orang tua dapat mengetahui siapa sajakah orang terdekat dari sang anak yang dapat orang tua hubungi apabila orangtua membutuhkannya.

Mengingat pentingnya pengawasan yang harus dilakukan orang tua terhadap anaknya sehubungan dengan penggunaan *smartphone* agar anak dapat terhindar dari bahaya konten negatif yang tersebar di internet perlu adanya suatu aplikasi yang dapat melakukan monitoring dan filter pada penggunaan *smartphone* anak, seperti yang pernah dilakukan sebelumnya oleh Taufik Hidayat, Mansur dan Rahmah [5] namun lebih berfokus pada *monitoring* pada *browser* untuk memantau situs yang dikunjungi, Nitin P. Jagtap, Kanchan A. Patil, Shaziya Sayyed Shakil, Nitin S. Ingle [6] tentang monitoring aktifitas menggunakan perangkat *smartphone* Android. Namun di dalam penelitian tersebut untuk melakukan monitoring perangkat Android diperlukan koneksi internet dan aplikasi tersebut tidak dapat digunakan ketika koneksi data dimatikan.

Penelitian juga telah dilakukan sebelumnya tentang pembangunan aplikasi parental control pada anak untuk pengawasan orang tua yang dilakukan oleh Lingga Agitya [7]. Namun, dalam penelitian tersebut belum menyertakan fungsionalitas aplikasi untuk memfilter konten negatif berupa gambar yang ada di *smartphone* anak.

Untuk menangani kelemahan tersebut menurut Kumar dan Qadeer [8], Zhang, Hui, Qizhen, dan Kim Tai-hoon [9], Kuppusamy dan Aghila [10], Rohitaksha Madhu Nalini, dan Nirupama [11], Punjabi, Pooja, Mantur, dan Sneha [12], Nilesh Dhawale, Mahesh Garad, Tushar Darwatkar [13], Vidhi Dave, Amit Welekar [14], Chandran [15], yaitu dengan menggunakan layanan Short Message Service(SMS) untuk melakukan kendali jarak jauh.

Untuk dapat melakukan identifikasi atau pengenalan pada gambar berkonten negatif, dibutuhkan sebuah *Application Programming Interface(API)* atau *Library* yang dapat melakukan pengenalan pada gambar yang memiliki kemampuan untuk secara otomatis memahami gambar atau video berdasarkan elemen dan pola visual yaitu dengan menggunakan API Clarifai [16].

Berdasarkan uraian masalah di atas maka dibutuhkan suatu aplikasi yang dapat membantu memudahkan orang tua dalam melakukan pemeriksaan atau pemindaian berkas gambar di *smartphone* anak dan pengawasan atau *monitoring smartphone* yaitu dengan pembangunan aplikasi monitoring dan pengendalian konten negatif pada perangkat pintar berbasis android menggunakan api Clarifai. Aplikasi akan dibangun berbasis Android karena berdasarkan statistik pengguna *smartphone* pada kuartal I tahun 2017 [17], sebanyak 85% pengguna *smartphone* menggunakan system operasi Android.

1.2 Identifikasi Masalah

Berdasarkan latar belakang masalah yang telah di paparkan, berikut adalah daftar identifikasi masalah dari latar belakang tersebut:

1. Orang tua kesulitan dalam melakukan pengawasan terhadap situs apa sajakah yang diakses anak dengan *smartphone* yang dimilikinya.
2. Orang tua kesulitan dalam memeriksa berkas gambar pada *smartphone* anak.
3. Orang tua kesulitan dalam mengetahui lokasi keberadaan anak.
4. Orang tua kesulitan dalam mengetahui teman-teman anak di dalam kontak yang dapat orang tua hubungi.

1.3 Maksud dan Tujuan Penelitian

Maksud dari penelitian yang dilakukan dan penulisan tugas akhir ini adalah membangun aplikasi monitoring dan pengendalian konten negatif pada perangkat pintar berbasis android menggunakan api Clarifai.

Dengan tujuan untuk sebagai berikut:

1. Memudahkan orang tua dalam melakukan pengawasan terhadap situs apa sajakah yang diakses anak dengan *smartphone* yang dimilikinya.
2. Memudahkan orang tua dalam memeriksa berkas gambar pada *smartphone* anak.
3. Memudahkan orang tua dalam mengetahui lokasi keberadaan anak.
4. Membantu orang tua dalam mengetahui teman-teman anak di dalam kontak yang dapat orang tua hubungi.

1.4 Batasan Masalah

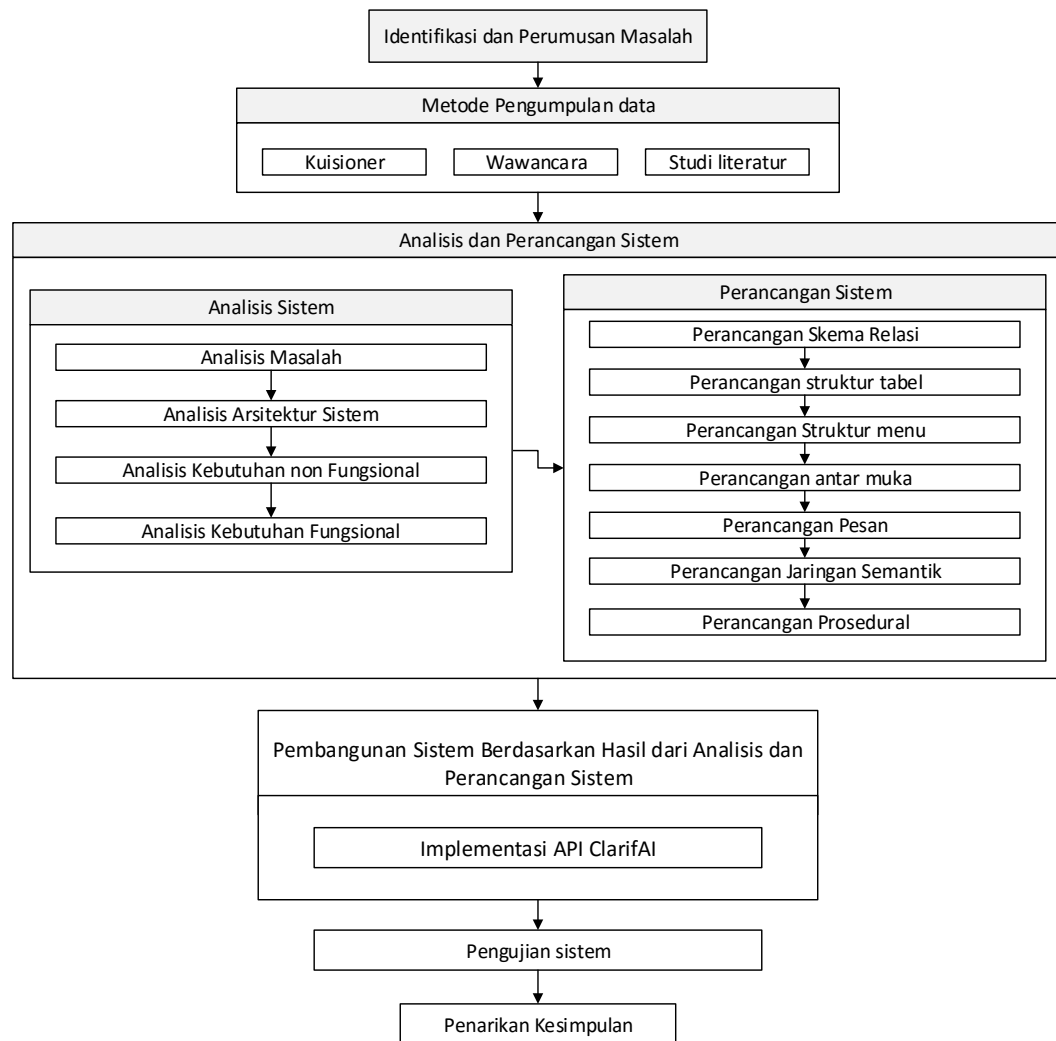
Batasan masalah dalam pembangunan aplikasi parental control untuk pencarian konten negatif pada pengguna *smartphone* dibawah umur adalah:

1. Aplikasi memiliki dua versi, versi anak dan versi orang tua.
2. Aplikasi terlebih dahulu harus dilakukan instalasi pada *smartphone* anak dan orang tua.
3. Berkas yang dicari dan diidentifikasi berupa gambar, serta *monitoring* pada *history browser* dan melakukan tindakan pada *smartphone* anak dari *smartphone* orang tua.
4. *Webservice* dibangun dengan bahasa pemrograman PHP dengan *framework* Codeigniter.
5. Pertukaran data dengan server yang dilakukan menggunakan JSON (*JavaScript Object Notation*).
6. DBMS yang digunakan adalah Mysql .
7. Model analisis yang digunakan dalam pembangunan aplikasi ini adalah OOAD (*Object Oriented Analysis and Design*) dengan UML (*Unified Modeling Language*).

8. *Tools* yang digunakan untuk pembangunan aplikasi adalah Android Studio dan Sublime text.
9. Fungsionalitas yang dapat dilakukan oleh sistem adalah sebagai berikut :
 - a. Melakukan pencarian konten negatif berupa gambar.
 - b. Melakukan monitoring pada *history browser*.
 - c. Orang tua dapat mengetahui posisi anak dari *gps smartphone*.
 - d. Orang tua dapat mengetahui nomor kontak yang ada di *smartphone* anak.
 - e. Orang tua dapat melakukan tindakan penghapusan berkas pada *smartphone* anak.
 - f. Orang tua akan mendapat notifikasi pemberitahuan apabila anak mengakses situs yang diblok oleh provider.

1.5 Metodologi Penelitian

Metode penelitian di gunakan untuk memberikan suatu pemecahan masalah yang sesuai dan logis, dimana membutuhkan data-data yang mendukung berjalannya suatu penelitian. Dalam pembuatan skripsi ini digunakan metodologi deskriptif, metode deskriptif merupakan sebuah metode yang bertujuan untuk membuat gambaran fakta-fakta dan informasi secara sistematis dan akurat. Metode penelitian ini memiliki tahapan-tahapan yang dapat di lihat seperti pada gambar berikut.



Gambar 1 Kerangka Kerja Penelitian

Berikut ini adalah penjelasan mengenai alur penelitian pada **Gambar 1.1**:

1.5.1 Identifikasi Masalah dan Perumusan masalah

Tahapan ini merupakan tahapan awal penelitian yang dimulai dengan melakukan identifikasi pada masalah-masalah yang mungkin timbul dan dirasakan oleh orang tua yang memiliki anak yang diberikan *smartphone* Android oleh orangtuanya. Setelah masalah ditemukan kemudian tahapan selanjutnya adalah melakukan perumusan masalah dan menentukan metode yang tepat untuk mengimplementasikan sistem agar masalah yang ada dapat terselesaikan.

1.5.2 Pengumpulan Data

Pada tahapan ini merupakan tahapan dimana peneliti menentukan bagaimana cara memperoleh data yang diperlukan untuk dapat digunakan pada sistem sebagai data masukan dalam penelitian. Berikut adalah tahapan yang digunakan pada pengumpulan data.

1. Kuisioner

Metode kuisioner digunakan peneliti untuk mengetahui perilaku orangtua dalam menggunakan *smartphone* Android dan tanggapan orang tua terhadap pola perilaku anak dalam menggunakan *smartphone*, dengan memberikan daftar pertanyaan yang berkaitan dengan judul penelitian.

2. Wawancara

Wawancara dilakukan kepada orang tua yang memiliki anak yang diberikan *smartphone* Android oleh orang tuanya, dalam hal ini Ibu Yanti yang memiliki anak laki-laki berusia 12 tahun yang telah diberikan *smartphone* Android dan beliau merasa kesulitan dalam pengawasan terhadap penggunaan *smartphone* oleh putranya tersebut. Wawancara dilakukan untuk memperoleh data dan fakta yang dibutuhkan.

3. Studi Literatur

Studi literatur dilakukan untuk menambah data dan kajian-kajian yang dibutuhkan yang berasal dari jurnal ilmiah maupun buku yang berhubungan dengan judul penelitian untuk mengetahui perbedaan dengan penelitian sebelumnya.

1.5.3 Analisa dan Perancangan Sistem

Pada tahap ini dilakukan analisis terhadap sistem yaitu dengan melakukan identifikasi dan evaluasi terhadap permasalahan-permasalahan, kesempatan, hambatan-hambatan serta kebutuhan yang diharapkan sehingga dapat sesuai dengan kebutuhan. Berikut adalah tahapan-tahapan yang digunakan dalam analisis dan perancangan.

1. Analisis Sistem
2. Analisis Masalah
3. Analisis Arsitektur Sistem
4. Analisis Kebutuhan Non Fungsional
 - a. Analisis Kebutuhan Perangkat Keras
 - b. Analisis Kebutuhan Perangkat Lunak
 - c. Analisis Pengguna
5. Analisis Kebutuhan Fungsional
 - a. *Use Case Diagram*
 - b. *Use Case Scenario*
 - c. *Activity Diagram*
 - d. *Class Diagram*
 - e. *Sequence Diagram*

1.5.4 Perancangan Sistem

Pada tahapan ini bertujuan untuk menemukan bentuk optimal dari aplikasi yang akan di bangun dengan mempertimbangkan masalah-masalah dan kebutuhan yang ada pada sistem seperti yang telah di tentukan dengan mengkombinasikan penggunaan teknologi perangkat keras dan perangkat lunak yang ada agar diperoleh aplikasi yang tepat dan optimal, mudah di implementasikan dan dapat memberikan solusi bagi masalah-masalah yang ada. Berikut adalah tahapan-tahapan pada perancangan system.

- a. Perancangan Skema Relasi
- b. Perancangan Struktur Tabel
- c. Perancangan Struktur Menu
- d. Perancangan Antar Muka
- e. Perancangan Pesan
- f. Perancangan Jaringan Semantik
- g. Perancangan Prosedural

1.5.5 Pembangunan Sistem

Pada tahap ini adalah tahap dimana pembangunan sistem dilakukan berdasarkan dari tahapan sebelumnya yaitu analisis dan perancangan sistem sehingga sistem yang dibangun akan sesuai dengan kebutuhan yang didapat dari analisis dan perancangan sistem.

1.5.6 Pengujian Sistem

Pada tahapan ini adalah tahapan dimana proses pengujian sistem yang telah dibangun, tahapan ini bertujuan untuk memastikan setiap pernyataan telah teruji dan sesuai dengan hasil yang dibutuhkan serta untuk mengungkap kesalahan yang mungkin terjadi

1.5.7 Penarikan Kesimpulan

Tahapan ini adalah tahapan akhir dimana akan ditarik kesimpulan terhadap sistem yang telah dibangun apakah memberikan hasil yang sesuai dengan tujuan awal dari penelitian atau belum.

1.6 Sistematika Penulisan

Sistematika penulisan laporan akhir dari penelitian ini disusun untuk memberikan gambaran umum tentang penelitian yang dilakukan. Sistematika penulisan tugas akhir ini adalah sebagai berikut :

BAB 1 PENDAHULUAN

Bab ini menguraikan tentang latar belakang permasalahan, bertujuan untuk mengidentifikasi dan merumuskan inti permasalahan yang dihadapi serta pendekatan yang di gunakan untuk mencapai hasil akhir dari penelitian, menentukan maksud dan tujuan penelitian, yang kemudian diikuti dengan pembatasan masalah, metodologi penelitian serta sistematika penulisan.

BAB 2 TINJAUAN PUSTAKA

Bab ini membahas tentang berbagai konsep dasar dan teori-teori yang berhubungan dengan topik penelitian seperti komponen-komponen yang terlibat

dengan system seperti *server*, jenis *database* yang digunakan, *web services* untuk bertukar data, API Clarifai, dan layanan *Short Message Service(SMS)* yang mendukung dalam pembangunan sistem.

BAB 3 ANALISIS DAN PERANCANGAN SISTEM

Bab ini berisi analisis sistem yang terdiri dari analisis sistem, analisis masalah, analisis kebutuhan fungsional maupun non fungsional untuk aplikasi yang akan dibangun, perancangan skema relasi, dan perancangan struktur tabel. Selain itu, pada bab ini juga akan digambarkan perancangan dan struktur antar muka untuk aplikasi yang akan dibangun.

BAB 4 IMPLEMENTASI SISTEM DAN PENGUJIAN

Bab ini berisi tentang batasan implementasi, spesifikasi *hardware*, spesifikasi *software*, dan *brainware*, serta tahapan-tahapan implementasi antarmuka hasil rancangan yang telah dibuat sebelumnya.

BAB 5 KESIMPULAN DAN SARAN

Bab ini merupakan penutup penyusunan laporan yang berisi rangkuman dari implementasi dan uji coba yang dilakukan. Selain itu berisi pula saran yang diharapkan dapat menjadi masukan untuk pengembangan aplikasi di masa yang akan datang.

BAB 2

LANDASAN TEORI

2.1 Aplikasi

Aplikasi adalah sebuah program atau perangkat lunak yang dirancang untuk memenuhi tujuan tertentu, menurut *English Oxford Living Dictionary* . Sedangkan menurut Jogiyanto [18] dalam penelitian Lingga Agitya [7] aplikasi merupakan penerapan, menyimpan, sesuatu hal, data permasalahan. Pekerjaan ke dalam suatu sarana atau media yang dapat digunakan untuk menerapkan atau mengimplementasikan hal atau permasalahan yang ada sehingga berubah menjadi suatu bentuk yang baru tanpa menghilangkan nilai-nilai dasar dari hal data, permasalahan, pekerjaan itu sendiri.

Dari pengertian diatas peneliti dapat menyimpulkan bahwa aplikasi adalah suatu program atau perangkat lunak yang dibuat untuk tujuan tertentu untuk dapat menyelesaikan suatu masalah dalam lingkup tertentu yang dihadapi baik itu dibuat dan dikembangkan dengan tujuan untuk melakukan tugas yang bersifat umum atau juga dapat dikembangkan untuk melakukan tugas khusus. Hal ini didasarkan dari aplikasi dengan paket-paket tertentu yang unik sehingga tidak ada paket-paket program yang sesuai yang dapat digunakan. Oleh karena itu pengembangan aplikasi itu sendiri perlu dilakukan.

2.2 Monitoring

Monitoring adalah proses menjaga atau pengawasan terhadap keberadaan dan besarnya perubahan keadaan dan arus data dalam sebuah sistem. *Monitoring* bertujuan untuk mengidentifikasi kesalahan dan kemajuan dalam menentukan keputusan selanjutnya. Teknik yang digunakan dalam *monitoring* informasi sistem memotong bidang pengolahan real-time, statistik, dan analisis data. Satu set Komponen perangkat lunak yang digunakan untuk pengumpulan data, pengolahan, dan presentasi disebut sistem *monitoring* [19].

2.3 Konten Negatif

Menurut Kamus Besar Bahasa Indonesia secara arti kata konten berarti informasi yang tersedia melalui media atau produk elektronik, sedangkan negatif berarti tidak pasti atau kurang pasti, tidak baik atau kurang baik [20]. Jadi, dapat diambil kesimpulan bahwa konten negatif adalah suatu informasi yang terdapat di pada media elektronik baik itu TV, Radio, *Smartphone*, atau dalam sebuah halaman web yang mengandung unsur-unsur yang menyimpang atau kurang baik untuk disajikan kepada masyarakat umum yang biasanya bermuatan kekerasan dan pornografi.

Menurut Kementerian Komunikasi dan Informatika RI (KOMINFO) sejak bulan April 2015, Menteri Komunikasi dan Informatika melalui SK Menkominfo Nomor 290 Tahun 2015 membentuk Forum Penanganan Situs Internet Bermuatan Negatif yang terdiri atas 4 (empat) Panel, yaitu :

1. Panel Pornografi, Kekerasan Pada Anak, dan Keamanan Internet.
2. Panel Terorisme, SARA, dan Kebencian.
3. Panel Investasi Ilegal, Penipuan, Perjudian, Obat & Makanan, dan Narkoba.
4. Panel Hak Kekayaan Intelektual.

Dalam penelitian ini peneliti lebih memfokuskan konten negatif pada perangkat *smartphone* Android berupa gambar yang mengandung unsur *pornografi*.

2.4 Perangkat Pintar

Perangkat pintar adalah sebuah alat elektronik interaktif yang mengerti perintah sederhana yang dikirim oleh pengguna dan membantu dalam aktivitas sehari-hari. Beberapa perangkat pintar yang paling umum digunakan adalah *smartphone*, tablet, *phablets*, *smartwatch*, kacamata pintar dan peralatan elektronik pribadi lainnya. Sementara banyak perangkat pintar kecil, elektronik pribadi portabel, perangkat tersebut sebenarnya ditentukan oleh kemampuan mereka untuk

terhubung ke jaringan untuk berbagi dan berinteraksi dari jarak jauh. Banyak TV dan kulkas juga dianggap perangkat cerdas [21].

Dalam penelitian ini peneliti menggunakan perangkat pintar berupa ponsel pintar atau *smartphone* dengan sistem operasi Android yang digunakan sebagai media dalam penelitian.

2.5 Smartphone

Menurut Williams dan Sawyer [22] *smartphone* adalah telepon seluler dengan mikroprosesor, memori, layar tampilan, dan modem *built-in*. *Smartphone* menggabungkan fungsionalitas PDA atau (*Pocket PC*) dengan telepon dalam satu alat yang sama sehingga menghasilkan *gadget* yang memiliki banyak fitur dan fungsionalitas, dimana terdapat pesan teks, kamera, pemutar musik, video, *game*, akses email, tv digital, *search engine*, pengelola informasi pribadi, fitur GPS, jasa telepon internet, dan bahkan terdapat telepon yang juga berfungsi sebagai kartu kredit.

Dalam penelitian ini, peneliti menggunakan *smartphone* dengan sistem operasi Android sebagai alat untuk dapat melakukan *monitoring* dengan mengakses perangkat keras melalui *Application Programming Interface (API)* guna mendapatkan data dan hasil yang diinginkan.

2.6 Android

Menurut Nazaruddin Safaat H [23] Android adalah Sebuah sistem informasi untuk perangkat *mobile* berbasis linux yang mencakup sistem operasi, *middleware* dan aplikasi. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Awalnya Google Inc. membeli Android Inc. yang merupakan pendatang baru yang membuat peranti lunak untuk ponsel/*smartphone*. Kemudian untuk mengembangkan android dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.

Pada saat perilisan Perdana Android, 5 November 2007 Android Bersama Open Handset Alliance menyatakan mendukung pengembangan *open source* pada

perangkat *mobile*. Di lain pihak, Google merilis kode-kode Android dibawah lisensi Apache, sebuah lisensi perangkat lunak dan *open platform* perangkat seluler.

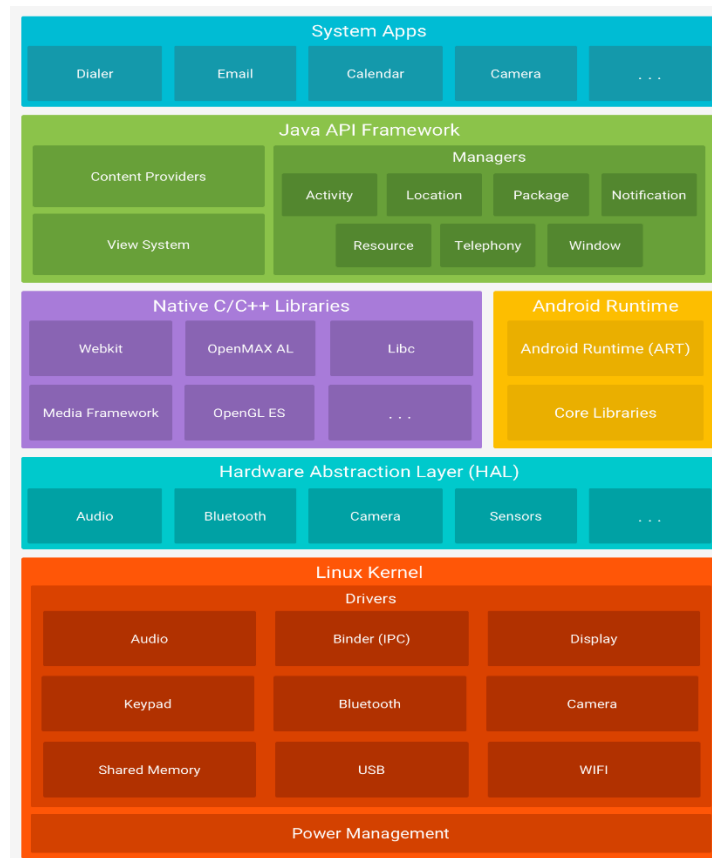
Hingga awal menjelang penghujung tahun 2017 terdapat beberapa versi Android yang tersedia untuk digunakan, diantaranya adalah sebagai berikut:

Tabel 2.1 Rilis Versi Android

Nama Kode	Versi	API level
Oreo	8.0	Api level 26
Nougat	7.1	API level 25
Nougat	7.0	API level 24
Marshmallow	6.0	API level 23
Lollipop	5.1	API level 22
Lollipop	5.0	API level 21
KitKat	4.4 - 4.4.4	API level 19
Jelly Bean	4.3.x	API level 18
Jelly Bean	4.2.x	API level 17
Jelly Bean	4.1.x	API level 16
Ice Cream Sandwich	4.0.3 - 4.0.4	API level 15, NDK 8
Ice Cream Sandwich	4.0.1 - 4.0.2	API level 14, NDK 7
Honeycomb	3.2.x	API level 13
Honeycomb	3.1	API level 12, NDK 6
Honeycomb	3.0	API level 11
Gingerbread	2.3.3 - 2.3.7	API level 10
Gingerbread	2.3 - 2.3.2	API level 9, NDK 5
Froyo	2.2.x	API level 8, NDK 4
Eclair	2.1	API level 7, NDK 3
Eclair	2.0.1	API level 6
Eclair	2.0	API level 5
Donut	1.6	API level 4, NDK 2
Cupcake	1.5	API level 3, NDK 1
(Tidak ada nama kode)	1.1	API level 2
(Tidak ada nama kode)	1.0	API level 1

2.7 Arsitektur Android

Secara garis besar arsitektur android terdiri dari komponen utama yang tersusun seperti pada Gambar 2.



Sumber Gambar : <https://developer.android.com/guide/platform/index.html?hl=id>

Gambar 2 Arsitektur Android

1. Linux Kernel

Fondasi platform Android adalah kernel Linux. Sebagai contoh, Android *Runtime* (ART) bergantung pada kernel Linux untuk fungsionalitas dasar seperti threading dan manajemen memori tingkat rendah.

Menggunakan kernel Linux memungkinkan Android untuk memanfaatkan fitur keamanan inti dan memungkinkan produsen perangkat untuk mengembangkan *driver* perangkat keras untuk kernel yang cukup dikenal [24].

2. *Hardware Abstraction Layer (HAL)*

Hardware Abstraction Layer (HAL) menyediakan antarmuka standar yang mengekspos kemampuan perangkat keras di perangkat ke kerangka kerja Java API yang lebih tinggi. HAL terdiri atas beberapa modul pustaka, masing-masing mengimplementasikan antarmuka untuk komponen perangkat keras tertentu, seperti modul kamera atau bluetooth. Bila API kerangka kerja melakukan panggilan untuk mengakses perangkat keras, sistem Android memuat modul pustaka untuk komponen perangkat keras tersebut [24].

3. *Android Runtime*

Untuk perangkat yang menjalankan Android versi 5.0 (API level 21) atau yang lebih tinggi, setiap aplikasi menjalankan proses masing-masing dengan tahap *Android Runtime* (ART). ART ditulis guna menjalankan beberapa mesin virtual pada perangkat bermemori rendah dengan mengeksekusi file DEX, format *bytecode* yang didesain khusus untuk Android yang dioptimalkan untuk *footprint* memori minimal. Buat rantai aplikasi, misalnya Jack, mengumpulkan sumber Java ke *bytecode* DEX, yang dapat berjalan pada platform Android.

Beberapa fitur utama ART mencakup:

1. Kompilasi mendahului waktu (AOT) dan tepat waktu (JIT).
2. Pengumpulan sampah (GC) yang dioptimalkan.
3. Dukungan debug yang lebih baik, mencakup profiler sampling terpisah, pengecualian diagnostik mendetail dan laporan kerusakan dan kemampuan untuk mengatur titik pantau guna memantau bidang tertentu.

Sebelum ke Android versi 5.0 (API level 21), Dalvik adalah waktu proses. Android. Jika aplikasi Anda berjalan baik pada ART, semestinya berfungsi baik juga pada Dalvik, tetapi mungkin tidak sebaliknya. Android juga menyertakan serangkaian pustaka waktu proses inti yang menyediakan sebagian besar fungsionalitas bahasa pemrograman Java, termasuk beberapa fitur bahasa Java 8, yang digunakan kerangka kerja Java API [24].

4. *Native C/C++ Library*

Banyak komponen dan layanan sistem Android inti seperti ART dan HAL dibuat dari kode asli yang memerlukan pustaka asli yang tertulis dalam C dan C++. Platform Android memungkinkan kerangka kerja Java API mengekspos fungsionalitas beberapa pustaka asli pada aplikasi. Misalnya, Anda bisa mengakses OpenGL ES melalui kerangka kerja Java OpenGL API Android guna menambahkan dukungan untuk menggambar dan memanipulasi grafik 2D dan 3D pada aplikasi Anda.

Jika Anda mengembangkan aplikasi yang memerlukan kode C atau C++, Anda bisa menggunakan Android NDK untuk mengakses beberapa pustaka *platform* asli langsung dari kode asli [24].

5. *Java API Framework*

Keseluruhan rangkaian fitur pada Android OS tersedia untuk Anda melalui API yang ditulis dalam bahasa Java. API ini membentuk elemen dasar yang Anda perlukan untuk membuat aplikasi Android dengan menyederhanakan penggunaan kembali inti, komponen dan layanan sistem modular, yang menyertakan berikut ini:

1. Tampilan Sistem yang kaya dan luas bisa Anda gunakan untuk membuat UI aplikasi, termasuk daftar, kisi, kotak teks, tombol, dan bahkan *browser* web yang dapat disematkan.
2. Pengelola Sumber Daya, memberikan akses ke sumber daya bukan kode seperti string yang dilokalkan, grafik, dan file *layout*.
3. Pengelola Notifikasi yang mengaktifkan semua aplikasi guna menampilkan lansiran khusus pada bilah status.
4. Pengelola Aktivitas yang mengelola daur hidup aplikasi dan memberikan back-stack navigasi yang umum.
5. Penyedia Materi yang memungkinkan aplikasi mengakses data dari aplikasi lainnya, seperti aplikasi Kontak, atau untuk berbagi data milik sendiri.

Developer memiliki akses penuh ke API kerangka kerja yang sama dengan yang digunakan oleh aplikasi sistem Android [24].

6. *System Apps*

Android dilengkapi dengan serangkaian aplikasi inti untuk email, perpesanan SMS, kalender, menjelajahi internet, kontak, dll. Aplikasi yang disertakan bersama platform tidak memiliki status khusus pada aplikasi yang ingin dipasang pengguna. Jadi, aplikasi pihak ketiga dapat menjadi browser web utama, pengolah pesan SMS atau bahkan keyboard utama (beberapa pengecualian berlaku, seperti aplikasi *Settings* sistem).

Aplikasi sistem berfungsi sebagai aplikasi untuk pengguna dan memberikan kemampuan kunci yang dapat diakses oleh *developer* dari aplikasi mereka sendiri. Misalnya, jika aplikasi Anda ingin mengirimkan pesan SMS, Anda tidak perlu membangun fungsionalitas tersebut sendiri-sebagai gantinya Anda bisa menjalankan aplikasi SMS mana saja yang telah dipasang guna mengirimkan pesan kepada penerima yang Anda tetapkan [24].

2.8 Komponen Aplikasi Android

Aplikasi android ditulis dalam bahasa pemrograman java. kode java dikompilasi bersama dengan data file yang dibutuhkan oleh aplikasi dimana prosesnya dikemas oleh *tools* yang dinamakan “*apt tools*” ke dalam paket android sehingga menghasilkan file dengan ekstensi apk. file apk itulah yang kita sebut dengan aplikasi dan nantinya dapat dipasang pada perangkat mobile [23].

terdapat empat jenis komponen pada aplikasi android yaitu:

1. Activity

Activity akan menyajikan *user interface* (ui) kepada pengguna sehingga pengguna dapat melakukan interaksi. Sebuah aplikasi android bisa jadi hanya memiliki satu *activity*, tetapi umumnya aplikasi memiliki banyak *activity* yang bergantung pada tujuan aplikasi dan desain aplikasi itu sendiri. Untuk berpindah dari satu *activity* ke *activity* lain dapat dilakukan menggunakan sebuah *trigger* seperti klik tombol pada tampilan aplikasi.

2. Service

Service tidak memiliki *graphic user interface* (gui), tetapi *service* berjalan secara *background*. Sebagai contoh dalam memutar musik, *service* mungkin

memutar musik atau mengambil data dari jaringan, tetapi setiap *service* harus berada dalam kelas induknya. Apabila sebuah pemutar musik sedang memutar lagu dari list yang ada, aplikasi akan memiliki dua atau lebih *activity* yang memungkinkan pengguna untuk memutar sambil memilih lagu baru. Untuk menjaga musik tetap berjalan sebuah *activity* dapat menjalankan *service*.

3. *Broadcast Receiver*

Broadcast receiver berfungsi menerima dan bereaksi untuk menyampaikan notifikasi. contoh *broadcast* seperti notifikasi zona waktu telah berubah, baterai lemah, gambar berhasil diambil oleh kamera, dan lain-lain. Aplikasi juga dapat meng-inisialisasi *broadcast* misalnya memberikan informasi pada aplikasi lain bahwa ada data yang telah diunduh ke perangkat dan siap untuk digunakan.

4. *Content Provider*

Content provider membuat kumpulan aplikasi data secara spesifik sehingga bisa digunakan oleh aplikasi lain. Data disimpan dalam file sistem seperti database sqlite. content provider menyediakan cara untuk mengakses data yang dibutuhkan oleh suatu *activity*.

2.9 **Android SDK(Software Development Kit)**

Android SDK (*Software Development Kit*) adalah *tools API (Application Programming Interface)* yang diperlukan untuk mulai mengembangkan aplikasi pada *platform* Android. Android SDK disediakan sebagai alat bantu untuk mulai mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java. Sebagai *platform* aplikasi-netral, Android memberi kesempatan untuk membuat aplikasi yang dibutuhkan [23].

Android SDK versi resmi dapat ditemukan dan diunduh di situs resmi Google. Google akan merilis SDK versi baru yang disesuaikan ketika Android versi terbaru juga. Untuk dapat mengembangkan aplikasi dengan fitur terbaru maka pengembang harus mengunduh Android SDK dengan versi terbaru pula yang disesuaikan dengan target *device* masing-masing.

2.10 Android Studio

Android Studio adalah Lingkungan Pengembangan Terpadu *Integrated Development Environment* (IDE) untuk pengembangan aplikasi Android, berdasarkan IntelliJ IDEA . Selain merupakan *editor* kode IntelliJ dan alat pengembang yang berdaya guna, Android Studio menawarkan fitur lebih banyak untuk meningkatkan produktivitas Anda saat membuat aplikasi Android, misalnya:

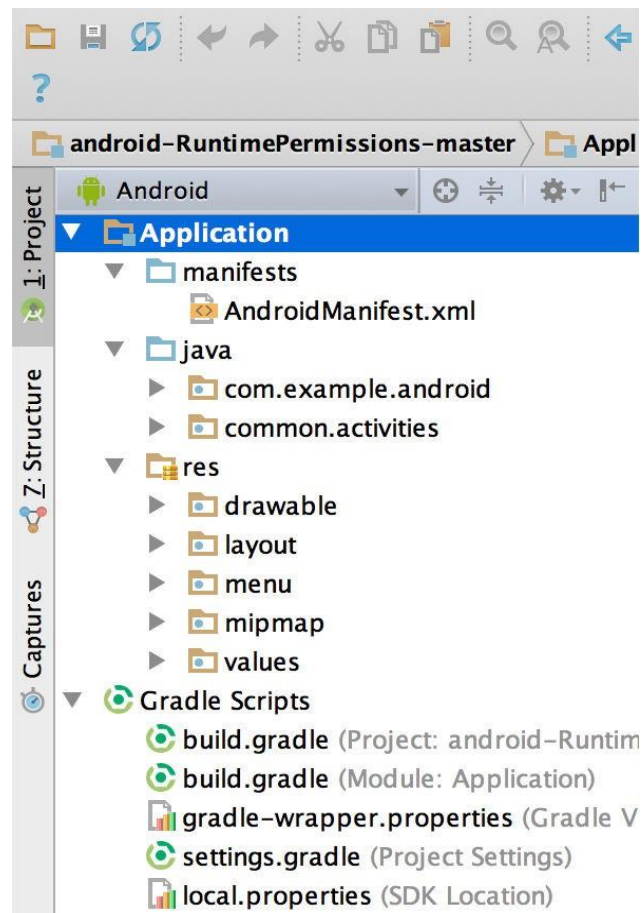
1. Sistem versi berbasis Gradle yang fleksibel.
2. *Emulator* yang cepat dan kaya fitur.
3. Lingkungan yang menyatu untuk pengembangan bagi semua perangkat Android.
4. *Instant Run* untuk mendorong perubahan ke aplikasi yang berjalan tanpa membuat APK baru.
5. Template kode dan integrasi GitHub untuk membuat fitur aplikasi yang sama dan mengimpor kode contoh.
6. Alat pengujian dan kerangka kerja yang ekstensif.
7. Alat Lint untuk meningkatkan kinerja, kegunaan, kompatibilitas versi, dan masalah-masalah lain.
8. Dukungan C++ dan NDK.
9. Dukungan bawaan untuk Google *Cloud Platform*, mempermudah pengintegrasian Google *Cloud Messaging* dan *App Engine*.

Setiap proyek di Android Studio berisi satu atau beberapa modul dengan file kode sumber dan file sumber daya. Jenis-jenis modul mencakup:

1. Modul aplikasi Android.
2. Modul Pustaka.
3. Modul Google *App Engine*.

Secara *default*, Android Studio akan menampilkan file proyek Anda dalam tampilan proyek Android, seperti yang ditampilkan dalam gambar 3. Tampilan

disusun berdasarkan modul untuk memberikan akses cepat ke *file* sumber utama proyek Anda.



Sumber Gambar : <https://developer.android.com/studio/intro/index.html?hl=id>

Gambar 3 File Proyek di Tampilan Android

Semua file versi terlihat di bagian atas di bawah *Gradle Scripts* dan masing-masing modul aplikasi berisi folder berikut:

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

Struktur proyek Android pada *disk* berbeda dari representasi rata ini. Untuk melihat struktur file sebenarnya dari proyek ini, pilih Project dari menu tarik turun Project (dalam *Gambar 3*, struktur ditampilkan sebagai Android) [25].

2.11 Bahasa Pemrograman Java

Java merupakan sebuah Bahasa pemrograman berorientasi objek yang dapat berjalan pada *platform* yang berbeda, baik di Windows, Linux, serta system operasi lainnya. Dengan menggunakan Java kita dapat mengembangkan banyak aplikasi yang dapat digunakan pada lingkungan yang berbeda, seperti pada *Desktop*, *Mobile*, *Internet* dan lain-lain.

Untuk menginstalasi dan menggunakan Java, Sun Micro System selaku pengembang Java menyediakan paket instalasi sesuai dengan kebutuhan kita dalam membangun suatu aplikasi. Berikut ini uraian singkat mengenai paket aplikasi Java yang tersedia.

1. J2ME (Java 2 *Micro Edition*)

Paket instalasi ini dapat digunakan untuk membangun *software* yang berjalan pada perangkat yang memiliki memori dan sumber daya yang kecil, seperti pada *handphone*, PDA, dan *Smartcard*.

2. J2SE (Java 2 *Standard Edition*)

Paket instalasi ini digunakan untuk mengembangkan aplikasi yang berjalan pada lingkungan *workstation*, seperti aplikasi desktop.

3. J2EE (Java 2 *Enterprise Edition*)

Paket instalasi ini dapat digunakan untuk mengembangkan aplikasi pada lingkungan internet maupun aplikasi skala *enterprise*.

Java juga merupakan bahasa pemrograman resmi yang digunakan untuk pembangunan aplikasi android yang didukung penuh oleh Google. Namun meskipun demikian saat ini java bukanlah satu-satunya Bahasa yang dapat digunakan untuk membangun aplikasi Android seperti Xamarin dengan

menggunakan Bahasa pemrograman C#, Cordova dengan menggunakan bahasa pemrograman web seperti HTML, CSS, dan Javascript dan lain-lain.

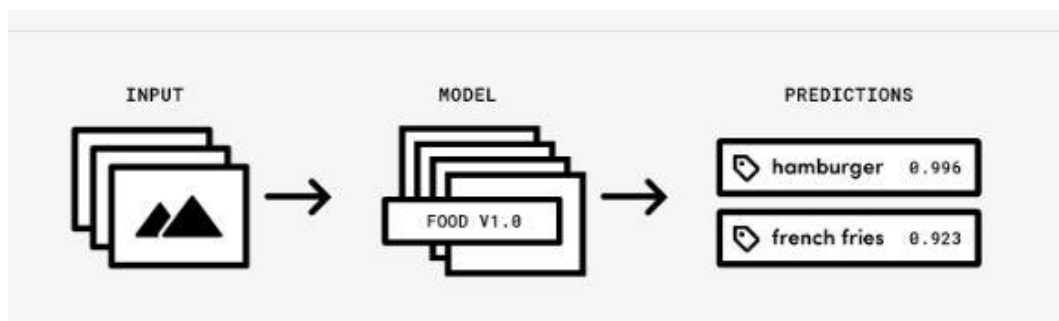
2.12 Clarifai

Clarifai adalah perusahaan kecerdasan buatan yang unggul dalam pengenalan visual (*Visual Recognition*), Clarifai didirikan pada tahun 2013 oleh Matthew Zeiler, seorang ahli terkemuka dalam *Machine Learning*, Clarifai telah menjadi pemimpin pasar sejak memenangkan lima tempat teratas dalam klasifikasi citra pada kompetisi ImageNet 2013 [16].

Clarifai adalah alat yang dapat mengidentifikasi atau mengenali gambar atau video yang dimasukkan sebagai inputan yang dapat memberikan hasil berupa prediksi tentang apa yang ada di dalam gambar atau video berupa besaran probabilitas kemungkinannya [26].

API Clarifai dibangun dengan konsep ide dasar yang sederhana dimana anda memasukkan inputan berupa gambar atau video dan kemudian anda akan mendapatkan hasil dari prediksi yang diminta.

Setiap tipe dari prediksi adalah berdasar dari model apa yang dipilih untuk menjalankan prediksi dari inputan gambar atau video. Sebagai contoh apabila anda memilih model makanan untuk gambar masukkan anda maka Clarifai akan memprediksi gambar berdasarkan model makanan yang telah diketahui dan begitu seterusnya.



Sumber Gambar : <https://www.clarifai.com/developer/guide/#getting-started>

Gambar 4 Konsep kerja Clarifai

2.12.1 Model Clarifai

Gambar atau video yang menjadi *data* masukkan yang dijalankan ditandai oleh model dimana sebuah model adalah *classifier* yang terlatih dan dapat mengenali apa yang terdapat di dalam gambar atau video sesuai dengan apa yang telah diketahui. Model satu dan yang lainnya digunakan untuk mengetahui hal-hal yang berbeda pula. Menjalankan model yang berbeda untuk mengenali suatu gambar atau video dapat memberikan hasil yang berbeda secara drastis.

Berikut adalah beberapa contoh dari model pada Clarifai :

1. *General Model*

General model adalah model dimana berisi tag yang lebih umum dan berisi berbagai topik yang berbeda dan akan menghasilkan prediksi yang umum yang terdapat di gambar.

2. *Face Model*

Face model memberikan hasil prediksi berupa besaran probabilitas yang terdeteksi berdasar model wajah yang diketahui dari gambar yang Anda masukkan

3. *Not Safe For Work (NSFW) model*

NSFW model analisis berupa gambar atau video berdasar pada besaran skor probabilitas hasilpengembalian kemungkinan bahwa gambar bermuatan pornografi.

Selain yang telah dipaparkrn di atas masih ada model-model yang lain seperti *color*, *weding*, *travel*, *celebrity*, *demographic*, *logo*, *food*, *apparel*, *focus*, *moderation* yang masing-masing memiliki fungsi dan fokus yang berbeda.

2.12.2 Authentication

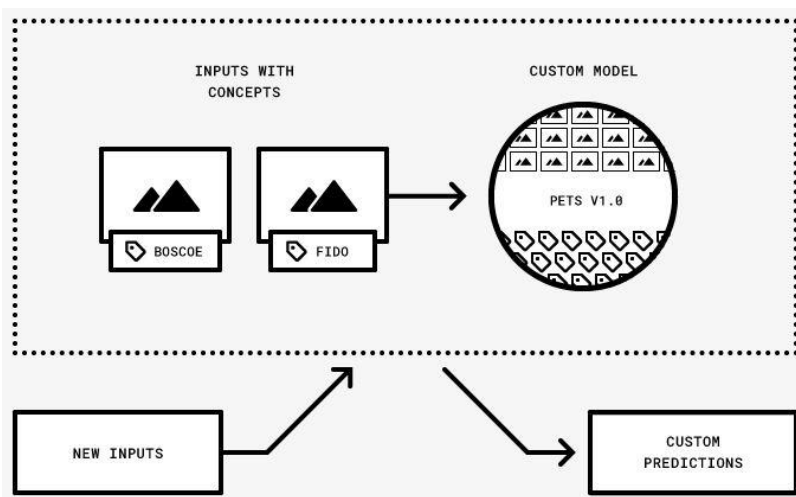
Otentikasi ke API ditangani melalui *API key*. Anda dapat membatasi cakupan *API key*, yang memungkinkan kunci untuk melakukan operasi yang sangat spesifik pada aplikasi yang diberikan, agar aplikasi Anda tetap aman.

2.12.3 Predict

Predict, melakukan prediksi dan analisa gambar yang Anda masukkan dan memberi tahu Anda apa isi yang terkandung di gambar. API akan mengembalikan daftar konsep dengan probabilitas yang sesuai dari seberapa besar kemungkinan konsep-konsep ini terkandung di dalam gambar. Bila Anda membuat prediksi melalui API, Anda memberi tahu model apa yang akan digunakan. Sebuah model berisi sekelompok konsep. Model hanya akan melihat konsep yang dikandungnya. Anda dapat menggunakan model yang berbeda untuk menganalisis gambar dengan berbagai cara.

2.12.4 Train

Train atau pelatihan memungkinkan Anda membuat model sendiri menggunakan konsep kustom Anda sendiri. Anda mulai dengan menambahkan masukan gambar yang sudah Anda ketahui berisi konsep yang Anda minati. Anda tidak memerlukan banyak gambar untuk memulai. Sebaiknya mulai dengan 10 dan tambahkan lebih banyak sesuai kebutuhan. Kemudian membuat model dan menentukan konsep apa yang dikandungnya. Setelah membuat model, Anda melatihnya berdasarkan gambar dan konsep yang Anda berikan. Operasi pelatihan ini bersifat asinkron. Mungkin perlu beberapa detik agar model Anda dilatih dan siap sepenuhnya. Setelah selesai latihan, Anda bisa menggunakan model itu untuk memprediksi konsep tersebut pada gambar baru.



Sumber Gambar : <https://www.clarifai.com/developer/guide/train#train>

Gambar 5 Train

Dalam penelitian ini peneliti menggunakan *Application Programming Interface* (API) Clarifai untuk melakukan identifikasi gambar dengan konten negatif yang telah dilakukan pelatihan sebelumnya.

2.13 Web Service

Web service adalah standar yang digunakan untuk pertukaran data antar aplikasi atau sistem. Standar diperlukan karena masing-masing aplikasi yang melakukan pertukaran data bisa ditulis dengan bahasa pemrograman yang berbeda atau berjalan pada platform yang berbeda. Contoh implementasi dari *web service* antara lain adalah SOAP dan REST [27].

Peneliti menggunakan *web service* yang ditempatkan di sebuah server untuk dapat menjalankan sistem sesuai dengan yang diharapkan sehingga aplikasi dapat mengambil serta menyimpan data yang diperlukan seperti data pengguna, data *monitoring* dan lain sebagainya. Adapun metode yang digunakan untuk format pertukaran data adalah *Javascript Object Notation* (JSON).

2.14 Representational State Transfer (REST)

REST (*REpresentational State Transfer*) merupakan standar arsitektur komunikasi berbasis web yang sering diterapkan dalam pengembangan layanan berbasis web. Umumnya menggunakan HTTP (*Hypertext Transfer Protocol*) sebagai protocol untuk komunikasi *data*. REST pertama kali diperkenalkan oleh Roy Fielding pada tahun 2000 [28].

Pada arsitektur REST, REST *server* menyediakan *resources* (sumber daya/data) dan REST *client* mengakses dan menampilkan *resource* tersebut untuk penggunaan selanjutnya. Setiap *resource* diidentifikasi oleh URIs (*Universal Resource Identifiers*) atau global ID. *Resource* tersebut direpresentasikan dalam bentuk format teks, JSON atau XML. Pada umumnya formatnya menggunakan JSON dan XML.

Keuntungan REST antara lain :

1. Bahasa dan *platform agnostic*.
2. Lebih sederhana/simpel untuk dikembangkan daripada SOAP.
3. Mudah dipelajari, tidak bergantung pada *tools*.
4. Ringkas, tidak membutuhkan *layer* pertukaran pesan (*messaging*) tambahan.
5. Secara desain dan filosofi lebih dekat dengan web.

Kelemahan REST antara lain :

1. Mengasumsi model *point-to-point* komunikasi tidak dapat digunakan untuk lingkungan komputasi terdistribusi di mana pesan akan melalui satu atau lebih perantara.
2. Kurangnya dukungan standar untuk keamanan, kebijakan, keandalan pesan, dll, sehingga layanan yang mempunyai persyaratan lebih canggih lebih sulit untuk dikembangkan ("dipecahkan sendiri").
3. Berkaitan dengan model *transport* HTTP(*Hypertext Transfer Protocol*).

Berikut metode HTTP(*Hypertext Transfer Protocol*) yang umum digunakan dalam arsitektur berbasis REST :

1. *GET*, menyediakan hanya akses baca pada *resource*.
2. *PUT*, digunakan untuk menciptakan *resource* baru.
3. *DELETE*, digunakan untuk menghapus *resource*.
4. *POST*, digunakan untuk memperbarui *resource* yang ada atau membuat *resource* baru.
5. *OPTIONS*, digunakan untuk mendapatkan operasi yang disupport pada *resource*.

Web service yang berbasis arsitektur REST kemudian dikenal sebagai *RESTful web services*. Layanan web ini menggunakan metode HTTP(*Hypertext Transfer Protocol*) untuk menerapkan konsep arsitektur REST.

Cara Kerja RESTful web services yaitu :

Sebuah *client* mengirimkan sebuah *data* atau *request* melalui *HTTP Request* dan kemudian *server* merespon melalui *HTTP Response*. Komponen dari *HTTP request* :

1. *Verb*, *HTTP method* yang digunakan misalnya *GET*, *POST*, *DELETE*, *PUT* dll.
2. *Uniform Resource Identifier* (URI) untuk mengidentifikasikan lokasi *resource* pada *server*.
3. *HTTP Version*, menunjukkan versi dari *HTTP* yang digunakan, contoh *HTTP v1.1*.
4. *Request Header*, berisi metadata untuk *HTTP Request*. Contoh, *type client/browser*, format yang didukung oleh *client*, format dari *body* pesan, seting *cache* dll.
5. *Request Body*, konten dari *data*.

Sedangkan komponen dari *http response* yaitu :

1. *Status/Response Code*, mengindikasikan status *server* terhadap *resource* yang *direquest*. misal : 404, artinya *resource* tidak ditemukan dan 200 *response OK*.
2. *HTTP Version*, menunjukkan versi dari *HTTP* yang digunakan, contoh *HTTP v1.1*.
3. *Response Header*, berisi *metadata* untuk *HTTP Response*. Contoh, *type server*, panjang *content*, tipe *content*, waktu *response*, dll
4. *Response Body*, konten dari data yang diberikan.

2.15 JavaScript Object Notation (JSON)

JavaScript Object Notation (JSON) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemrograman JavaScript, Standar ECMA-262 Edisi ke-3 - Desember 1999. JSON

merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran-data [29].

JSON terbuat dari dua struktur:

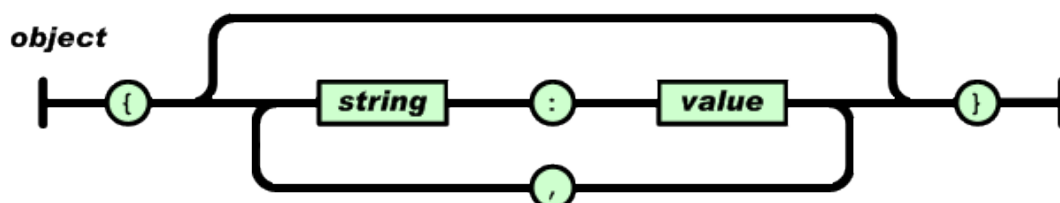
1. Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (*object*), rekaman (*record*), struktur (*struct*), kamus (*dictionary*), tabel hash (*hash table*), daftar berkunci (*keyed list*), atau *associative array*.
2. Daftar nilai terurutkan (*an ordered list of values*). Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik (*array*), vektor (*vector*), daftar (*list*), atau urutan (*sequence*).

Struktur-struktur data ini disebut sebagai struktur data *universal*. Pada dasarnya, semua bahasa pemrograman moderen mendukung struktur data ini dalam bentuk yang sama maupun berlainan. Hal ini pantas disebut demikian karena format data mudah dipertukarkan dengan bahasa-bahasa pemrograman yang juga berdasarkan pada struktur data ini.

JSON menggunakan bentuk sebagai berikut:

1. Objek

Objek adalah sepasang nama/nilai yang tidak terurutkan. Objek dimulai dengan { (kurung kurawal buka) dan diakhiri dengan } (kurung kurawal tutup). Setiap nama diikuti dengan : (titik dua) dan setiap pasangan nama/nilai dipisahkan oleh , (koma).

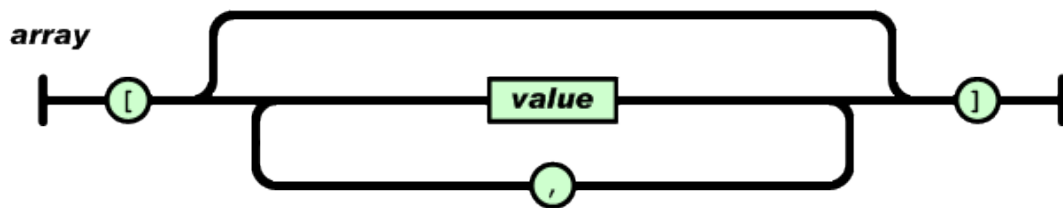


Sumber Gambar : <http://www.json.org/json-id.html>

Gambar 6 JSON Object

2. Larik

Larik adalah kumpulan nilai yang terurutkan. Larik dimulai dengan [(kurung kotak buka) dan diakhiri dengan] (kurung kotak tutup). Setiap nilai dipisahkan oleh , (koma).

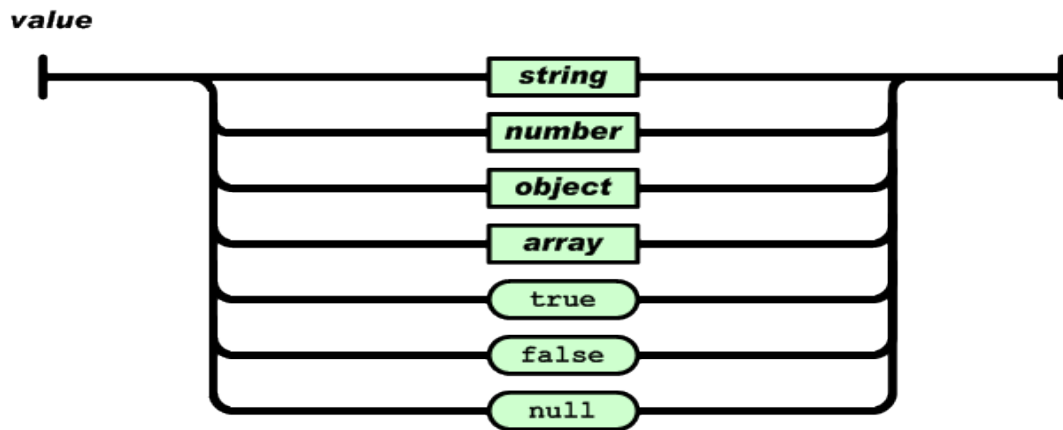


Sumber Gambar : <http://www.json.org/json-id.html>

Gambar 7 JSON Array

3. Nilai

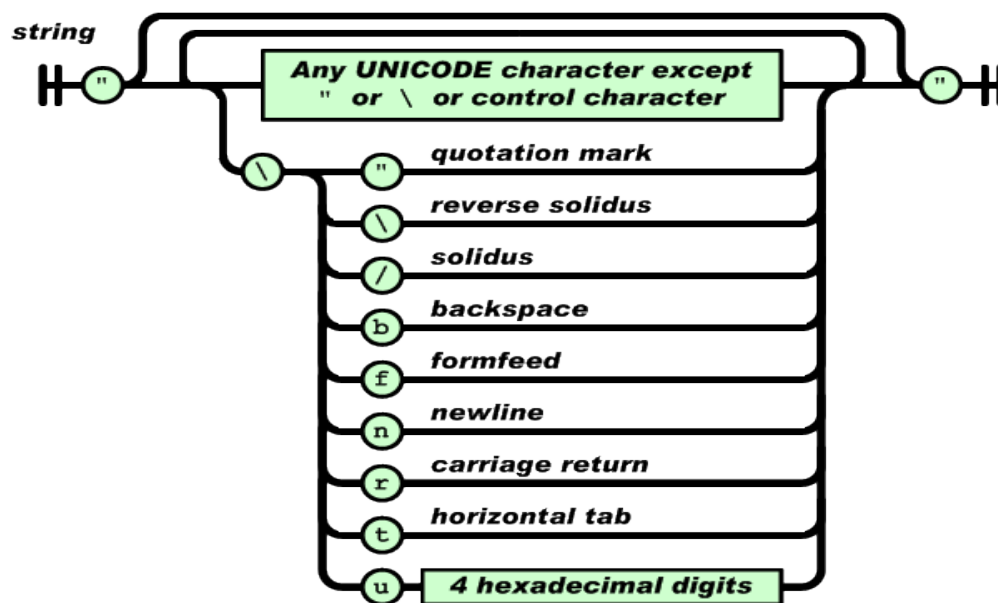
Nilai (*value*) dapat berupa sebuah *string* dalam tanda kutip ganda, atau angka, atau *true* atau *false* atau *null*, atau sebuah objek atau sebuah larik. Struktur-struktur tersebut dapat disusun bertingkat.



Sumber Gambar : <http://www.json.org/json-id.html>

Gambar 8 Value JSON4. *String*

String adalah kumpulan dari nol atau lebih karakter *Unicode*, yang dibungkus dengan tanda kutip ganda. Di dalam string dapat digunakan *backslash escapes* "\" untuk membentuk karakter khusus. Sebuah karakter mewakili karakter tunggal pada *string*. *String* sangat mirip dengan *string* C atau Java.

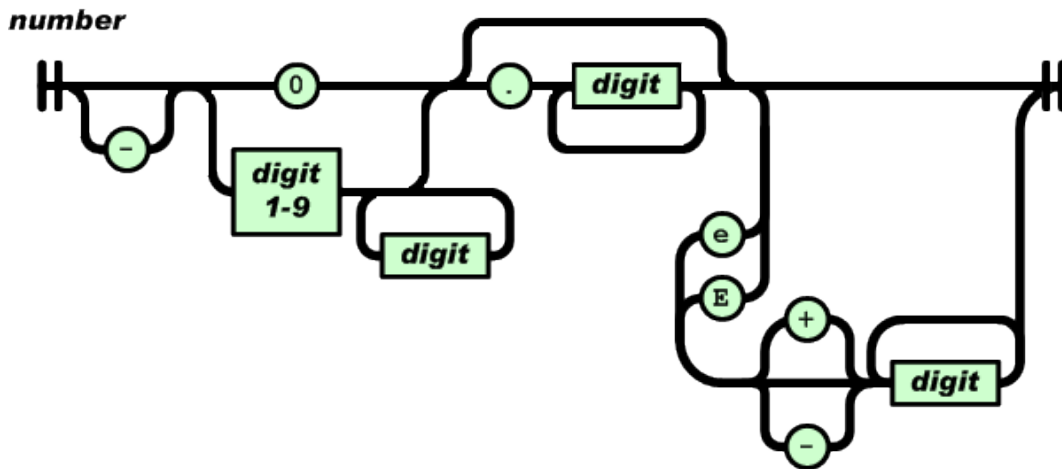


Sumber Gambar : <http://www.json.org/json-id.html>

Gambar 9 String JSON

5. Angka

Angka adalah sangat mirip dengan angka di C atau Java, kecuali format oktal dan heksadesimal tidak digunakan.



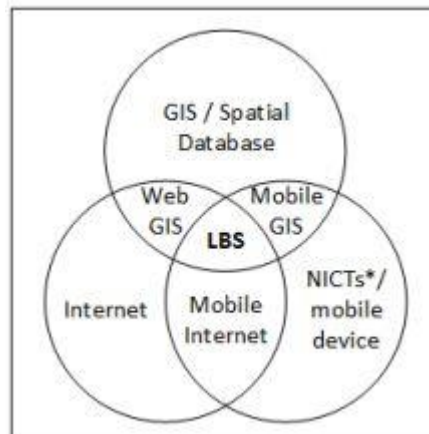
Sumber Gambar : <http://www.json.org/json-id.html>

Gambar 10 JSON Angka

Spasi kosong (*whitespace*) dapat disisipkan di antara pasangan tanda-tanda tersebut, kecuali beberapa detail *encoding* yang secara lengkap dipaparkan oleh bahasa pemrograman yang bersangkutan.

2.16 Location Base Service (LBS)

Layanan Berbasis lokasi adalah layanan informasi yang dapat diakses melalui *mobile device* dengan menggunakan *mobile network*, yang dilengkapi kemampuan untuk memanfaatkan lokasi dari *mobile device* tersebut. LBS memberikan kemungkinan komunikasi dan interaksi dua arah. Oleh karena itu pengguna memberitahu penyedia layanan untuk mendapatkan informasi yang dia butuhkan, dengan referensi posisi pengguna tersebut. Layanan berbasis lokasi dapat digambarkan sebagai suatu layanan yang berada pada pertemuan tiga teknologi yaitu : *Geographic Information System*, *Internet Service*, dan *Mobile Devices*, hal ini dapat dilihat pada gambar LBS adalah pertemuan dari tiga teknologi [30].

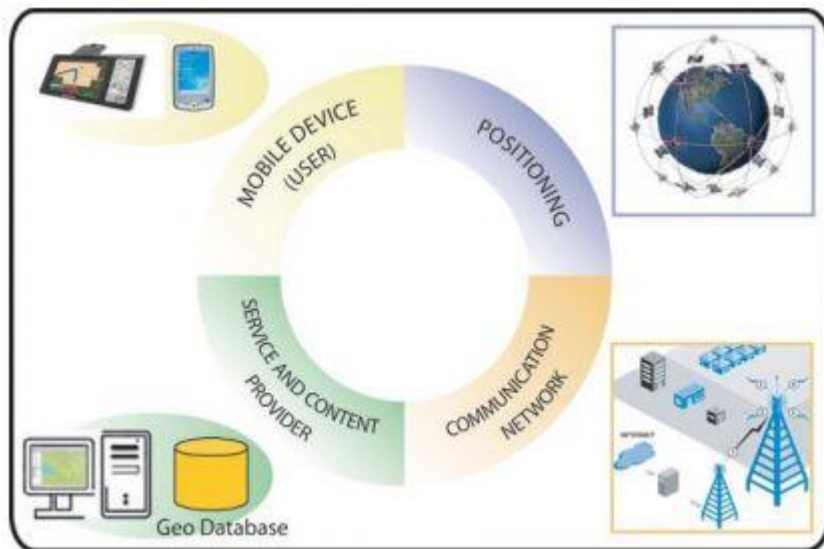


Sumber gambar: Jurnal Aplikasi Location Based Service untuk Sistem Informasi Publikasi Acara pada Platform Android [30]

Gambar 11 LBS Sebagai simpang tiga teknologi

Secara garis besar jenis Layanan Berbasis Lokasi juga dapat dibagi menjadi dua, yaitu:

1. *Pull Service*: Layanan diberikan berdasarkan permintaan dari pelanggan akan kebutuhan suatu informasi. Jenis layanan ini dapat dianalogikan seperti mengakses suatu web pada jaringan internet.
2. *Push Service*: Layanan ini diberikan langsung oleh *service provider* tanpa menunggu permintaan dari pelanggan, tentu saja informasi yang diberikan tetap berkaitan dengan kebutuhan pelanggan.
3. *Positioning Component*: Untuk memproses suatu layanan maka posisi pengguna harus diketahui
4. *Service and Application Provider*: Penyedia layanan menawarkan berbagai macam layanan kepada pengguna dan bertanggung jawab untuk memproses informasi yang diminta oleh pengguna.
5. *Data and Content Provider*: Penyedia layanan tidak selalu menyimpan semua data yang dibutuhkan yang bisa diakses oleh pengguna. Untuk itu, data dapat diminta dari *data and content provider*.



Sumber gambar: Jurnal Aplikasi Location Based Service untuk Sistem Informasi
Publikasi Acara pada Platform Android [30]

Gambar 12 Komponen dasar LBS

Selanjutnya mengirim informasi yang telah diolah melalui jaringan internet dan jaringan komunikasi. Pada akhirnya pengguna dapat menerima informasi yang diinginkan.

2.17 Mysql

MySQL merupakan *software RDBMS (Relational Database Management System)* atau server *database* yang dapat mengelola *database* dengan sangat cepat, dapat menampung data dalam jumlah sangat besar, dapat diakses oleh banyak user (*multi-user*), dan dapat melakukan suatu proses secara sinkron atau berbarengan (*multi-threaded*) [31].

Saat ini, MySQL banyak digunakan di berbagai kalangan untuk melakukan penyimpanan dan pengolahan data, mulai dari kalangan akademis sampai ke industry, baik industri kecil, menengah, maupun besar.

Lisensi MySQL terbagi menjadi dua. Anda dapat menggunakan MySQL sebagai produk *open source* di bawah GNU (*General Public License*) secara gratis atau dapat membeli lisensi dari versi komersialnya. MySQL versi komersial tentu memiliki nilai lebih atau kemampuan-kemampuan yang tidak disematkan pada

versi gratis. Pada kenyataannya, untuk keperluan industri menengah kebawah, versi gratis masih dapat digunakan dengan baik.

2.18 Structured Query Language (SQL)

SQL adalah pendekatan dari *Structural Query Language*, yang merupakan Bahasa atau kumpulan perintah standar yang digunakan untuk berkomunikasi dengan *database*. Perintah dalam SQL diklasifikasikan menjadi lima bagian besar [31], yaitu :

1. Data Definition Language (DDL)

Data Definition Language (DDL) adalah sekumpulan perintah SQL yang berkaitan dengan pembuatan, perubahan dan penghapusan *database* prosedur/fungsi, *trigger*, dan sebagainya.

Perintah SQL yang termasuk kategori DDL adalah :

1. CREATE DATABASE, Berfungsi untuk membuat database dan objek-objek di dalam *database* berikut sintaks yang digunakan.

```
CREATE DATABASE <nama database>;
```

2. CREATE TABLE, Berfungsi untuk membuat tabel di dalam database, berikut sintaks yang digunakan.

```
CREATE TABLE <nama tabel>(  
  <field 1> <type data>,  
  <field 2> <type data>,  
  ...  
)[TYPE = <type tabel>;
```

3. CREATE INDEX , digunakan untuk membuat index berikut sintaks yang digunakan.

```
CREATE INDEX <nama index> ON <nama tabel>(<nama kolom>)
```

4. CREATE VIEW, digunakan untuk membuat view. Berikut sintaks yang digunakan.

```
CREATE VIEW <nama view> AS
```

```
SELECT <column1>, <column2>,
...
FROM <nama table>
WHERE <kondisi>;
```

5. DROP DATABASE, digunakan untuk menghapus basis data, berikut sintaks yang digunakan.

```
DROP DATABASE [IF EXIST] <nama database>;
```

6. DROP TABLE, digunakan untuk menghapus tabel. Berikut sintaks yang digunakan.

```
DROP TABLE [IF EXIST] <nama table>;
```

7. DROP INDEX, digunakan untuk menghapus index. Berikut sintaks yang digunakan.

```
DROP INDEX <nama index> ON <nama tabel>;
```

8. DROP VIEW, digunakan untuk menghapus view. Berikut sintaks yang digunakan.

```
DROP VIEW <nama view>;
```

9. ALTER TABLE, digunakan untuk mengubah struktur suatu tabel. Berikut beberapa sintaks yang digunakan.

Merubah nama tabel

```
ALTER TABLE <nama tabel lama> RENAME TO <nama tabel baru>;
```

Menambah kolom

```
ALTER TABLE ADD COLUMN <nama kolom> <type data>;
```

Mengubah kolom

```
ALTER TABLE <nama tabel> CHANGE <nama kolom lama> <nama kolom baru> <type data baru>;
```

Menghapus kolom

```
ALTER TABLE <nama tabel> DROP COLUMN <nama kolom yang akan dihapus>;
```

Mendefinisikan *primary key*

```
ALTER TABLE <nama tabel> ADD PRIMARY KEY (<nama kolom>;
```

Mendefinisikan *foreign key*

```
ALTER TABLE <nama tabel> ADD FOREIGN KEY (<nama kolom>) REFERENCES <nama tabel referensi> (<nama kolom referensi>;
```

Menghapus *primary key*

```
ALTER TABLE <nama tabel> DROP PRIMARY KEY;
```

Menghapus *Foreign key*

```
ALTER TABLE <nama tabel> DROP FOREIGN KEY <nama foreign key>;
```

Menghapus *index*

```
ALTER TABLE <nama tabel> DROP INDEX <nama index>;
```

2. Data Manipulation Language (DML)

Data Manipulation Language (DML) adalah kumpulan perintah SQL yang berkaitan dengan data atau isi dari suatu tabel. Dengan perintah-perintah di dalam DML, kita dapat memanipulasi (menambah, mengubah, dan menghapus) data yang terdapat pada suatu tabel secara mudah.

Perintah-perintah yang termasuk ke dalam DML adalah :

1. INSERT, berfungsi untuk menambah atau memasukan data baru ke dalam tabel. Berikut sintaks *insert*:

```
INSERT INTO [(<nama kolom>, <nama kolom>,...)]  
VALUES (<nilai 1>, <nilai 2>,...);
```

2. UPDATE, berfungsi untuk mengubah data dalam tabel dengan nilai baru. Berikut sintaks *update* :

```
UPDATE <nama tabel>
SET <nama kolom 1> = <nilai 1>,
    <nama kolom 2> = <nilai 2>,
    ...
WHERE <kriteria atau kondisi>;
```

3. DELETE, berfungsi untuk menghapus data dari suatu tabel. Berikut sintaks *delete* :

```
DELETE <nama tabel> WHERE <kriteria atau kondisi>;
```

3. Data Control Language (DCL)

Data Control Language (DCL) adalah salah satu komponen SQL yang berfungsi untuk mengontrol hak akses *user*. Perintah yang termasuk ke dalam DCL adalah GRANT dan REVOKE. Berikut penjelasannya.

1. GRANT, Perintah GRANT digunakan untuk memberikan hak akses (*privilege*) kepada *user* tertentu. Melalui perintah hak akses semacam ini, seseorang *user* memiliki keterbatasan dalam menggunakan *database* sehingga data akan aman dari pihak-pihak yang tidak berkepentingan. Berikut contoh penggunaan perintah GRANT dalam MySQL:

```
GRANT ALL
ON BUKUDB.*
TO rudianto@localhost
IDENTIFIED BY 's3c23t';
```

Untuk melihat daftar hak akses yang dimiliki oleh seorang *user* , gunakan perintah SHOW GRANTS.

```
SHOW GRANTS FOR 'rudianto'@'localhost';
```

Untuk mengeksekusi perintah GRANT, anda perlu memiliki hak akses GRANT OPTION yang sebelumnya diberikan oleh administrator kepada anda.

2. REVOKE, Perintah REVOKE merupakan kebalikan dari perintah GRANT, yang berfungsi untuk mencabut salah satu atau beberapa hak akses dari *user* tertentu di dalam *database*. Sama seperti perintah GRANT, untuk

menjalankan perintah REVOKE anda perlu memiliki hak akses GRANT OPTION.

Berikut contoh penggunaan perintah REVOKE dalam MYSQL:

```
REVOKE ALL ON BUKUDB.* rudianto;
```

4. *Transaction control Language* (TCL)

Selain pengolahan hak akses user, dalam SQL kita juga dapat mengontrol transaksi data yang telah dilakukan. Perintah-perintah yang berkaitan dengan pengontrolan transaksi data digolongkan ke dalam *Transactional Control Language* (TCL). Perintah yang termasuk ke dalam TCL adalah COMMIT dan ROLLBACK. Berikut penjelasannya.

1. COMMIT, perintah COMMIT berfungsi untuk menyimpan perubahan-perubahan yang dilakukan terhadap *database* (melalui perintah INSERT, UPDATE, atau DELETE) secara permanent.
2. ROLLBACK, perintah ROLLBACK merupakan kebalikan dari perintah COMMIT, yang berfungsi untuk membatalkan transaksi atau perubahan-perubahan yang telah dilakukan ke dalam *database* (melalui perintah INSERT, UPDATE, atau DELETE). Dengan melakukan pembatalan transaksi, data di dalam *database* akan kembali ke keadaan awal (keadaan sebelum dilakukan perubahan). Sehingga dengan kata lain, perubahan yang dilakukan tidak akan pernah disimpan ke dalam *database*.

2.19 Personal Home Page (PHP)

PHP Pertama kali ditemukan pada 1995 oleh seorang *Software Developer* bernama Rasmus Lerdorf. Ide awal PHP adalah ketika itu Rasmus ingin mengetahui jumlah pengunjung yang membaca resume onlinenya. Script yang dikembangkan baru dapat melakukan dua pekerjaan, yakni merekam informasi *visitor*, dan menampilkan jumlah pengunjung dari suatu *website*. Dan sampai sekarang kedua tugas tersebut masih tetap populer digunakan oleh dunia web saat ini. Kemudian, dari situ banyak orang di milis mendiskusikan script buatan Rasmus Lerdorf,

hingga akhirnya rasmus mulai membuat sebuah *tool/script*, bernama *Personal Home Page* (PHP) [32].

Kebutuhan PHP sebagai *tool* yang serba guna membuat Lerdorf melanjutkan untuk mengembangkan PHP hingga menjadi suatu bahasa tersendiri yang mungkin dapat mengkonversikan data yang di masukkan melalui *Form HTML* menjadi suatu variabel, yang dapat dimanfaatkan oleh sistem lainnya. Untuk merealisasinya, akhirnya Lerdorf mencoba mengembangkan PHP menggunakan bahasa C ketimbang menggunakan Perl. Tahun 1997, PHP versi 2.0 di rilis, dengan nama *Personal Home Page Form Interpreter* (PHP-FI). PHP Semakin populer, dan semakin diminati oleh programmer web dunia.

Rasmus Lerdorf benar-benar menjadikan PHP sangat populer, dan banyak sekali *Team Developer* yang ikut bergabung dengan Lerdorf untuk mengembangkan PHP hingga menjadi seperti sekarang, Hingga akhirnya dirilis versi ke 3-nya, pada Juni 1998, dan tercatat lebih dari 50.000 programmer menggunakan PHP dalam membuat website dinamis.

Pengembangan demi pengembangan terus berlanjut, ratusan fungsi ditambahkan sebagai fitur dari bahasa PHP, dan di awal tahun 1999, netcraft mencatat, ditemukan 1.000.000 situs di dunia telah menggunakan PHP. Ini membuktikan bahwa PHP merupakan bahasa yang paling populer digunakan oleh dunia *web development*. Hal ini mengagetkan para developernya termasuk Rasmus sendiri, dan tentunya sangat diluar dugaan sang pembuatnya. Kemudian Zeev Suraski dan Andi Gutsman selaku *core developer* (programmer inti) mencoba untuk menulis ulang PHP Parser, dan diintegrasikan dengan menggunakan Zend scripting engine, dan mengubah jalan alur operasi PHP. Dan semua fitur baru tersebut di rilis dalam PHP 4.

13 Juli 2004, evolusi PHP, PHP telah mengalami banyak sekali perbaikan di segala sisi, dan wajar jika netcraft mengumumkan PHP sebagai bahasa web populer di dunia, karena tercatat 19 juta domain telah menggunakan PHP sebagai *server side* scriptingnya. PHP saat ini telah Mendukung XML dan Web Services, Mendukung SQLite. Tercatat lebih dari 19 juta domain telah menggunakan PHP sebagai *server* scriptingnya. Benarbenar PHP sangat mengejutkan.

Yang menjadikan PHP berbeda dengan HTML adalah proses dari PHP itu sendiri. HTML merupakan bahasa statis yang apabila kita ingin merubah konten/isinya maka yang harus dilakukan pertama kali nya adalah, membuka file-nya terlebih dahulu, kemudian menambahkan isi kedalam file tersebut. Beda hal nya dengan PHP. Bagi anda yang pernah menggunakan CMS seperti wordpress atau joomla yang dibangun dengan PHP tentunya, ketika akan menambahkan konten kedalam website, anda tinggal masuk kedalam halaman admin, kemudian pilih *new artikel* untuk membuat halaman/*content* baru. Artinya hal ini, seorang user tidak berhubungan langsung dengan scriptnya. Sehingga seorang pemula sekalipun dapat menggunakan aplikasi seperti itu.

Keunggulan PHP antara lain :

1. Gratis

Apa yang membuat PHP begitu berkembang sangat pesat hingga jutaan domain menggunakan PHP, begitu populernya PHP? Jawabannya adalah karena PHP itu gratis.

2. *Cross platform*

Artinya dapat di gunakan di berbagai sistem operasi, mulai dari linux, windows, mac os dan os yang lain.

3. Mendukung banyak *database*

PHP telah mendukung banyak *database*, ini mengapa banyak *developer web* menggunakan PHP Adabas D Adabas D, dBase dBase, Empress Empress, FilePro (*read-only*) FilePro (*read-only*) Hyperwave, IBM DB2, Informix, Ingres, InterBase, FrontBase mSQL, Direct MS-SQL, MySQL MySQL, ODBC, Oracle (OCI7 and OCI8), Ovrimos, PostgreSQL SQLite, Solid, Sybase, Velocis, Unix dbm.

4. *On The Fly*

PHP sudah mendukung *on the fly*, artinya dengan php anda dapat membuat *document text*, Word, Excel, PDF, menciptakan *image* dan *flash*, juga menciptakan file-file seperti zip, XML, dan banyak lagi.

2.20 CodeIgniter (CI)

CodeIgniter adalah sebuah Application Development Framework (*toolkit*) bagi orang-orang yang ingin membangun website menggunakan PHP. Tujuannya adalah untuk memungkinkan Anda mengembangkan proyek-proyek lebih cepat daripada Anda menulis kode dari awal, tersedia banyak *library* untuk tugas-tugas yang biasa diperlukan, serta antarmuka dan struktur logis yang sederhana untuk mengakses *library* ini. CodeIgniter memungkinkan Anda fokus pada proyek Anda dengan meminimalkan jumlah kode yang dibutuhkan untuk tugas yang diberikan [33].

1. Sejarah singkat Codeigniter

CodeIgniter lahir pada tahun 2006. CodeIgniter adalah kerangka kerja PHP open source yang hebat dengan *foot print* yang sangat kecil, dibuat oleh Rick Ellis pada tahun 2006. CodeIgniter lahir dari Expression Engine, yang merupakan koleksi kelas *refactored* yang aslinya ditulis untuk CMS unggulan EllisLab. Diambil dari fungsi khusus aplikasi, CodeIgniter dibuat sebagai *toolkit* yang sederhana dan elegan, yang memungkinkan perkembangan pesat dari situs web dan aplikasi web, menarik ribuan pengembang PHP berbakat [34].

Pada tahun 2008 pemimpin industri Dalam lingkungan yang sekarang dan merasa jenuh dengan kerangka kerja PHP lain yang telah ada, CodeIgniter adalah satu-satunya pilihan padat bagi pengembang yang berkecimpung hidup di dunia akun *shared hosting* setiap harinya dan klien dengan *deadline*. CodeIgniter terbang dalam menghadapi kerangka kerja yang sangat besar dan tidak terdokumentasi dengan benar, yang menyebabkan pencipta PHP Rasmus Lerdorf, seorang kritikus kerangka kerja vokal, untuk memuji CodeIgniter.



Sumber Gambar: <https://ellislab.com/codeigniter>

Gambar 13 Logo CodeIgniter

Pada tahun 2009, EllisLab meluncurkan ExpressionEngine 2.0, dibangun kembali pada framework CodeIgniter. Basis kode fleksibel ini memungkinkan pengembang pihak ketiga untuk lebih mudah membangun alat dan *add-on* yang hebat untuk ExpressionEngine, yang menghasilkan ledakan bakat berkualitas tinggi di komunitas ExpressionEngine.

2. Keunggulan Codeigniter

Fitur dalam dan dari diri mereka sendiri adalah cara yang sangat buruk untuk menilai sebuah aplikasi karena mereka belum memberitahu Anda apa-apa tentang pengalaman pengguna, atau bagaimana itu dirancang secara cerdas atau intuitif. Fitur tidak mengungkapkan apa-apa tentang kualitas kode, atau kinerja, atau perhatian terhadap detail, atau penerapan keamanan. Satu-satunya cara untuk benar-benar menilai sebuah aplikasi adalah dengan mencobanya dan mengenal kodenya. Instalasi CodeIgniter sangat mudah, oleh karena itu kami mendorong Anda untuk melakukan hal itu. Sementara itu, ini adalah daftar fitur utama CodeIgniter [33].

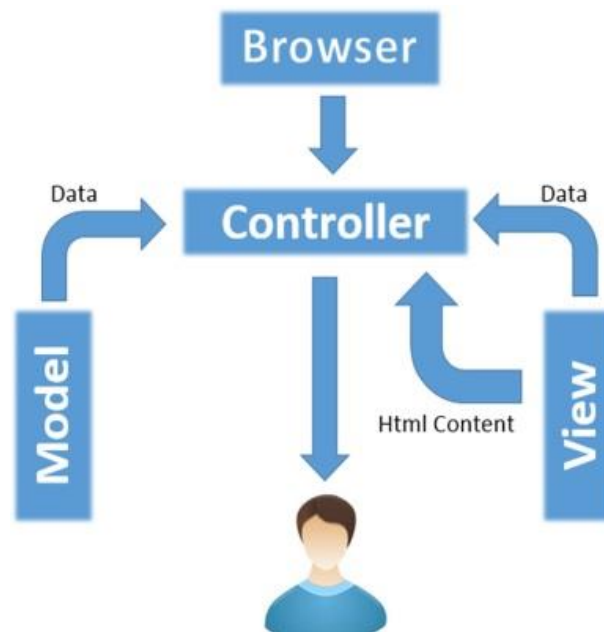
1. Sistem berbasis MVC
2. Sangat ringan
3. Database Class yang lengkap dengan dukungan untuk beberapa *platform*.
4. Dukungan query *builder* untuk *database*
5. *Form* dan validasi data

6. Keamanan dan penyaringan XSS
7. Manajemen *sessions*
8. *Email Sending Class*. Mendukung lampiran, HTML/Text email, beberapa protokol (*sendmail*, SMTP, dan *Mail*) dan banyak lagi.
9. *Image Manipulation Library* (cropping, mengubah ukuran, memutar, dll). Mendukung GD, *ImageMagick*, dan Netpbm
10. *File Uploading Class*
11. *FTP Class*
12. *Localization*
13. *Pagination*
14. Enkripsi data
15. *Benchmarking*
16. *Full Page Caching*
17. *Error Logging*
18. *Application Profiling*
19. *Calendaring Class*
20. *User Agent Class*
21. *Zip Encoding Class*
22. *Template Engine Class*
23. *Trackback Class*
24. *XML-RPC Library*
25. *Unit Testing Class*
26. *Search-engine Friendly* URLs
27. *Flexible* URI Routing
28. Dukungan untuk *Hooks* dan *Class Extensions*
29. Kumpulan *library* fungsi-fungsi “*helper*” yang besar

3. *Model-View-Controller*

CodeIgniter didasarkan pada pola pengembangan *Model-View-Controller*. MVC adalah pendekatan perangkat lunak yang memisahkan logika aplikasi dari

presentasi. Dalam prakteknya, itu memungkinkan halaman web Anda memiliki *scripting* yang minimal karena presentasi terpisah dari *scripting* PHP.



Sumber Gambar : http://www.w3ii.com/id/codeigniter/codeigniter_mvc_framework.html

Gambar 14 Konsep Model View Controller(MVC)

1. *Model*, Model mewakili struktur data Anda. Biasanya *class* model Anda akan berisi fungsi yang membantu Anda mengambil, menyimpan, dan memperbarui informasi dalam *database* Anda.
2. *View* adalah informasi yang disajikan kepada pengguna. *View* yang biasanya akan menjadi halaman web, tetapi dalam CodeIgniter, view juga bisa menjadi bagian dari sebuah halaman seperti *header* atau *footer*. Hal ini juga dapat menjadi halaman RSS, atau jenis-jenis lain dari “halaman”.
3. *Controller* berfungsi sebagai perantara antara *Model*, *View*, dan *resource* lain yang diperlukan untuk memproses *HTTP request* dan menghasilkan halaman web.

CodeIgniter memiliki pendekatan yang cukup longgar untuk MVC karena Model tidak selalu diperlukan. Jika Anda tidak perlu menambahkan pemisahan, atau menemukan bahwa mempertahankan sebuah model memerlukan kompleksitas lebih dari yang Anda inginkan, Anda bisa mengabaikan mereka dan membangun aplikasi Anda dengan minimal menggunakan *Controller* dan *View*. CodeIgniter juga memungkinkan Anda untuk memasukkan script Anda sendiri, atau bahkan mengembangkan *library* inti untuk sistem, memungkinkan Anda untuk bekerja dengan cara yang paling masuk akal bagi Anda.

4. Desain and Sasaran Arsitektur

Tujuan kami untuk CodeIgniter adalah kinerja yang maksimum, kemampuan, dan fleksibilitas dalam hal terkecil, *package* seringan mungkin. Untuk memenuhi tujuan ini kami berkomitmen untuk melakukan benchmarking, *refactoring*, dan menyederhanakan setiap langkah dari proses pembangunan, menolak semua yang tidak menjadi tujuan berikutnya. Dari sudut pandang teknis dan arsitektur, CodeIgniter diciptakan dengan tujuan sebagai berikut:

1. *Dynamic Instantiation*. Dalam CodeIgniter, komponen dimuat dan rutinitas dieksekusi hanya jika diminta, bukan secara global. Tidak ada asumsi yang dibuat oleh sistem tentang apa yang mungkin diperlukan di luar resource utama minimal, sehingga sistem ini sangat ringan secara *default*. *Event*, yang dipicu oleh HTTP *request*, dan *controller* dan *view* yang Anda rancang akan menentukan apa yang dipanggil.
2. *Loose Coupling*. *Coupling* adalah sejauh mana komponen-komponen dari sistem mengandalkan satu sama lain. Semakin sedikit komponen yang bergantung satu sama lain, maka komponen tersebut lebih dapat digunakan kembali dan sistem menjadi fleksibel. Tujuan kami adalah sistem yang sangat longgar (*very loosely coupled system*).
3. *Component Singularity*. *Singularity* adalah sejauh mana komponen memiliki tujuan yang difokuskan secara sempit. Dalam CodeIgniter, setiap *class* dan fungsinya sangat otonom supaya memungkinkan kegunaan yang maksimal.

CodeIgniter itu *dynamically instantiated*, sistem yang *loosely coupled* dengan singularitas komponen yang tinggi. Codeigniter berusaha untuk sederhana, fleksibel, dan kinerja tinggi dalam paket dengan *footprint* yang kecil.

Dalam Penelitian ini peneliti menggunakan *framework* CodeIgniter untuk membangun *Web Service* yang akan digunakan aplikasi dalam pertukaran data baik mengirim atau menerima data ke aplikasi dari *database*.

2.21 Unified Modeling language (UML)

UML (*Unifed Modeling Language*) adalah salah satu standar Bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. Ada beberapa diagram yang digunakan proses pembuatan perangkat lunak berorientasi objek diantaranya, *use case diagram*, *activity diagram*, *class diagram* dan *sequence diagram* [35].

2.21.1 Use Case Diagram

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

Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*.

1. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibangun.
2. *Use Case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

2.21.2 Class Diagram

Diagram kelas atau *Class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas

memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variable-variabel yang dimiliki oleh suatu kelas. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

2.21.3 Sequence Diagram

Diagram Sekuen atau *Sequence diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

Banyaknya diagram sekuen yang harus digambar adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak. Penomoran pesan berdasarkan urutan interaksi pesan. Penggambaran letak pesan harus berurutan, pesan yang lebih atas dari lainnya adalah pesan yang berjalan terlebih dahulu.

2.21.4 Activity Diagram

Diagram Aktivitas atau *Activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut:

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
2. Urutan atau pengelompokan tampilan dari sistem/*user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.
4. Rancangan menu yang ditampilkan pada perangkat lunak.

BAB 3

ANALISA DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Analisis sistem merupakan tahapan yang sistematis untuk mendapatkan aplikasi yang baik dan bersesuaian dengan maksud dan tujuan pembuatan aplikasi. Analisis sistem bertujuan untuk mengidentifikasi masalah-masalah serta hambatan-hambatan yang terdapat pada sistem dan kebutuhan-kebutuhan dari sistem yang akan di bangun.

3.1.1 Analisis Masalah

Analisis masalah didapat berdasarkan pada identifikasi masalah yang telah dikaji sebelumnya. Adapun identifikasi masalah pada Pembangunan Aplikasi *Monitoring* dan Pengendalian Konten Negatif pada Perangkat Pintar Berbasis Android Menggunakan API Clarifai ini yaitu di zaman sekarang ini dimana hampir setiap orang tua telah memberikan *smartphone* kepada anak, baik pada tingkat kanak-kanak ataupun remaja. Namun, kemajuan teknologi komunikasi seperti pada *smartphone* harus diimbangi dengan perhatian dari orang tua agar anak dapat terkontrol dan terawasi dari hal-hal negatif yang terkandung pada *smartphone* dan bisa saja diakses oleh anak. Masalah timbul dari orang tua yang merasa kesulitan untuk melakukan pemeriksaan pada penggunaan *smartphone* pada anak.

Berdasarkan dari hasil pengolahan data kuisisioner tentang pola perilaku orang tua dalam menggunakan *smartphone* yang diperoleh dari 20 orang responden dapat diketahui bahwa orang tua kesulitan dalam melakukan pemeriksaan kepada anak dengan alasan yang berbeda-beda seperti tidak terlalu mengerti pada pencarian berkas dan lainnya serta memeriksa *history browser* pada *smartphone*.

Kemudian pada penelitian ini peneliti mengusulkan untuk melengkapi aplikasi yang akan dibangun dengan fitur-fitur yang dapat membantu orangtua dalam rangka memberikan perhatian dan pengawasan terhadap anak. Pencarian konten negatif pada penelitian ini yaitu di fokuskan pada berkas gambar dengan

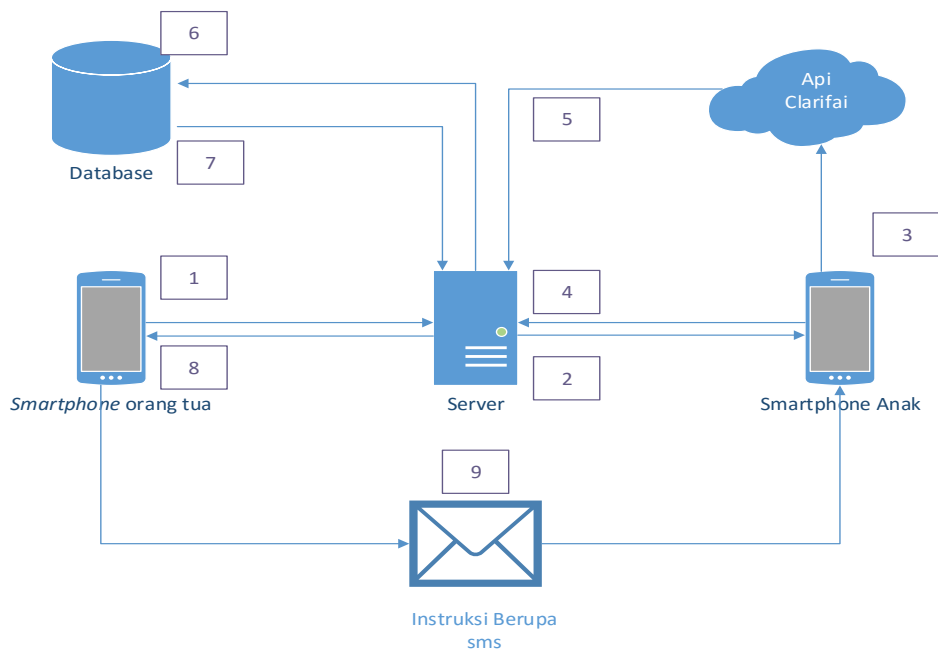
menggunakan API Clarifai yang dapat mendeteksi konten negatif pada gambar yang mungkin terkandung di dalam berkas *smartphone* anak.

3.1.2 Analisis Arsitektur Sistem

Sebelum melakukan perancangan aplikasi, terlebih dahulu dilakukan analisis pada arsitektur sistem yang akan dibangun. Hal ini dilakukan dengan tujuan agar aplikasi yang akan dibangun nanti sesuai dengan cakupan sistem yang telah ada.

3.1.2.1 Model Sistem

Model sistem memberikan gambaran umum mengenai proses yang terjadi saat pengguna melakukan monitoring dan pemeriksaan konten negatif berupa gambar pada *smartphone* anak menggunakan API Clarifai. Adapun model sistem dari aplikasi dapat dilihat pada gambar berikut:



Gambar 15 model sistem

Tahapan pada gambar model sistem aplikasi yang dibangun pada gambar di atas adalah sebagai berikut:

1. Proses pemberian instruksi dari *smartphone* orangtua untuk melakukan pencarian konten negatif, instruksi dikirim ke *server*.
2. Proses *server* meneruskan instruksi kepada *smartphone* anak.
3. Proses *smartphone* anak mengirimkan data ke API Clarifai untuk dilakukan identifikasi konten negatif.
4. Proses *smartphone* anak mengirimkan data ke server kemudian server berupa nama *file* dan lokasi dimana gambar berada dan memberikan status belum teridentifikasi dan menyimpannya ke *database*.
5. Proses hasil identifikasi yang dilakukan API Clarifai dikirimkan ke *server*.
6. Proses server menyimpan hasil dari identifikasi Clarifai ke *database*.
7. Proses server mengambil data dari hasil identifikasi Clarifai dari *database*.
8. Proses *smartphone* orang tua menerima laporan hasil pencarian yang dikirim *server* dari identifikasi gambar yang dilakukan API Clarifai.
9. Proses *smartphone* orang tua mengirimkan instruksi perintah berupa SMS kepada *smartphone* anak.

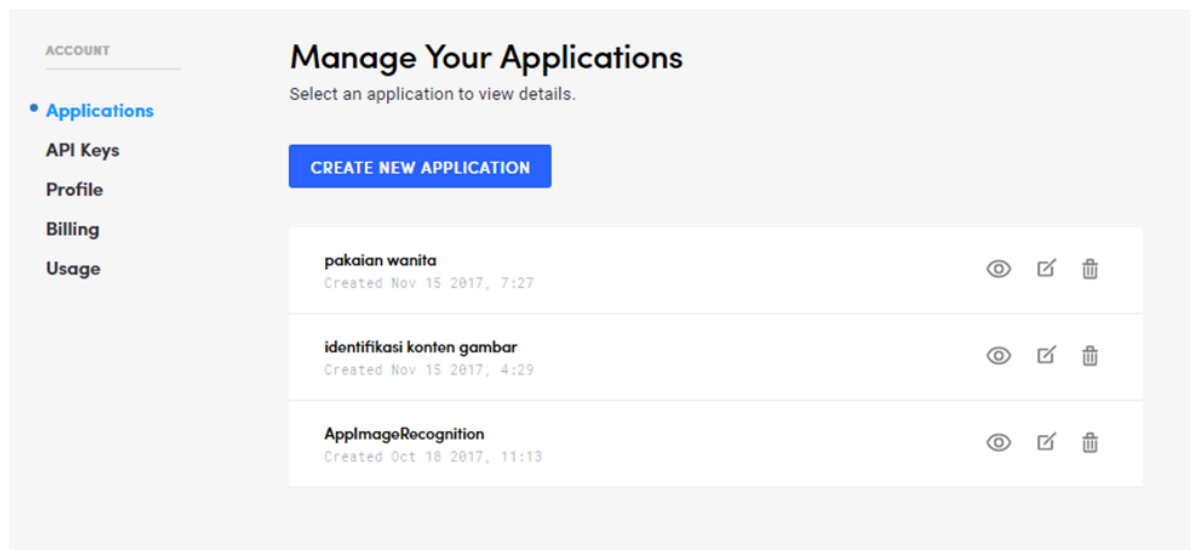
3.1.3 Analisis Teknologi

Analisis teknologi dilakukan untuk mengetahui teknologi apa saja yang digunakan dalam pembuatan aplikasi.

3.1.3.1 Analisis API Clarifai

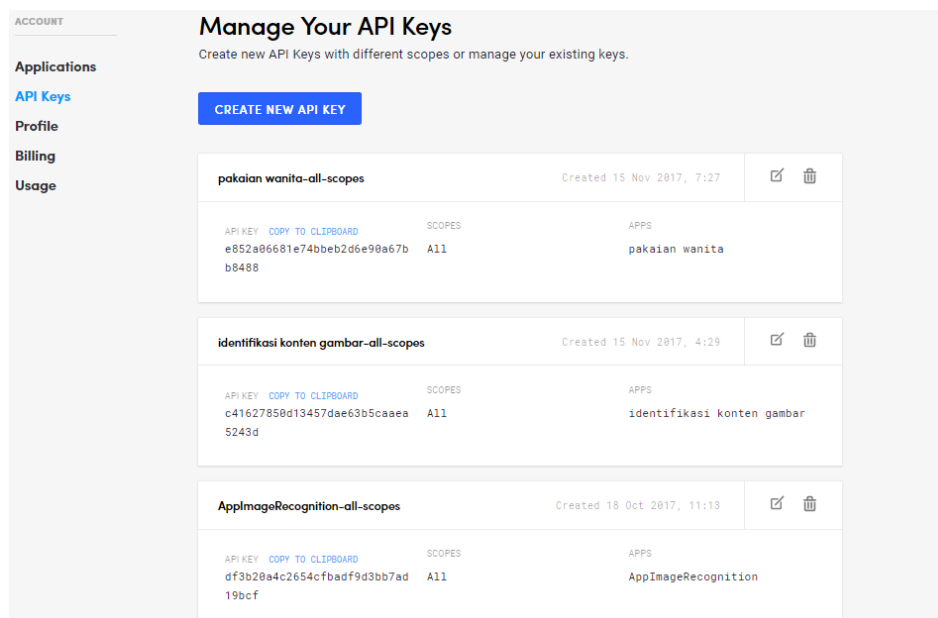
API Clarifai digunakan untuk mengenali sebuah citra, untuk dapat mengenali sebuah citra menggunakan api Clarifai dengan melalui beberapa tahap yaitu pelatihan dan pengujian. Pelatihan merupakan proses yang akan dijadikan sebagai acuan pengenalan dalam mengenali citra, setelah dilakukan pelatihan kemudian dilakukan pengujian dengan menggunakan citra baru yang akan di uji.

1. Membuat project baru di alamat <https://clarifai.com/developer/account/applications>.



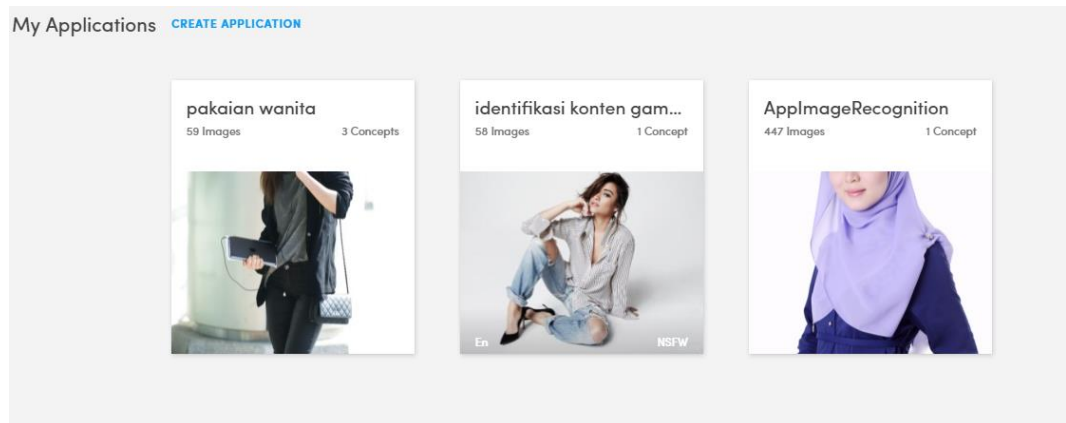
Gambar 16 Membuat Proyek Baru di Situs Clarifai

2. Mendapatkan API key yang telah di buat di alamat <https://clarifai.com/developer/account/keys>, dimana api key digunakan untuk autentikasi library.



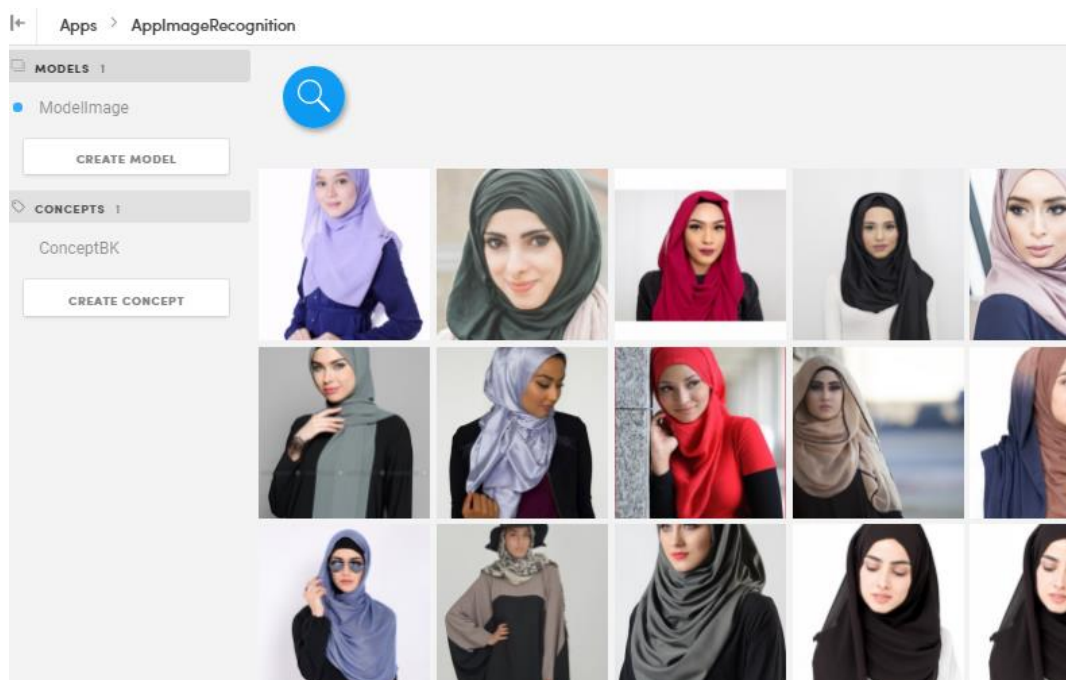
Gambar 17 Mendapatkan Autentikasi Key

3. Setelah membuat *project application*, selanjutnya melihat project yang telah dibuat di menu explorer dengan alamat <https://clarifai.com/explorer/>.



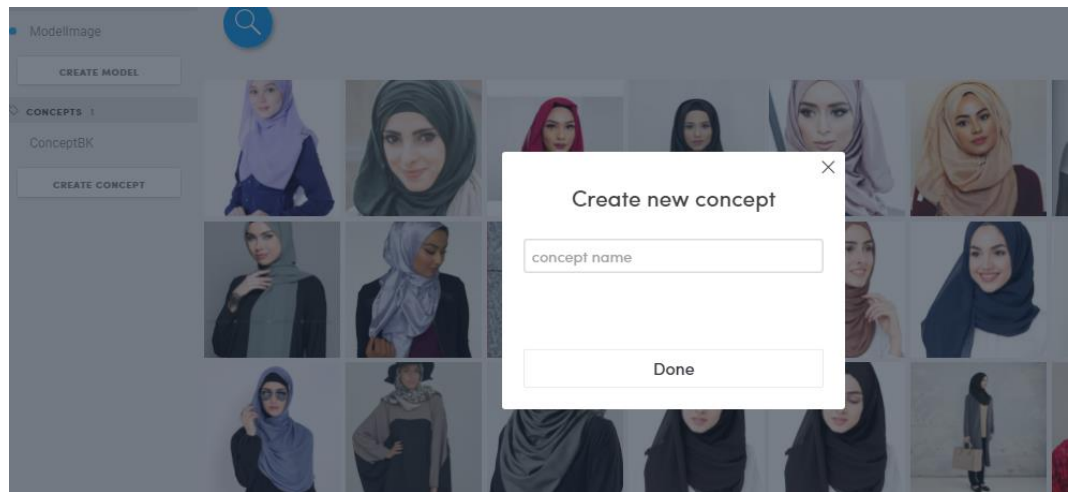
Gambar 18 Gambar menu *Explorer di Clarifai*

4. Pilih project yang sebelumnya telah dibuat. Lalu akan menampilkan halaman berikut ini:



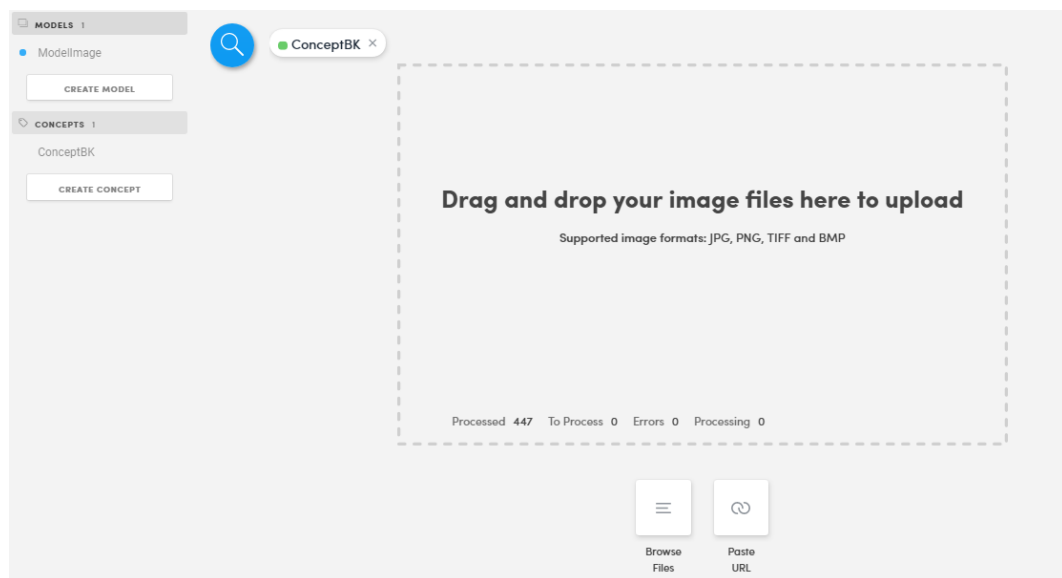
Gambar 19 Tampilan Projek yang telah dibuat di menu *explorer*

5. Pilih Create new concept, lalu akan menampilkan halaman berikut:



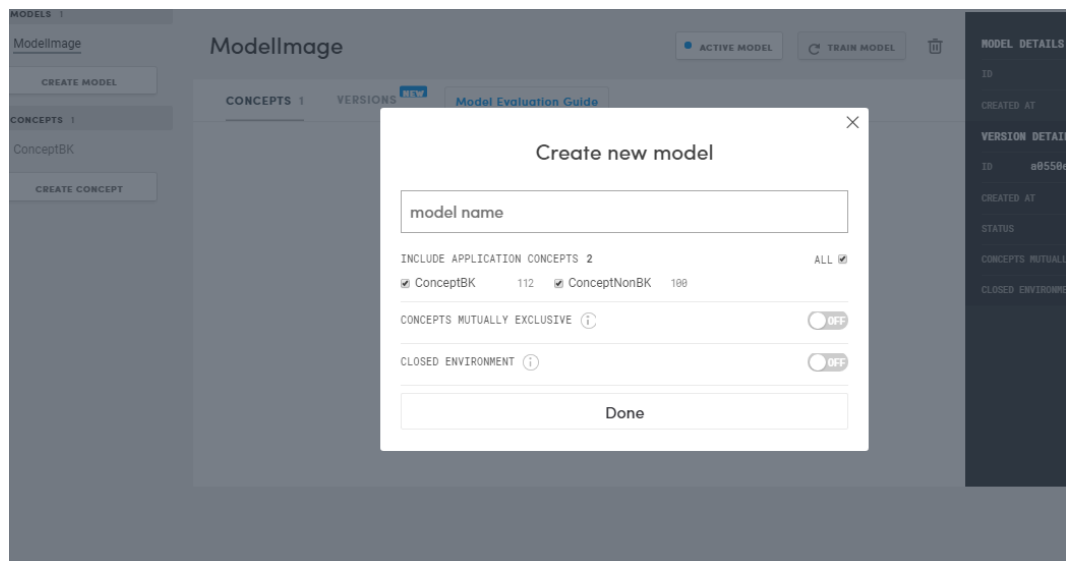
Gambar 20 Membuat Konsep baru di Clarifai

6. Setelah concept dibuat, lalu masukan citra pada concept yang telah dibuat.



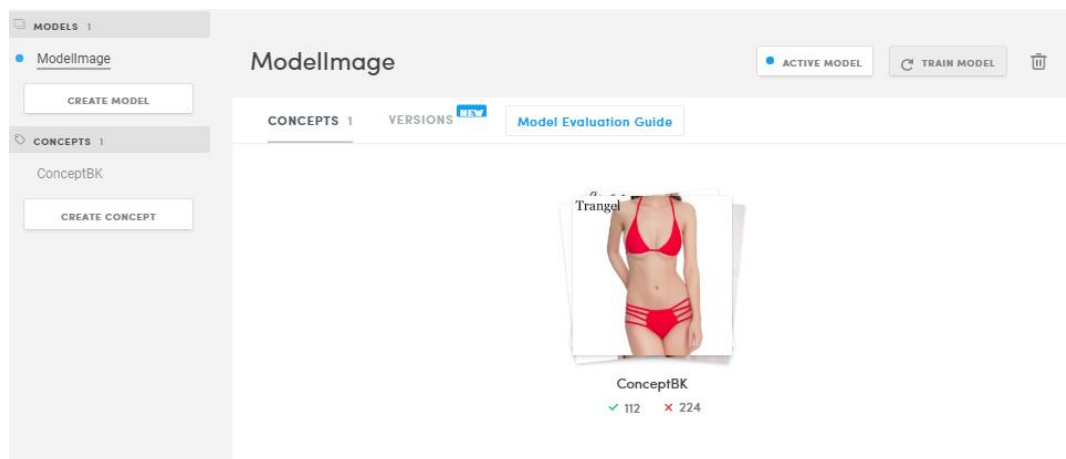
Gambar 21 Menambahkan Citra pada *Concept* yang telah dibuat

7. Pilih create model dengan membuat model baru untuk concept yang telah dibuat.



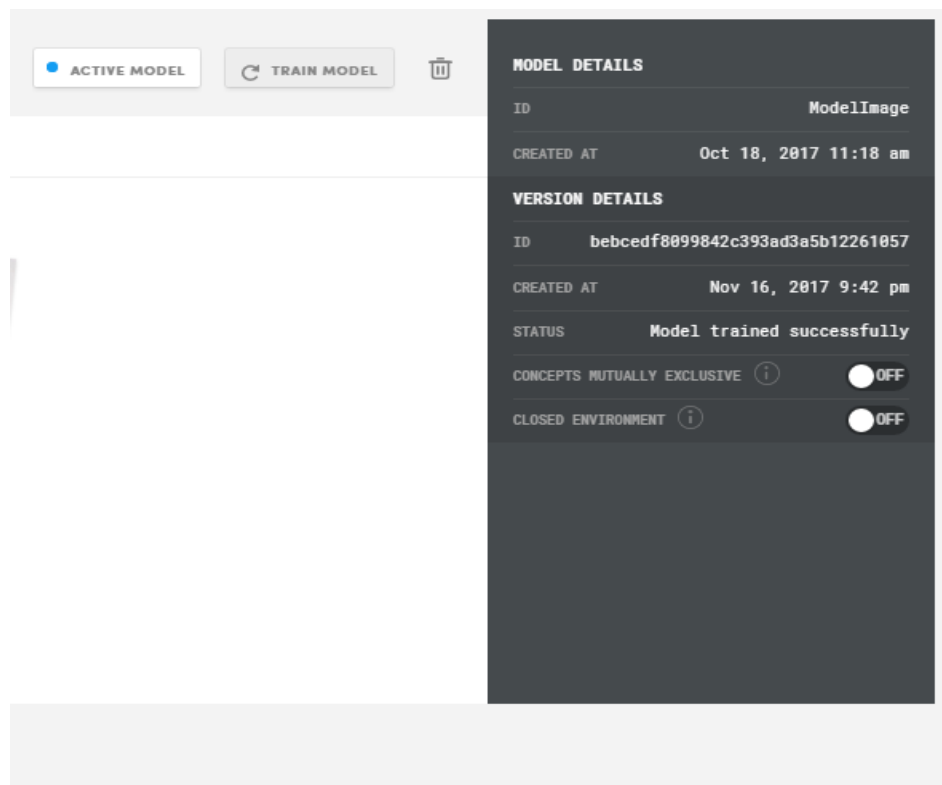
Gambar 22 Membuat Model Baru Calarifai

8. Setelah model telah dibuat lalu train pada model, seperti pada gambar berikut:



Gambar 23 Training Pada Model

9. Setelah model berhasil di lakukan *training* kemudian akan didapatkan model id yang digunakan pada library java.



Gambar 24 Model id Clarifai

10. Melakukan pengujian dengan library java



Gambar 25 Pengujian API Clarifai dengan Library Java

11. Hasil response dari library Api Clarifai dari java

```

{
  "status": {
    "code": 10000,
    "description": "Ok"
  },
  "outputs": [
    {
      "id": "e8b6eb27de764f3fa8d4f7752a3a2dfc",
      "status": {
        "code": 10000,
        "description": "Ok"
      },
      "created_at": "2016-11-22T17:22:23Z",
      "model": {
        "name": "pets",
        "id": "a10f0cf48cf3426cbb8c4805e246c214",
        "created_at": "2016-11-22T17:17:36Z",
        "app_id": "f09abb8a57c041cbb94759ebb0cf1b0d",
        "output_info": {
          "message": "Show output_info with: GET /models/{model_id}/output_info",
          "type": "concept",
          "output_config": {
            "concepts_mutually_exclusive": false,
            "closed_environment": false
          }
        },
        "model_version": {
          "id": "d1b38fd2251148d08675c5542ef00c7b",
          "created_at": "2016-11-22T17:21:13Z",
          "status": {
            "code": 21100,
            "description": "Model trained successfully"
          }
        }
      },
      "input": {
        "id": "e8b6eb27de764f3fa8d4f7752a3a2dfc",
        "data": {
          "image": {
            "url": "https://samples.clarifai.com/puppy.jpeg"
          }
        }
      },
      "data": {
        "concepts": [
          {
            "id": "boscoe",
            "name": "boscoe",
            "app_id": "f09abb8a57c041cbb94759ebb0cf1b0d",
            "value": 0.98308545
          }
        ]
      }
    }
  ]
}

```

Gambar 26 Response dari Library Api Clarifai dari Java

12. Pada object **data** dengan array concepts dengan berisi atribut id, name, app_id dan value. Dimana value merupakan nilai alpha dengan range 0-1. Dimana nilai terbesar mendekati satu merupakan hasil prediksi pengenalan citra yang dianggap mengenali.

3.1.4 Analisis Kebutuhan Spesifikasi Perangkat Lunak

Spesifikasi kebutuhan perangkat lunak terbagi menjadi dua bagian yaitu spesifikasi kebutuhan perangkat lunak kebutuhan fungsional dan spesifikasi

kebutuhan perangkat lunak kebutuhan non fungsional. Spesifikasi kebutuhan perangkat lunak dapat dilihat seperti pada Tabel 3.2 dan Tabel 3.3.

Tabel 3.1 Tabel Spesifikasi Kebutuhan Perangkat Lunak Fungsional

Kode SKPL	Spesifikasi Kebutuhan Perangkat Lunak
SKPL-F-001	Sistem dapat melakukan fungsi daftar akun
SKPL-F-002	Sistem dapat memproses permintaan lupa <i>password</i>
SKPL-F-003	Sistem dapat melakukan login
SKPL-F-004	Sistem dapat melakukan log out
SKPL-F-005	Sistem dapat menampilkan menu profil
SKPL-F-006	Sistem dapat menyunting nama akun
SKPL-F-007	Sistem dapat menyunting password akun
SKPL-F-008	Sistem dapat melakukan monitoring konten negatif
SKPL-F-009	Sistem dapat melakukan penghapusan berkas gambar
SKPL-F-010	Sistem dapat melakukan monitoring <i>lokasi anak</i>
SKPL-F-011	Sistem dapat melakukan monitoring <i>history browser</i>
SKPL-F-012	Sistem dapat melakukan <i>backup data</i> kontak
SKPL-F-013	Sistem dapat mengirimkan notifikasi
SKPL-F-014	Sistem dapat mengirimkan notifikasi monitoring konten negatif
SKPL-F-015	Sistem dapat mengirimkan notifikasi <i>monitoring history browser</i>
SKPL-F-016	Sistem dapat mengirimkan notifikasi <i>monitoring</i> lokasi anak

Tabel 3.2 Tabel Spesifikasi Kebutuhan Perangkat Lunak Non-Fungsional

Kode SKPL	Spesifikasi Kebutuhan Perangkat Lunak
SKPL-NF-001	Sistem berbasis <i>mobile</i> dengan minimum Android Jelly Bean 4.1.x (API level 16)
SKPL-NF-002	Sistem menggunakan bahasa Indonesia

SKPL-NF-003	Sistem menggunakan <i>web server</i> untuk menyimpan konten <i>monitoring</i>
SKPL-NF-004	<i>Smartphone</i> tidak dalam keadaan <i>airplane mode</i> atau mati.

3.1.5 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional merupakan analisis yang dibutuhkan dalam pembangunan aplikasi untuk dapat menentukan spesifikasi dan kebutuhan sistem. Spesifikasi tersebut merupakan elemen-elemen atau komponen-komponen apa saja yang dibutuhkan sistem yang akan dibangun sehingga sistem dapat diimplementasikan.

3.1.5.1 Analisis Kebutuhan Perangkat Keras

Analisis perangkat keras merupakan gambaran dari kebutuhan perangkat keras yang dibutuhkan untuk pembangunan aplikasi dan agar aplikasi *monitoring* dan pengendalian konten negatif pada perangkat pintar berbasis android menggunakan api clarifai dapat dijalankan, spesifikasi perangkat keras yang direkomendasikan dapat dilihat pada tabel 3.4 yaitu sebagai berikut:

Tabel 3.3 Spesifikasi Kebutuhan Perangkat Keras

No.	Perangkat Keras	Spesifikasi
1	CPU	1 GHz
2	<i>Disk Space</i>	45 MB
3	RAM	512 MB
4	Dimensi Layar	4,0 <i>inch</i>

Selain itu spesifikasi perangkat keras yang dibutuhkan untuk pembangunan aplikasi *monitoring* dan pengendalian konten negatif pada perangkat pintar berbasis android menggunakan api clarifai yang direkomendasikan dapat dilihat pada tabel 3.5 yaitu berikut:

Tabel 3.4 Spesifikasi Kebutuhan Perangkat Keras

No.	Perangkat Keras	Spesifikasi
1	CPU	Intel® Core™ i3 M350 dengan kecepatan 2.27GHz (4CPU)
2	RAM	4 GB
3	Hard Disk	500GB
4	VGA	Intel® HD Graphicc 1696 MB

3.1.5.2 Analisis Kebutuhan perangkat Lunak

Analisis kebutuhan perangkat lunak merupakan gambaran tentang spesifikasi perangkat lunak yang dibutuhkan untuk membangun maupun menjalankan aplikasi *monitoring* dan pengendalian konten negatif pada perangkat pintar berbasis Android, spesifikasi perangkat lunak yang dibutuhkan dapat dilihat pada tabel 3.6 sebagai berikut:

Tabel 3.5 Spesifikasi Kebutuhan Perangkat Lunak

No.	Perangkat Lunak	Spesifikasi
1	Sistem Operasi	Android Oreo (API Level 26)
2	Google Play Services	Versi Terbaru

Selain itu, spesifikasi kebutuhan perangkat lunak yang dibutuhkan untuk pembangunan aplikasi monitoring dan pengendalian konten negatif pada perangkat pintar berbasis Android menggunakan API Clarifai dapat dilihat pada tabel 3.7 sebagai berikut:

Tabel 3.6 Tabel Spesifikasi Kebutuhan Perangkat Lunak

No.	Perangkat Lunak	Spesifikasi
1	Sistem Operasi	Windows 7
2	Bahasa Pemrograman	PHP, Java, JSON
3	IDE	Android Studio, Sublime Text
4	<i>Database</i>	MySQL

5	<i>Browser</i>	Google Chrome
---	----------------	---------------

3.1.5.3 Analisis Kebutuhan Pengguna

Analisis pengguna merupakan gambaran mengenai karakteristik pengguna yang dibutuhkan untuk aplikasi yang akan dibangun. Karakteristik pengguna yang dianalisis berdasarkan pada empat kategori, yaitu pengguna, hak akses, tingkat keterampilan, dan bahasa. Karakteristik pengguna yang terlibat dalam aplikasi *monitoring* dan pengendalian konten negatif pada perangkat pintar berbasis android menggunakan API Clarifai dapat dilihat pada tabel 3.8 sebagai berikut:

Tabel 3.7 Karakteristik Pengguna

Pengguna	Hak Akses	Tingkat keterampilan	Bahasa
Pengguna	Dapat Mengakses semua fitur yang terdapat pada aplikasi	Dapat mengoperasikan <i>smartphone</i> Android	Mengerti bahasa Indonesia

3.1.6 Analisis Kebutuhan Fungsional

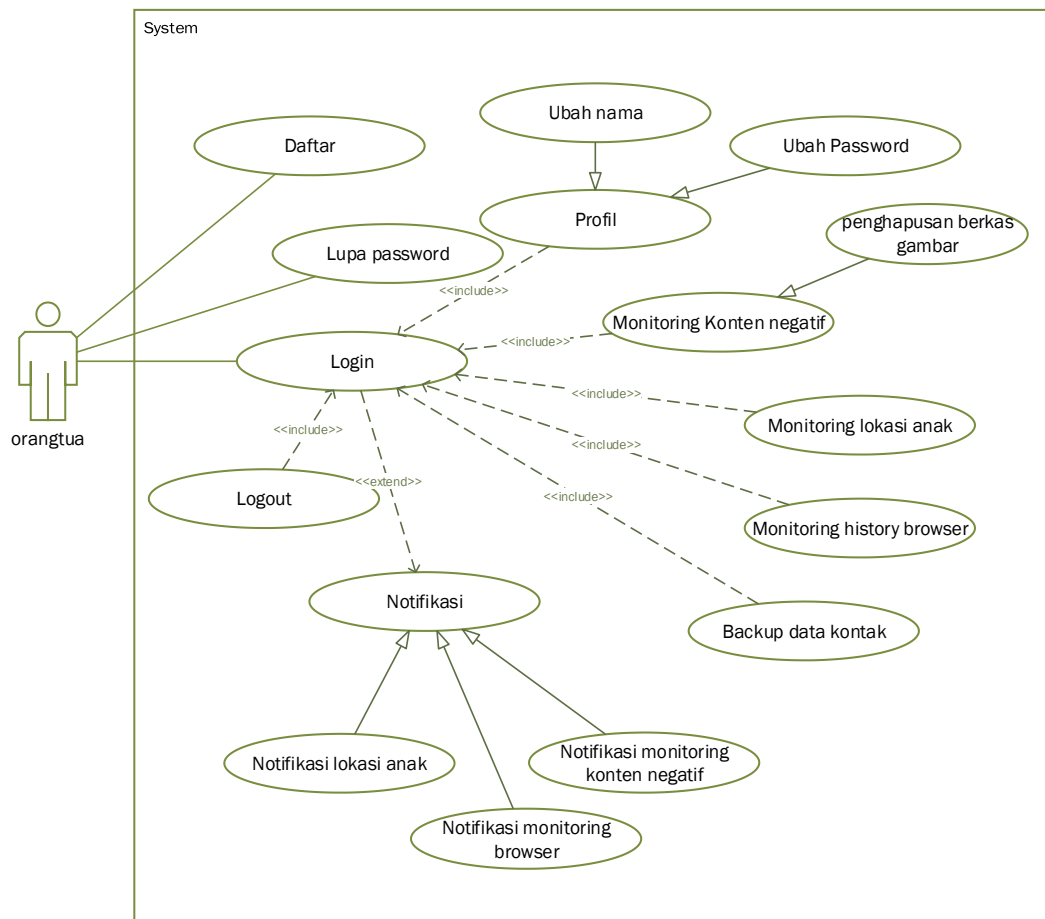
Analisis kebutuhan fungsional menggambarkan proses kegiatan dan tahapan-tahapan yang akan diterapkan dalam sebuah sistem serta menjelaskan kebutuhan yang diperlukan sistem agar sistem yang akan dibangun dapat berjalan dengan baik dan sesuai dengan kebutuhan. Pemodelan yang digunakan sistem dimodelkan dengan menggunakan UML (*Unified Modeling Language*).

Tahapan-tahapan pemodelan dari sistem dalam analisis tersebut antara lain *Use Case Diagram*, *Use Case Scenario*, *Activity Diagram*, *Class Diagram*, dan *Sequence Diagram*.

3.1.6.1 Use Case Diagram

Use case diagram menggambarkan bagaimana aktor atau pengguna akan menggunakan dan memanfaatkan sistem. Diagram *use case* juga mendeskripsikan apa yang akan dilakukan sistem.

Use case diagram terdiri dari tiga bagian yaitu, definisi aktor, identifikasi *use case* dan skenario *use case*. *Use case diagram* dari aplikasi *monitoring* dan pengendalian konten negatif pada perangkat pintar berbasis android menggunakan API Clarifai dapat dilihat pada gambar



Gambar 27 Use Case Diagram

3.1.6.1.1 Definisi Aktor

Dalam sistem yang akan dibangun terdapat satu aktor, yaitu orang tua. Penjelasan mengenai definisi aktor dapat dilihat pada tabel 3.9 sebagai berikut:

Tabel 3.8 Definisi Aktor

Aktor	Definisi
Orang tua	Orangtua adalah aktor yang masuk ke dalam sistem dengan tujuan untuk melakukan <i>monitoring</i> dan menjalankan instruksi yang terdapat pada sistem.

3.1.6.1.2 Definisi Use Case

Terdapat 13 *use case* di dalam aplikasi ini. Dimana setiap *use case* menggambarkan operasi yang berbeda. Definisi dari *use case* yang terdapat pada sistem dapat dilihat pada Tabel 3.10 sebagai berikut:

Tabel 3.9 Definisi Use Case

No.	Use Case	Deskripsi
1	Daftar	Fungsionalitas untuk mendaftarkan pengguna yaitu orangtua kedalam sistem.
2	Lupa Password	Fungsionalitas untuk membantu pengguna atau orang tua mengatur ulang password agar dapat masuk ke dalam sistem.
3	Login	Fungsionalitas untuk masuk ke dalam sistem.
4	Logout	Fungsionalitas untuk keluar dari sistem.
5	Profil	Fungsionalitas menu profil pengguna
6	Ubah nama	Fungsionalitas ubah nama pengguna pada profil pengguna
7	Ubah password	Fungsionalitas untuk merubah password pengguna atau orang tua
8	Monitoring Konten negatif	Fungsionalitas untuk melakukan <i>monitoring</i> konten negatif oleh pengguna atau orang tua.
9	Instruksi penghapusan berkas gambar	Fungsionalitas untuk melakukan penghapusan berkas gambar yang dipilih pengguna atau orang tua.

10	Monitoring lokasi anak	Fungsionalitas untuk melakukan <i>monitoring</i> lokasi anak pada sistem.
11	<i>Monitoring</i> history browser	Fungsionalitas untuk melakukan <i>monitoring</i> history browser pada <i>smartphone</i> anak melalui sistem.
12	<i>Backup</i> data kontak	Fungsionalitas untuk melakukan <i>backup</i> kontak pada <i>smartphone</i> anak.
13	Notifikasi	Fungsionalitas notifikasi yang muncul pada sistem.
14	Notifikasi <i>monitoring</i> konten negatif	Fungsionalitas notifikasi <i>monitoring</i> konten negatif yang muncul pada sistem.
15	Notifikasi <i>monitoring</i> browser	Fungsionalitas notifikasi <i>monitoring</i> history browser yang muncul pada sistem.
16	Notifikasi lokasi anak	Fungsionalitas notifikasi lokasi anak yang muncul pada sistem.

3.1.6.2 Use Case Scenario

Use case skenario mendeskripsikan tahapan-tahapan atau langkah-langkah pada proses bisnis yang terlibat di dalam sistem, baik yang dilakukan aktor terhadap sistem maupun sebaliknya. Berdasarkan *use case* diagram pada Gambar 21, maka skenario *use case* untuk aplikasi yang dibangun akan dijelaskan pada Tabel

1. Skenario *Use Case* Daftar

Skenario *Use Case* Daftar akan dijelaskan pada Tabel 3.11 berikut ini:

Tabel 3.10 Skenario *Use Case* Daftar

<i>Use Case Name</i>	Daftar
<i>Related Requirements</i>	-
<i>Goal in Context</i>	Pengguna aplikasi yaitu orang tua ingin mendaftar ke dalam sistem dengan membuat akun
<i>Preconditions</i>	Pengguna belum terdaftar dan belum memiliki akun

<i>Successful End Condition</i>	Pengguna berhasil terdaftar kedalam sistem	
<i>Failed End Condition</i>	pengguna gagal terdaftar ke dalam sistem	
<i>Primary Actors</i>	Orang tua	
<i>Secondary Actor</i>	-	
<i>Trigger</i>	Pengguna memilih menu daftar untuk membuat akun	
<i>Main Flow</i>	<i>Step</i>	<i>Action</i>
	1	Sistem menampilkan <i>form</i> daftar akun
	2	Pengguna mengisi <i>form</i>
	3	Sistem mengirimkan kode pendaftaran melalui <i>email</i>
	4	Pengguna memasukkan kode pendaftaran Pada <i>form</i> pendaftaran yang disediakan
<i>Extension</i>	<i>Step</i>	<i>Branching Action</i>
		Sistem gagal mengirimkan kode pendaftaran melalui <i>email</i>

2. Skenario Use Case Lupa Password

Skenario Use Case Lupa Password akan dijelaskan pada Tabel 3.12 berikut ini:

Tabel 3.11 Skenario Use Case Lupa Password

<i>Use Case Name</i>	Lupa Password
<i>Related Requirements</i>	-
<i>Goal in Context</i>	Pengguna lupa <i>password</i> untuk masuk kedalam sistem dan ingin mengatur ulang <i>password</i> agar dapat masuk ke sistem
<i>Preconditions</i>	Pengguna telah terdaftar namun lupa <i>password</i> untuk masuk sistem
<i>Successful End Condition</i>	Sistem berhasil mengatur ulang <i>password</i> dan mengirimkan password sementara user melalui <i>email</i>

<i>Failed End Condition</i>	Sistem gagal menset ulang <i>password</i> pengguna	
<i>Primary Actors</i>	Orang tua	
<i>Secondary Actor</i>	-	
<i>Trigger</i>	Pengguna memilih menu lupa <i>password</i> untuk mendapatkan <i>password</i> baru	
<i>Main Flow</i>	<i>Step</i>	<i>Action</i>
	1	Sistem menampilkan <i>form</i> Lupa <i>password</i>
	2	Pengguna mengisi <i>form</i> dengan <i>email</i> dari akun orang tua
	3	Sistem melakukan pencocokan email yang dimasukan orang tua dengan database di <i>server</i>
	4	Sistem mengirimkan <i>password</i> baru pengguna melalui <i>email</i>
	5	Pengguna <i>login</i> kedalam sistem menggunakan <i>password</i> Baru
<i>Extension</i>	<i>Step</i>	<i>Branching Action</i>
		Sistem gagal mengirimkan <i>password</i> baru pengguna melalui <i>email</i>

3. Skenario *Use Case Login*

Skenario *Use Case Login* akan dijelaskan pada Tabel 3.13 berikut:

Tabel 3.12 Skenario *Use Case Login*

<i>Use Case Name</i>	Login
<i>Related Requirements</i>	Profil, <i>Monitoring</i> konten negatif, <i>monitoring</i> lokasi anak, <i>monitoring history browser</i> , <i>backup data</i> kontak, notifikasi
<i>Goal in Context</i>	Pengguna ingin masuk kedalam sistem
<i>Preconditions</i>	Sistem menampilkan <i>form login</i>

<i>Successful End Condition</i>	Pengguna berhasil masuk kedalam sistem	
<i>Failed End Condition</i>	pengguna gagal masuk ke dalam sistem	
<i>Primary Actors</i>	Orang tua	
<i>Secondary Actor</i>	-	
<i>Trigger</i>	Pengguna memilih menu daftar untuk membuat akun	
<i>Main Flow</i>	<i>Step</i>	<i>Action</i>
	1	Pengguna yaitu orangtua mengisi <i>form login</i>
	2	Sistem mencocokkan data <i>login</i> orangtua dengan data orang tua yang ada di <i>database server</i>
	3	Sistem meng-generate <i>token key</i> orangtua dan anak
	4	Orangtua mengonfirmasi sesi <i>login</i>
	5	Orang tua berhasil masuk ke dalam sistem
<i>Extension</i>	<i>Step</i>	<i>Branching Action</i>
		Sistem gagal memproses informasi <i>login</i> orangtua

4. Skenario Use Case Log out

Skenario Use Case Log Out akan dijelaskan pada Tabel 3.14 berikut ini:

Tabel 3.13 Skenario Use Case Log Out

<i>Use Case Name</i>	Log Out
<i>Related Requirements</i>	Login
<i>Goal in Context</i>	Pengguna ingin keluar dari sistem
<i>Preconditions</i>	Pengguna yaitu orang tua berada di dalam sistem
<i>Successful End Condition</i>	Pengguna berhasil keluar dari sistem
<i>Failed End Condition</i>	Pengguna gagal keluar dari sistem

<i>Primary Actors</i>	Orang tua	
<i>Secondary Actor</i>	-	
<i>Trigger</i>	Orangtua mengklik menu <i>logout</i>	
<i>Main Flow</i>	<i>Step</i>	<i>Action</i>
	<i>Include::Login</i>	
	1	Pengguna yaitu orangtua memilih menu <i>log out</i>
	2	Orang tua berhasil keluar dari sistem
<i>Extension</i>	<i>Step</i>	<i>Branching Action</i>
	1	Sistem gagal membuat koneksi ke server

5. Skenario *Use Case* Profil

Skenario *Use Case* Profil akan dijelaskan pada Tabel 3.15 berikut ini:

Tabel 3.14 Skenario *Use Case* Profil

<i>Use Case Name</i>	Profil	
<i>Related Requirements</i>	<i>Login</i>	
<i>Goal in Context</i>	Orang tua ingin melihat profil akun	
<i>Preconditions</i>	Sistem menampilkan menu utama	
<i>Successful End Condition</i>	Orang tua dapat melihat data akun orangtua di sistem	
<i>Failed End Condition</i>	Data profil akun orang tua gagal ditampilkan	
<i>Primary Actors</i>	Orang tua	
<i>Secondary Actor</i>	-	
<i>Trigger</i>	Orangtua meng-klik menu profil	
<i>Main Flow</i>	<i>Step</i>	<i>Action</i>
	<i>Include::Login</i>	
	1	Pengguna yaitu orangtua memilih menu <i>profil</i>

	2	Sistem menampilkan data profil dari akun orang tua
<i>Extension</i>	Step	<i>Branching Action</i>
	1	Sistem gagal mengambil data profil orang tua dari server

6. Skenario *Use Case* Ubah Nama

Skenario *Use Case* Ubah Nama akan dijelaskan pada Tabel 3.16 berikut ini:

Tabel 3.15 Skenario *Use Case* Ubah Nama

<i>Use Case Name</i>	Ubah Nama	
<i>Related Requirements</i>	<i>Login</i>	
<i>Goal in Context</i>	Orang tua ingin merubah nama di akun orang tua	
<i>Preconditions</i>	Sistem menampilkan menu profil	
<i>Successful End Condition</i>	Orang tua berhasil merubah data nama akun orangtua di sistem	
<i>Failed End Condition</i>	Data nama profil akun orang tua gagal dirubah	
<i>Primary Actors</i>	Orang tua	
<i>Secondary Actor</i>	-	
<i>Trigger</i>	Orangtua meng-klik menu ubah nama akun	
<i>Main Flow</i>	<i>Step</i>	<i>Action</i>
	<i>Include::Login</i>	
	1	Sistem menampilkan informasi profil akun
	2	Orang tua mengubah informasi nama akun pada <i>form</i> yang disediakan
	3	Pengguna berhasil menyimpan informasi nama dari profil akun
<i>Extension</i>	Step	<i>Branching Action</i>

	1	Sistem gagal menyimpan perubahan data nama profil akun
--	---	--

7. Skenario *Use Case* Ubah Password

Skenario *Use Case* Ubah Password akan dijelaskan pada Tabel 3.17 berikut ini:

Tabel 3.16 Skenario *Use Case* Ubah Password

<i>Use Case Name</i>	Ubah Password	
<i>Related Requirements</i>	<i>Login</i>	
<i>Goal in Context</i>	Orang tua ingin merubah password di akun orang tua	
<i>Preconditions</i>	Sistem menampilkan menu profil	
<i>Successful End Condition</i>	Orang tua berhasil merubah data <i>password</i> akun orangtua di sistem	
<i>Failed End Condition</i>	Data <i>password</i> profil akun orang tua gagal dirubah	
<i>Primary Actors</i>	Orang tua	
<i>Secondary Actor</i>	-	
<i>Trigger</i>	Orangtua meng-klik menu ubah <i>password</i> akun	
<i>Main Flow</i>	<i>Step</i>	<i>Action</i>
	<i>Include::Login</i>	
	1	Sistem menampilkan informasi profil akun
	2	Orang tua mengubah informasi <i>password</i> akun pada <i>form</i> yang disediakan
	3	Pengguna berhasil menyimpan informasi perubahan <i>password</i> dari profil akun
<i>Extension</i>	<i>Step</i>	<i>Branching Action</i>
	1	Sistem gagal menyimpan perubahan data <i>password</i> profil akun

8. Skenario *Use Case Monitoring* Konten Negatif

Skenario *Use Case Monitoring* Konten Negatif akan dijelaskan pada Tabel 3.18 berikut ini:

Tabel 3.17 Skenario *Use Case Monitoring* Konten Negatif

<i>Use Case Name</i>	Monitoring Konten Negatif	
<i>Related Requirements</i>	<i>Login</i>	
<i>Goal in Context</i>	Orang tua ingin melihat hasil <i>monitoring</i> konten negatif <i>smartphone Android</i>	
<i>Preconditions</i>	Orang tua mengirim instruksi <i>monitoring</i> konten negatif	
<i>Successful End Condition</i>	Orang tua dapat melihat konten hasil <i>monitoring</i>	
<i>Failed End Condition</i>	Orang tua gagal melihat konten hasil <i>monitoring</i>	
<i>Primary Actors</i>	Orang tua	
<i>Secondary Actor</i>	-	
<i>Trigger</i>	Orangtua meng-klik menu <i>monitoring</i> konten negatif	
<i>Main Flow</i>	<i>Step</i>	<i>Action</i>
	<i>Include::Login</i>	
	1	Sistem memberikan instruksi ke server
	2	Sistem menampilkan konten hasil <i>monitoring</i>
<i>Extension</i>	<i>Step</i>	<i>Branching Action</i>
	1	Sistem gagal mengunggah konten sehingga orang tua tidak dapat melihat informasi hasil <i>monitoring</i>

9. Skenario *Use Case* Penghapusan Berkas Gambar

Skenario *Use Case* Penghapusan Berkas Gambar akan dijelaskan pada Tabel 3.19 berikut ini:

Tabel 3.18 Skenario *Use Case* Penghapusan Berkas Gambar

<i>Use Case Name</i>	Penghapusan Berkas Gambar	
<i>Related Requirements</i>	<i>Monitoring</i> konten negatif	
<i>Goal in Context</i>	Sistem dapat menjalankan fungsi hapus berkas gambar direktori eksternal	
<i>Preconditions</i>	Orang tua lokasi berkas yang akan dihapus	
<i>Successful End Condition</i>	Sistem berhasil menghapus konten pada direktori eksternal	
<i>Failed End Condition</i>	Sistem gagal menghapus konten pada direktori eksternal	
<i>Primary Actors</i>	Orang tua	
<i>Secondary Actor</i>	-	
<i>Trigger</i>	Sistem menerima instruksi penghapusan berkas gambar	
<i>Main Flow</i>	<i>Step</i>	<i>Action</i>
	1	Orang tua memilih instruksi penghapusan berkas gambar
	2	Sistem menerima instruksi penghapusan berkas gambar
	3	Sistem membaca lokasi penyimpanan berkas gambar
	4	Sistem berhasil menghapus konten gambar
<i>Extension</i>	<i>Step</i>	<i>Branching Action</i>
	1	Sistem gagal menghapus konten gambar

10. Skenario *Use Case Monitoring* Lokasi Anak

Skenario *Use Case Monitoring Lokasi Anak* akan dijelaskan pada Tabel 3.20 berikut ini:

Tabel 3.19 Skenario *Use Case Monitoring Lokasi Anak*

<i>Use Case Name</i>	<i>Monitoring Konten Lokasi Anak</i>	
<i>Related Requirements</i>	<i>Login</i>	
<i>Goal in Context</i>	Sistem dapat menjalankan fungsi pencarian lokasi pada <i>smartphone</i> Android	
<i>Preconditions</i>	Orang tua memilih menu <i>monitoring</i> lokasi anak	
<i>Successful End Condition</i>	Orang tua dapat melihat konten hasil <i>monitoring</i>	
<i>Failed End Condition</i>	Orang tua gagal melihat konten hasil <i>monitoring</i>	
<i>Primary Actors</i>	Orang tua	
<i>Secondary Actor</i>	-	
<i>Trigger</i>	Orangtua meng-klik menu <i>monitoring</i> lokasi anak	
<i>Main Flow</i>	<i>Step</i>	<i>Action</i>
	<i>Include::Login</i>	
	1	Sistem memberikan instruksi ke server
	2	Sistem menampilkan konten hasil <i>monitoring lokasi</i>
<i>Extension</i>	<i>Step</i>	<i>Branching Action</i>
	1	Sistem gagal mengunggah konten sehingga orang tua tidak dapat melihat informasi hasil <i>monitoring lokasi</i>

11. Skenario *Use Case Monitoring History Browser*

Skenario *Use Case Log Out* akan dijelaskan pada Tabel 3.21 berikut ini:

Tabel 3.20 *Monitoring History Browser*

<i>Use Case Name</i>	<i>Monitoring History Browser</i>	
<i>Related Requirements</i>	<i>Login</i>	
<i>Goal in Context</i>	Sistem dapat menjalankan fungsi <i>monitoring history browser</i> pada <i>smartphone</i> Android	
<i>Preconditions</i>	Orang tua masuk menu aplikasi	
<i>Successful End Condition</i>	Orang tua dapat melihat konten hasil <i>monitoring history browser</i>	
<i>Failed End Condition</i>	Orang tua gagal melihat konten hasil <i>monitoring history browser</i>	
<i>Primary Actors</i>	Orang tua	
<i>Secondary Actor</i>	-	
<i>Trigger</i>	Orangtua meng-klik menu <i>monitoring history browser</i>	
<i>Main Flow</i>	<i>Step</i>	<i>Action</i>
	<i>Include::Login</i>	
	1	Sistem memberikan instruksi ke server untuk melakukan fungsi <i>monitoring history browser</i>
	2	Sistem menampilkan konten hasil <i>monitoring history browser</i>
<i>Extension</i>	<i>Step</i>	<i>Branching Action</i>
	1	Sistem gagal mengunggah konten sehingga orang tua tidak dapat melihat informasi hasil <i>monitoring history browser</i>

12. Skenario Use Case Backup Data Kontak

Skenario Use Case Backup Data Kontak akan dijelaskan pada Tabel 3.22 berikut ini:

Tabel 3.21 Backup data Kontak

<i>Use Case Name</i>	<i>Backup Data Kontak</i>	
<i>Related Requirements</i>	<i>Login</i>	
<i>Goal in Context</i>	Sistem dapat menjalankan fungsi <i>backup data</i> kontak pada <i>Smartphone</i> Android	
<i>Preconditions</i>		
<i>Successful End Condition</i>	Orang tua dapat menjalankan fungsi <i>Backup Data Kontak</i>	
<i>Failed End Condition</i>	Orang tua gagal melakukan <i>Backup Data Kontak</i>	
<i>Primary Actors</i>	Orang tua	
<i>Secondary Actor</i>	-	
<i>Trigger</i>	Orangtua meng-klik menu <i>Backup Data Kontak</i>	
<i>Main Flow</i>	<i>Step</i>	<i>Action</i>
	<i>Include::Login</i>	
	1	Sistem memberikan instruksi ke server untuk menjalankan fungsi backup kontak
	2	Sistem Menjalankan fungsi <i>backup kontak</i>
	3	Orangtua berhasil melakukan <i>backup data</i> kontak
<i>Extension</i>	<i>Step</i>	<i>Branching Action</i>
	1	Sistem gagal menjalankan fungsi <i>backup data</i> kontak

13. Skenario *Use Case* Notifikasi

Skenario *Use Case* Notifikasi akan dijelaskan pada Tabel 3.23 berikut ini:

Tabel 3.22 Notifikasi

<i>Use Case Name</i>	Notifikasi
<i>Related Requirements</i>	<i>Login</i>

<i>Goal in Context</i>	Orang tua dapat menerima notifikasi	
<i>Preconditions</i>	Orang masuk ke dalam sistem	
<i>Successful End Condition</i>	Orang tua menerima notifikasi laporan hasil monitoring	
<i>Failed End Condition</i>	Orang tua tidak dapat menerima notifikasi <i>monitoring</i>	
<i>Primary Actors</i>	Orang tua	
<i>Secondary Actor</i>	-	
<i>Trigger</i>	Fungsi dari operasi <i>monitoring</i>	
<i>Main Flow</i>	<i>Step</i>	<i>Action</i>
	<i>Include::Login</i>	
	1	Sistem menjalankan fungsi monitoring pada aplikasi
	2	Sistem membaca hasil <i>monitoring</i>
	3	Sistem menjalankan notifikasi pada aplikasi
<i>Extension</i>	<i>Step</i>	<i>Branching Action</i>
	1	Sistem gagal memberikan notifikasi laporan hasil monitoring

14. Skenario *Use Case* Notifikasi *Monitoring* Konten negatif

Skenario *Use Case Monitoring* Konten negatif akan dijelaskan pada Tabel 3.24 berikut ini:

Tabel 3.23 Notifikasi *Monitoring* Konten Negatif

<i>Use Case Name</i>	Notifikasi <i>Monitoring</i> Konten Negatif
<i>Related Requirements</i>	<i>Login</i>
<i>Goal in Context</i>	Sistem dapat mengirimkan Notifikasi hasil <i>monitoring</i> konten negatif pada <i>smartphone Android</i>
<i>Preconditions</i>	Orang tua berada di dalam sistem

<i>Successful End Condition</i>	Orang tua dapat menerima notifikasi hasil <i>monitoring</i> konten negatif	
<i>Failed End Condition</i>	Orang tua gagal menerima notifikasi hasil <i>monitoring</i> konten negatif	
<i>Primary Actors</i>	Orang tua	
<i>Secondary Actor</i>	-	
<i>Trigger</i>	Sistem menjalankan fungsi <i>monitoring</i> konten negatif	
<i>Main Flow</i>	<i>Step</i>	<i>Action</i>
	1	Sistem menjalankan fungsi <i>monitoring</i> konten negatif
	2	Sistem membaca hasil <i>monitoring</i> konten negatif
	3	Sistem mengirimkan notifikasi hasil <i>monitoring</i> konten negatif
	4	Orangtua berhasil menerima notifikasi hasil <i>monitoring</i> konten negatif
<i>Extension</i>	<i>Step</i>	<i>Branching Action</i>
	1	Sistem gagal mengirimkan notifikasi hasil <i>monitoring</i> konten negatif

15. Skenario *Use Case* Notifikasi *Monitoring History Browser*

Skenario *Use Case* Notifikasi *Monitoring History Browser* akan dijelaskan pada Tabel 3.25 berikut ini:

Tabel 3.24 Case Notifikasi *Monitoring History Browser*

<i>Use Case Name</i>	Notifikasi <i>Monitoring History Browser</i>
<i>Related Requirements</i>	<i>Login</i>
<i>Goal in Context</i>	Sistem dapat mengirimkan notifikasi hasil <i>monitoring history browser</i> pada <i>smartphone</i> Android.
<i>Preconditions</i>	Orang tua berada di dalam sistem

<i>Successful End Condition</i>	Orang tua dapat melihat notifikasi hasil <i>monitoring history browser</i>	
<i>Failed End Condition</i>	Sistem gagal mengirimkan notifikasi <i>monitoring history browser</i>	
<i>Primary Actors</i>	Orang tua	
<i>Secondary Actor</i>	-	
<i>Trigger</i>	Sistem menjalankan fungsi <i>monitoring history browser</i>	
<i>Main Flow</i>	<i>Step</i>	<i>Action</i>
	1	Sistem menjalankan fungsi <i>monitoring history browser</i>
	2	Sistem membaca hasil <i>monitoring history browser</i>
	3	Sistem mengirimkan notifikasi hasil <i>monitoring history browser</i> kepada orang tua
	4	Orangtua berhasil menerima notifikasi hasil <i>monitoring history browser</i>
<i>Extension</i>	<i>Step</i>	<i>Branching Action</i>
	1	Sistem gagal mengirimkan notifikasi hasil <i>monitoring</i>

16. Skenario *Use Case* Notifikasi *Monitoring Lokasi*

Skenario *Use Case Notifikasi Monitoring Lokasi* akan dijelaskan pada Tabel 3.26 berikut ini:

Tabel 3.25 Notifikasi *Monitoring Lokasi*

<i>Use Case Name</i>	<i>Notifikasi Monitoring Lokasi</i>
<i>Related Requirements</i>	<i>Login</i>
<i>Goal in Context</i>	Sistem dapat mengirimkan notifikasi hasil <i>Monitoring Lokasi</i> anak
<i>Preconditions</i>	Orang tua berada di dalam sistem

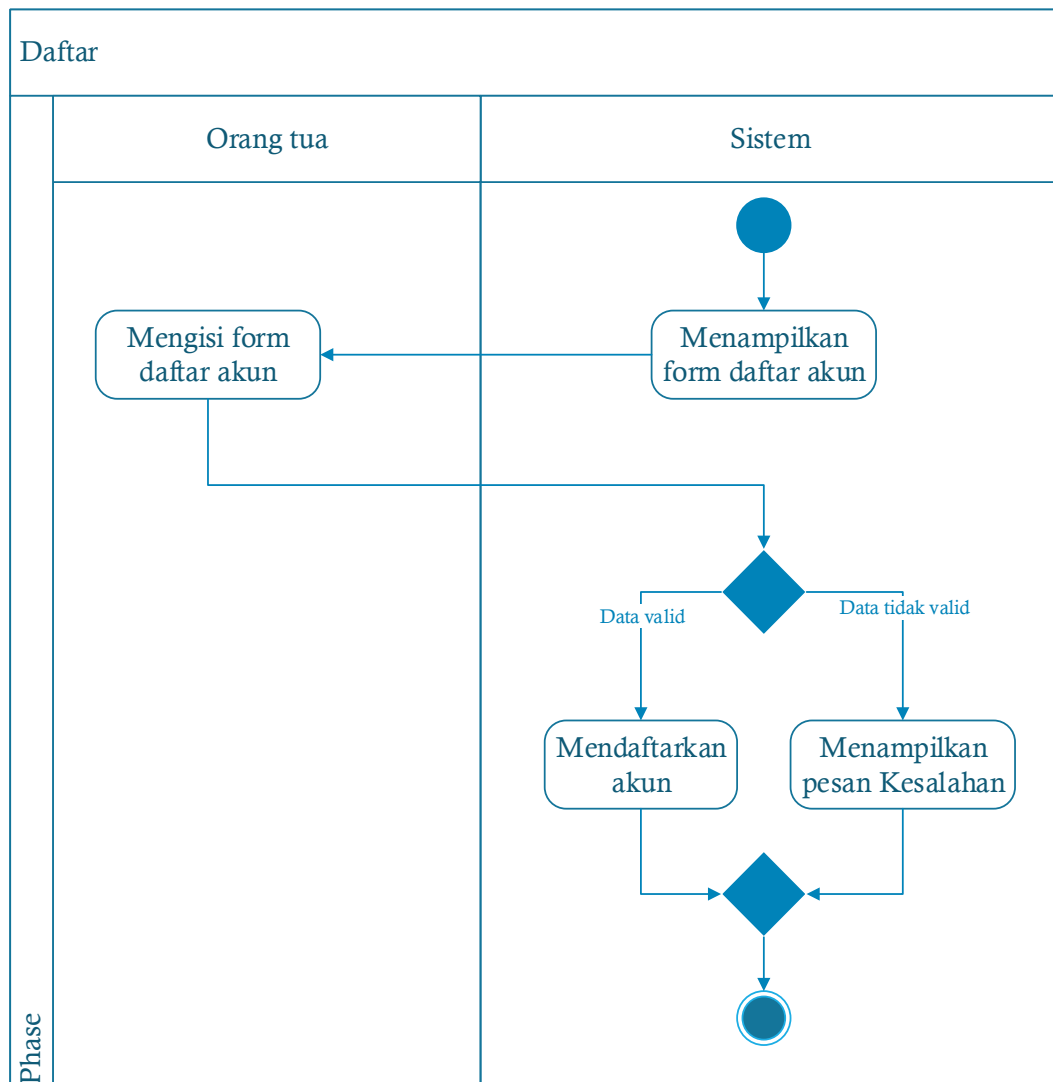
<i>Successful End Condition</i>	Orang tua dapat melihat notifikasi hasil <i>Monitoring</i> Lokasi anak	
<i>Failed End Condition</i>	Sistem gagal mengirimkan notifikasi <i>monitoring</i> lokasi anak	
<i>Primary Actors</i>	Orang tua	
<i>Secondary Actor</i>	-	
<i>Trigger</i>	Sistem menjalankan fungsi <i>monitoring lokasi anak</i>	
<i>Main Flow</i>	<i>Step</i>	<i>Action</i>
	1	Sistem menjalankan fungsi <i>monitoring</i> lokasi anak
	2	Sistem membaca hasil <i>monitoring</i> lokasi anak
	3	Sistem mengirimkan notifikasi hasil <i>monitoring</i> lokasi anak kepada orangtua
	4	Orangtua berhasil menerima notifikasi hasil <i>monitoring</i> Lokasi anak
<i>Extension</i>	<i>Step</i>	<i>Branching Action</i>
	1	Sistem gagal mengirimkan notifikasi hasil <i>monitoring</i> lokasi anak

3.1.6.3 Activity Diagram

Activity diagram memodelkan aliran kerja atau *workflow* dari urutan aktivitas dalam suatu proses yang mengacu pada *use case diagram* yang dibuat sebelumnya.

1. Activity Diagram Daftar

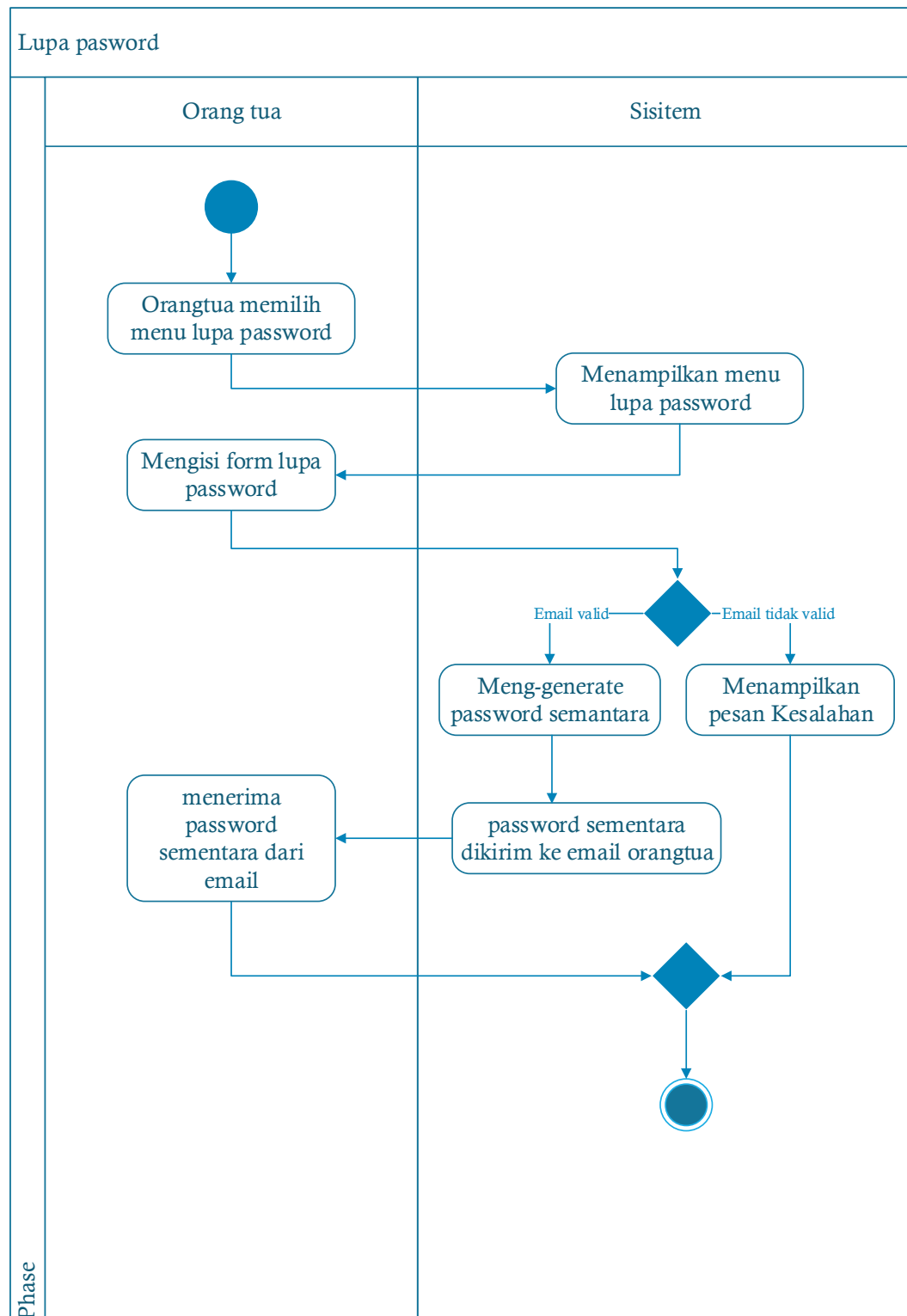
Activity daftar menjelaskan aliran kerja aktor pengguna yaitu orang tua pada saat akan mendaftar ke dalam sistem. *Activity diagram* daftar dapat dilihat pada Gambar 22.



Gambar 28 Activity Diagram Daftar

2. Activity Diagram Lupa Password

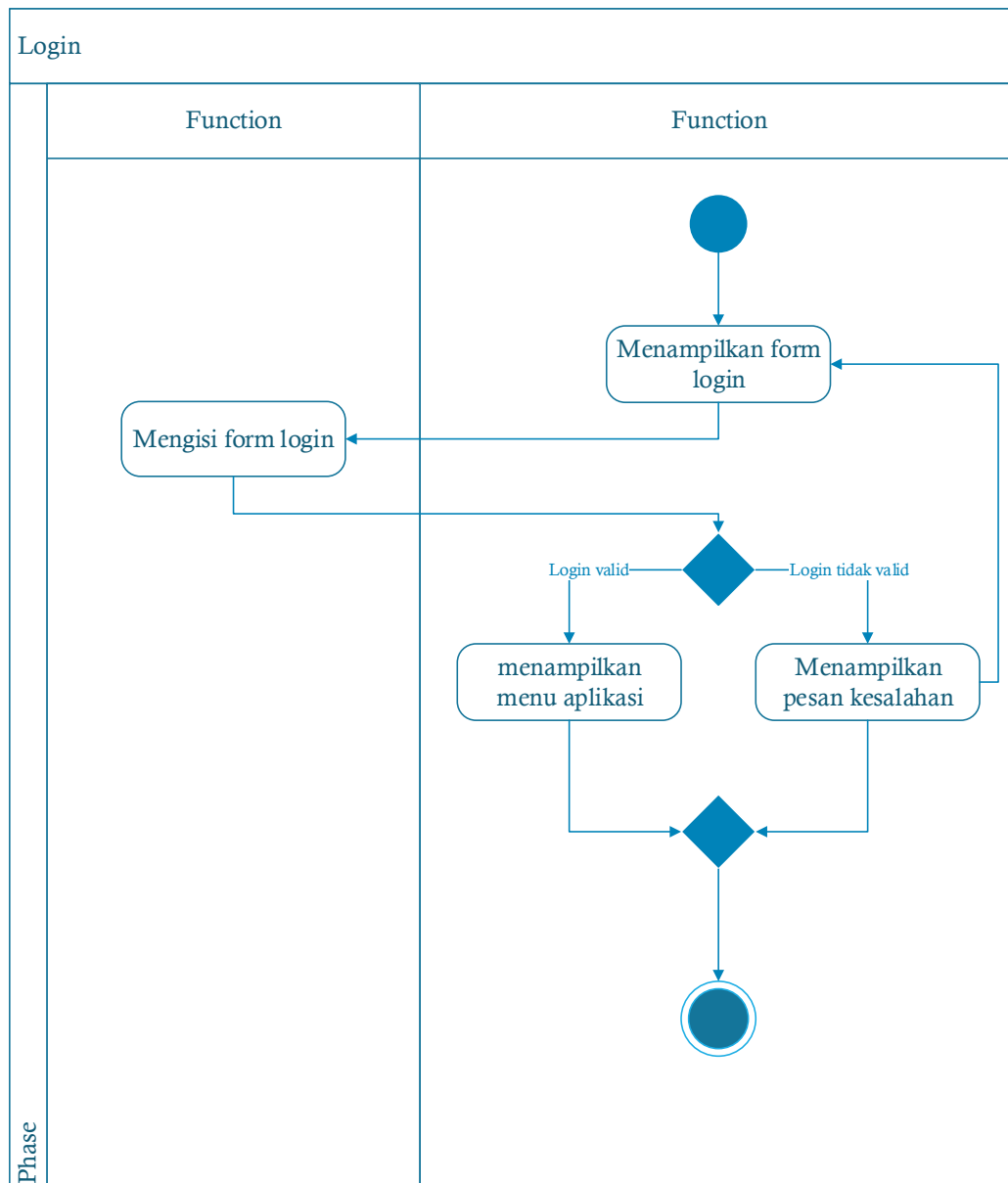
Activity diagram Lupa Password menjelaskan aliran kerja aktor orang tua pada saat hendak masuk kedalam sistem namun terkendala lupa pada *password*. *Activity diagram Lupa password* dapat dilihat pada Gambar 23.



Gambar 29 Activity Diagram Lupa password

3. Activity Diagram Login

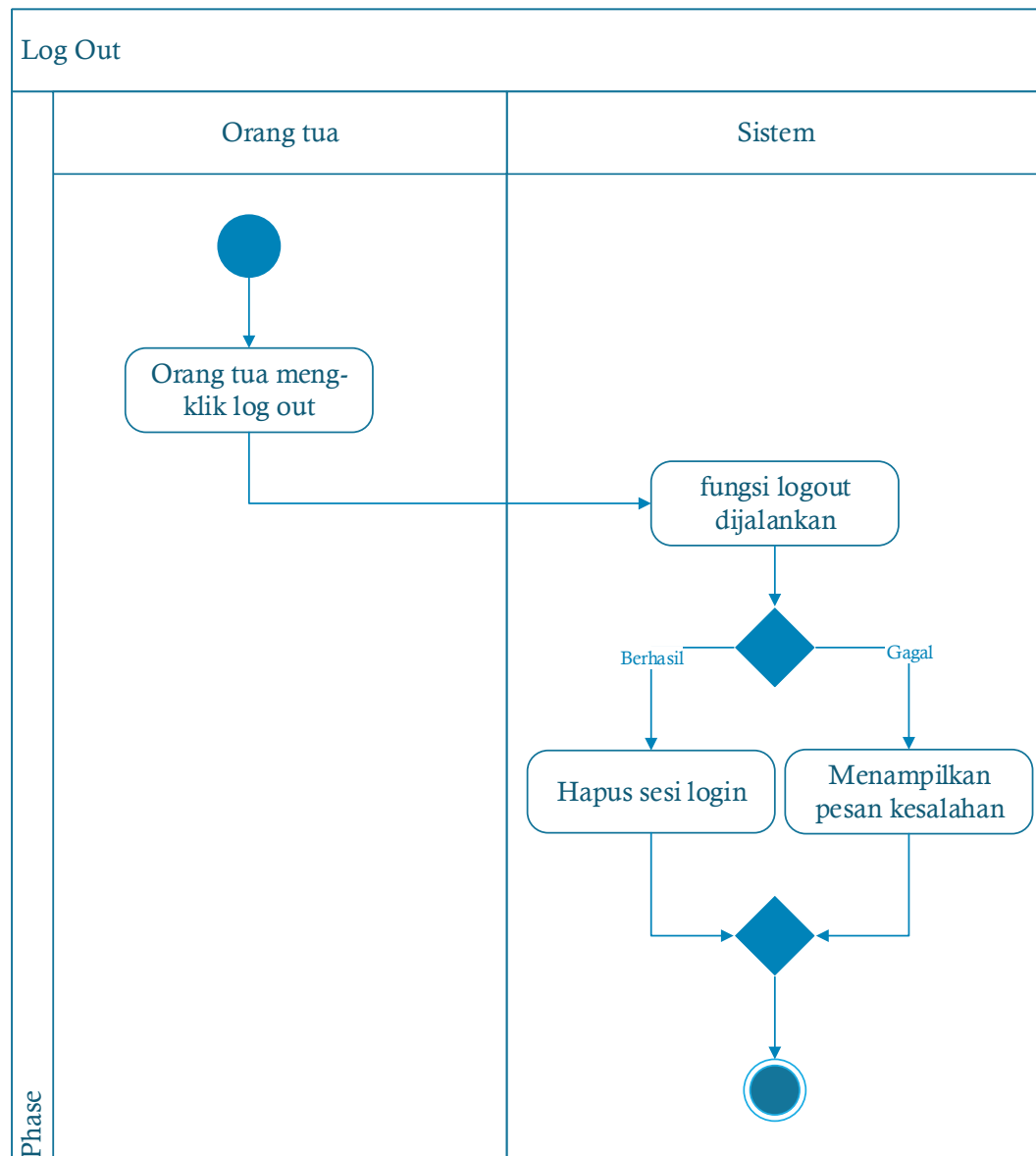
Activity diagram Login menjelaskan aliran kerja aktor orang tua pada saat hendak masuk kedalam sistem dengan melakukan *login*. *Activity diagram Login* dapat dilihat pada Gambar 24.



Gambar 30 Activity Diagram Login

4. Activity Diagram Log Out

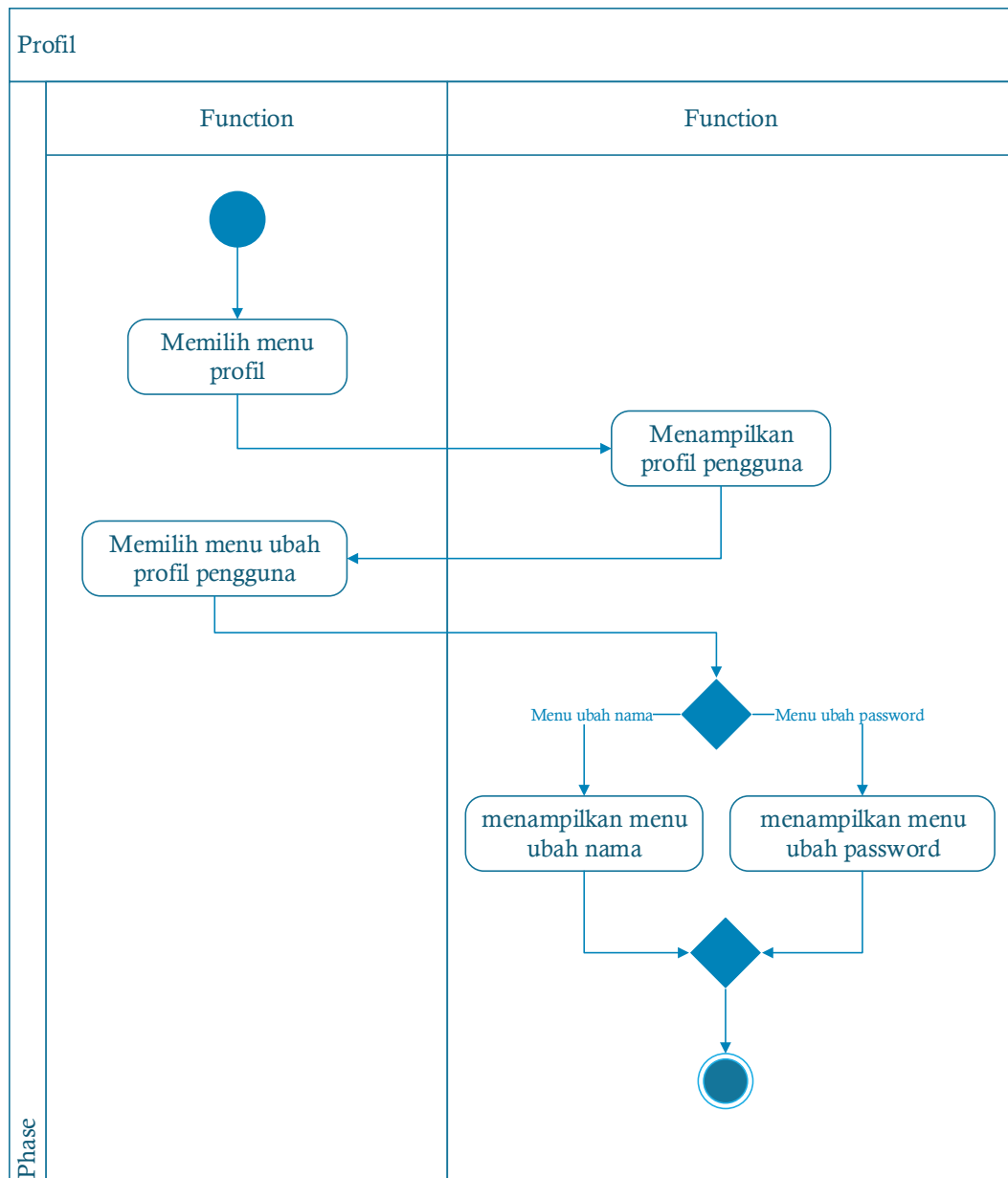
Activity diagram Log Out menjelaskan aliran kerja aktor orang tua pada saat akan keluar dari sistem dengan melakukan *log Out*. *Activity diagram Log Out* dapat dilihat pada Gambar 25.



Gambar 31 Activity Diagram Log Out

5. Activity Diagram Profil

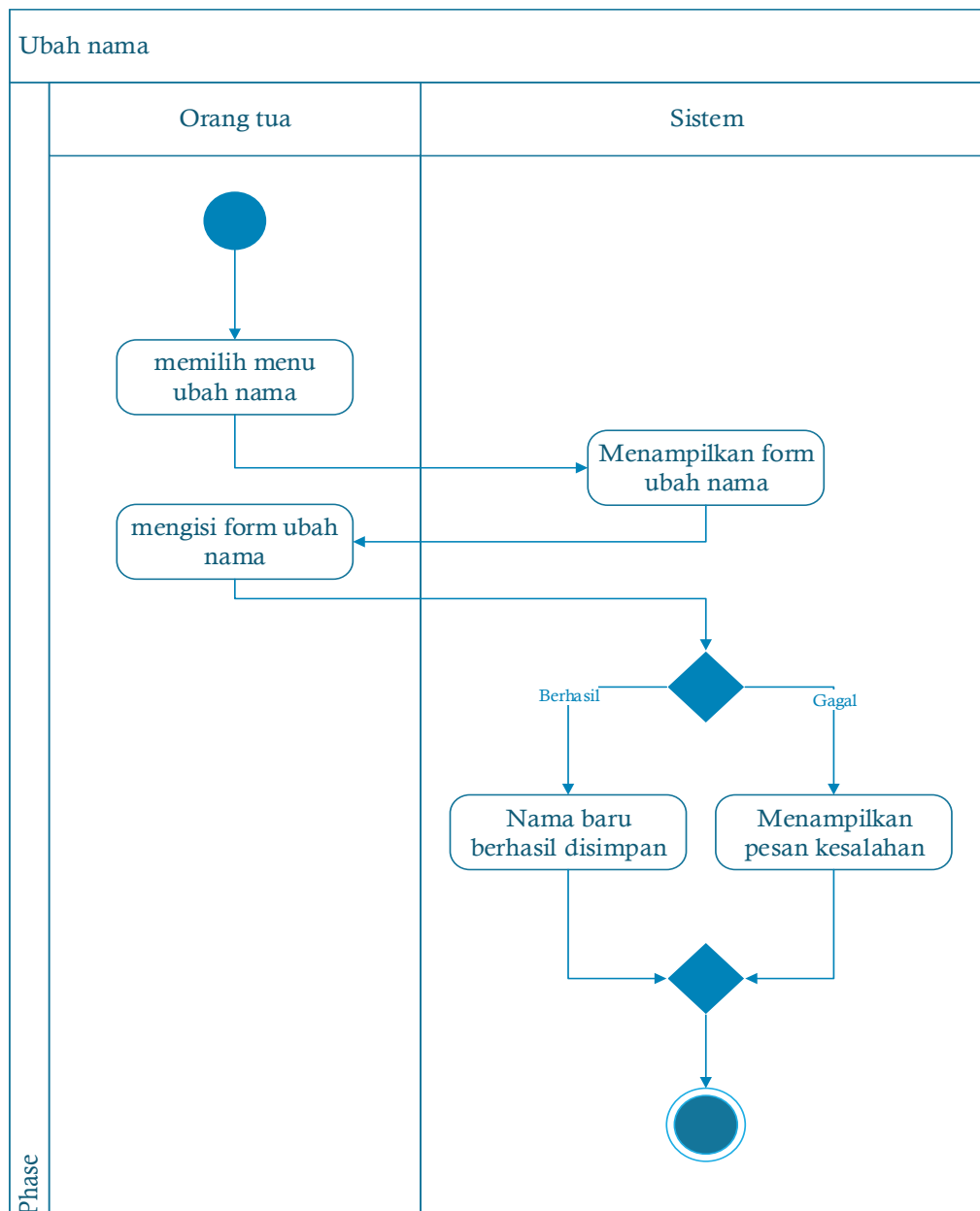
Activity diagram profil menjelaskan aliran kerja aktor orang tua pada saat menjalankan menu profil. Activity diagram profil dapat dilihat pada Gambar 26.



Gambar 32 Activity Diagram Profil

6. Activity Diagram Ubah nama

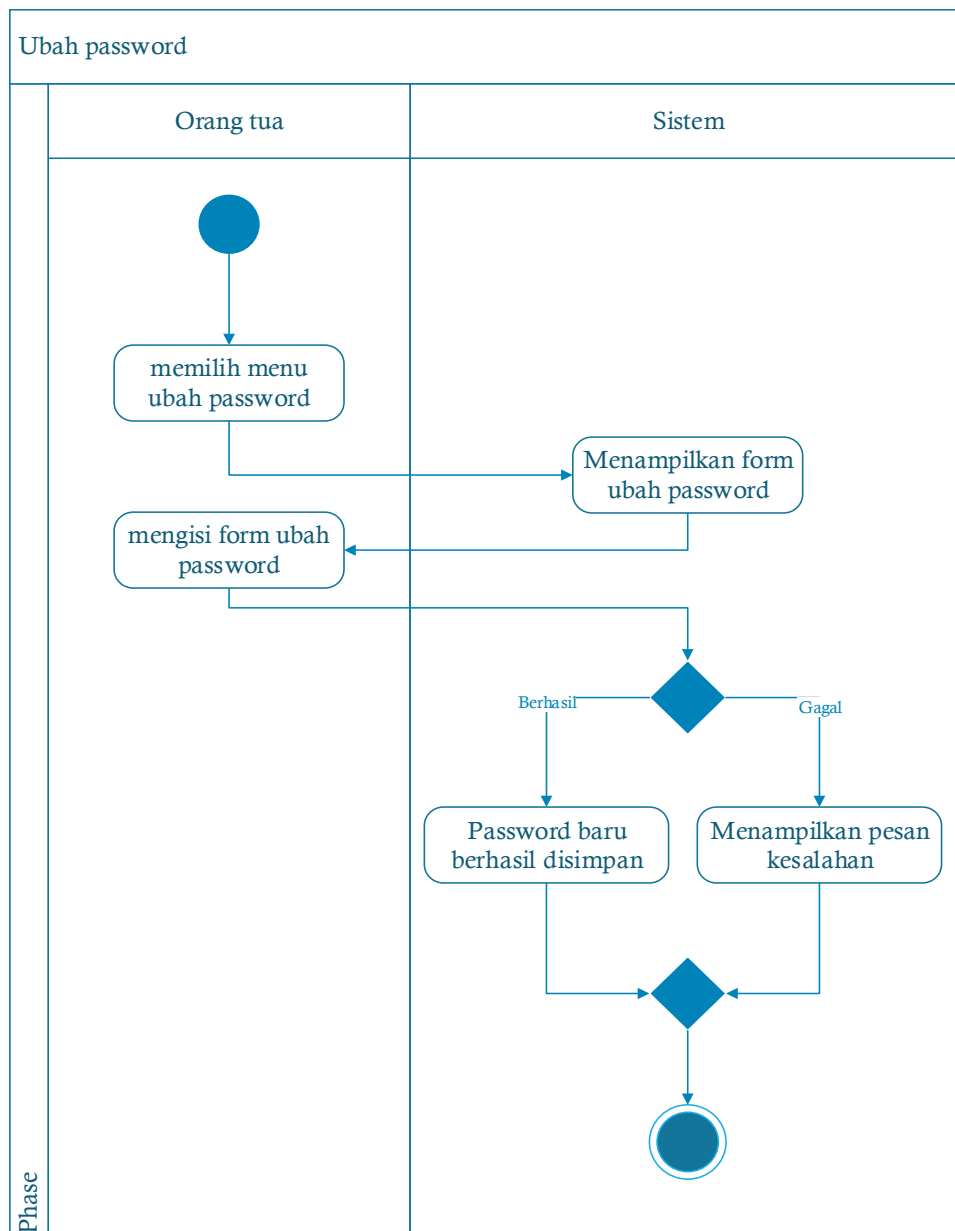
Activity diagram profil menjelaskan aliran kerja aktor orang tua pada saat akan mengubah nama pada akun profil orang tua. Activity diagram ubah nama dapat dilihat pada Gambar 27.



Gambar 33 Activity Diagram Ubah nama

7. Activity Diagram Ubah Password

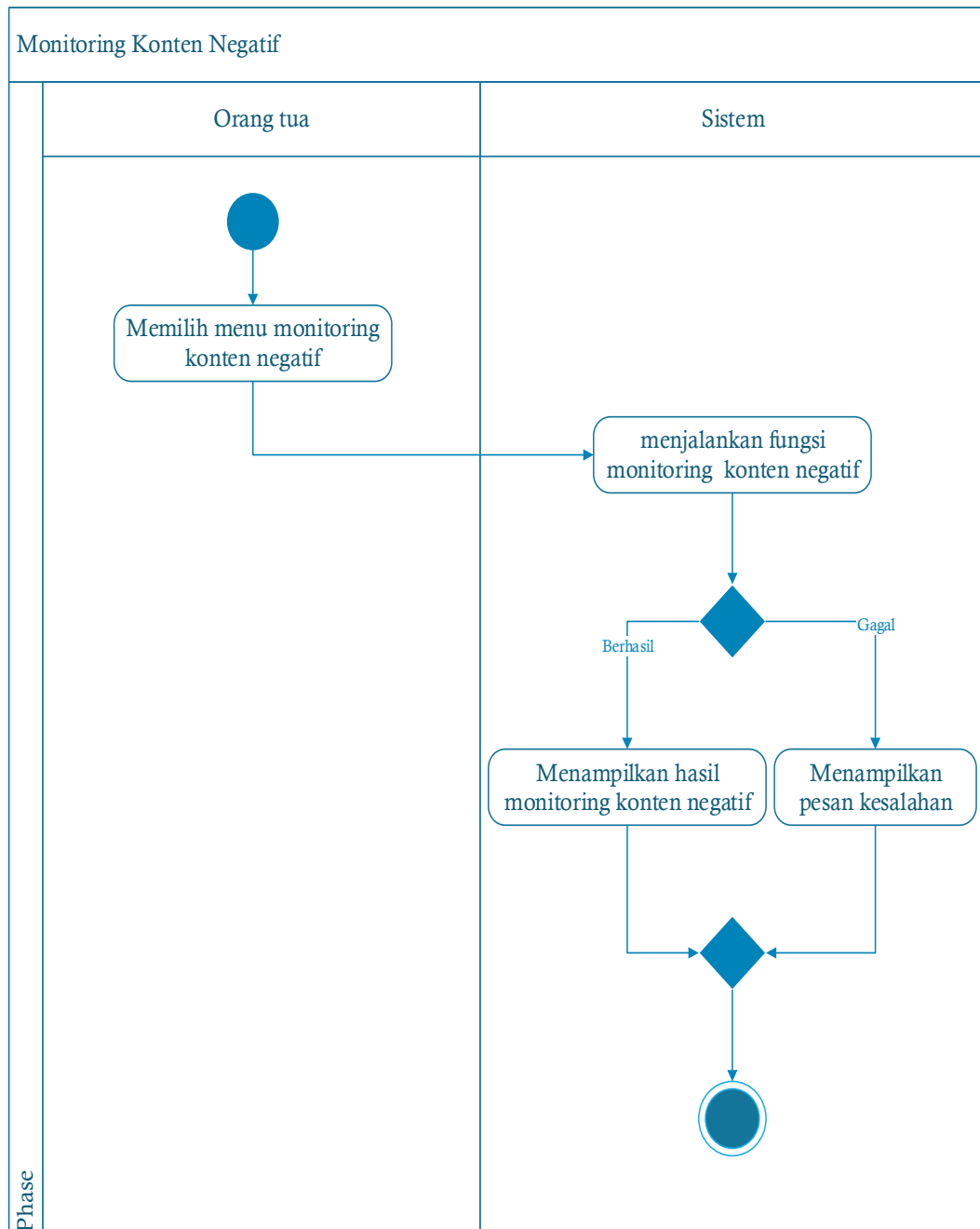
Activity diagram profil menjelaskan aliran kerja aktor orang tua pada saat akan mengubah *password* pada akun profil orang tua. Activity diagram ubah *password* dapat dilihat pada Gambar 28.



Gambar 34 Activity Diagram Ubah Password

8. *Activity Diagram Monitoring* Konten Negatif

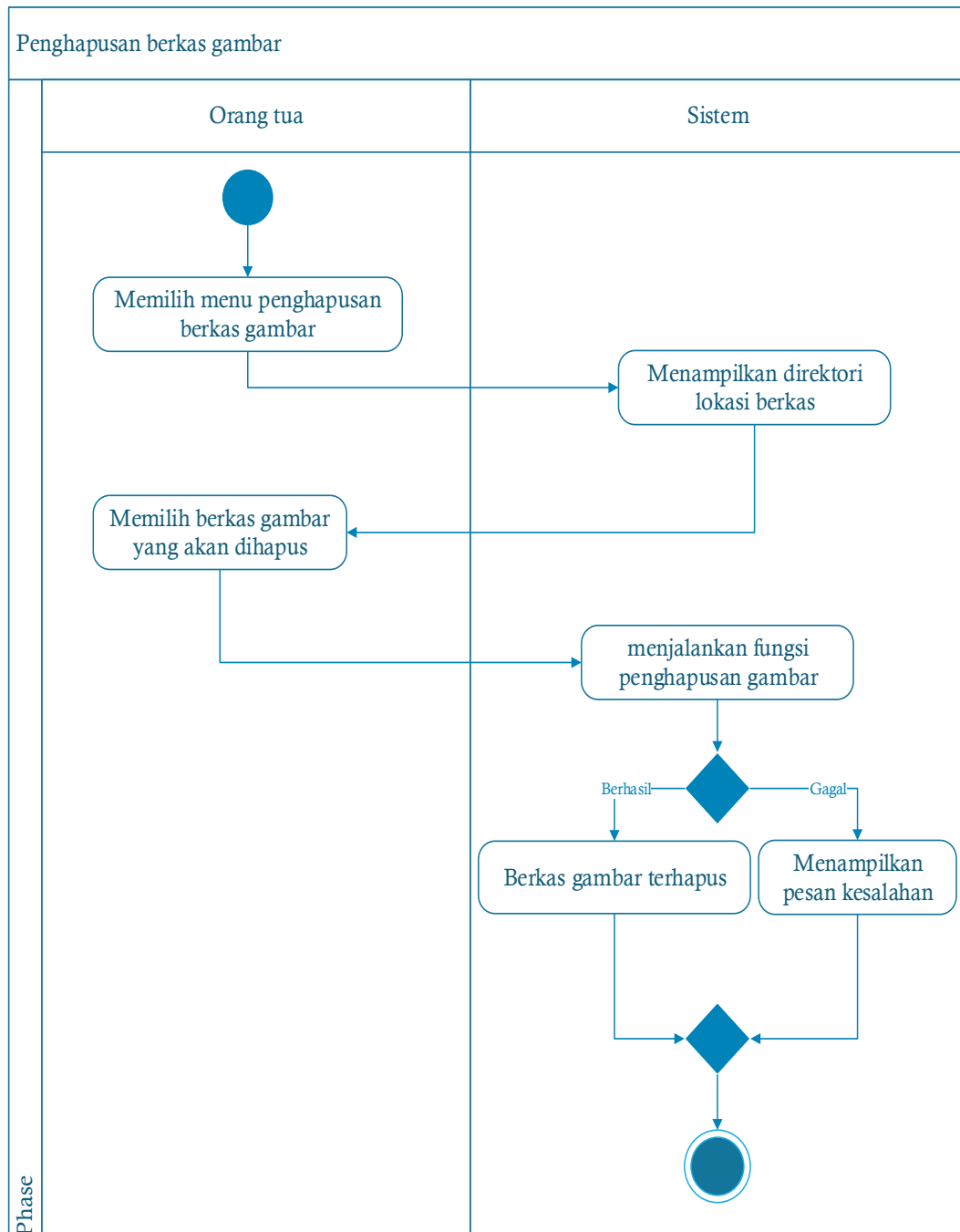
Activity diagram monitoring Konten Negatif menjelaskan aliran kerja sistem dalam menampilkan konten hasil *monitoring*. *Activity diagram monitoring* konten negatif dapat dilihat pada Gambar 29.



Gambar 35 *Activity Diagram Monitoring* Konten Negatif

9. Activity Diagram Penghapusan Berkas Gambar

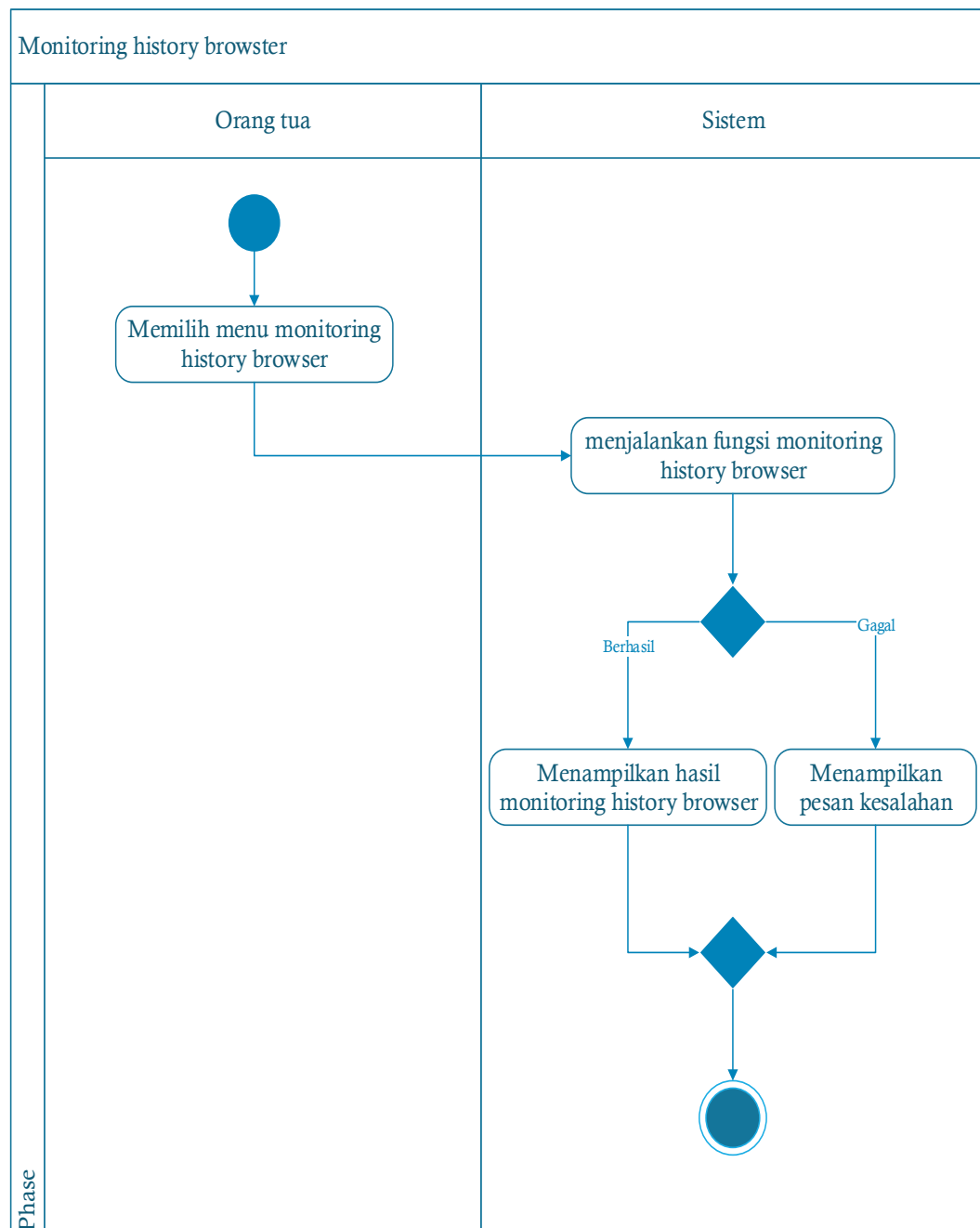
Activity diagram penghapusan berkas gambar menjelaskan aliran kerja sistem dalam menjalankan fungsi penghapusan berkas gambar. Activity diagram penghapusan berkas gambar dapat dilihat pada Gambar 30.



Gambar 36 Activity Diagram Penghapusan Berkas Gambar

10. Activity Diagram Monitoring History Browser

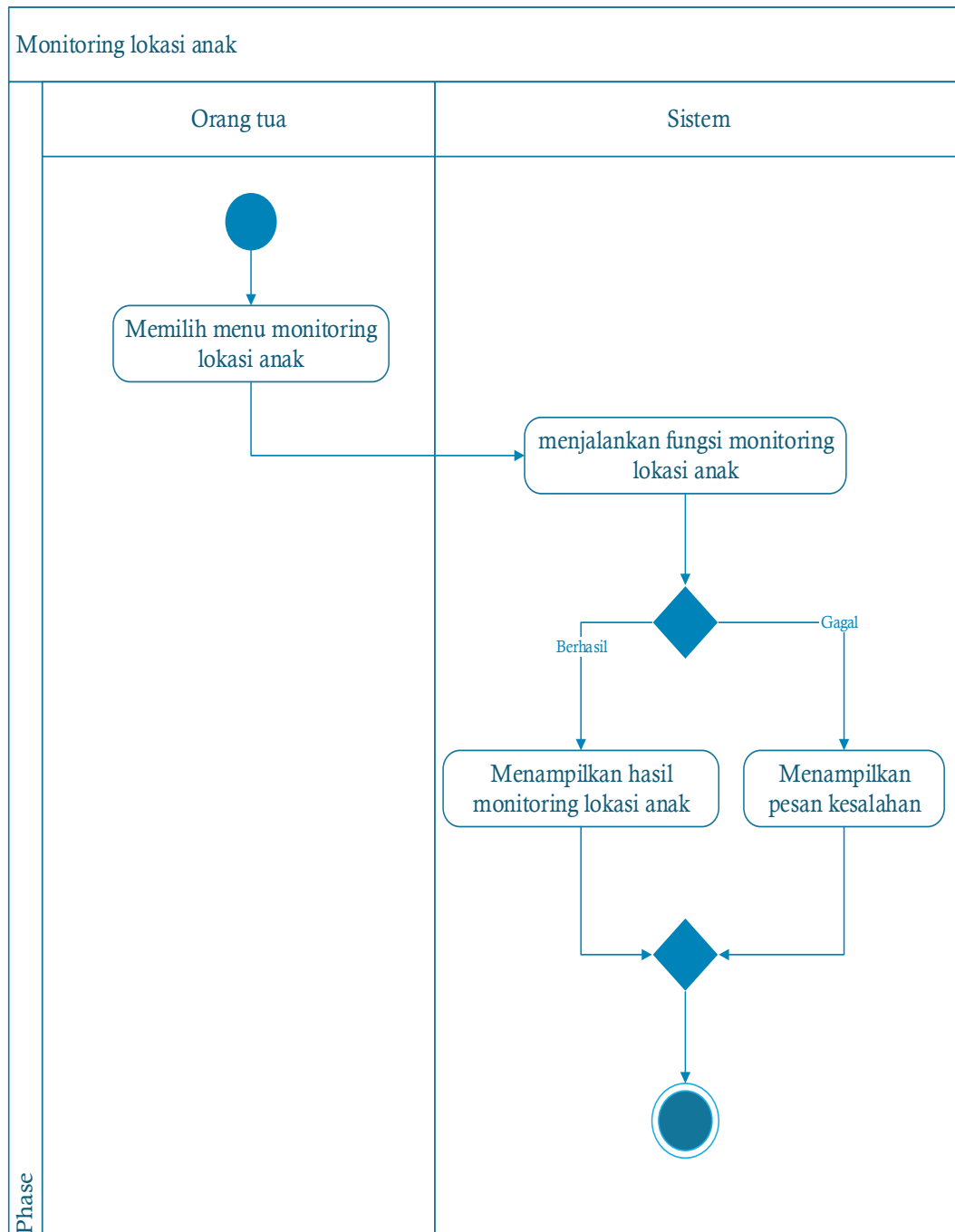
Activity diagram monitoring history browser menjelaskan aliran kerja sistem dalam menjalankan *monitoring history browser*. Activity diagram monitoring history browser dapat dilihat pada Gambar 31.



Gambar 37 Activity Diagram Monitoring History Browser

11. Activity Diagram Monitoring Lokasi Anak

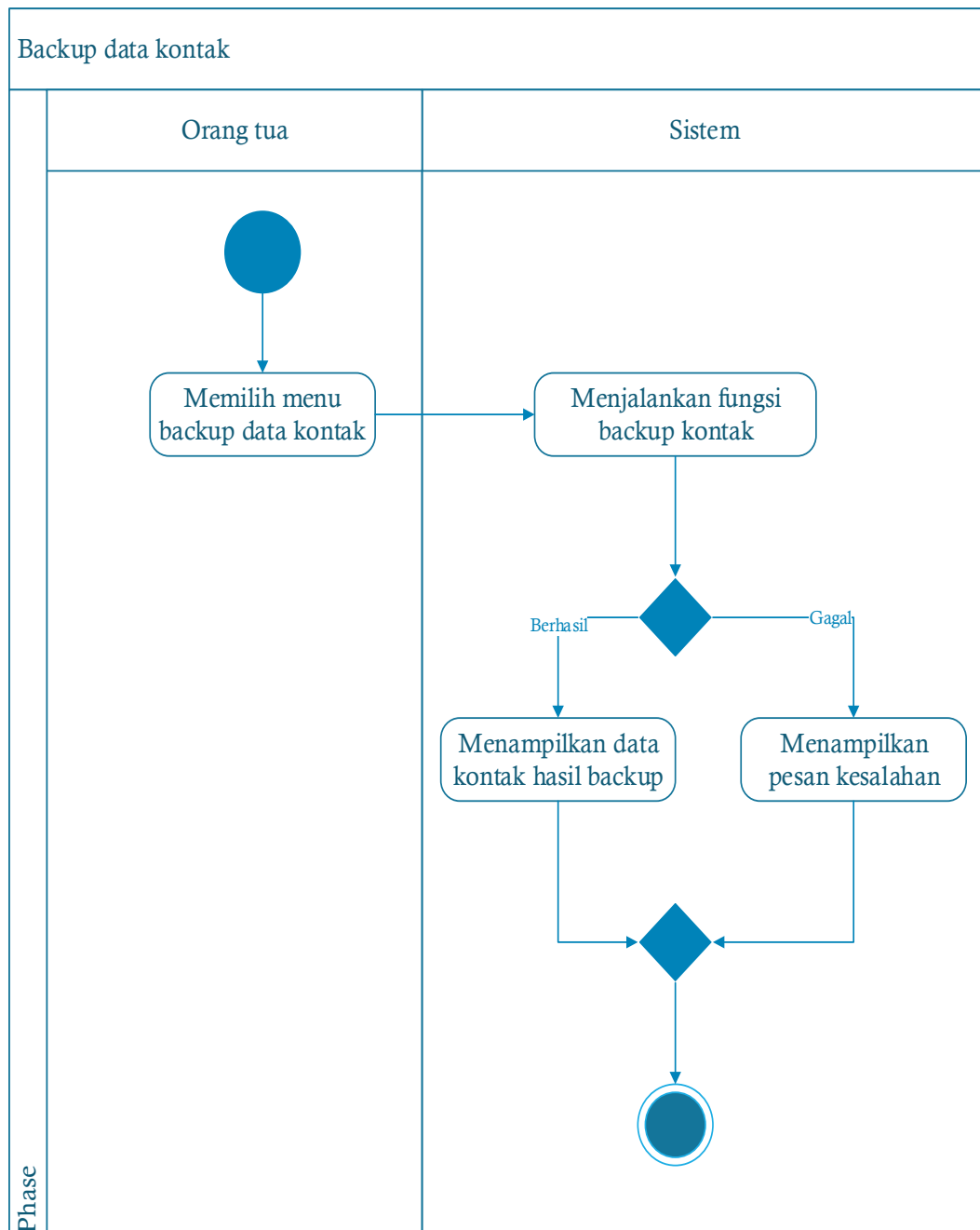
Activity diagram Activity Diagram Monitoring Lokasi Anak menjelaskan aliran kerja sistem dalam menjalankan *Monitoring Lokasi Anak*. Activity Diagram *Monitoring Lokasi Anak* dapat dilihat pada Gambar 32.



Gambar 38 Activity Diagram Monitoring Lokasi Anak

12. Activity Diagram Backup Data kontak

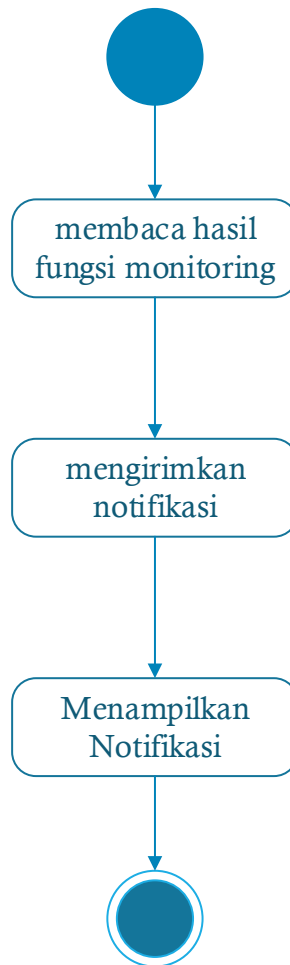
Activity diagram Backup Data kontak menjelaskan aliran kerja sistem dalam menjalankan Backup Data kontak. Activity diagram Backup Data kontak dapat dilihat pada Gambar 33.



Gambar 39 Activity Diagram Backup Data kontak

13. *Activity Diagram* Notifikasi

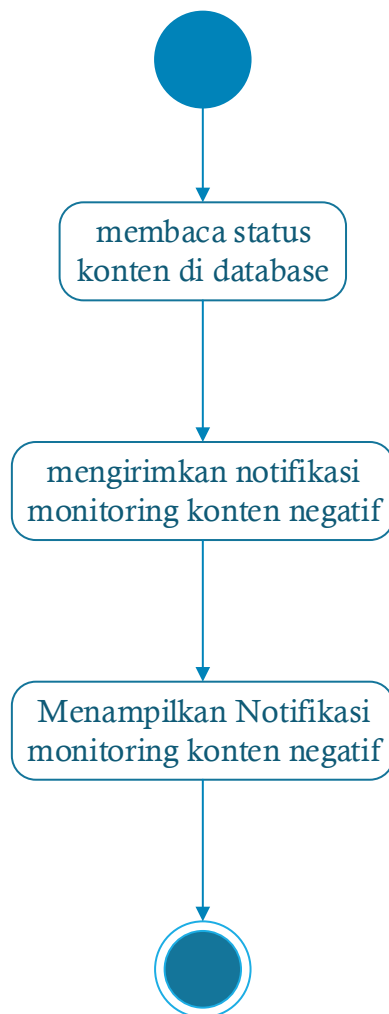
Activity diagram Notifikasi menjelaskan aliran kerja sistem dalam menjalankan Notifikasi kepada pengguna. *Activity diagram* Notifikasi dapat dilihat pada Gambar 34.



Gambar 40 *Activity Diagram* Notifikasi

14. *Activity Diagram* Notifikasi *Monitoring* Konten Negatif

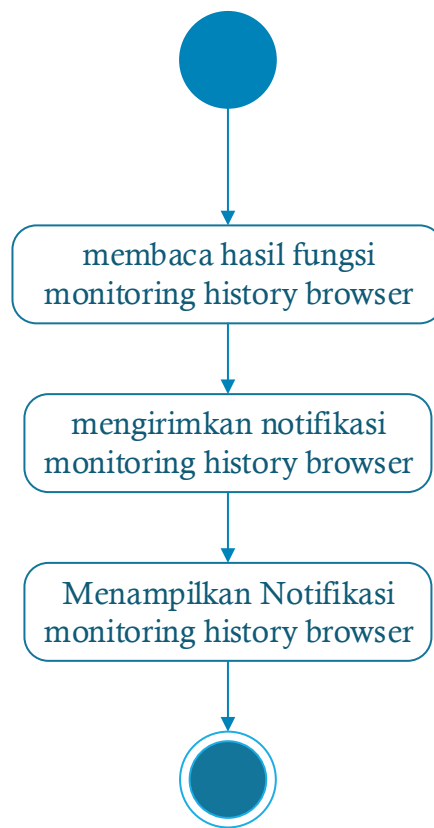
Activity diagram Notifikasi *Monitoring* Konten Negatif menjelaskan aliran kerja sistem dalam menjalankan Notifikasi *Monitoring* Konten Negatif kepada pengguna. *Activity diagram* Notifikasi *Monitoring* Konten Negatif dapat dilihat pada Gambar 35.



Gambar 41 Activity Diagram Notifikasi Monitoring Konten Negatif

15. *Activity Diagram Notifikasi Monitoring History Browser*

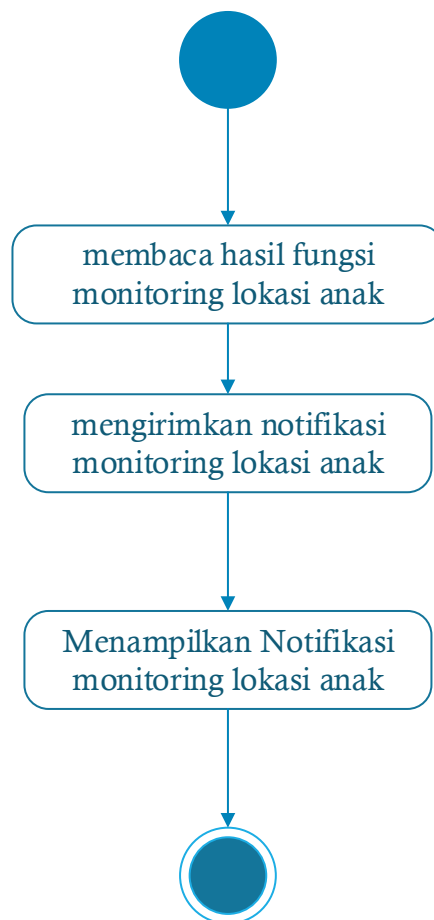
Activity diagram Notifikasi Monitoring History Browser menjelaskan aliran kerja sistem dalam menjalankan Notifikasi *Monitoring History Browser* kepada pengguna. *Activity diagram Notifikasi Monitoring History Browser* dapat dilihat pada Gambar 36.



Gambar 42 Activity Diagram Notifikasi Monitoring History Browser

16. *Activity Diagram* Notifikasi Lokasi Anak

Activity diagram notifikasi lokasi anak menjelaskan aliran kerja sistem saat menjalankan Notifikasi lokasi anak kepada pengguna. *Activity diagram* Notifikasi lokasi anak dapat dilihat pada Gambar 37.



Gambar 43 Activity Diagram Notifikasi Lokasi Anak

3.1.6.4 Class Diagram

Class Diagram dari aplikasi *monitoring* dan pengendalian konten negatif pada perangkat pintar berbasis Android menggunakan API Clarifai dapat dilihat pada Gambar 44 berikut :

Gambar 44 Aplikasi Monitoring Dan Pengendalian Konten Negatif Pada Perangkat Pintar Berbasis Android Menggunakan API Clarifai

3.1.6.5 Sequence Diagram

Sequence Diagram merupakan interaksi dari tiap objek pada setiap *use case* dalam tahapan urutan waktu. Interaksi yang berlangsung di dalam sistem ini berupa

rangkaian data antar objek-objek yang saling berinteraksi. Adapun *sequence diagram* pada aplikasi monitoring dan pengendalian konten negatif pada perangkat pintar berbasis android menggunakan API Clarifai yaitu sebagai berikut.

1. *Sequence Diagram* Daftar

Sequence diagram daftar menunjukkan rangkaian pesan yang dikirim antara objek pengguna dengan objek lainnya saat proses mendaftar untuk membuat akun. *Sequence diagram* daftar dapat dilihat pada Gambar 45

Gambar 45 *Sequence Diagram* Daftar

2. *Sequence Diagram* Lupa Password

Sequence diagram daftar menunjukkan rangkaian pesan yang dikirim antara objek pengguna dengan objek lainnya saat mengakses lupa *password* untuk dapat masuk ke aplikasi. *Sequence diagram* lupa password dapat dilihat pada Gambar 46

Gambar 46 *Sequence Diagram* Lupa Password

3. *Sequence Diagram* Login

Sequence diagram login menunjukkan rangkaian pesan yang dikirim antara objek pengguna dengan objek lainnya saat mengakses *login* untuk dapat masuk ke aplikasi. *Sequence diagram login* dapat dilihat pada Gambar 47

Gambar 47 *Sequence Diagram* Login

4. *Sequence Diagram* Log Out

Sequence diagram log out menunjukkan rangkaian pesan yang dikirim antara objek pengguna dengan objek lainnya saat mengakses *log out* untuk dapat keluar dari aplikasi. *Sequence diagram log out* dapat dilihat pada Gambar 48

Gambar 48 *Sequence Diagram* Log Out

5. *Sequence Diagram* Profil

Sequence diagram profil menunjukkan rangkaian pesan yang dikirim antara objek pengguna dengan objek lainnya saat mengakses menu profil untuk dapat melihat profil akun. *Sequence diagram* profil dapat dilihat pada Gambar 49

Gambar 49 Sequence Diagram Profil

6. Sequence Diagram Ubah Nama

Sequence diagram ubah nama menunjukkan rangkaian pesan yang dikirim antara objek pengguna dengan objek lainnya saat mengakses menu ubah nama untuk dapat melakukan fungsi ubah nama dari akun profil. *Sequence diagram* ubah nama dapat dilihat pada Gambar 50

Gambar 50 Sequence Diagram Ubah Nama

7. Sequence Diagram Ubah Password

Sequence diagram ubah *password* menunjukkan rangkaian pesan yang dikirim antara objek pengguna dengan objek lainnya saat mengakses menu ubah *password* untuk dapat melakukan fungsi ubah *password* dari akun profil. *Sequence diagram* ubah *password* dapat dilihat pada Gambar 51

Gambar 51 Sequence Diagram Ubah Password

8. Sequence Diagram Monitoring Konten Negatif

Sequence diagram monitoring konten negatif menunjukkan rangkaian pesan yang dikirim antara objek pengguna dengan objek lainnya saat mengakses menu *monitoring* konten negatif untuk dapat melakukan fungsi *monitoring* konten negatif dari *smartphone*. *Sequence monitoring* konten negatif dapat dilihat pada Gambar 52

Gambar 52 Sequence Diagram Monitoring Konten Negatif

9. Sequence Diagram Penghapusan Berkas Gambar

Sequence diagram penghapusan berkas gambar menunjukkan rangkaian pesan yang dikirim antara objek pengguna dengan objek lainnya saat menjalankan fungsi penghapusan berkas gambar. *Sequence diagram* penghapusan berkas gambar dapat dilihat pada Gambar 53

Gambar 53 Sequence Diagram Penghapusan Berkas Gambar

10. Sequence Diagram Monitoring Lokasi Anak

Sequence diagram monitoring lokasi anak menunjukkan rangkaian pesan yang dikirim antara objek pengguna dengan objek lainnya saat mengakses menu *monitoring lokasi anak* untuk dapat melakukan fungsi *monitoring lokasi anak* dari *smartphone*. *Sequence monitoring lokasi anak* dapat dilihat pada Gambar 54

Gambar 54 Sequence Diagram Monitoring Lokasi Anak

11. *Sequence Diagram Monitoring History Browser*

Sequence diagram monitoring history browser menunjukkan rangkaian pesan yang dikirim antara objek pengguna dengan objek lainnya saat mengakses menu *monitoring history browser* untuk dapat melakukan fungsi *monitoring history browser* dari *smartphone*. *Sequence monitoring history browser anak* dapat dilihat pada Gambar 55

Gambar 55 Sequence Diagram Monitoring History Browser

12. *Sequence Diagram Backup Data Kontak*

Sequence diagram backup data kontak menunjukkan rangkaian pesan yang dikirim antara objek pengguna dengan objek lainnya saat mengakses menu *backup data kontak* untuk dapat melakukan fungsi *backup data kontak* dari *smartphone*. *Sequence diagram backup data kontak anak* dapat dilihat pada Gambar 56

Gambar 56 Sequence Diagram Backup Data Kontak

13. *Sequence Diagram Notifikasi*

Sequence diagram notifikasi menunjukkan rangkaian pesan yang dikirim antara objek pengguna dengan objek lainnya saat sistem mengirimkan notifikasi sebagai pemberitahuan pada aplikasi untuk pengguna aplikasi. *Sequence diagram notifikasi* dapat dilihat pada Gambar 57

Gambar 57 Sequence Diagram Notifikasi

14. *Sequence Diagram Notifikasi Monitoring Konten Negatif*

Sequence diagram notifikasi monitoring konten negatif menunjukkan rangkaian pesan yang dikirim antara objek pengguna dengan objek lainnya saat

sistem mengirimkan notifikasi *monitoring* konten negatif sebagai pemberitahuan pada aplikasi untuk pengguna aplikasi. *Sequence diagram* notifikasi *monitoring* konten negatif dapat dilihat pada Gambar 58

Gambar 58 *Sequence Diagram* Notifikasi *Monitoring* Konten Negatif

15. *Sequence Diagram* Notifikasi *Monitoring History Browser*

Sequence diagram notifikasi *monitoring history browser* menunjukkan rangkaian pesan yang dikirim antara objek pengguna dengan objek lainnya saat sistem mengirimkan notifikasi *monitoring history browser* sebagai pemberitahuan pada aplikasi untuk pengguna aplikasi. *Sequence diagram* notifikasi *monitoring history browser* dapat dilihat pada Gambar 59

Gambar 59 *Sequence Diagram* Notifikasi *Monitoring History Browser*

16. *Sequence Diagram* Notifikasi Lokasi Anak

Sequence diagram notifikasi lokasi anak menunjukkan rangkaian pesan yang dikirim antara objek pengguna dengan objek lainnya saat sistem mengirimkan notifikasi lokasi anak sebagai pemberitahuan pada aplikasi untuk pengguna aplikasi. *Sequence diagram* notifikasi lokasi anak dapat dilihat pada Gambar 60

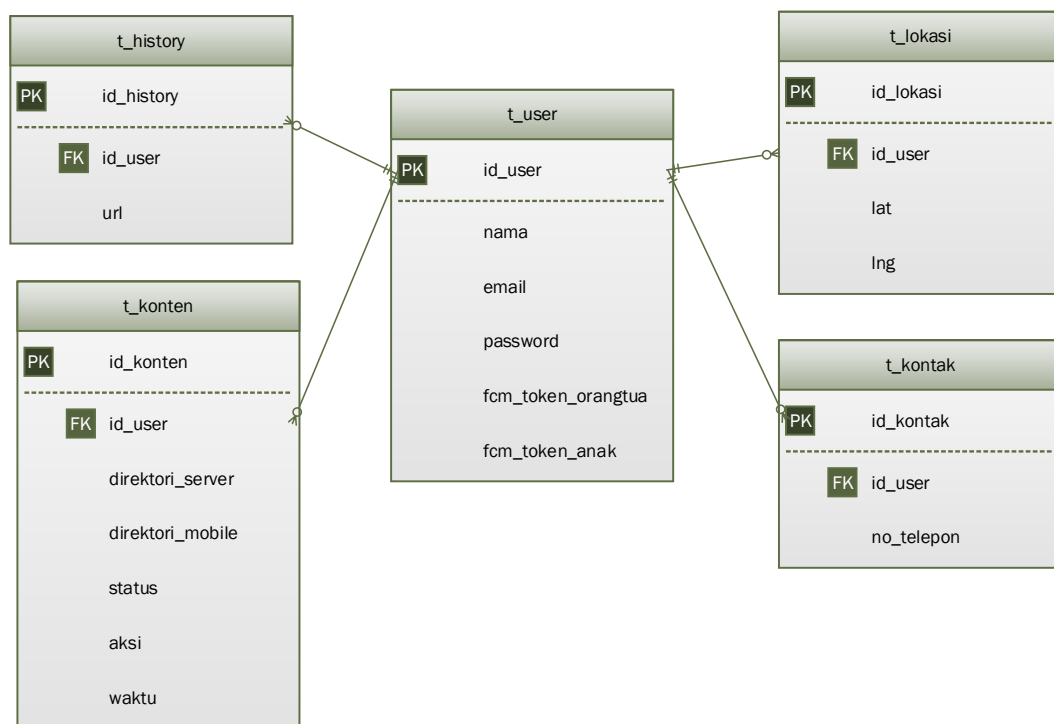
Gambar 60 *Sequence Diagram* Notifikasi Lokasi Anak

3.2 Perancangan Sistem

Perancangan sistem merupakan rancangan atau penggambaran, perencanaan, serta pembuatan dari elemen-elemen pada sistem yang terpisah kedalam satu kesatuan model yang utuh berdasar analisis sistem yang telah dilakukan sebelumnya. Tahapan perancangan dalam pembangunan aplikasi *monitoring* dan pengendalian konten negatif pada perangkat pintar berbasis Android menggunakan API Clarifai meliputi perancangan sistem relasi, struktur tabel, struktur menu, antarmuka, pesan, jaringan semantik, dan perancangan prosedural.

3.2.1 Perancangan Skema Relasi

Dalam perancangan skema relasi setiap tabel harus memiliki hubungan dengan tabel lainnya. Bila tidak ada tabel yang saling berhubungan maka dapat dipastikan pemodelan dari pembuatan basis data yang dilakukan adalah gagal. Adapun gambaran skema relasi yang ada dapat dilihat pada Gambar 61 berikut:



Gambar 61 Skema Relasi

3.2.2 Perancangan Struktur Tabel

Berikut adalah tabel-tabel yang terdapat pada basis data yang digunakan dalam pembangunan aplikasi *monitoring* dan pengendalian konten negatif pada perangkat pintar berbasis Android menggunakan API Clarifai :

1. Tabel *t_user*

Tabel *t_user* digunakan untuk menyimpan data pengguna. Struktur tabel *t_user* dapat dilihat pada Tabel 3.7 sebagai berikut.

Nama Field	Tipe Data	Panjang	Atribut	Keterangan
------------	-----------	---------	---------	------------

id_user	Int	5	<i>Primary Key</i>	
nama	Varchar	45		<i>Not null</i>
email	Varchar	45		<i>Not null</i>
password	Text			<i>Not null</i>
Fcm_token_orangtua	Varchar	225		<i>Default Not null</i>
Fcm token anak	varchar	225		<i>Default Not nul</i>

Tabel 3.26 Struktur Tabel *t_user*

2. Tabel *t_konten*

Tabel *t_konten* digunakan untuk menyimpan data konten, seperti nama serta direktori tempat konten tersimpan. Struktur tabel *t_konten* dapat dilihat pada Tabel 3.28 sebagai berikut:

Nama Field	Tipe Data	Panjang	Atribut	Keterangan
id_konten	Int	10	<i>Primary Key</i>	
Id_user	Int	11	<i>Foreign Key</i>	<i>Not null</i>
Direktori_server	Text			<i>Not null</i>
Direktori_mobile	Text			<i>Not null</i>
status	<i>enum</i>	0, 1		<i>Not null</i>
aksi	<i>enum</i>	0, 1, 2, 3		<i>Default 0</i>
waktu	<i>datetime</i>			<i>Not null</i>

Tabel 3.27 Struktur Tabel *t_konten*

3. Tabel *t_history*

Tabel *t_history* digunakan untuk menyimpan data dari *history browser* di *smartphone* anak, seperti nama serta direktori tempat konten tersimpan. Struktur tabel *t_history* dapat dilihat pada Tabel 3.29 sebagai berikut:

Nama Field	Tipe Data	Panjang	Atribut	Keterangan
id_history	Int	10	<i>Primary Key</i>	
Id_user	Int	11	<i>Foreign Key</i>	<i>Not null</i>
url	Text			<i>Not null</i>

Tabel 3.28 Struktur Tabel *t_history*

4. Tabel t_lokasi

Tabel t_lokasi digunakan untuk menyimpan data dari lokasi anak dari *smartphone* anak, seperti latitude dan longitude-nya. Struktur tabel t_lokasi dapat dilihat pada Tabel 3.29 sebagai berikut:

Nama Field	Tipe Data	Panjang	Atribut	Keterangan
id_lokasi	Int	10	<i>Primary Key</i>	
Id_user	Int	11	<i>Foreign Key</i>	<i>Not null</i>
lat	double			<i>Not null</i>
lng	double			<i>Not null</i>

Tabel 3.29 Struktur Tabel t_lokasi

5. Tabel t_kontak

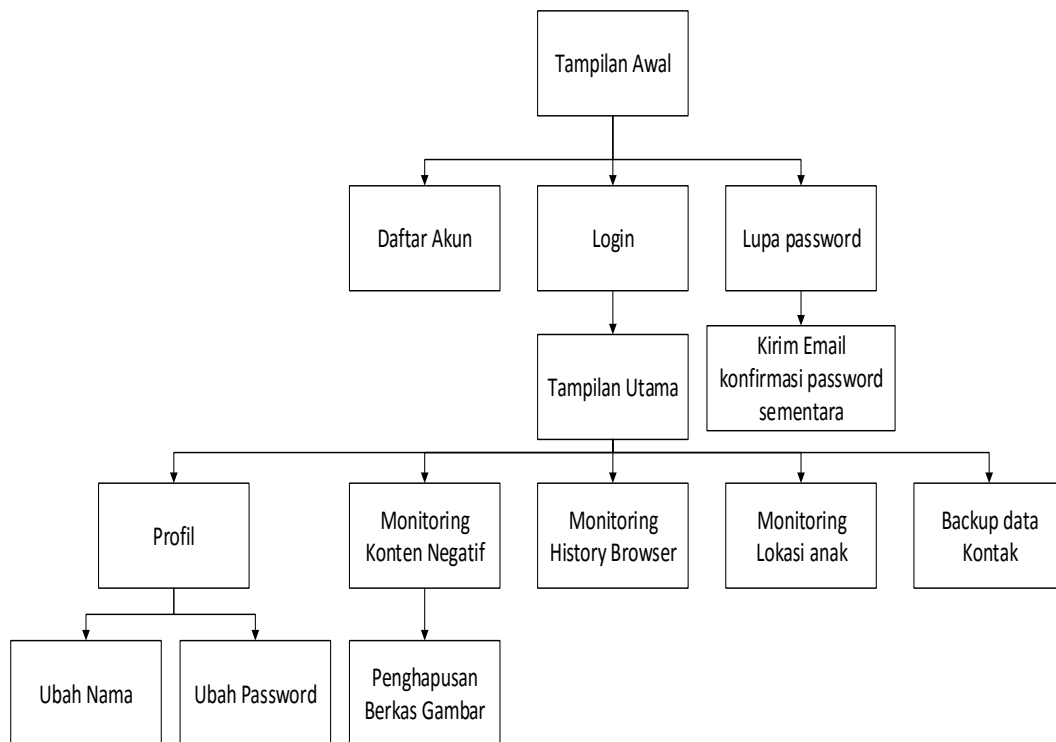
Tabel t_kontak digunakan untuk menyimpan data dari kontak pada *smartphone* anak, seperti nama dan nomor *handphone*-nya. Struktur tabel t_kontak dapat dilihat pada Tabel 3.30 sebagai berikut:

Nama Field	Tipe Data	Panjang	Atribut	Keterangan
id_kontak	Int	11	<i>Primary Key</i>	
Id_user	Int	11	<i>Foreign Key</i>	<i>Not null</i>
nama	varchar	50		<i>Not null</i>
No_telepon	varchar	15		<i>Not null</i>

Tabel 3.30 Struktur Tabel t_kontak

3.2.3 Perancangan Struktur Menu

Perancangan struktur menu merupakan gambaran dari jalur dalam pemakaian aplikasi. Perancangan struktur menu dari aplikasi *monitoring* dan pengendalian konten negatif pada perangkat pintar berbasis android menggunakan API Clarifai dapat dilihat pada gambar 62 berikut :



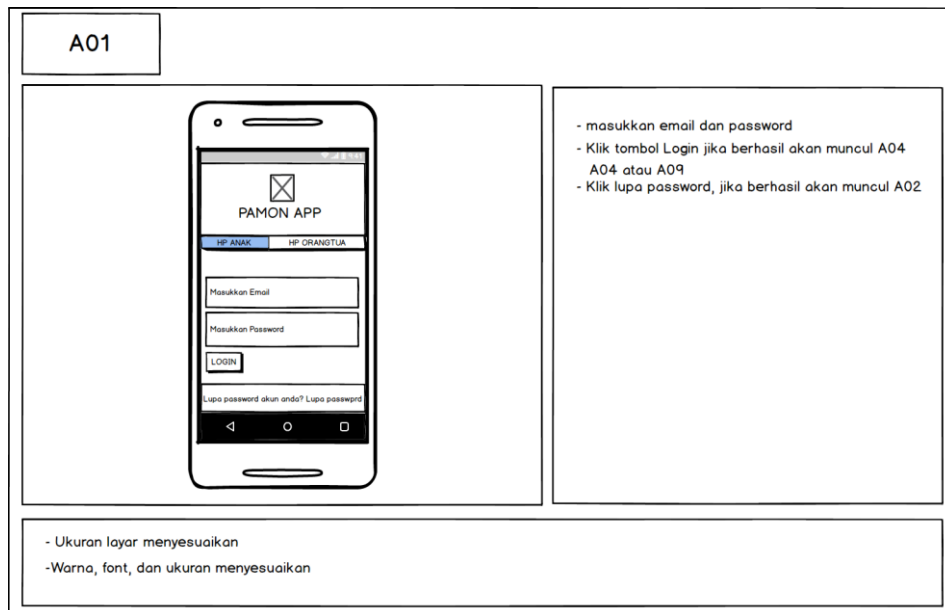
Gambar 62 Struktur Menu

3.2.4 Perancangan Antar Muka

Perancangan antarmuka bertujuan untuk memberikan gambaran tentang aplikasi yang akan dibangun sehingga akan mempermudah dalam membangun aplikasi. Pada tahap ini mendeskripsikan rencana tampilan dari setiap *form* yang akan digunakan pada tampilan aplikasi *monitoring* dan pengendalian konten negatif pada perangkat pintar berbasis Android menggunakan API Clarifai. Berikut ini gambaran dari perancangan *form* aplikasi *monitoring* dan pengendalian konten negatif pada perangkat pintar berbasis Android menggunakan API Clarifai yang dapat dilihat pada Gambar 63-72.

1. Perancangan Antarmuka *Login*

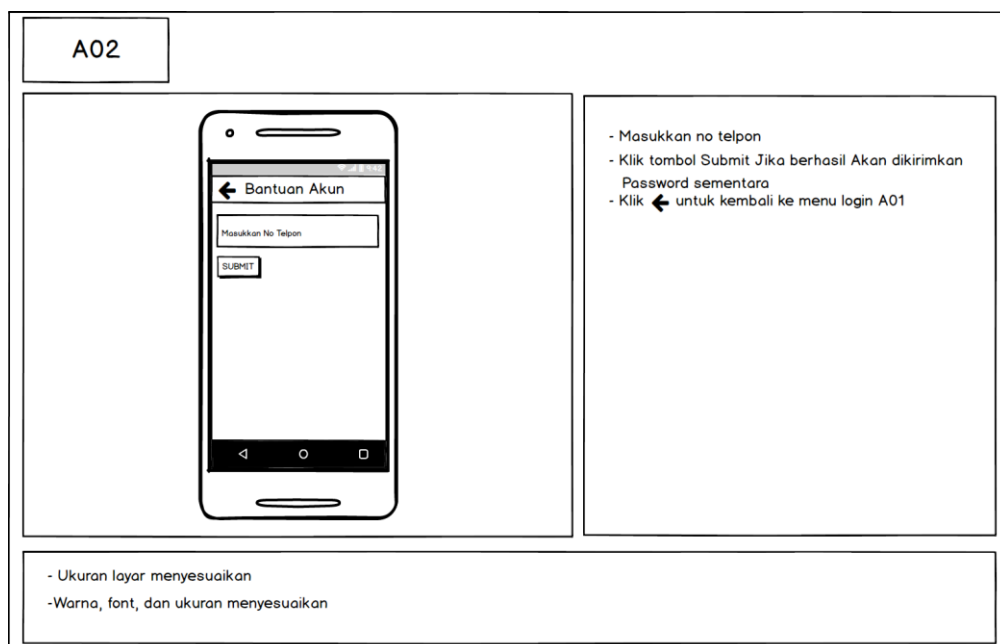
Perancangan antarmuka *login* merupakan rancangan yang akan muncul pertama kali pada saat Orangtua membuka aplikasi. Adapun perancangan antarmuka *login* dapat dilihat pada Gambar 44.



Gambar 63 Perancangan Antar Muka Login

2. Perancangan Antarmuka Lupa password

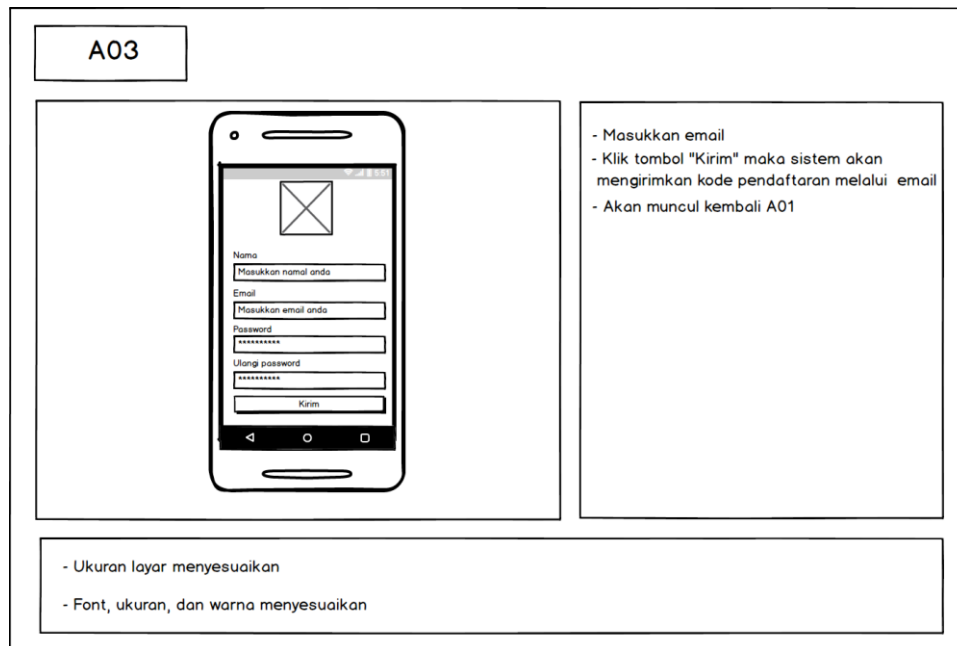
Perancangan antarmuka lupa *password* merupakan rancangan yang akan muncul saat pengguna ingin melakukan set ulang pada *password* akun yang hendak dikirim melalui email konfirmasi. Adapun perancangan antarmuka lupa *password* dapat dilihat pada Gambar 45.



Gambar 64 Perancangan Antarmuka Lupa *password*

3. Perancangan Antarmuka Daftar

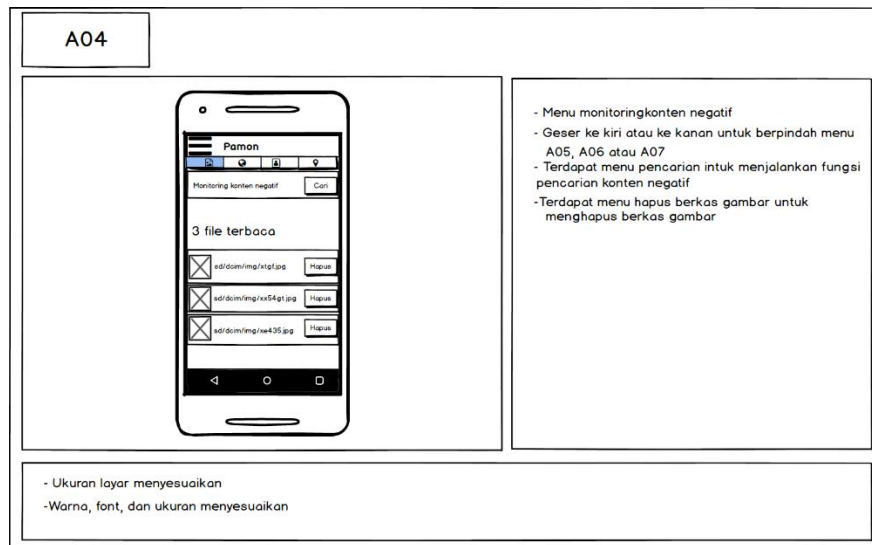
Perancangan antarmuka daftar merupakan rancangan yang akan muncul saat pengguna ingin melakukan pendaftaran atau membuat akun untuk dapat masuk ke dalam aplikasi. Adapun perancangan antarmuka daftar dapat dilihat pada Gambar 46.



Gambar 65 Perancangan Antarmuka Daftar

4. Perancangan Antarmuka *Monitoring* Konten Negatif

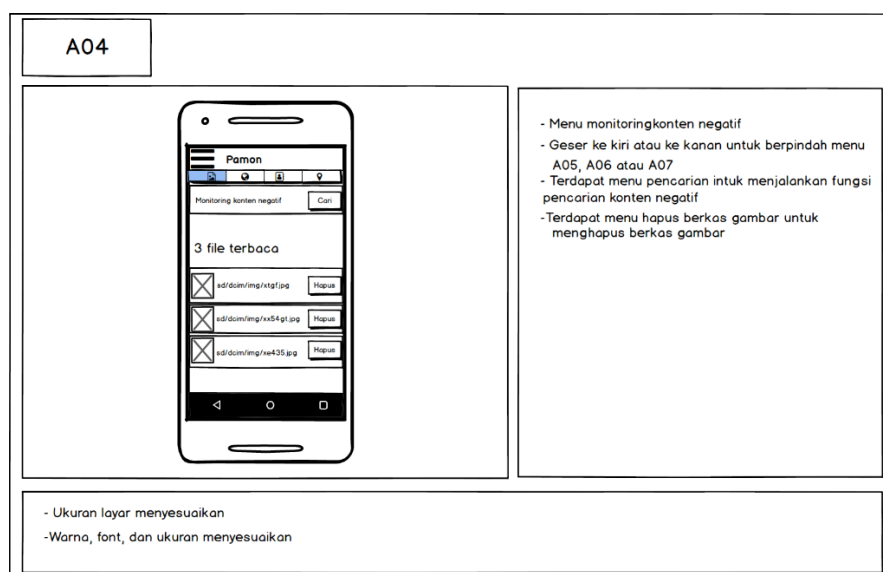
Perancangan antarmuka *Monitoring* Konten Negatif merupakan rancangan yang akan muncul saat pengguna memilih menu *Monitoring* Konten Negatif untuk dapat melihat hasil pencarian konten negatif yang nungkin terdapat di *smartphone* anak. Adapun perancangan antarmuka *Monitoring* Konten Negatif dapat dilihat pada Gambar 66.



Gambar 66 Perancangan Antarmuka *Monitoring Konten Negatif*

5. Perancangan Antarmuka Penghapusan berkas gambar

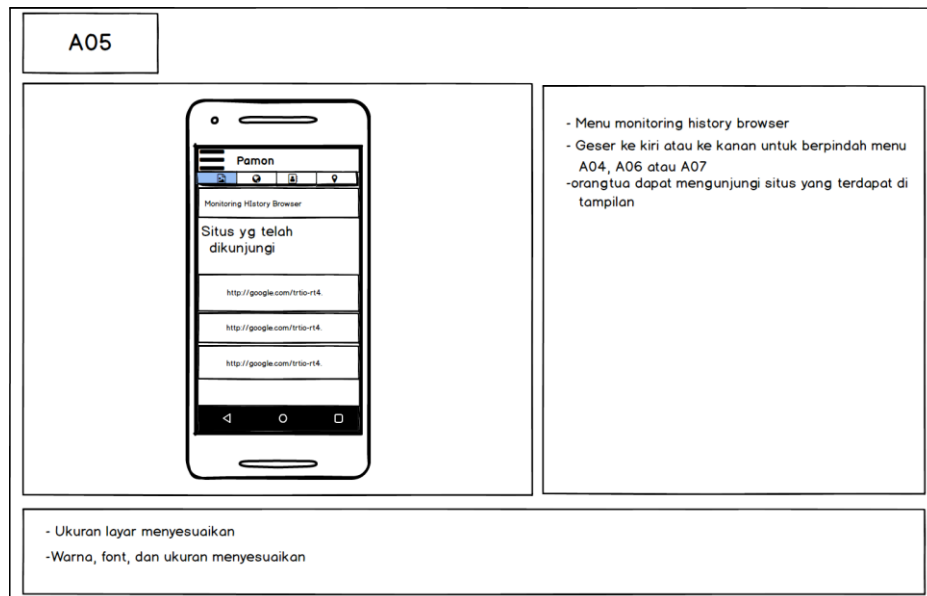
Perancangan antarmuka penghapusan berkas gambar merupakan rancangan yang akan muncul saat pengguna memilih menu penghapusan berkas gambar untuk dapat melakukan fungsi penghapusan berkas gambar pada *smartphone* anak. Menu penghapusan berkas gambar berada pada menu *monitoring konten negatif*. Adapun perancangan antarmuka penghapusan berkas gambar dapat dilihat pada Gambar 67 berikut.



Gambar 67 Perancangan Antarmuka Penghapusan berkas gambar

6. Perancangan Antarmuka *Monitoring History Browser*

Perancangan antarmuka *Monitoring History Browser* merupakan rancangan yang akan muncul saat pengguna memilih menu *Monitoring History Browser* untuk dapat melihat situs apa saja yang telah dikunjungi. Adapun perancangan antarmuka *Monitoring History Browser* dapat dilihat pada Gambar 68 berikut.



Gambar 68 Perancangan Antarmuka *Monitoring History Browser*

7. Perancangan Antarmuka *Monitoring Lokasi*

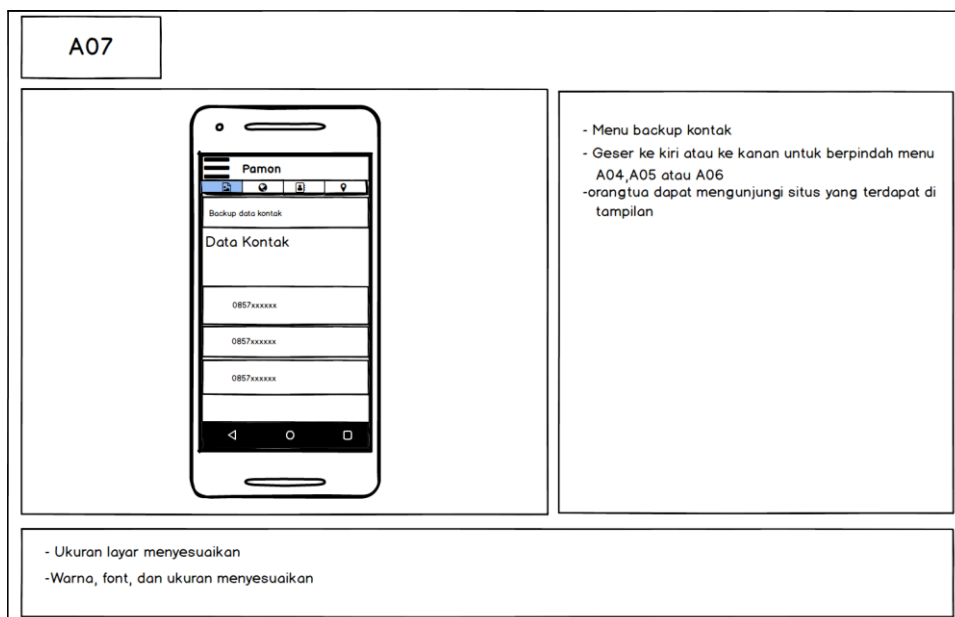
Perancangan antarmuka *Monitoring Lokasi* merupakan rancangan yang akan muncul saat pengguna memilih menu *Monitoring Lokasi* untuk dapat melihat posisi anak. Adapun perancangan antarmuka *Monitoring Lokasi* dapat dilihat pada Gambar 69 berikut.



Gambar 69 Perancangan Antarmuka *Monitoring Lokasi*

8. Perancangan Antarmuka *Backup Kontak*

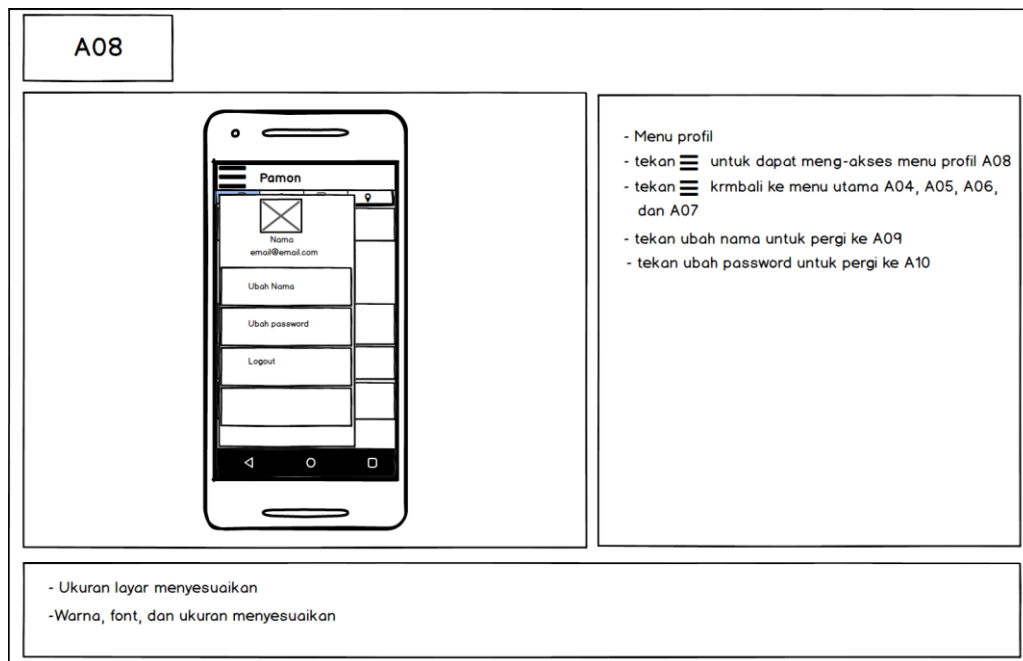
Perancangan antarmuka *backup* kontak merupakan rancangan yang akan muncul saat pengguna memilih menu *backup* kontak untuk dapat melihat kontak pada *smartphone* anak. Adapun perancangan antarmuka *backup* kontak dapat dilihat pada Gambar 70 berikut.



Gambar 70 Perancangan Antar Muka *Backup Kontak*

9. Perancangan Antarmuka Profil

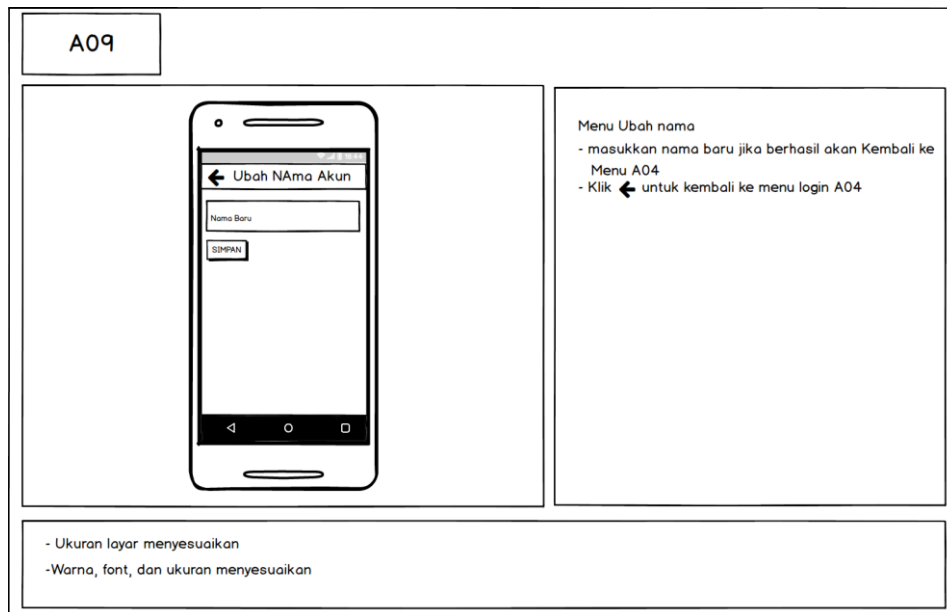
Perancangan antarmuka profil merupakan rancangan yang akan muncul saat pengguna memilih menu profil untuk dapat melihat profil dari akun. Adapun perancangan antarmuka profil dapat dilihat pada Gambar 71.



Gambar 71 Perancangan Antarmuka Profil

10. Perancangan Antarmuka Ubah nama

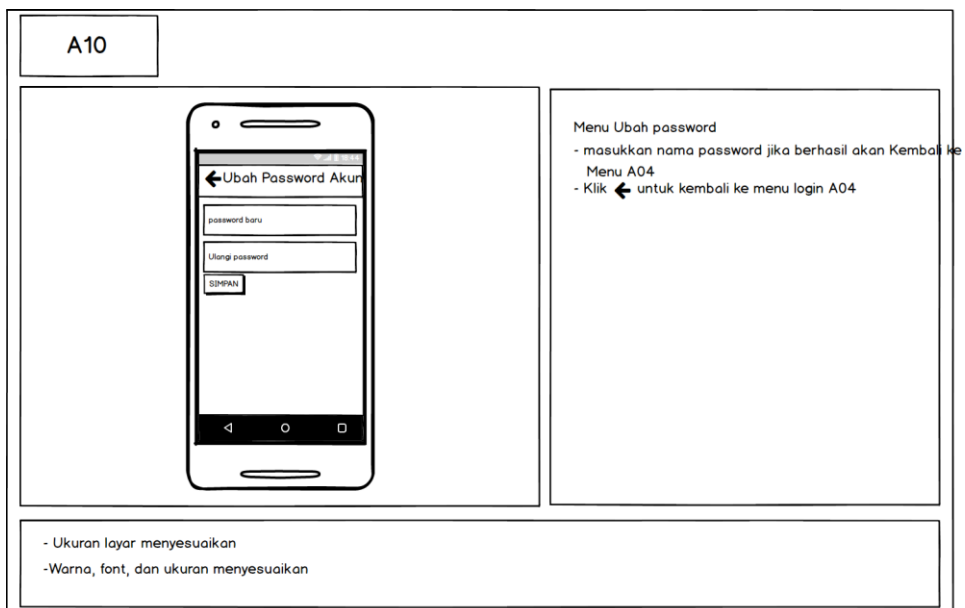
Perancangan antarmuka Ubah nama merupakan rancangan yang akan muncul saat pengguna memilih menu profil untuk dapat melihat profil dari akun kemudian memilih ubah nama untuk menyunting nama dari akun. Adapun perancangan antarmuka Ubah nama dapat dilihat pada Gambar 72.



Gambar 72 Perancangan Antar Muka Ubah Nama

11. Perancangan Antarmuka Ubah *Password*

Perancangan antarmuka Ubah *password* merupakan rancangan yang akan muncul saat pengguna memilih menu profil untuk dapat melihat profil dari akun kemudian memilih ubah *password* untuk menyunting nama dari akun. Adapun perancangan antarmuka Ubah *password* dapat dilihat pada Gambar 73.



Gambar 73 Perancangan Antarmuka Ubah *Password*

3.2.5 Perancangan Pesan

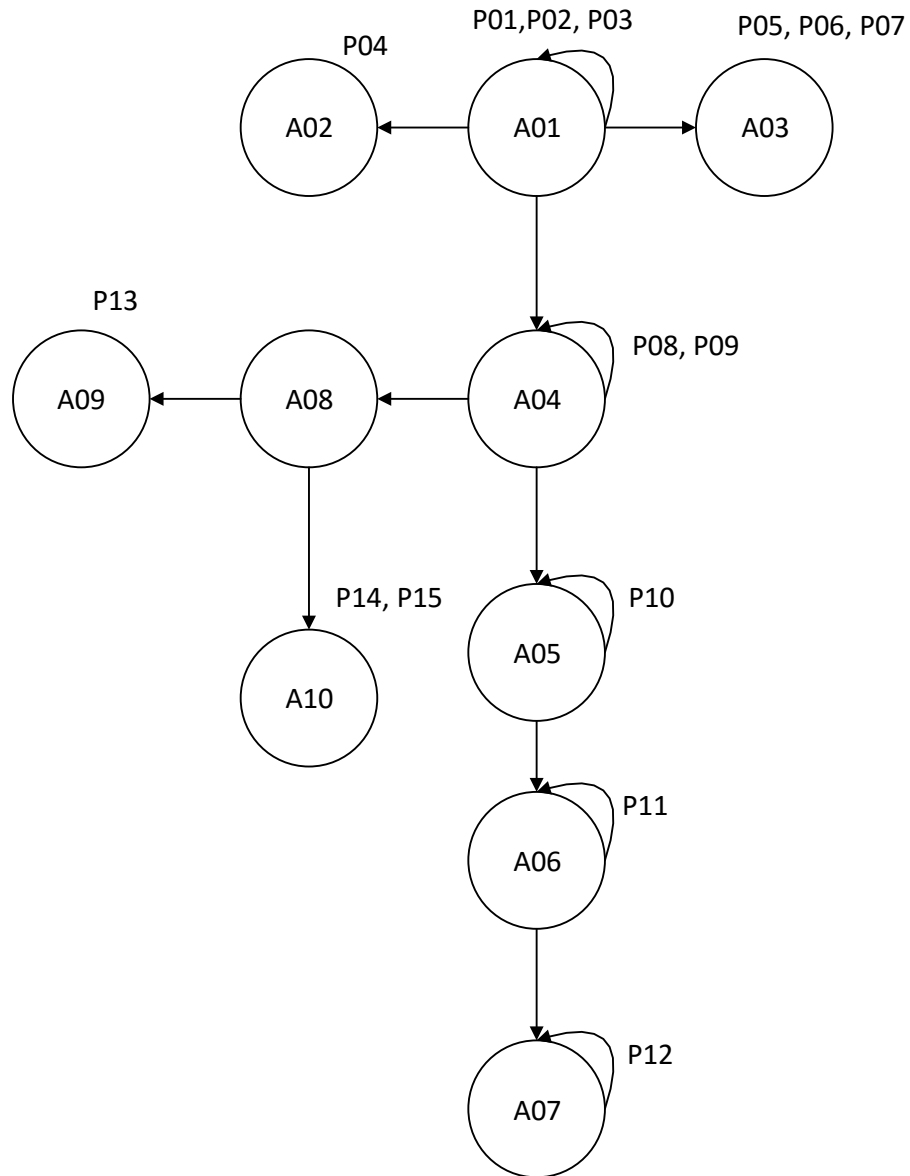
Perancangan pesan mendeskripsikan setiap tampilan pesan yang akan muncul dan digunakan pada aplikasi *monitoring* dan pengendalian konten negatif pada perangkat pintar berbasis android menggunakan API Clarifai. Berikut merupakan gambaran dari perancangan pesan pada aplikasi *monitoring* dan pengendalian konten negatif pada perangkat pintar berbasis android menggunakan API Clarifai yang dapat dilihat pada Tabel 3.30.

Tabel 3.31 Perancangan Pesan

Kode	Pesan
P01	Masukkan <i>email</i>
P02	Masukkan <i>password</i>
P03	<i>Email dan password yang anda gunakan salah</i>
P04	Masukkan No Telepon
P05	Masukkan <i>Email</i>
P06	Masukkan Nama
P07	Masukkan <i>Password</i>
P08	... <i>File</i> ditemukan
P09	Hapus gambar
P10	Situs yang telah dikunjungi
P11	Lokasi anak saat ini
P12	Data kontak
P13	Masukkan nama baru
P14	Password baru
P15	Ulangi Password

3.2.6 Perancangan Jaringan Semantik

Jaringan semantik merupakan gambaran tentang pengetahuan grafis yang menunjukkan hubungan antar objek (lingkaran) dan garis yang menggambarkan sebagai formasi antar objek tersebut yang saling berhubungan. Adapun perancangan jaringan semantik pada aplikasi *monitoring* dan pengendalian konten negatif pada perangkat pintar berbasis android menggunakan API Clarifai dapat dilihat pada Gambar 73 berikut.



Gambar 74 Jaringan Semantik

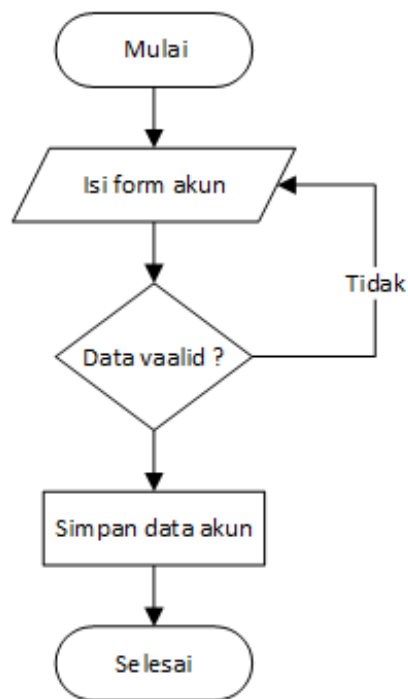
3.2.7 Perancangan Prosedural

Perancangan prosedural merupakan perancangan dari elemen-elemen struktural pada arsitektur program kedalam suatu deskripsi prosedural dari setiap komponen perangkat lunak.

Perancangan prosedural digambarkan dalam bentuk *flowchart*. Perancangan prosedural dari aplikasi *monitoring* dan pengendalian konten negatif pada perangkat pintar berbasis android menggunakan API Clarifai yaitu sebagai berikut:

1. Prosedural Daftar

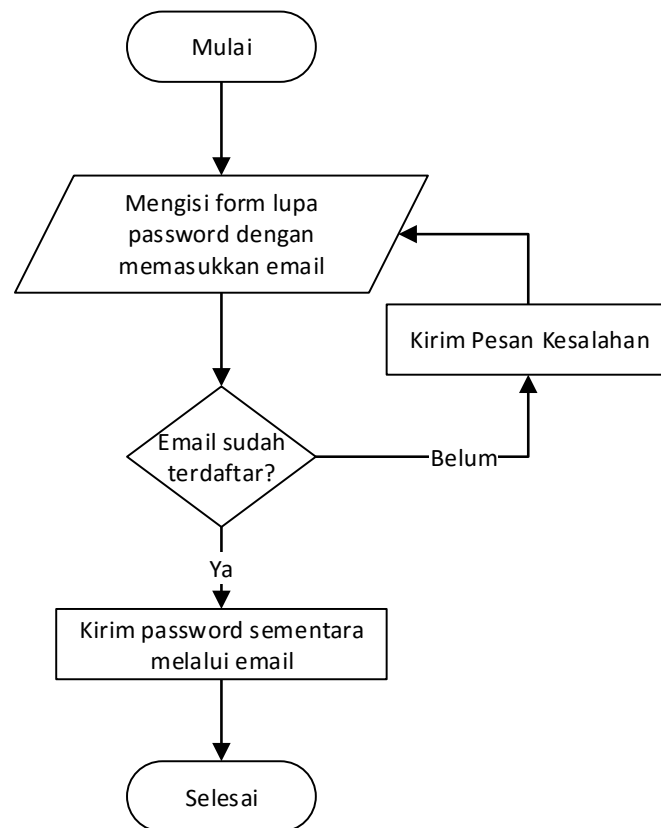
Berikut adalah perancangan prosedural daftar yang dapat dilihat pada Gambar 74.



Gambar 75 Prosedural Daftar

2. Prosedural Lupa *Password*

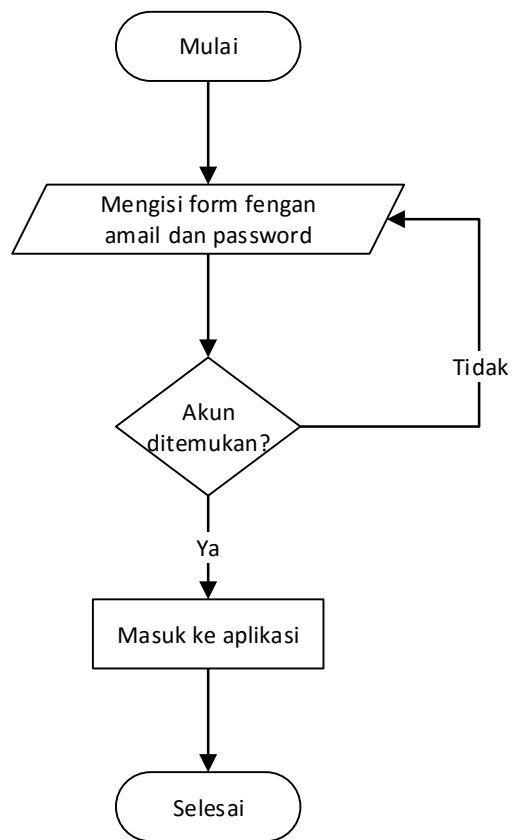
Berikut adalah perancangan prosedural Lupa *Password* yang dapat dilihat pada Gambar 75.



Gambar 76 Prosedural Lupa *Password*

3. Prosedural *Login*

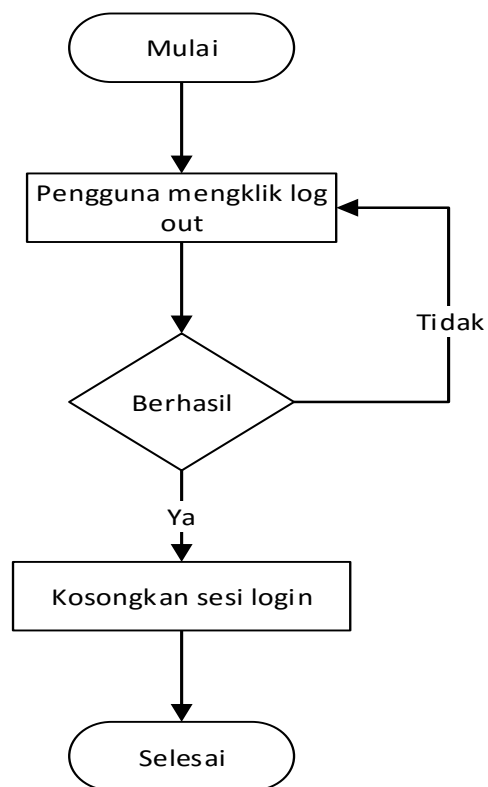
Berikut adalah perancangan prosedural *login* yang dapat dilihat pada Gambar 76.



Gambar 77 Prosedural *Login*

4. Prosedural *Log Out*

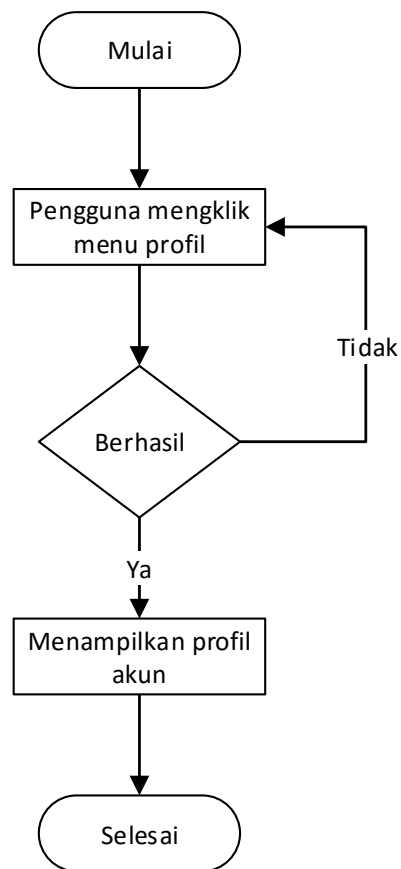
Berikut adalah perancangan prosedural *log Out* yang dapat dilihat pada Gambar 77.



Gambar 78 Prosedural *Log Out*

5. Prosedural Profil

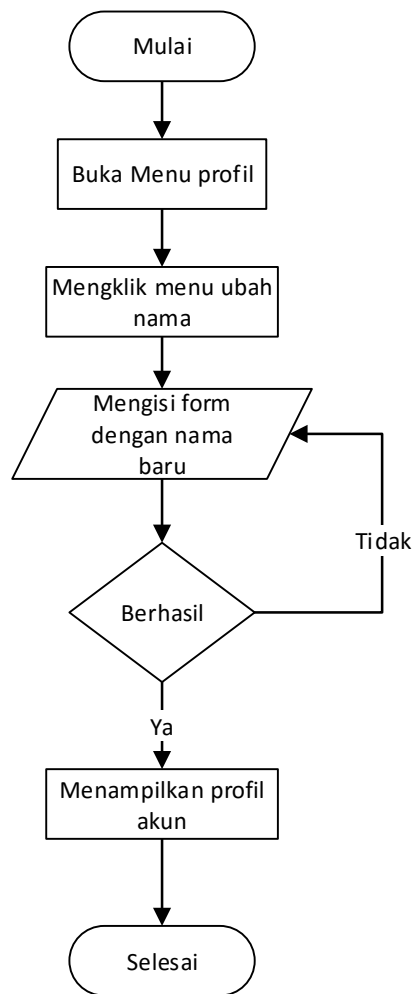
Berikut adalah perancangan prosedural profil yang dapat dilihat pada Gambar 78.



Gambar 79 Prosedural Profil

6. Prosedural Ubah Nama

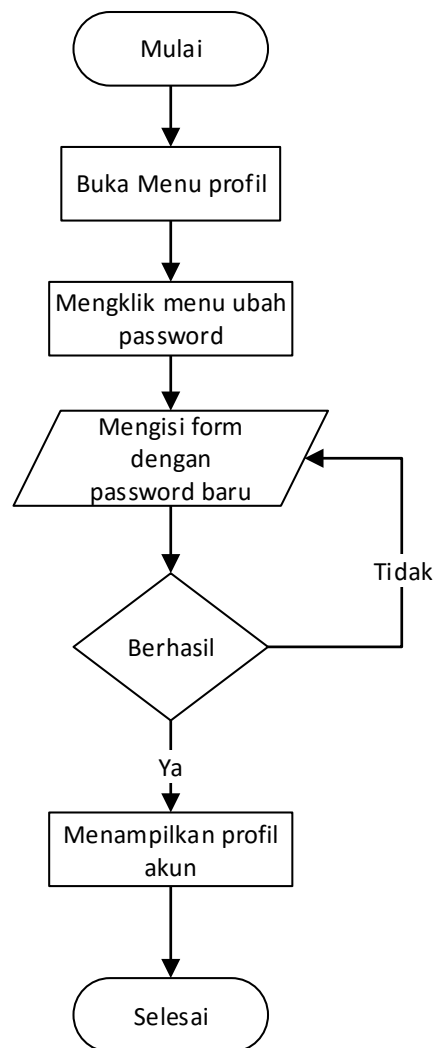
Berikut adalah perancangan prosedural Ubah Nama yang dapat dilihat pada Gambar 79.



Gambar 80 Prosedural Ubah Nama

7. Prosedural Ubah Password

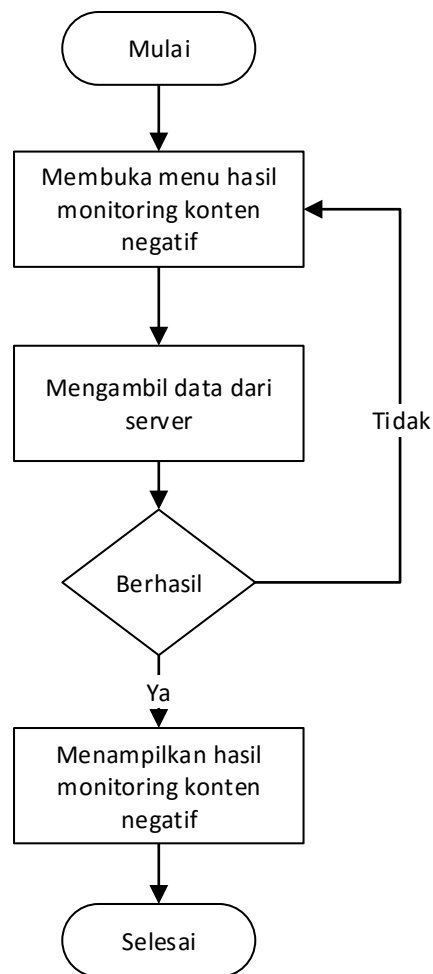
Berikut adalah perancangan prosedural ubah *password* yang dapat dilihat pada Gambar 80.



Gambar 81 Prosedural Ubah Password

8. Prosedural *Monitoring* Konten Negatif

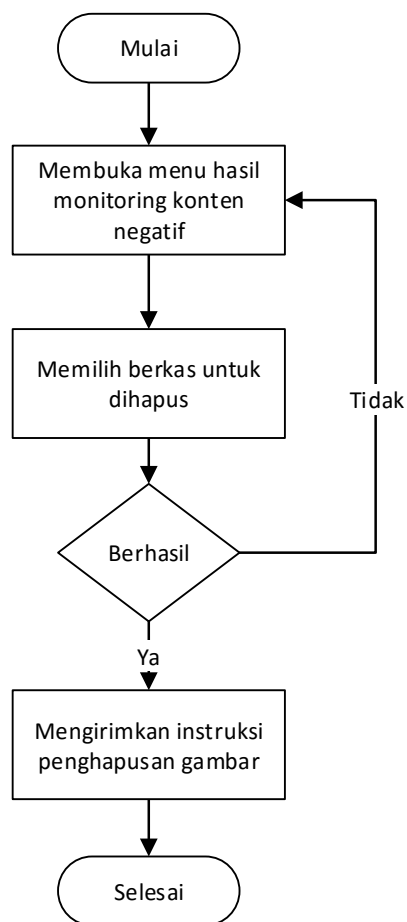
Berikut adalah perancangan prosedural *monitoring* konten negatif yang dapat dilihat pada Gambar 81.



Gambar 82 Prosedural *Monitoring* Konten Negatif

9. Prosedural Penghapusan Berkas Gambar

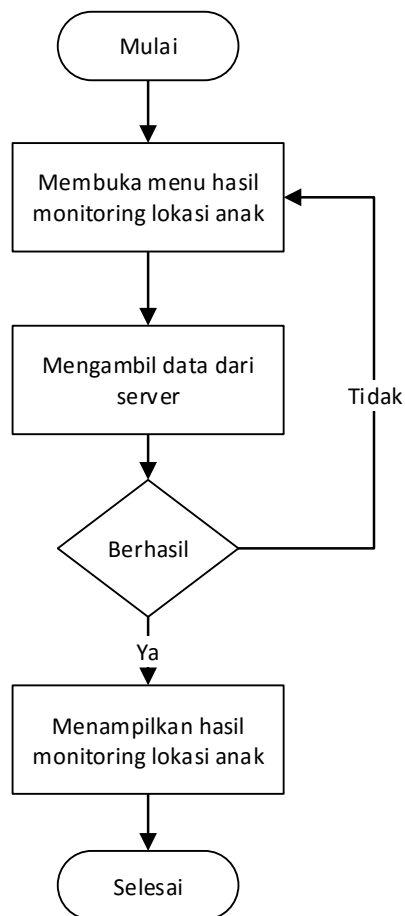
Berikut adalah perancangan prosedural penghapusan berkas gambar yang dapat dilihat pada Gambar 82.



Gambar 83 Prosedural Penghapusan Berkas Gambar

10. Prosedural *Monitoring* Lokasi Anak

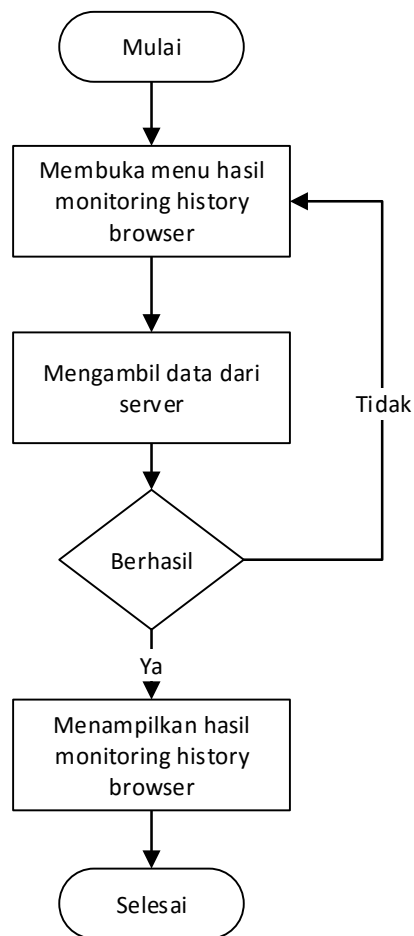
Berikut adalah perancangan prosedural *monitoring* lokasi anak yang dapat dilihat pada Gambar 83.



Gambar 84 Prosedural *Monitoring Lokasi Anak*

11. Prosedural Daftar *Monitoring History Browser*

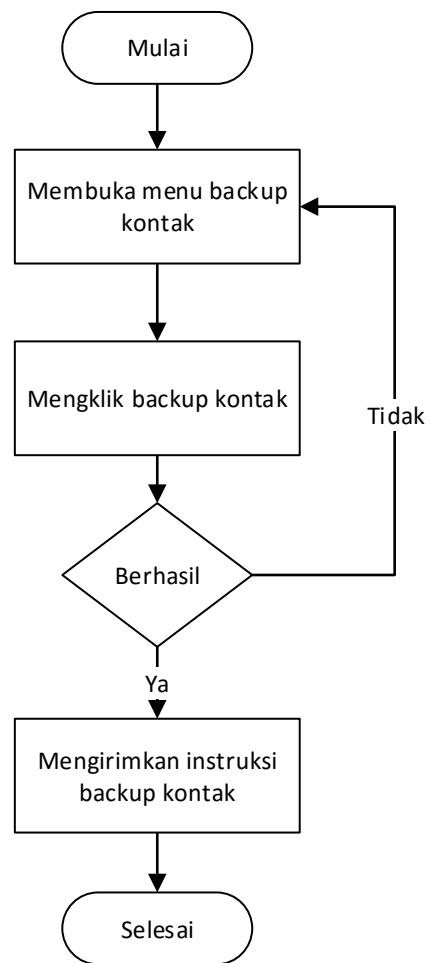
Berikut adalah perancangan prosedural *monitoring history browser* yang dapat dilihat pada Gambar 84.



Gambar 85 Prosedural *Monitoring History Browser*

12. Prosedural *Backup Data Kontak*

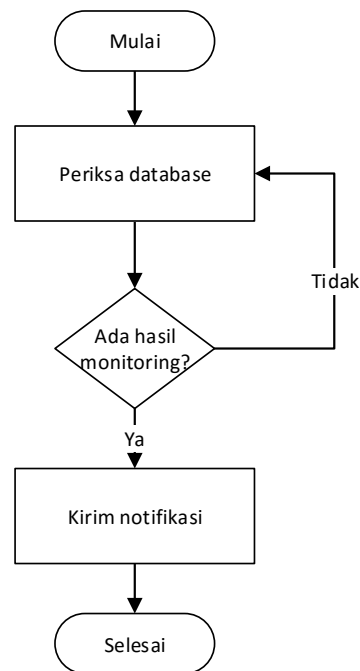
Berikut adalah perancangan prosedural *backup data* kontak yang dapat dilihat pada Gambar 85.



Gambar 86 Prosedural *Backup Data Kontak*

13. Prosedural Notifikasi

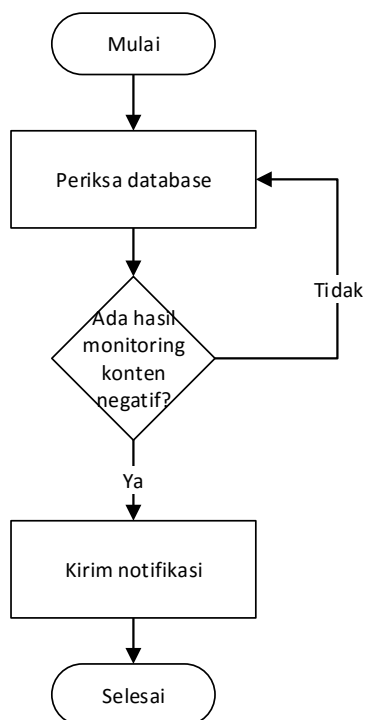
Berikut adalah perancangan prosedural Notifikasi yang dapat dilihat pada Gambar 86.



Gambar 87 Prosedural Notifikasi

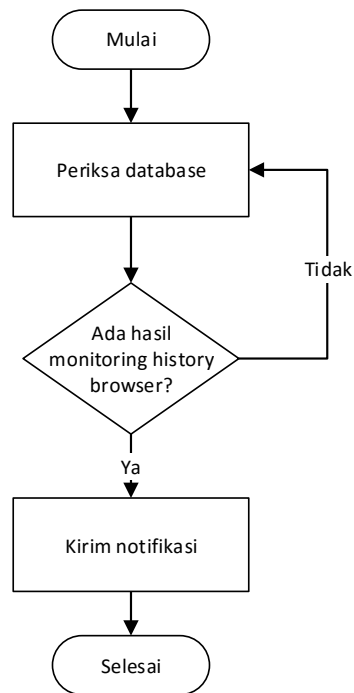
14. Prosedural Notifikasi *Monitoring* konten Negatif

Berikut adalah perancangan prosedural notifikasi *monitoring* konten negatif yang dapat dilihat pada Gambar 87.

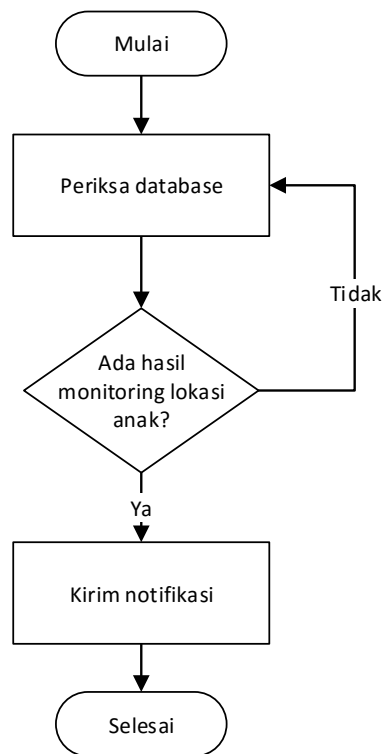


Gambar 88 Prosedural Notifikasi *Monitoring Konten Negatif***15. Prosedural Notifikasi *Monitoring History Browser***

Berikut adalah perancangan prosedural *monitoring history browser* yang dapat dilihat pada Gambar 88.

**Gambar 89 Prosedural Notifikasi *Monitoring History Browser*****16. Prosedural Notifikasi Lokasi Anak**

Berikut adalah perancangan prosedural notifikasi lokasi anak yang dapat dilihat pada Gambar 89.



Gambar 90 Prosedural Notifikasi Lokasi Anak

DAFTAR PUSTAKA

- [1] Asosiasi Penyelenggara Jasa Internet Indonesia(APJII), "Penetrasi & Perilaku Pengguna Internet Indonesia," Asosiasi Penyelenggara Jasa Internet Indonesia(APJII), Indonesia, 2016.
- [2] A. Pramiyanti, I. P. Putri and R. Nureni, "Motif Remaja Dalam Menggunakan Media Baru(Studi Pada Remaja Di Daerah Sub-Urban Kota Bandung)," *Motif Remaja dalam Menggunakan Media Baru*, vol. 6, no. 2, pp. 95-103, 2017.
- [3] R. Aditya, "Survei: 97% Remaja Indonesia Mengakses Situs Porno," Okezone, 24 September 2013. [Online]. Available: <https://techno.okezone.com/read/2013/09/24/55/870832/survei-97-remaja-indonesia-mengakses-situs-porno>. [Accessed 02 September 2017].
- [4] A. S. Dyah, P. I. Perdini and F. S. Ali, "Strategi Komunikasi Program Internet Sehat Dan Aman Kementerian Komunikasi Dan Informatika Republik Indonesia," *Jurnal Komunikasi*, vol. 8, pp. 25-35, 2016.
- [5] T. Hidayat, Mansur and Rahman, "Desain prototipe aplikasi sistem monitoring browser ponsel," *Jurnal Teknologi Informasi & Komunikasi Digital Zone*, vol. 08, pp. 43-49, 2017.
- [6] N. P. Jagtap, A. Patil, S. S. Shakil and S. Ingle, "Mobile Activity Monitoring System Using Android Spy," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 4, no. 2, pp. 158-162, 2015.
- [7] L. Agitya, "Pembangunan Aplikasi Parental Supervision Untuk Pengawasan Orangtua Terhadap Anak Berbasis Android," *Jurnal Ilmiah Komputer dan Informatika (KOMPUTA)*, vol. 20, pp. 1-8, 2016.

- [8] D. Kumar and M. A. Qadeer, "SMS Based Emerging Techniques for Monitoring and Controlling android Mobile," *IACSIT International Journal of*, vol. 4, pp. 789-802, 2012.
- [9] W. Zhang, H. He, Q. Zhang and T.-h. Kim, "Protecting User Privacy on the Android-Based Mobile Platform," *International Journal of Distributed Sensor Networks*, vol. 2014, pp. 1-10, 2014.
- [10] K. Kuppusamy, R.Senthilraja and G. Aghila, "A Model for Remote Access and Protection of Smartphones Using Short Message Service," *International Journal of Computer Science, Engineering and Information Technology (IJCSEIT)*, vol. 2, pp. 91-100, 2012.
- [11] R. k, M. C G, N. B G and N. C V, "Android Application for Vehicle Theft Prevention and Tracking System," (*IJCSIT*) *International Journal of Computer Science and Information Technologies*, vol. 5, pp. 3754-3758, 2014.
- [12] K. Punjabi, P. Bolaj, P. Mantur and S. Wali, "Bus Locator via SMS Using Android Application," (*IJCSIT*) *International Journal of Computer Science and Information Technologies*, vol. 5, pp. 1603-1603, 2014.
- [13] N. Dhawale, M. Garad and T. Darwatkar, "GPS and GPRS Based Cost Effective Human Tracking System Using Mobile Phones," *International Journal of Innovations & Advancement in Computer Science*, vol. 3, no. 4, pp. 5-10, 2014.
- [14] V. Dave and A. Welekar, "Design of an Android Application to provide Emergency Service," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 3, no. 3, pp. 1484-1488, 2011.

- [15] A. Chandran, "Smartphone Monitoring System," *International Journal of Computer Science & Engineering Technology (IJCSET)*, vol. 4, pp. 451-455, 2013.
- [16] Clarifai, "What is visual Recognition?," *Visual Recognition*, 28 September 2013. [Online]. Available: <https://www.clarifai.com/technology>. [Accessed 2 September 2017].
- [17] IDC, "Smartphone OS," IDC, May 2017. [Online]. Available: <https://www.idc.com/promo/smartphone-market-share/os>. [Accessed 03 October 2017].
- [18] j. Hartono, *Analisis Dan Disain: Sistem Informasi: Pendekatan Terstruktur Teori Dan Praktek Aplikasi Bisnis*, Yogyakarta: Andi Offset, 2001.
- [19] L. Stawek, *Effective Monitoring and Alerting*, California: O'reilly, 2012.
- [20] Kemendikbud, "Kamus Besar Bahasa Indonesia Online," [Online]. Available: <https://kbbi.kemdikbud.go.id/entri/>. [Accessed 30 October 2017].
- [21] Techopedia, "Techopedia: Smart Device," *Information and news of technology*, 21 October 2015. [Online]. Available: <https://www.techopedia.com/definition/31463/smart-device>. [Accessed 23 October 2017].
- [22] B. K. Williams and S. C. Sawyer, *Using Information Technology: A Practical Introduction To Computers & Communications*. (9th edition), Newyork: McGraw-Hill, 2011.
- [23] S. N. H, *Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android(Revisi Kedua)*, Bandung: Informatika, 2015.

- [24] Android Developer, "Platform Architecture," [Online]. Available: <https://developer.android.com/guide/platform/index.html?hl=id>. [Accessed 23 October 2017].
- [25] Android Developer, "Panduan Pengguna Android Studio," [Online]. Available: <https://developer.android.com/studio/intro/index.html?hl=id>. [Accessed 24 October 2017].
- [26] Clarifai, "Getting Started," [Online]. Available: <https://www.clarifai.com/developer/guide/#getting-started>. [Accessed 31 October 2017].
- [27] H. Mukhlisin, "Mengenal Restful Web Service?," 8 December 2015. [Online]. Available: <http://www.hafidmukhlisin.com/2015/12/08/mengenal-restful-web-service/>. [Accessed 28 October 2017].
- [28] Feridi, "Mengenal RESTful Web Services," 26 February 2016. [Online]. Available: <https://www.codepolitan.com/mengenal-restful-web-services>. [Accessed 28 October 2017].
- [29] Json.org, "Pengenalan JSON," [Online]. Available: <http://www.json.org/json-id.html>. [Accessed 24 October 2017].
- [30] J. Imaniar, A. ST MT and A. S. Khalilullah, "Aplikasi Location Based Service untuk Sistem Informasi Publikasi Acara pada Platform Android," 2011.
- [31] B. Raharjo, Belajar Otodidak MySQL, Bandung: Informatika, 2015.
- [32] L. Dwiartara, Menyelam dan Menaklukan Samdera PHP, Ilmu Website.
- [33] British Columbia Institute of Technology, "Sekilas Tentang CodeIgniter," 6 December 2015. [Online]. Available: <https://codeigniter->

id.github.io/user-guide/overview/at_a_glance.html. [Accessed 30 October 2017].

- [34] EllisLab, "About CodeIgniter," 10 February 2015. [Online]. Available: <https://ellislab.com/codeigniter>. [Accessed 30 October 2017].
- [35] Rosa A. S and M. Shalahuddin, Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek), Bandung: Informatika Bandung, 2016.
- [36] "CodeIgniter MVC Framework," [Online]. Available: http://www.w3ii.com/id/codeigniter/codeigniter_mvc_framework.html. [Accessed 30 October 2017].