**PEMBANGUNAN APLIKASI MONITORING**

**DAN PENGENDALIAN KONTEN NEGATIF**

**PADA PERANGKAT PINTAR BERBASIS ANDROID**

**MENGGUNAKAN 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](#_Toc499948641)

[Daftar Gambar iv](#_Toc499948642)

[Daftar Tabel vii](#_Toc499948643)

[Daftar Simbol ix](#_Toc499948644)

[BAB 1 1](#_Toc499948645)

[PENDAHULUAN 1](#_Toc499948646)

[1.1 Latar Belakang Masalah 1](#_Toc499948647)

[1.2 Identifikasi Masalah 3](#_Toc499948648)

[1.3 Maksud dan Tujuan Penelitian 4](#_Toc499948649)

[1.4 Batasan Masalah 4](#_Toc499948650)

[1.5 Metodologi Penelitian 5](#_Toc499948651)

[1.5.1 Identifikasi Masalah dan Perumusan masalah 6](#_Toc499948652)

[1.5.2 Pengumpulan Data 7](#_Toc499948653)

[1.5.3 Analisa dan Perancangan Sistem 7](#_Toc499948654)

[1.5.4 Perancangan Sistem 8](#_Toc499948655)

[1.5.5 Pembangunan Sistem 9](#_Toc499948656)

[1.5.6 Pengujian Sistem 9](#_Toc499948657)

[1.5.7 Penarikan Kesimpulan 9](#_Toc499948658)

[1.6 Sistematika Penulisan 9](#_Toc499948659)

[BAB 2 11](#_Toc499948660)

[Landasan teori 11](#_Toc499948661)

[2.1 Aplikasi 11](#_Toc499948662)

[2.2 Monitoring 11](#_Toc499948663)

[2.3 Konten Negatif 12](#_Toc499948664)

[2.4 Perangkat Pintar 12](#_Toc499948665)

[2.5 Smartphone 13](#_Toc499948666)

[2.6 Android 13](#_Toc499948667)

[2.7 Arsitektur Android 15](#_Toc499948668)

[2.8 Komponen Aplikasi Android 18](#_Toc499948669)

[2.9 Android SDK(Software Development Kit) 19](#_Toc499948670)

[2.10 Android Studio 20](#_Toc499948671)

[2.11 Bahasa Pemrograman Java 22](#_Toc499948672)

[2.12 Clarifai 23](#_Toc499948673)

[2.12.1 Model Clarifai 24](#_Toc499948674)

[2.12.2 Authentication 24](#_Toc499948675)

[2.12.3 Predict 25](#_Toc499948676)

[2.12.4 Train 25](#_Toc499948677)

[2.13 Web Service 26](#_Toc499948678)

[2.14 Representational State Transfer (REST) 26](#_Toc499948679)

[2.15 JavaScript Object Notation (JSON) 28](#_Toc499948680)

[2.16 Location Base Service (LBS) 32](#_Toc499948681)

[2.17 Mysql 34](#_Toc499948682)

[2.18 Structured Query Language (SQL) 35](#_Toc499948683)

[2.19 Personal Home Page (PHP) 39](#_Toc499948684)

[2.20 CodeIgniter (CI) 42](#_Toc499948685)

[2.21 Unified Modeling language (UML) 47](#_Toc499948686)

[2.21.1 Use Case Diagram 47](#_Toc499948687)

[2.21.2 Class Diagram 47](#_Toc499948688)

[2.21.3 Sequence Diagram 48](#_Toc499948689)

[2.21.4 Activity Diagram 48](#_Toc499948690)

[BAB 3 49](#_Toc499948691)

[Analisa dan perancangan sistem 49](#_Toc499948692)

[3.1 Analisis Sistem 49](#_Toc499948693)

[3.1.1 Analisis Masalah 49](#_Toc499948694)

[3.1.2 Analisis Arsitektur Sistem 50](#_Toc499948695)

[3.1.3 Analisis Teknologi 51](#_Toc499948696)

[3.1.4 Analisis Kebutuhan Spesifikasi Perangkat Lunak 62](#_Toc499948697)

[3.1.5 Analisis Kebutuhan Non Fungsional 63](#_Toc499948698)

[3.1.6 Analisis Kebutuhan Fungsional 65](#_Toc499948699)

[3.2 Perancangan Sistem 117](#_Toc499948700)

[3.2.1 Perancangan Skema Relasi 117](#_Toc499948701)

[3.2.2 Perancangan Struktur Tabel 118](#_Toc499948702)

[3.2.3 Perancangan Struktur Menu 120](#_Toc499948703)

[3.2.4 Perancangan Antar Muka 121](#_Toc499948704)

[3.2.5 Perancangan Pesan 129](#_Toc499948705)

[3.2.6 Perancangan Jaringan Semantik 129](#_Toc499948706)

[3.2.7 Perancangan Prosedural 130](#_Toc499948707)

[Daftar Pustaka 146](#_Toc499948708)

Daftar Gambar

[Gambar 1 Kerangka Kerja Penelitian 6](#_Toc499948847)

[Gambar 2 Arsitektur Android 15](#_Toc499948848)

[Gambar 3 File Proyek di Tampilan Android 21](#_Toc499948849)

[Gambar 4 Konsep kerja Clarifai 23](#_Toc499948850)

[Gambar 5 Train 26](#_Toc499948851)

[Gambar 6 JSON Object 30](#_Toc499948852)

[Gambar 7 JSON Array 30](#_Toc499948853)

[Gambar 8 Value JSON 30](#_Toc499948854)

[Gambar 9 String JSON 31](#_Toc499948855)

[Gambar 10 JSON Angka 32](#_Toc499948856)

[Gambar 11 LBS Sebagai simpang tiga teknologi 33](#_Toc499948857)

[Gambar 12 Komponen dasar LBS 34](#_Toc499948858)

[Gambar 13 Logo CodeIgniter 43](#_Toc499948859)

[Gambar 14 Konsep Model View Controller(MVC) 45](#_Toc499948860)

[Gambar 15 model sistem 50](#_Toc499948861)

[Gambar 16 Membuat Projek Baru di Situs Clarifai 52](#_Toc499948862)

[Gambar 17 Mendapatkan Autentikasi Key 52](#_Toc499948863)

[Gambar 18 Gambar menu Explorer di Clarifai 53](#_Toc499948864)

[Gambar 19 Tampilan Projek yang telah dibuat di menu ecplorer 53](#_Toc499948865)

[Gambar 20 Membuat Konsep baru di Clarifai 54](#_Toc499948866)

[Gambar 21 Menambahkan Citra pada Concept yang telah dibuat 54](#_Toc499948867)

[Gambar 22 Membuat Model Baru Calarifai 55](#_Toc499948868)

[Gambar 23 Training Pada Model 55](#_Toc499948869)

[Gambar 24 Model id Clarifai 56](#_Toc499948870)

[Gambar 25 Pengujian API Clarifai dengan Library Java 56](#_Toc499948871)

[Gambar 26 Response dari Library Api Clarifai dari Java 57](#_Toc499948872)

[Gambar 27 Tampilan Default CodeIgniter Setelah di Install 59](#_Toc499948873)

[Gambar 28 Tampilan Default CodeIgniter Setelah di Install Library REST 60](#_Toc499948874)

[Gambar 29 Use Case Diagram 66](#_Toc499948875)

[Gambar 30 Activity Diagram Daftar 84](#_Toc499948876)

[Gambar 31 Activity Diagram Lupa password 85](#_Toc499948877)

[Gambar 32 Activity Diagram Login 86](#_Toc499948878)

[Gambar 33 Activity Diagram Log Out 87](#_Toc499948879)

[Gambar 34 Activity Diagram Profil 88](#_Toc499948880)

[Gambar 35 Activity Diagram Ubah nama 89](#_Toc499948881)

[Gambar 36 Activity Diagram Ubah Password 90](#_Toc499948882)

[Gambar 37 Activity Diagram Monitoring Konten Negatif 91](#_Toc499948883)

[Gambar 38 Activity Diagram Penghapusan Berkas Gambar 92](#_Toc499948884)

[Gambar 39 Activity Diagram Monitoring History Browser 93](#_Toc499948885)

[Gambar 40 Activity Diagram Monitoring Lokasi Anak 94](#_Toc499948886)

[Gambar 41 Activity Diagram Backup Data kontak 95](#_Toc499948887)

[Gambar 42 Activity Diagram Notifikasi 96](#_Toc499948888)

[Gambar 43 Activity Diagram Notifikasi Monitoring Konten Negatif 97](#_Toc499948889)

[Gambar 44 Activity Diagram Notifikasi Monitoring History Browser 98](#_Toc499948890)

[Gambar 45 Activity Diagram Notifikasi Lokasi Anak 99](#_Toc499948891)

[Gambar 46 Aplikasi Monitoring Dan Pengendalian Konten Negatif Pada Perangkat Pintar Berbasis Android Menggunakan API Clarifai 100](#_Toc499948892)

[Gambar 47 ActivityMain 101](#_Toc499948893)

[Gambar 48 AdapterViewPagerPublic 101](#_Toc499948894)

[Gambar 49 FragmentPublicA 102](#_Toc499948895)

[Gambar 50 FragmentPublicB 102](#_Toc499948896)

[Gambar 51 AdapterViewPagerPrivate 103](#_Toc499948897)

[Gambar 52 FragmentPrivateA 103](#_Toc499948898)

[Gambar 53 FragmentPrivateB 104](#_Toc499948899)

[Gambar 54 History 104](#_Toc499948900)

[Gambar 55 FragmentPrivateC 105](#_Toc499948901)

[Gambar 56 Kontak 105](#_Toc499948902)

[Gambar 57 FragmentPrivateD 106](#_Toc499948903)

[Gambar 58 Lokasi 106](#_Toc499948904)

[Gambar 59 ActivityLogin 107](#_Toc499948905)

[Gambar 60 ActivityAnak 107](#_Toc499948906)

[Gambar 61 AdapterViewPagerLogin 107](#_Toc499948907)

[Gambar 62 FragmentLoginA 107](#_Toc499948908)

[Gambar 63 FragmentLoginB 108](#_Toc499948909)

[Gambar 64 HelperMan 108](#_Toc499948910)

[Gambar 65 ApiNetwork 109](#_Toc499948911)

[Gambar 66 User 110](#_Toc499948912)

[Gambar 67 ResponseStatusCode 110](#_Toc499948913)

[Gambar 68 SessionUser 111](#_Toc499948914)

[Gambar 69 ActivityForgetPassword 111](#_Toc499948915)

[Gambar 70 Sequence Diagram Lupa Password 112](#_Toc499948916)

[Gambar 71 Sequence Diagram Login 113](#_Toc499948917)

[Gambar 72 Sequence Diagram Log Out 114](#_Toc499948918)

[Gambar 73 Sequence Diagram Monitoring Konten Negatif 115](#_Toc499948919)

[Gambar 74 Sequence Diagram Monitoring Lokasi Anak 116](#_Toc499948920)

[Gambar 75 Sequence Diagram Monitoring History Browser 116](#_Toc499948921)

[Gambar 76 Sequence Diagram Backup Data Kontak 117](#_Toc499948922)

[Gambar 77 Skema Relasi 118](#_Toc499948923)

[Gambar 78 Struktur Menu 121](#_Toc499948924)

[Gambar 79 Perancangan Antar Muka Login 122](#_Toc499948925)

[Gambar 80 Perancangan Antarmuka Lupa password 122](#_Toc499948926)

[Gambar 81 Perancangan Antarmuka Daftar 123](#_Toc499948927)

[Gambar 82 Perancangan Antarmuka Monitoring Konten Negatif 124](#_Toc499948928)

[Gambar 83 Perancangan Antarmuka Penghapusan berkas gambar 124](#_Toc499948929)

[Gambar 84 Perancangan Antarmuka Monitoring History Browser 125](#_Toc499948930)

[Gambar 85 Perancangan Antarmuka Monitoring Lokasi 126](#_Toc499948931)

[Gambar 86 Perancangan Antar Muka Backup Kontak 126](#_Toc499948932)

[Gambar 87 Perancangan Antarmuka Profil 127](#_Toc499948933)

[Gambar 88 Perancangan Antar Muka Ubah Nama 128](#_Toc499948934)

[Gambar 89 Perancangan Antarmuka Ubah Password 128](#_Toc499948935)

[Gambar 90 Jaringan Semantik 130](#_Toc499948936)

[Gambar 91 Prosedural Daftar 131](#_Toc499948937)

[Gambar 92 Prosedural Lupa Password 132](#_Toc499948938)

[Gambar 93 Prosedural Login 133](#_Toc499948939)

[Gambar 94 Prosedural Log Out 134](#_Toc499948940)

[Gambar 95 Prosedural Profil 135](#_Toc499948941)

[Gambar 96 Prosedural Ubah Nama 136](#_Toc499948942)

[Gambar 97 Prosedural Ubah Password 137](#_Toc499948943)

[Gambar 98 Prosedural Monitoring Konten Negatif 138](#_Toc499948944)

[Gambar 99 Prosedural Penghapusan Berkas Gambar 139](#_Toc499948945)

[Gambar 100 Prosedural Monitoring Lokasi Anak 140](#_Toc499948946)

[Gambar 101 Prosedural Monitoring History Browser 141](#_Toc499948947)

[Gambar 102 Prosedural Backup Data Kontak 142](#_Toc499948948)

[Gambar 103 Prosedural Notifikasi 143](#_Toc499948949)

[Gambar 104 Prosedural Notifikasi Monitoring Konten Negatif 144](#_Toc499948950)

[Gambar 105 Prosedural Notifikasi Monitoring History Browser 144](#_Toc499948951)

[Gambar 106 Prosedural Notifikasi Lokasi Anak 145](#_Toc499948952)

Daftar Tabel

[Tabel 2.1 Rilis Versi Android 14](#_Toc499948709)

[Tabel 3.1 Tabel Spesifikasi Kebutuhan Perangkat Lunak Fungsional 62](#_Toc499948710)

[Tabel 3.2 Tabel Spesifikasi Kebutuhan Perangkat Lunak Non-Fungsional 62](#_Toc499948711)

[Tabel 3.3 Spesifikasi Kebutuhan Perangkat Keras 63](#_Toc499948712)

[Tabel 3.4 Spesifikasi Kebutuhan Perangkat Keras 64](#_Toc499948713)

[Tabel 3.5 Spesifikasi Kebutuhan Perangkat Lunak 64](#_Toc499948714)

[Tabel 3.6 Tabel Spesifikasi Kebutuhan Perangkat Lunak 64](#_Toc499948715)

[Tabel 3.7 Karakteristik Pengguna 65](#_Toc499948716)

[Tabel 3.8 Definisi Aktor 67](#_Toc499948717)

[Tabel 3.9 Definisi Use Case 67](#_Toc499948718)

[Tabel 3.10 Skenario Use Case Daftar 68](#_Toc499948719)

[Tabel 3.11 Skenario Use Case Lupa Password 69](#_Toc499948720)

[Tabel 3.12 Skenario Use Case Login 70](#_Toc499948721)

[Tabel 3.13 Skenario Use Case Log Out 71](#_Toc499948722)

[Tabel 3.14 Skenario Use Case Profil 72](#_Toc499948723)

[Tabel 3.15 Skenario Use Case Ubah Nama 73](#_Toc499948724)

[Tabel 3.16 Skenario Use Case Ubah Password 74](#_Toc499948725)

[Tabel 3.17 Skenario Use Case Monitoring Konten Negatif 75](#_Toc499948726)

[Tabel 3.18 Skenario Use Case Penghapusan Berkas Gambar 76](#_Toc499948727)

[Tabel 3.19 Skenario Use Case Monitoring Lokasi Anak 77](#_Toc499948728)

[Tabel 3.20 Monitoring History Browser 78](#_Toc499948729)

[Tabel 3.21Backup data Kontak 79](#_Toc499948730)

[Tabel 3.22 Notifikasi 80](#_Toc499948731)

[Tabel 3.23 Notifikasi Monitoring Konten Negatif 80](#_Toc499948732)

[Tabel 3.24 Case Notifikasi monitoring History Browser 81](#_Toc499948733)

[Tabel 3.25 Notifikasi Monitoring Lokasi 82](#_Toc499948734)

[Tabel 3.26 Struktur Tabel t\_user 119](#_Toc499948735)

[Tabel 3.27 Struktur Tabel t\_konten 119](#_Toc499948736)

[Tabel 3.28 Struktur Tabel t\_history 119](#_Toc499948737)

[Tabel 3.29 Struktur Tabel t\_lokasi 120](#_Toc499948738)

[Tabel 3.30 Struktur Tabel t\_kontak 120](#_Toc499948739)

[Tabel 3.31 Perancangan Pesan 129](#_Toc499948740)

Daftar Simbol

Simbol *Use Case Diagram*

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

Simbol *Activity Diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| 1 |  | *Action* | *State* dari sistem yang mencerminkan eksekusi dari suatu aksi |
| 2 |  | *Initial Node* | Menunjukan bagaimana  *activity* diawali. |
| 3 |  | *Activity Final Node* | Menunjukan bagaimana  *activity* berakhir |
| 4 |  | *Node* | Menunjukan arah/arus dari satu *action* ke *action* selanjutnya |

Simbol *Class Diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| 1 |  | *Class* | Himpunan dari objek- objek yang berbagi atribut serta operasi yang sama. |
| 2 |  | *Association* | Apa yang menghubungkan antara objek satu dengan objek lainnya |
| 3 |  | *Composition* | Sebuah kelas tidak bisa berdiri sendiri dan harus merupakan bagian dari *class* yang lain, maka *class* tersebut memiliki relasi *composition* |
| 4 |  | *Dependency* | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri |
| 5 | *Screen Clipping* | *Direct Association* | Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan multiplicity |
| 6 | *Screen Clipping* | *Aggregation* | Relasi antar kelas dengan makna semua-bagian (Whole-part) |

Simbol *Sequence Diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| 1 |  | *Participants* | Menggambarkan bagian dari sistem yang saling berinteraksi dengan bagian lainnya |
|  |  | *Lifeline* | Menunjukan kapan suatu bagian terlibat didalam urutan rantai sekuensial |
| 2 |  | *Activation Bar* | Untuk menunjukan sebuah partisipan sedang aktif menerima atau mengirim pesan |
| 3 |  | *Synchronous Message* | Menunjukan komunikasi antar partisipan yang bersifat sinkronus |
| 4 |  | *Asynchronous Message* | Menunjukan komunikasi antar partisipan yang bersifat asinkronus |
| 5 |  | *Return Message* | Menunjukan pesan balasan komunikasi antar partisipan |

Simbol *Flowchart*

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| 1 |  | *Terminator* | Digunakan untuk memulai dan mengakhiri sebuah proses |
| 2 |  | *Process* | Digunakan untuk menunjukan sebuah proses kegiatan |
| 3 |  | *Decision* | Menentukan arah proses berikutnya berdasarkan kondisi yang ada |

# 

PENDAHULUAN

## 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 di sajikan 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 di lakukan 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 dapatmelakukan 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* anakdan 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.**

## ****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.

## 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.

## 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 :
10. Melakukan pencarian konten negatif berupa gambar.
11. Melakukan monitoring pada *history browser.*
12. Orang tua dapat mengetahui posisi anak dari gps *smartphone*.
13. Orang tua dapat mengetahui nomor kontak yang ada di *smartphone* anak.
14. Orang tua dapat melakukan tindakan penghapusan berkas pada *smartphone* anak.
15. Orangtua akan mendapat notifikasi pemberitahuan apabila anak mengakses situs yang diblok oleh provider.

## 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 Kerangka Kerja Penelitian

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

### 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.

### 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.

1. 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.

1. 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.

### 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
5. Analisis Kebutuhan Perangkat Keras
6. Analisis Kebutuhan Perangkat Lunak
7. Analisis Pengguna
8. Analisis Kebutuhan Fungsional
9. *Use Case Diagram*
10. *Use Case Scenario*
11. *Activity Diagram*
12. *Class Diagram*
13. *Sequence Diagram*

### 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.

1. Perancangan Skema Relasi
2. Perancangan Struktur Tabel
3. Perancangan Struktur Menu
4. Perancangan Antar Muka
5. Perancangan Pesan
6. Perancangan Jaringan Semantik
7. Perancangan Prosedural

### 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.

### 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

### 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.

## 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.

# 

Landasan teori

## 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.

## 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].

## 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*.

## 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.

## 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.

## 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 Aliance*, 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.

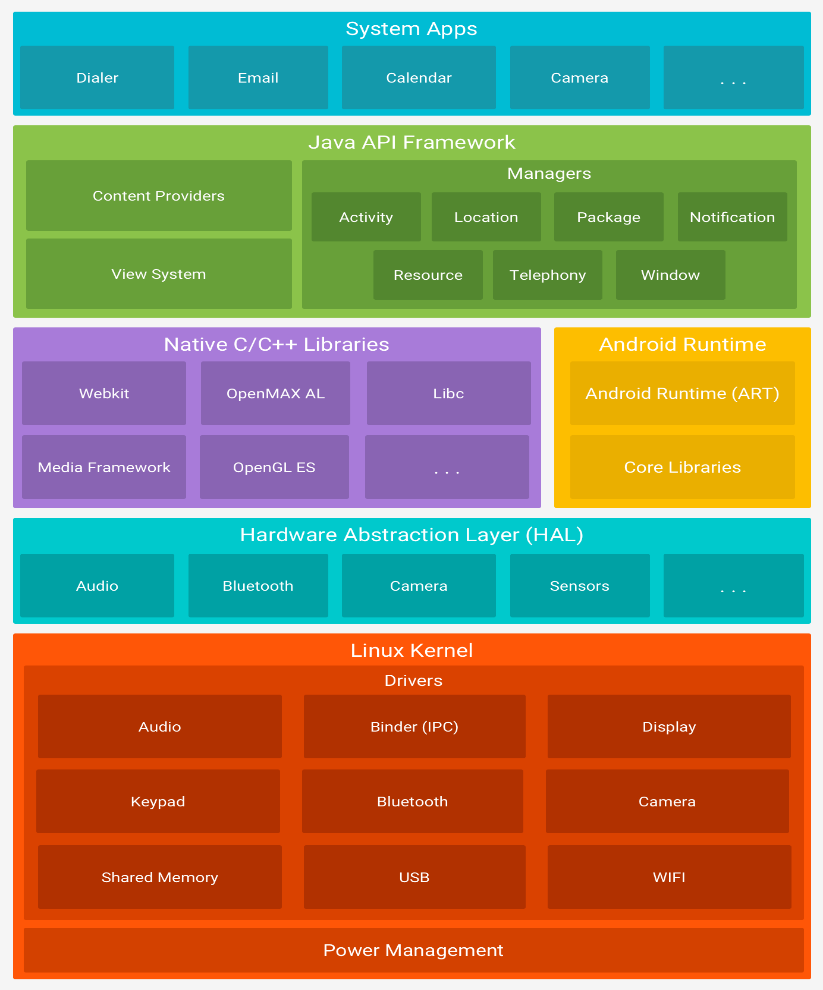
Hingga awal menjelang penghujung tahun 2017 terdapat beberapa versi Android yang tersedia untuk digunakan, diantaranya adalah sebagai berikut:

Tabel . Rilis Versi Android

|  |  |  |
| --- | --- | --- |
| **Nama Kode** | **Versi** | **APIlevel** |
| 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 |

## 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 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].

1. *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].

1. *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].

1. *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].

1. *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].

1. *Systen 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].

## 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.

1. *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*.

1. *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.

1. *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*.

## 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.

## Android Studio

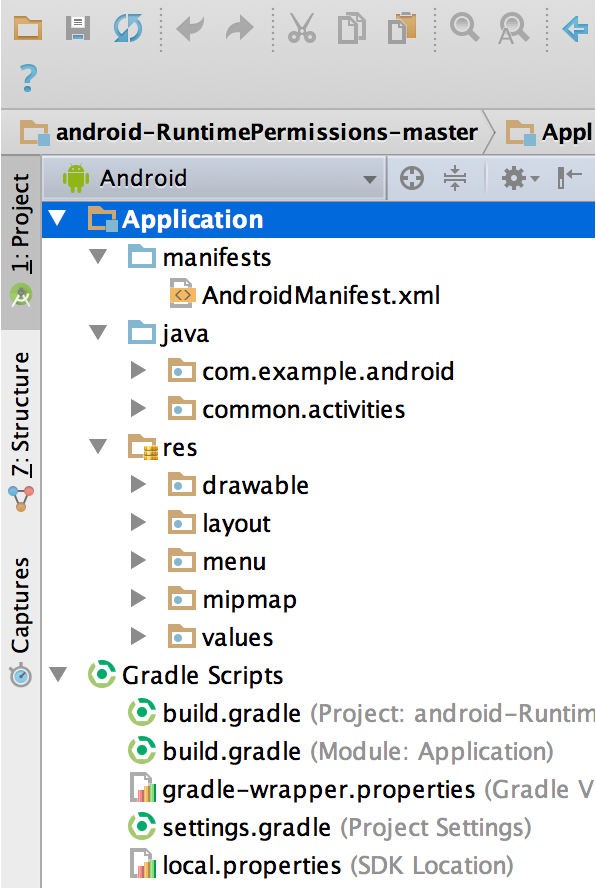
Android Studio adalah Lingkungan Pengembangan Terpadu *Integrated Development Environment* (IDE) untuk pengembangan aplikasi Android, berdasarkan IntelliJ IDEA . Selain merupakan *editor* kode IntelliJ dan alat pengembang yang berdaya guna, Android Studio menawarkan fitur lebih banyak untuk meningkatkan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

## 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, seprti 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.*

1. J2SE (Java 2 *Standard Edition*)

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

1. 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.

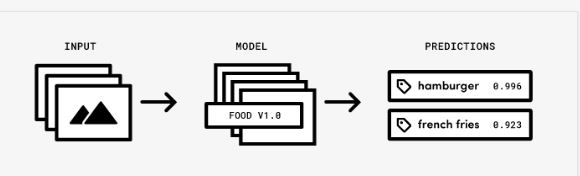
## Clarifai

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 kemungkinan nya [26].

API Clarifai dibangun dengan konsep ide dasar yang sederhana dimana anda memesukkan 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 Konsep kerja Clarifai

### 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.

1. *Face Model*

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

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

NSFW model analisis berupa gambar atau video berdasar pada besaran skor probabilitas hasilpengembalian kemunginan 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.

### Authentication

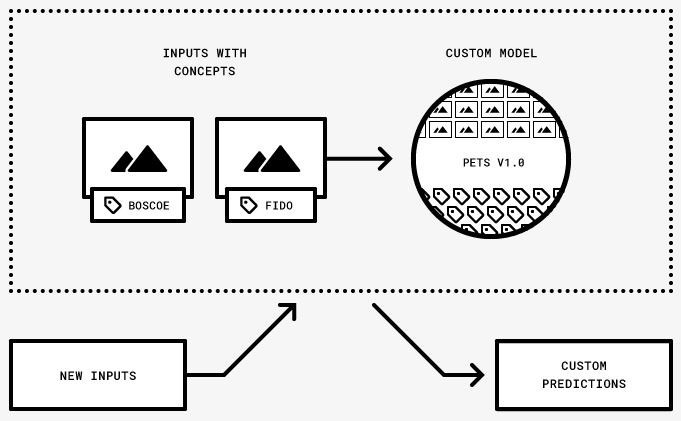
Otentikasi ke API ditangani melalui API *key*. Anda dapat membatasi cakupan API *key*, yang memungkinkan kunci untuk melakukan operasi yang sangat spesifik pada aplikasi yang diberikan, agar aplikasi Anda tetap aman.

### Predict

*Predict,* melakukanprediksi 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.

### 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 Train

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

## 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. Contohimplementasidari *web service* antara lain adalahSOAPdanREST [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)*.*

## 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)* sebagaiprotocol 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.

## 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 Pemprograman 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 pemprograman 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).

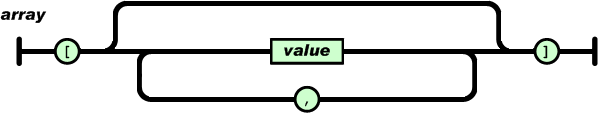


*Sumber Gambar : http://www.json.org/json-id.html*

Gambar JSON Object

1. Larik

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

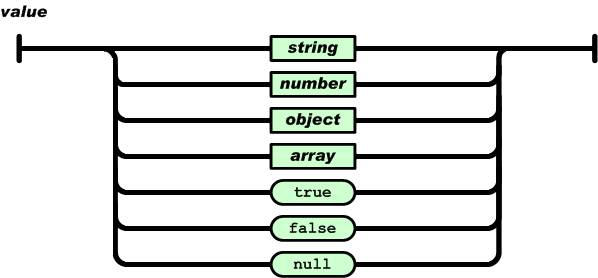


*Sumber Gambar : http://www.json.org/json-id.html*

Gambar JSON Array

1. 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.

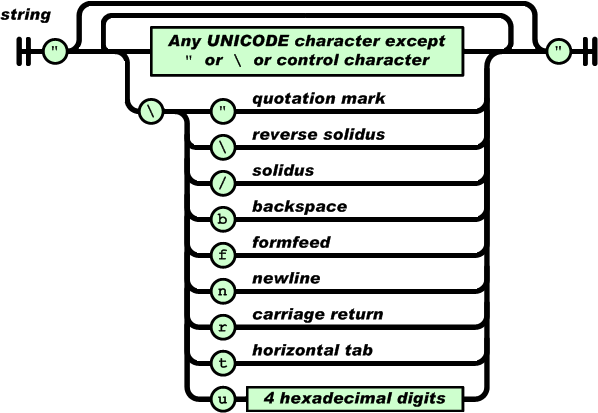


*Sumber Gambar : http://www.json.org/json-id.html*

Gambar Value JSON

1. *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.

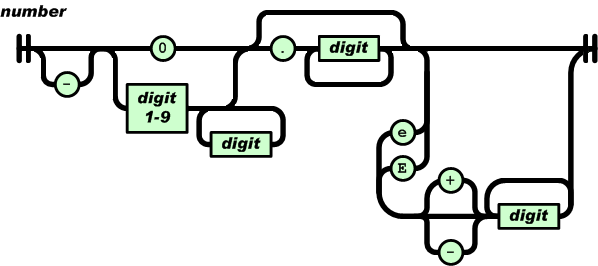


*Sumber Gambar : http://www.json.org/json-id.html*

Gambar String JSON

1. Angka

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



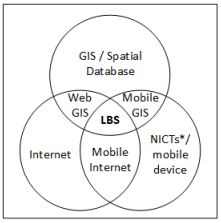
*Sumber Gambar : http://www.json.org/json-id.html*

Gambar JSON Angka

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

## 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].

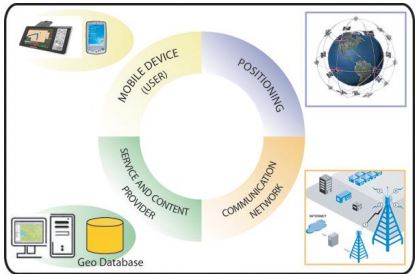


*Sumber gambar: Jurnal Aplikasi Location Based Service untuk Sistem Informasi Publikasi Acara pada Platform Android* [30]

Gambar 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 menggakses suatu web pada jaringan internet.
2. *Push Service*: Layanan ini diberikan langsung oleh *sevice 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 Aplication 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 Komponen dasar LBS

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

## 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 *opn 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.

## 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*>; |

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

|  |
| --- |
| CREATE TABLE <*nama tabel*>(  <*field 1*> <*tipe data*>,  <*field 2*> <*tipe data*>,  …  )[TYPE = <*tipe tabel*>]; |

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

|  |
| --- |
| CREATE INDEX <*nama index*> ON <*nama tabel*>(<*nama kolom*>) |

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

|  |
| --- |
| CREATE VIEW <*nama view*> AS  SELECT <*column1*>, <*column2*>,  ...  FROM <*nama table*>  WHERE <*kondisi*>; |

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

|  |
| --- |
| DROP DATABASE [IF EXIST] <*nama database*>; |

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

|  |
| --- |
| DROP TABLE [IF EXIST] <*nama table*>; |

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

|  |
| --- |
| DROP INDEX <*nama index*> ON <*nama tabel*>; |

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

|  |
| --- |
| DROP VIEW <*nama view*>; |

1. 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*> <*tipe data*>; |

Mengubah kolom

|  |
| --- |
| ALTER TABLE <*nama tabel*> CHANGE <*nama kolom lama*> <*nama kolom baru*> <*tipe 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 table*> ADD FOREIGN KEY (<*nama kolom>*) REFERENCES <*nama tabel referensi*> (<*nama kolom reerensi*>); |

Menghapus *primary key*

|  |
| --- |
| ALTER TABLE <*nama tabel*> DROP PRIMARY KEY; |

Menghapus *Foreign key*

|  |
| --- |
| ALTER TABLE <*nama table*> DROP FOREIGN KEY <*nama foreign key*>; |

Menghapus *index*

|  |
| --- |
| ALTER TABLE <*nama tabel*> DROP INDEX <*nama index*>; |

1. *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*>,…); |

1. 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*>; |

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

|  |
| --- |
| DELETE <*nama tabel*> WHERE<*kriteria atau kondisi*>; |

1. *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.

1. 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; |

1. *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 perintanh 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.*

## Personal Home Page (PHP)

PHP Pertama kali ditemukan pada 1995 oleh seorang *Software Developer* bernama Rasmus Lerdrof. 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 Lerdrof, 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 merealisasikannya, akhirnya Lerdrof 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 popular, dan semakin diminati oleh programmer web dunia.

Rasmus Lerdrof benar-benar menjadikan PHP sangat populer, dan banyak sekali *Team Developer* yang ikut bergabung dengan Lerdrof 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.

1. *Cross platform*

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

1. 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.

1. *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.

## 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 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.

1. 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
30. *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 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.

1. 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, *re-factoring*, 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.*

## 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].

### 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 actor 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.

### 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.

### 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.

### 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.

# 

Analisa dan perancangan sistem

## 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.

### Analisis Masalah

Analisis masalah didapat berdasarkan pada identifikasi masalah yang telah dikaji sebelumnya. Adapun identifikasi masalah pada Pembangunan Aplikasi *Monitoring* danPengendalian Konten Negatif pada Perangkat Pintar Berbasis AndroidMenggunakanAPI Clarifaiini 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 kuisioner 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.

### 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.

#### 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 model sistem

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

1. Proses pemberikan instruksi dari *smartphone* orangtua untuk melakukan pencarian konten negatif, instruksi dikirim ke *server.*
2. Proses s*erver* 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 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.

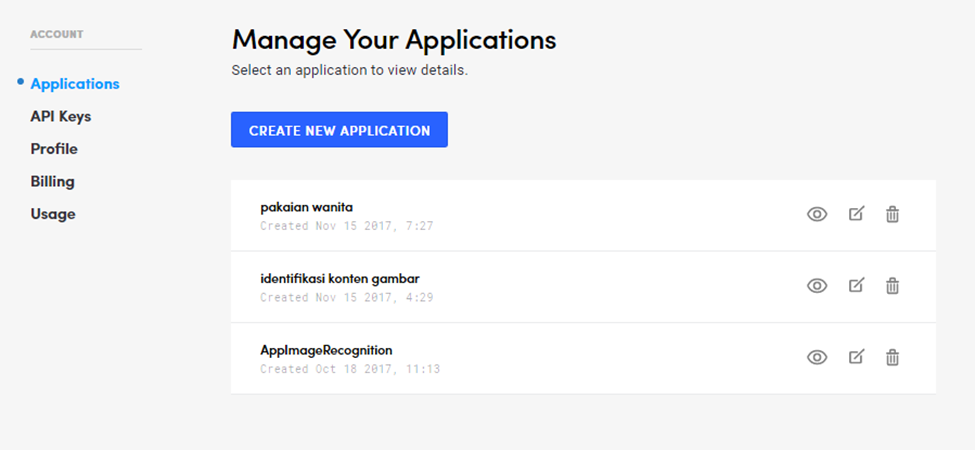
### Analisis Teknologi

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

#### Analisis API Clarifai

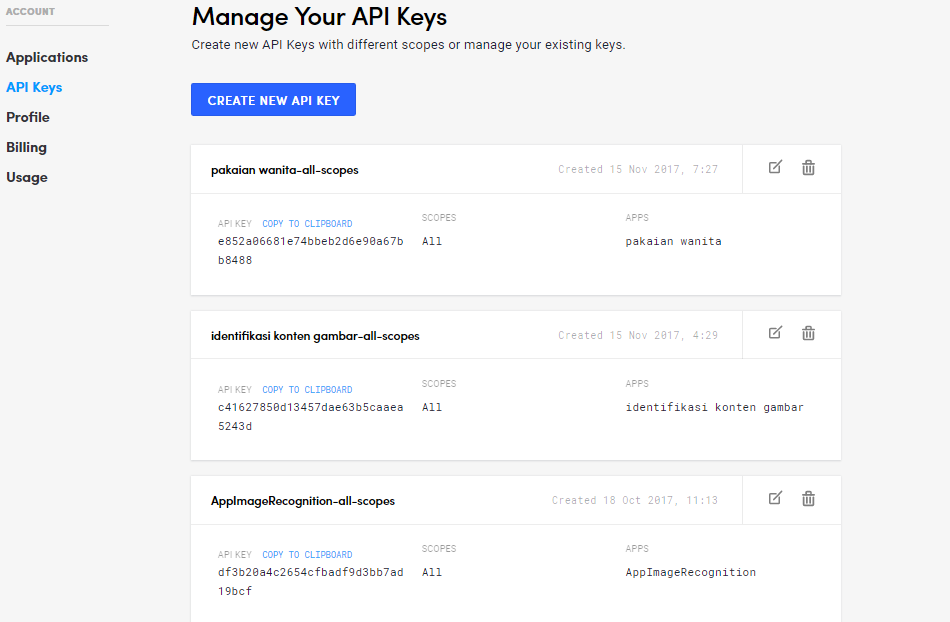
API Clarifai digunakan untuk mengenali sebuah citra, untuk dapat mengenali sebuah citra menggunakan api Clarifai dengan melalui beberapa tahap yaitu pelatihan dan pengujian. Pelatihan merupakan proses yang akan dijadikan sebagai acuan pengenalan dalam mengenali citra, setelah dilakukan pelatihan kemudian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

Langkah pertama yang harus dilakukan yaitu dengan membuat project baru di alamat https://clarifai.com/developer/account/applications.



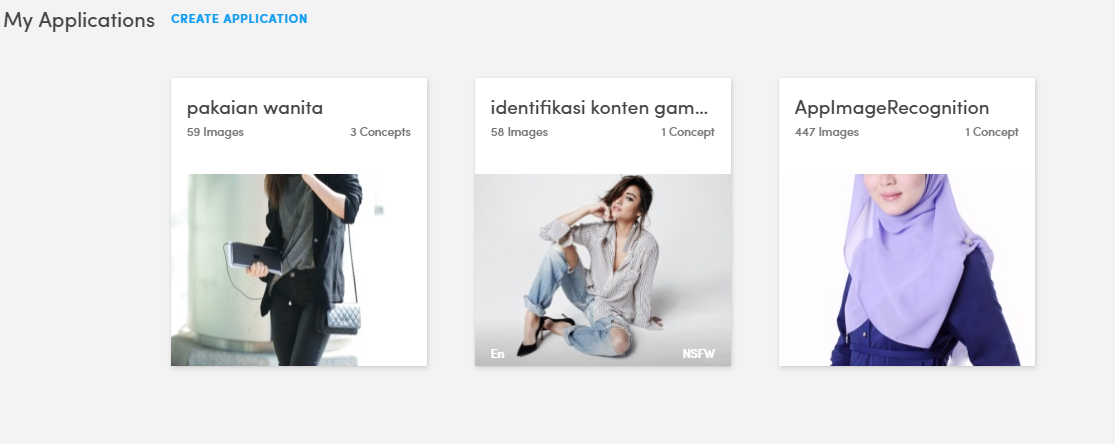
Gambar Membuat Projek Baru di Situs Clarifai

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



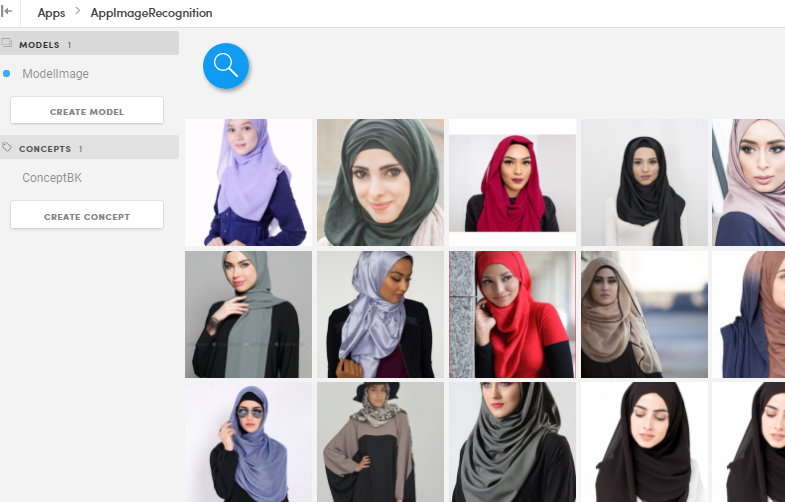
Gambar Mendapatkan Autentikasi Key

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



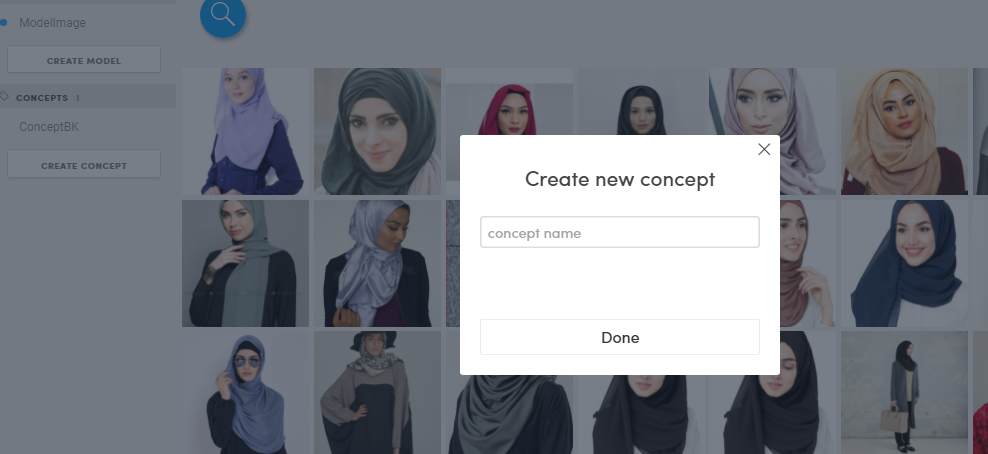
Gambar Gambar menu *Explorer di Clarifai*

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



Gambar Tampilan Projek yang telah dibuat di menu *ecplorer*

Pilih Create new concept, lalu akan menampilkan halaman berikut:



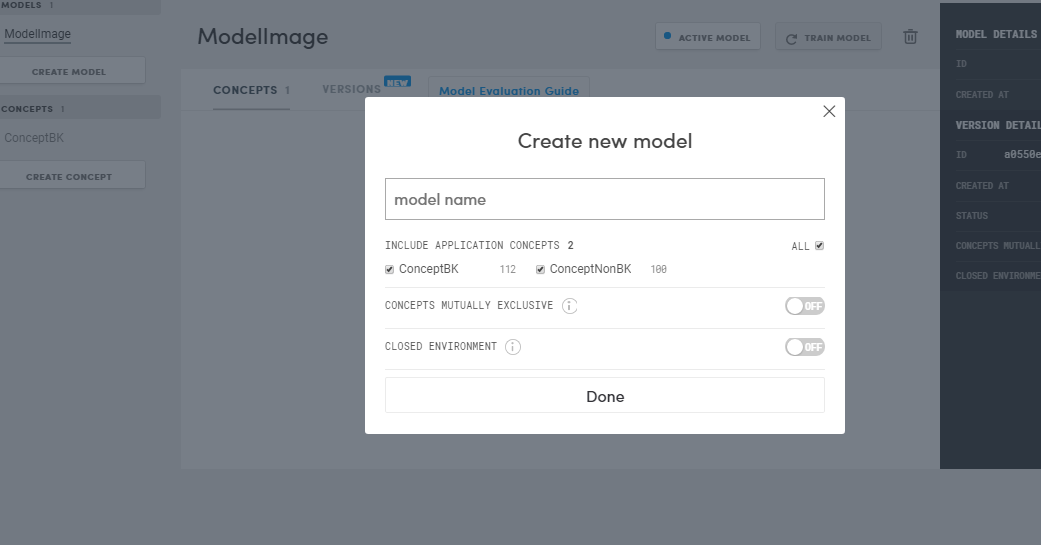
Gambar Membuat Konsep baru di Clarifai

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



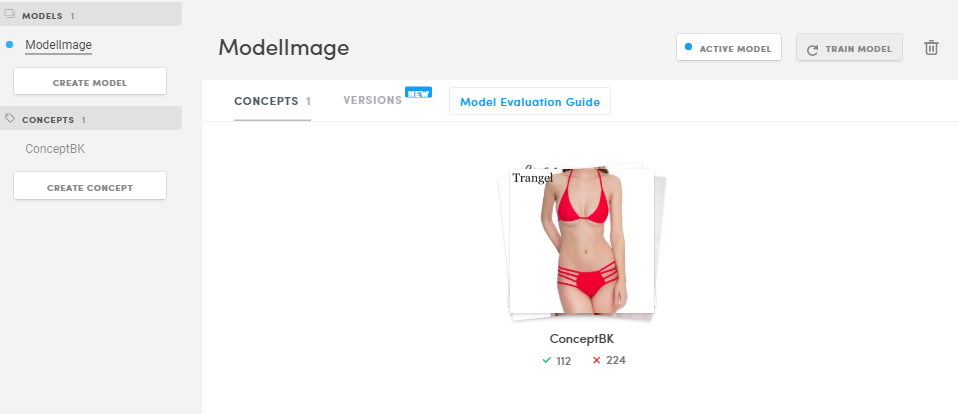
Gambar Menambahkan Citra pada *Concept*  yang telah dibuat

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



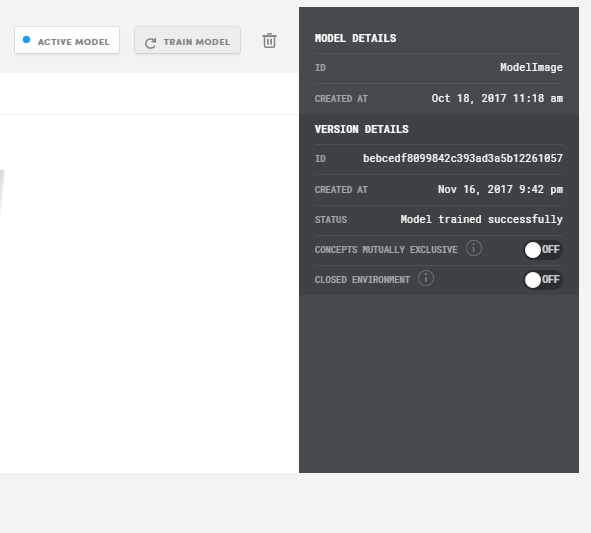
Gambar Membuat Model Baru Calarifai

Setalah model telah dibuat lalu train pada model, seperti pada gambar berikut:



Gambar *Training* Pada Model

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



Gambar Model id Clarifai

Selanjutnya melakukan pengujian dengan library java



Gambar Pengujian API Clarifai dengan Library Java

Hasil respone dari library Api Clarifai dari java



Gambar *Response* dari *Library* Api Clarifai dari Java

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.

#### Analisis *Web Service* dengan *Framework* CodeIgniter

CodeIgniter merupakan kerangka kerja PHP MVC(*Model View* Controller) yang memudahkan pengembang dalam pembangunan aplikasi berbasis web dengan bahasa pemrograman PHP yang menyediakan banyak fungsi dan library yang dapat digunakan sehingga pengembang tidak perlu membangunnya dari awal.

Pada penelitian ini peneliti menggunakan *framework* CodeIgniter sebagai *web service-*nya dimana arsitektur yang digunakan adalah REST( *Representational State Transfer*). Arsitektur REST, umumnya dijalankan via HTTP (*Hypertext Transfer Protocol*), melibatkan proses pembacaan laman web yang memuat sebuah file XML atau JSON.

REST memiliki kekhasan pada interaksi antar klien dengan *server* yang difasilitasi oleh metode yang digunakan untuk menghindari ambiguitas yaitu GET, POST, PUT dan DELETE dengan fungsinya masing-masing. Adapun tahapan-tahapan dalam penggunaan CodeIgniter sebagai *web service* yang digunakan dalam pembangunan aplikasi monitoring dan pengendalian konten negatif pada perangkat pintar berbasis android menggunakan API Clarifai adalah sebagai berikut:

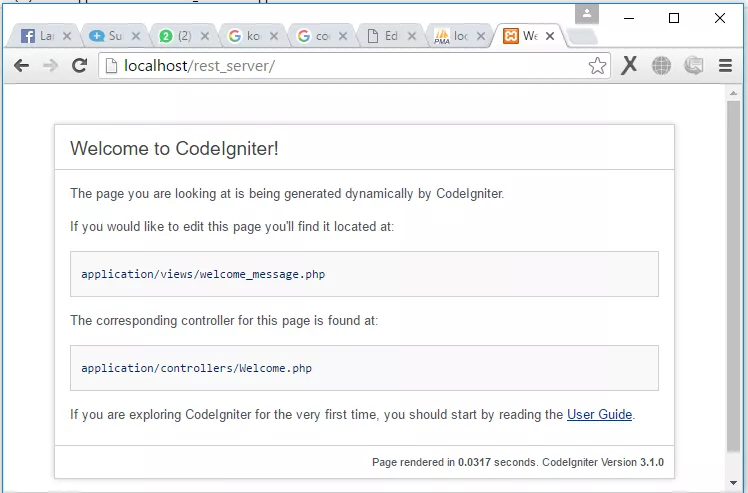
1. Unduh *framework* CodeIgniter pada situs resminya [www.codeigniter.com](http://www.codeigniter.com).
2. Ekstrak dan berinama sesuai kebutuhan.
3. Simpan di *folder* htdocs atau server lokal.
4. Sesuaikan *config* pada CodeIgniter yang berada pada folder\_root/application/config/config.php sebagai berikut

|  |
| --- |
| $config['base\_url'] = 'http://localhost/isi dengan nama projek yang sedang dibangun/'; |

1. Sesuaikan *database* dengan *database* yang telah dibuat pada folder\_root/application/config/database.php seperti berikut:

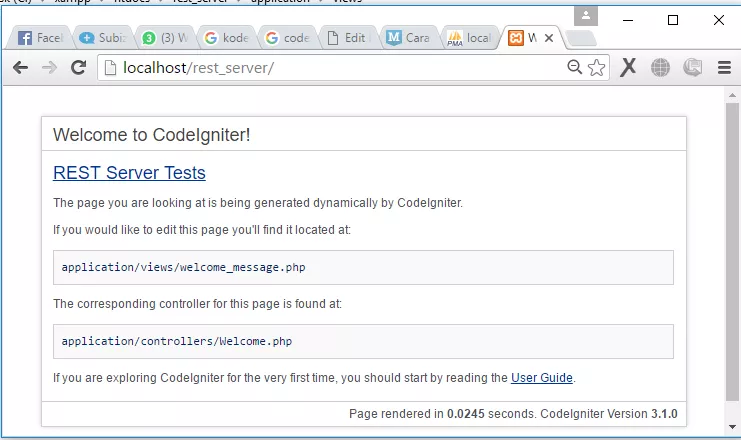
|  |
| --- |
| $db['default'] = array(  'dsn' => '',  'hostname' => 'localhost',//diisi dengan host server  'username' => 'root',// username database yang digunakan  'password' => '',// password database yang digunakan  'database' => ' db\_pamon ',//nama database yang digunakan  'dbdriver' => 'mysqli',  'dbprefix' => '',  'pconnect' => FALSE,  'db\_debug' => (ENVIRONMENT !== 'production'),  'cache\_on' => FALSE,  'cachedir' => '',  'char\_set' => 'utf8',  'dbcollat' => 'utf8\_general\_ci',  'swap\_pre' => '',  'encrypt' => FALSE,  'compress' => FALSE,  'stricton' => FALSE,  'failover' => array(),  'save\_queries' => TRUE  ); |

1. Tes dengan mengaksesnya di *browser* <http://localhost/folder_root_projek/> dan jika berhasil akan tampil seperti berikut:



Gambar Tampilan *Default* CodeIgniter Setelah di *Install*

1. Langkah selanjutnya yaitu mengunduh dan menginstall *Library* REST untuk membuat *web service* yang di buat oleh Chris Kacergius pada *repository-*nya <https://github.com/chriskacerguis/codeigniter-restserver>.
2. Ekstrak dan salin ke folder projek tadi.jika sampai tahapan ini berhasil maka akan tampil seperti berikut ini:



Gambar Tampilan *Default* CodeIgniter Setelah di *Install Library* REST

1. Selanjutnya kita tinggal membuat *controller* sesuai dengan yang dibutuhkan seperti pada contoh berikut:

*Method GET* (Mengambil *Data*)

|  |
| --- |
| function index\_get() {  $id = $this->get('id\_user');  if ($id == '') {  $user = $this->db->get('t\_user')->result();  } else {  $this->db->where('id\_user', $id);  $user = $this->db->get('t\_user')->result();  }  $this->response($user, 200);  } |

*Method POST* (Tambah *Data*)

|  |
| --- |
| function index\_post() {  $ql = $this->db->select('id\_user ')->from('t\_user')->where('id\_user',$this->post('id\_user'))->get();  if( $ql->num\_rows() > 0) {  $this->response(array('status' => 'fail','code' => 502));  } else {  $data = array(  //'id\_user' => $this->post('id\_user'),  'nama' => $this->post('nama'),  'email' => $this->post('email'),  'password' => md5($this->post('password')));  $insert = $this->db->insert('t\_user', $data);  $lastin = $this->db->insert\_id();    if ($insert) {  $datainserted = array(  'id\_user' => $lastin,  'nama' => $this->post('nama'),  'email' => $this->post('email'),  'password' => md5($this->post('password')));  $this->response($datainserted, 200);  }  }  } |

*Method PUT* (Edit *Data*)

|  |
| --- |
| function index\_put() {  $id = $this->put('id\_user');  $data = array(  'id\_user' => $this->put('id\_user'),  'nama' => $this->put('nama'),  'email' => $this->put('email'),  'password' => md5($this->put('password')));  $this->db->where('id\_user', $id);  $update = $this->db->update('t\_user', $data);  if ($update) {  $this->response($data, 200);  } else {  $this->response(array('status' => 'fail', 502));  }  } |

*Method DELETE* ( Hapus *Data*)

|  |
| --- |
| function index\_delete() {  $id = $this->delete('id\_user');  $this->db->where('id\_user', $id);  $delete = $this->db->delete('t\_user');  if ($delete) {  $this->response(array('status' => 'success','code' => 201));  } else {  $this->response(array('status' => 'fail', 502));  }  } |

### 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 . Tabel Spesifikasi Kebutuhan Perangkat Lunak Fungsional

|  |  |
| --- | --- |
| **Kode SKPL** | **Spesifikasi Kebutuhan Perangkat Luna** |
| SKPL-F-001 | Sistem dapat melakukan fungsi daftar akun |
| SKPL-F-002 | Sistem dapat memproses permintaan lupa *password* |
| 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 *lokasi anak* |
| SKPL-F-011 | Sistem dapat melakukan monitoring *history browser* |
| SKPL-F-012 | Sistem dapat melakukan *backup data* 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 *monitoring history browser* |
| SKPL-F-016 | Sistem dapat mengirimkan notifikasi *monitoring* lokasi anak |

Tabel . Tabel Spesifikasi Kebutuhan Perangkat Lunak Non-Fungsional

|  |  |
| --- | --- |
| **Kode SKPL** | **Spesifikasi Kebutuhan Perangkat Lunak** |
| SKPL-NF-001 | Sistem berbasis *mobile* dengan minimum Android Jelly Bean 4.1.x (API level 16) |
| SKPL-NF-002 | Sistem menggunakan bahasa Indonesia |
| SKPL-NF-003 | Sistem menggunakan *web server* untuk menyimpan konten *monitoring* |
| SKPL-NF-004 | *Smartphone* tidak dalam keadaan *airplane mode* atau mati. |

### 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.

#### 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 . Spesifikasi Kebutuhan Perangkat Keras

|  |  |  |
| --- | --- | --- |
| **No.** | **Perangkat Keras** | **Spesifikasi** |
| 1 | CPU | 1 GHz |
| 2 | *Disk Space* | 45 MB |
| 3 | RAM | 512 MB |
| 4 | Dimensi Layar | 4,0 *inch* |

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 . 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 |

#### 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 . 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 . 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 | *Database* | MySQL |
| 5 | *Browser* | Google Chrome |

#### 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 . Karakteristik Pengguna

|  |  |  |  |
| --- | --- | --- | --- |
| **Pengguna** | **Hak Akses** | **Tingkat keterampilan** | **Bahasa** |
| Pengguna | Dapat Mengakses semua fitur yang terdapat pada aplikasi | Dapat mengoperasikan *smartphone* Android | Mengerti bahasa Indonesia |

### 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.*

#### 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 *Use Case Diagram*

##### 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 . Definisi Aktor

|  |  |
| --- | --- |
| **Aktor** | **Definisi** |
| Orang tua | Orangtua adalah aktor yang masuk ke dalam sistem dengan tujuan untuk melakukan *monitoring* dan menjalankan instruksi yang terdapat pada sistem. |

##### 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 . 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 *monitoring* 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 *monitoring* lokasi anak pada sistem. |
| 11 | *Monitoring* history browser | Fungsionalitas untuk melakukan *monitoring* history browser pada *smartphone* anak melalui sistem. |
| 12 | *Backup* data kontak | Fungsionalitas untuk melakukan *backup* kontak pada *smartphone* anak. |
| 13 | Notifikasi | Fungsionalitas notifikasi yang muncul pada sistem. |
| 14 | Notifikasi *monitoring* konten negatif | Fungsionalitas notifikasi *monitoring* konten negatif yang muncul pada sistem. |
| 15 | Notifikasi *monitoring* browser | Fungsionalitas notifikasi *monitoring* history browser yang muncul pada sistem. |
| 16 | Notifikasi lokasi anak | Fungsionalitas notifikasi lokasi anak yang muncul pada sistem. |

#### 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 . Skenario *Use Case* Daftar

|  |  |  |
| --- | --- | --- |
| *Use Case Name* | **Daftar** | |
| *Related Requirements* | - | |
| *Goal in Context* | Pengguna aplikasi yaitu orang tua ingin mendaftar ke dalam sistem dengan membuat akun | |
| *Preconditions* | Pengguna belum terdaftar dan belum memiliki akun | |
| *Successful End Condition* | Pengguna berhasil terdaftar kedalam sistem | |
| *Failed End Condition* | pengguna gagal terdaftar ke dalam sistem | |
| *Primary Actors* | Orang tua | |
| *Secondary Actor* | - | |
| *Trigger* | Pengguna memilih menu daftar untuk membuat akun | |
| *Main Flow* | *Step* | *Action* |
|  | 1 | Sistem menampilkan *form* daftar akun |
| 2 | Pengguna mengisi *form* |
| 3 | Sistem mengirimkan kode pendaftaran melalui *email* |
| 4 | Pengguna memasukkan kode pendaftaran Pada *form* pendaftaran yang disediakan |
| *Extension* | Step | *Branching Action* |
|  |  | Sistem gagal mengirimkan kode pendaftaran melalui *email* |

1. Skenario *Use Case* Lupa Password

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

Tabel . Skenario *Use Case* Lupa *Password*

|  |  |  |
| --- | --- | --- |
| *Use Case Name* | **Lupa *Password*** | |
| *Related Requirements* | - | |
| *Goal in Context* | Pengguna lupa *password* untuk masuk kedalam sistem dan ingin mengatur ulang *password* agar dapat masuk ke sistem | |
| *Preconditions* | Pengguna telah terdaftar namun lupa *password* untuk masuk sistem | |
| *Successful End Condition* | Sistem berhasil mengatur ulang *password* dan mengirimkan password sementara user melalui *email* | |
| *Failed End Condition* | Sistem gagal menset ulang *password* pengguna | |
| *Primary Actors* | Orang tua | |
| *Secondary Actor* | - | |
| *Trigger* | Pengguna memilih menu lupa *password* untuk mendapatkan *password* baru | |
| *Main Flow* | *Step* | *Action* |
|  | 1 | Sistem menampilkan *form* Lupa *password* |
| 2 | Pengguna mengisi *form* dengan *email* dari akun orang tua |
| 3 | Sistem melakukan pencocokan email yang dimasukan orang tua dengan database di *server* |
| 4 | Sistem mengirimkan *password* baru pengguna melalui *email* |
| 5 | Pengguna *login* kedalam sistem menggunakan *password Baru* |
| *Extension* | Step | *Branching Action* |
|  |  | Sistem gagal mengirimkan *password* baru pengguna melalui *email* |

1. Skenario *Use Case Login*

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

Tabel . Skenario *Use Case Login*

|  |  |  |
| --- | --- | --- |
| *Use Case Name* | **Login** | |
| *Related Requirements* | Profil, *Monitoring* konten negatif, *monitoring* lokasi anak, *monitoring history browser, backup data* kontak, notifikasi | |
| *Goal in Context* | Pengguna ingin masuk kedalam sistem | |
| *Preconditions* | Sistem menampilkan *form login* | |
| *Successful End Condition* | Pengguna berhasil masuk kedalam sistem | |
| *Failed End Condition* | pengguna gagal masuk ke dalam sistem | |
| *Primary Actors* | Orang tua | |
| *Secondary Actor* | - | |
| *Trigger* | Pengguna memilih menu daftar untuk membuat akun | |
| *Main Flow* | *Step* | *Action* |
|  | 1 | Pengguna yaitu orangtua mengisi *form* *login* |
| 2 | Sistem mencocokan data *login* orangtua dengan data orang tua yang ada di *database server* |
| 3 | Sistem meng-*generate* *token key* orangtua dan anak |
| 4 | Orangtua mengonfirmasi sesi *login* |
| 5 | Orang tua berhasil masuk ke dalam sistem |
| *Extension* | Step | *Branching Action* |
|  |  | Sistem gagal memproses informasi *login* orangtua |

1. Skenario *Use Case Log out*

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

Tabel . Skenario *Use Case Log Out*

|  |  |  |
| --- | --- | --- |
| *Use Case Name* | **Log Out** | |
| *Related Requirements* | Login | |
| *Goal in Context* | Pengguna ingin keluar dari sistem | |
| *Preconditions* | Pengguna yaitu orang tua berada di dalam sistem | |
| *Successful End Condition* | Pengguna berhasil keluar dari sistem | |
| *Failed End Condition* | Pengguna gagal keluar dari sistem | |
| *Primary Actors* | Orang tua | |
| *Secondary Actor* | - | |
| *Trigger* | Orangtua mengklik menu *logout* | |
| *Main Flow* | *Step* | *Action* |
|  | *Include::Login* |  |
|  | 1 | Pengguna yaitu orangtua memilih menu *log out* |
| 2 | Orang tua berhasil keluar dari sistem |
| *Extension* | Step | *Branching Action* |
|  | 1 | Sistem gagal membuat koneksi ke server |

1. Skenario *Use Case* Profil

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

Tabel . Skenario *Use Case* Profil

|  |  |  |
| --- | --- | --- |
| *Use Case Name* | **Profil** | |
| *Related Requirements* | *Login* | |
| *Goal in Context* | Orang tua ingin melihat profil akun | |
| *Preconditions* | Sistem menampilkan menu utama | |
| *Successful End Condition* | Orang tua dapat melihat data akun orangtua di sistem | |
| *Failed End Condition* | Data profil akun orang tua gagal ditampilkan | |
| *Primary Actors* | Orang tua | |
| *Secondary Actor* | - | |
| *Trigger* | Orangtua meng-klik menu profil | |
| *Main Flow* | *Step* | *Action* |
|  | *Include::Login* |  |
|  | 1 | Pengguna yaitu orangtua memilih menu *profil* |
| 2 | Sistem menampilkan data profil dari akun orang tua |
| *Extension* | Step | *Branching Action* |
|  | 1 | Sistem gagal mengambil data profil orang tua dari server |

1. Skenario *Use Case* Ubah Nama

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

Tabel . Skenario *Use Case* Ubah Nama

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

1. Skenario *Use Case* Ubah Password

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

Tabel . Skenario *Use Case* Ubah *Password*

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

1. Skenario *Use Case Monitoring* Konten Negatif

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

Tabel . Skenario *Use Case Monitoring* Konten Negatif

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

1. Skenario *Use Case* Penghapusan Berkas Gambar

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

Tabel . Skenario *Use Case* Penghapusan Berkas Gambar

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

1. Skenario *Use Case Monitoring* Lokasi Anak

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

Tabel . Skenario *Use Case Monitoring* Lokasi Anak

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

1. Skenario *Use Case Monitoring History Browser*

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

Tabel . *Monitoring History Browser*

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

1. Skenario *Use Case Backup Data Kontak*

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

Tabel .*Backup data Kontak*

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

1. Skenario *Use Case* Notifikasi

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

Tabel . Notifikasi

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

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

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

Tabel . Notifikasi *Monitoring* Konten Negatif

|  |  |  |
| --- | --- | --- |
| *Use Case Name* | **Notifikasi *Monitoring* Konten Negatif** | |
| *Related Requirements* | *Login* | |
| *Goal in Context* | Sistem dapat mengirimkan Notifikasi hasil *monitoring* konten negatif pada *smartphone Android* | |
| *Preconditions* | Orang tua berada di dalam sistem | |
| *Successful End Condition* | Orang tua dapat menerima notifikasi hasil *monitoring* konten negatif | |
| *Failed End Condition* | Orang tua gagal menerima notifikasi hasil *monitoring* konten negatif | |
| *Primary Actors* | Orang tua | |
| *Secondary Actor* | - | |
| *Trigger* | Sistem menjalankan fungsi *monitoring* konten negatif | |
| *Main Flow* | *Step* | *Action* |
|  | 1 | Sistem menjalankan fungsi *monitoring* konten negatif |
| 2 | Sistem membaca hasil *monitoring* konten negatif |
|  | 3 | Sistem mengirimkan notifikasi hasil *monitoring* konten negatif |
|  | 4 | Orangtua berhasil menerima notifikasi hasil *monitoring* konten negatif |
| *Extension* | Step | *Branching Action* |
|  | 1 | Sistem gagal mengirimkan notifikasi hasil *monitoring* konten negatif |

1. Skenario *Use Case*Notifikasi *monitoring History Browser*

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

Tabel . *Case* Notifikasi *monitoring History Browser*

|  |  |  |
| --- | --- | --- |
| *Use Case Name* | **Notifikasi *Monitoring History Browser*** | |
| *Related Requirements* | *Login* | |
| *Goal in Context* | Sistem dapat mengirimkan notifikasi hasil *monitoring* *history browser* pada *smartphone* Android. | |
| *Preconditions* | Orang tua berada di dalam sistem | |
| *Successful End Condition* | Orang tua dapat melihat notifikasi hasil *monitoring history browser* | |
| *Failed End Condition* | Sistem gagal mengirimkan notifikasi *monitoring history browser* | |
| *Primary Actors* | Orang tua | |
| *Secondary Actor* | - | |
| *Trigger* | Sistem menjalankan fungsi *monitoring history browser* | |
| *Main Flow* | *Step* | *Action* |
|  | 1 | Sistem menjalankan fungsi *monitoring history browser* |
| 2 | Sistem membaca hasil *monitoring history browser* |
|  | 3 | Sistem mengirimkan notifikasi hasil *monitoring history browser* kepada orang tua |
|  | 4 | Orangtua berhasil menerima notifikasi hasil *monitoring history browser* |
| *Extension* | Step | *Branching Action* |
|  | 1 | Sistem gagal mengirimkan notifikasi hasil *monitoring* |

1. Skenario *Use Case Notifikasi Monitoring* Lokasi

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

Tabel . *Notifikasi Monitoring* Lokasi

|  |  |  |
| --- | --- | --- |
| *Use Case Name* | ***Notifikasi Monitoring* Lokasi** | |
| *Related Requirements* | *Login* | |
| *Goal in Context* | Sistem dapat mengirimkan notifikasi hasil *Monitoring* Lokasi anak | |
| *Preconditions* | Orang tua berada di dalam sistem | |
| *Successful End Condition* | Orang tua dapat melihat notifikasi hasil *Monitoring* Lokasi anak | |
| *Failed End Condition* | Sistem gagal mengirimkan notifikasi *monitoring* lokasi anak | |
| *Primary Actors* | Orang tua | |
| *Secondary Actor* | - | |
| *Trigger* | Sistem menjalankan fungsi *monitoring lokasi anak* | |
| *Main Flow* | *Step* | *Action* |
|  | 1 | Sistem menjalankan fungsi *monitoring* lokasi anak |
| 2 | Sistem membaca hasil *monitoring* lokasi anak |
|  | 3 | Sistem mengirimkan notifikasi hasil *monitoring* lokasi anak kepada orangtua |
|  | 4 | Orangtua berhasil menerima notifikasi hasil *monitoring* Lokasi anak |
| *Extension* | Step | *Branching Action* |
|  | 1 | Sistem gagal mengirimkan notifikasi hasil *monitoring* lokasi anak |

#### *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 saatakan mendaftar ke dalam sistem. *Activity diagram* daftar dapat dilihat pada Gambar 22.



Gambar *Activity Diagram* Daftar

1. *Activity Diagram* Lupa Password

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



Gambar *Activity Diagram* Lupa *password*

1. *Activity Diagram Login*

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



Gambar *Activity Diagram Login*

1. *Activity Diagram Log Out*

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



Gambar *Activity Diagram Log Out*

1. *Activity Diagram Profil*

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



Gambar *Activity Diagram* Profil

1. *Activity Diagram* Ubah nama

*Activity diagram* profilmenjelaskan 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 *Activity Diagram* Ubah nama

1. *Activity Diagram* Ubah *Password*

*Activity diagram* profilmenjelaskan 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  *Activity Diagram* Ubah *Password*

1. *Activity Diagram Monitoring* Konten Negatif

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



Gambar *Activity Diagram Monitoring* Konten Negatif

1. *Activity Diagram* Penghapusan Berkas Gambar

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



Gambar *Activity Diagram* Penghapusan Berkas Gambar

1. *Activity Diagram Monitoring History Browser*

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



Gambar Activity Diagram Monitoring History Browser

1. *Activity Diagram Monitoring* Lokasi Anak

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



Gambar  *Activity Diagram Monitoring* Lokasi Anak

1. *Activity Diagram Backup Data* kontak

*Activity diagram Backup Data* kontak menjelaskan aliran kerja sistemdalam menjalankan *Backup Data* kontak. *Activity diagram* *Backup Data* kontak dapat dilihat pada Gambar 33.



Gambar  *Activity Diagram Backup Data* kontak

1. *Activity Diagram* Notifikasi

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



Gambar  *Activity Diagram* Notifikasi

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

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



Gambar Activity Diagram Notifikasi Monitoring Konten Negatif

1. *Activity Diagram Notifikasi Monitoring History Browser*

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



Gambar Activity Diagram Notifikasi Monitoring History Browser

1. *Activity Diagram* Notifikasi Lokasi Anak

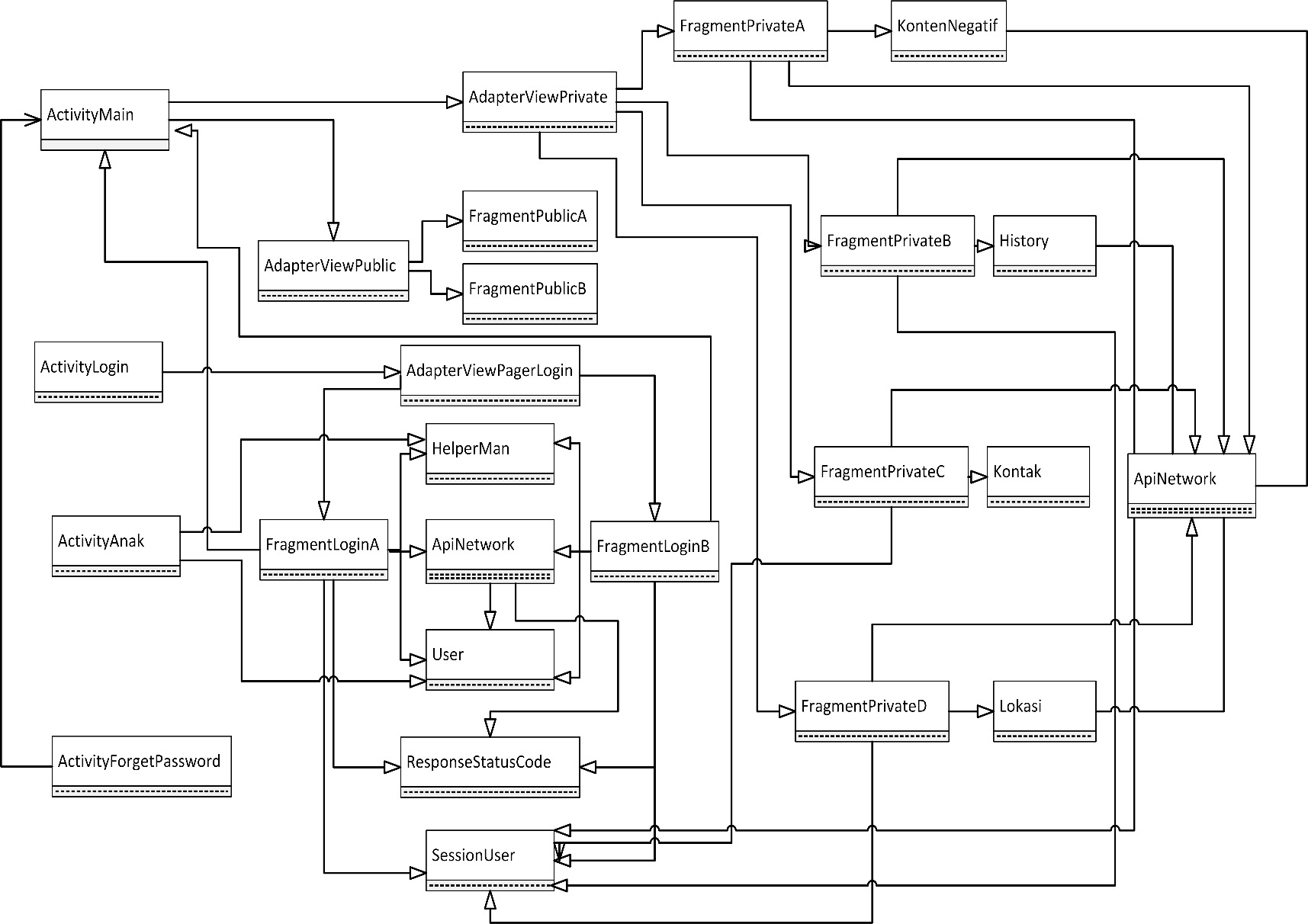
*Activity diagram* notifikasi lokasi anak menjelaskan aliran kerja sistemsaat menjalankan Notifikasi lokasi anak kepada pengguna. *Activity diagram* Notifikasi lokasi anak dapat dilihat pada Gambar 37.



Gambar *Activity Diagram* Notifikasi Lokasi Anak

#### *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 Aplikasi *Monitoring* Dan Pengendalian Konten Negatif Pada Perangkat Pintar Berbasis Android Menggunakan API Clarifai

Berikut penjelasan Gambar 46 *Class Diagram* aplikasi *monitoring* dan pengendalian konten negatif pada perangkat pintar berbasis Android menggunakan API Clarifai :

1. ActivityMain



Gambar ActivityMain

1. AdapterViewPagerPublic



Gambar AdapterViewPagerPublic

1. FragmentPublicA



Gambar FragmentPublicA

1. FragmentPublicB



Gambar FragmentPublicB

1. AdapterViewPagerPrivate



Gambar AdapterViewPagerPrivate

1. FragmentPrivateA



Gambar FragmentPrivateA

1. FragmentPrivateB



Gambar FragmentPrivateB

1. History



Gambar History

1. FragmentPrivateC



Gambar FragmentPrivateC

1. Kontak



Gambar Kontak

1. FragmentPrivateD



Gambar FragmentPrivateD

1. Lokasi



Gambar Lokasi

1. ActivityLogin



Gambar ActivityLogin

1. ActivityAnak



Gambar ActivityAnak

1. AdapterViewPagerLogin



Gambar AdapterViewPagerLogin

1. FragmentLoginA



Gambar FragmentLoginA

1. FragmentLoginB



Gambar FragmentLoginB

1. HelperMan



Gambar HelperMan

1. ApiNetwork



Gambar ApiNetwork

1. User



Gambar User

1. ResponseStatusCode



Gambar ResponseStatusCode

1. SessionUser



Gambar SessionUser

1. ActivityForgetPassword



Gambar ActivityForgetPassword

#### *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.

*Sequence Diagram* Lupa Password

*Sequence diagram* daftar menunjukan 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 70



Gambar  *Sequence Diagram* Lupa Password

*Sequence Diagram Login*

*Sequence diagram login* menunjukan 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  *Sequence Diagram Login*

*Sequence Diagram Log Out*

*Sequence diagram log out* menunjukan 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  *Sequence Diagram Log Out*

*Sequence Diagram Monitoring* Konten Negatif

*Sequence diagram monitoring* konten negatif menunjukan 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 73



Gambar  *Sequence Diagram Monitoring* Konten Negatif

*Sequence Diagram Monitoring* Lokasi Anak

*Sequence diagram monitoring* lokasi anak menunjukan 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  *Sequence Diagram Monitoring* Lokasi Anak

*Sequence Diagram Monitoring History Browser*

*Sequence diagram monitoring history browser* menunjukan 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  *Sequence Diagram Monitoring History Browser*

*Sequence Diagram Backup Data* Kontak

*Sequence diagram backup data* kontak menunjukan 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  *Sequence Diagram Backup Data* Kontak

## 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.

### 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 Skema Relasi

### 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 tabei *t\_user* dapat dilihat pada Tabel .7 sebagai berikut.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nama Field** | **Tipe Data** | **Panjang** | **Atribut** | **Keterangan** |
| id\_user | Int | 5 | *Primary Key* |  |
| nama | Varchar | 45 |  | *Not null* |
| email | Varchar | 45 |  | *Not null* |
| password | Text |  |  | *Not null* |
| Fcm\_token\_orangtua | Varchar | 225 |  | *Default* *Not null* |
| Fcm token anak | varchar | 225 |  | *Default* *Not nul* |

Tabel . Struktur Tabel *t\_user*

1. 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 | *Primary Key* |  |
| Id\_user | Int | 11 | *Foreign Key* | *Not null* |
| Direktori\_server | Text |  |  | *Not null* |
| Direktori\_mobile | Text |  |  | *Not null* |
| status | *enum* | 0, 1 |  | *Not null* |
| aksi | *enum* | 0, 1, 2, 3 |  | *Default 0* |
| waktu | *datetime* |  |  | *Not null* |

Tabel . Struktur Tabel t\_konten

1. 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 | *Primary Key* |  |
| Id\_user | Int | 11 | *Foreign Key* | *Not null* |
| url | Text |  |  | *Not null* |

Tabel . Struktur Tabel *t\_history*

1. Tabel t\_lokasi

Tabel t\_lokasi digunakan untuk menyimpan data dari lokasi anakdari *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 | *Primary Key* |  |
| Id\_user | Int | 11 | *Foreign Key* | *Not null* |
| lat | double |  |  | *Not null* |
| lng | double |  |  | *Not null* |

Tabel . Struktur Tabel t\_lokasi

1. 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 | *Primary Key* |  |
| Id\_user | Int | 11 | *Foreign Key* | *Not null* |
| nama | varchar | 50 |  | *Not null* |
| No\_telepon | varchar | 15 |  | *Not null* |

Tabel . Struktur Tabel t\_kontak

### 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 :



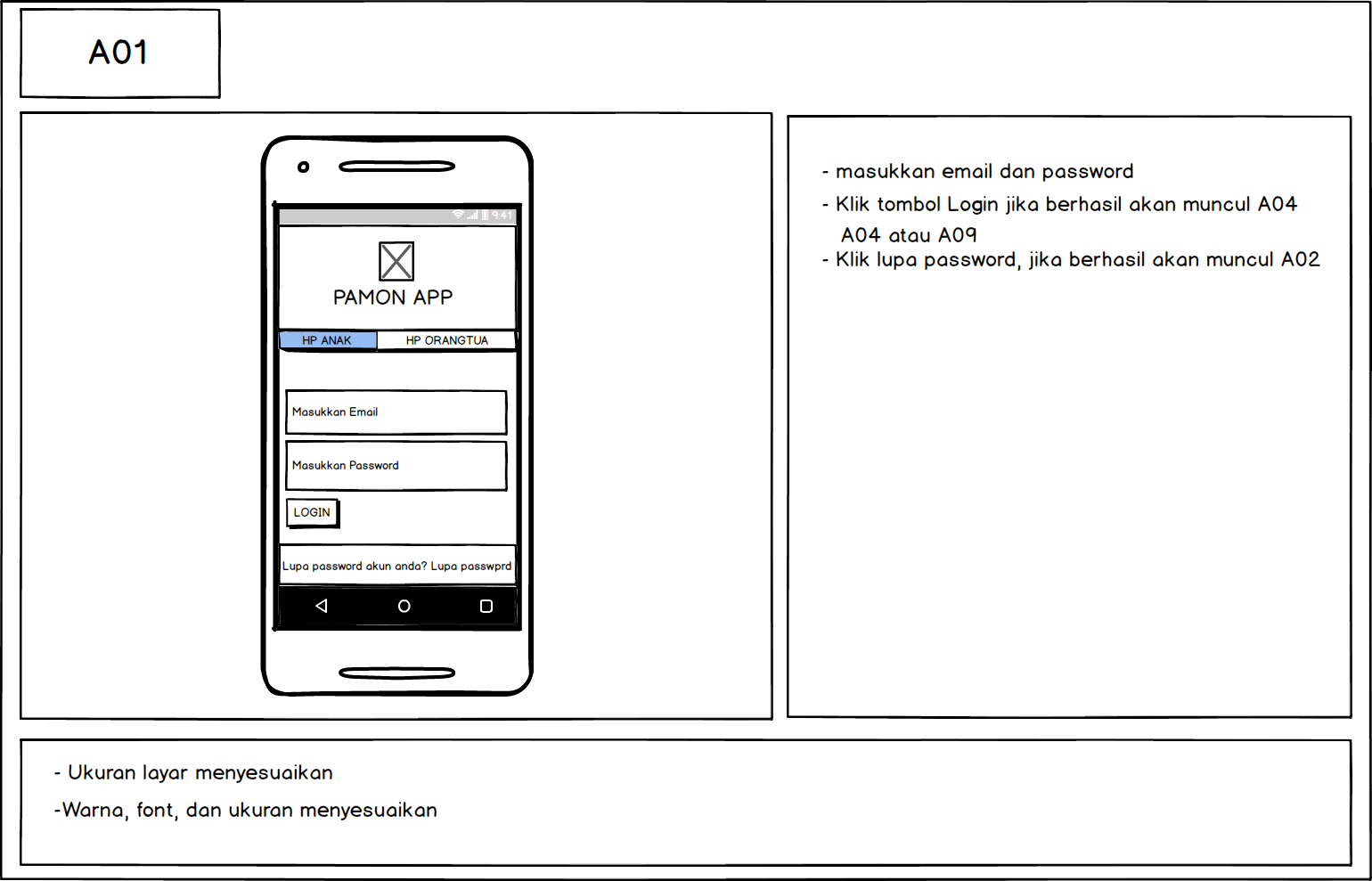
Gambar Struktur Menu

### 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*

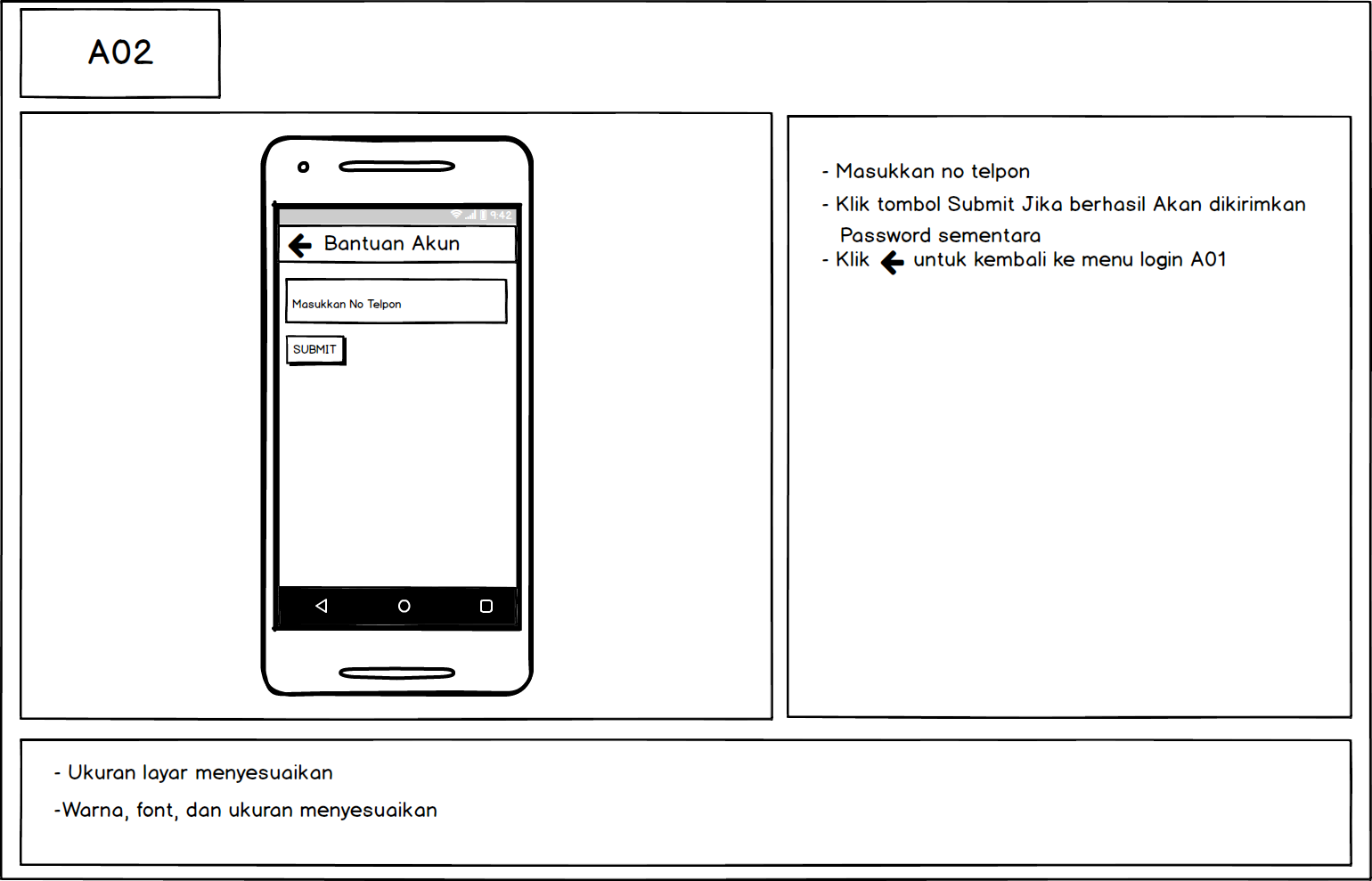
Perancangan antarmuka *login* merupakan rancangan yang akan muncul pertama kali pada saat Orangtua membuka aplikasi. Adapun perancangan antarmuka *login* dapat dilihat pada Gambar 44.



Gambar Perancangan Antar Muka *Login*

1. Perancangan Antarmuka Lupa password

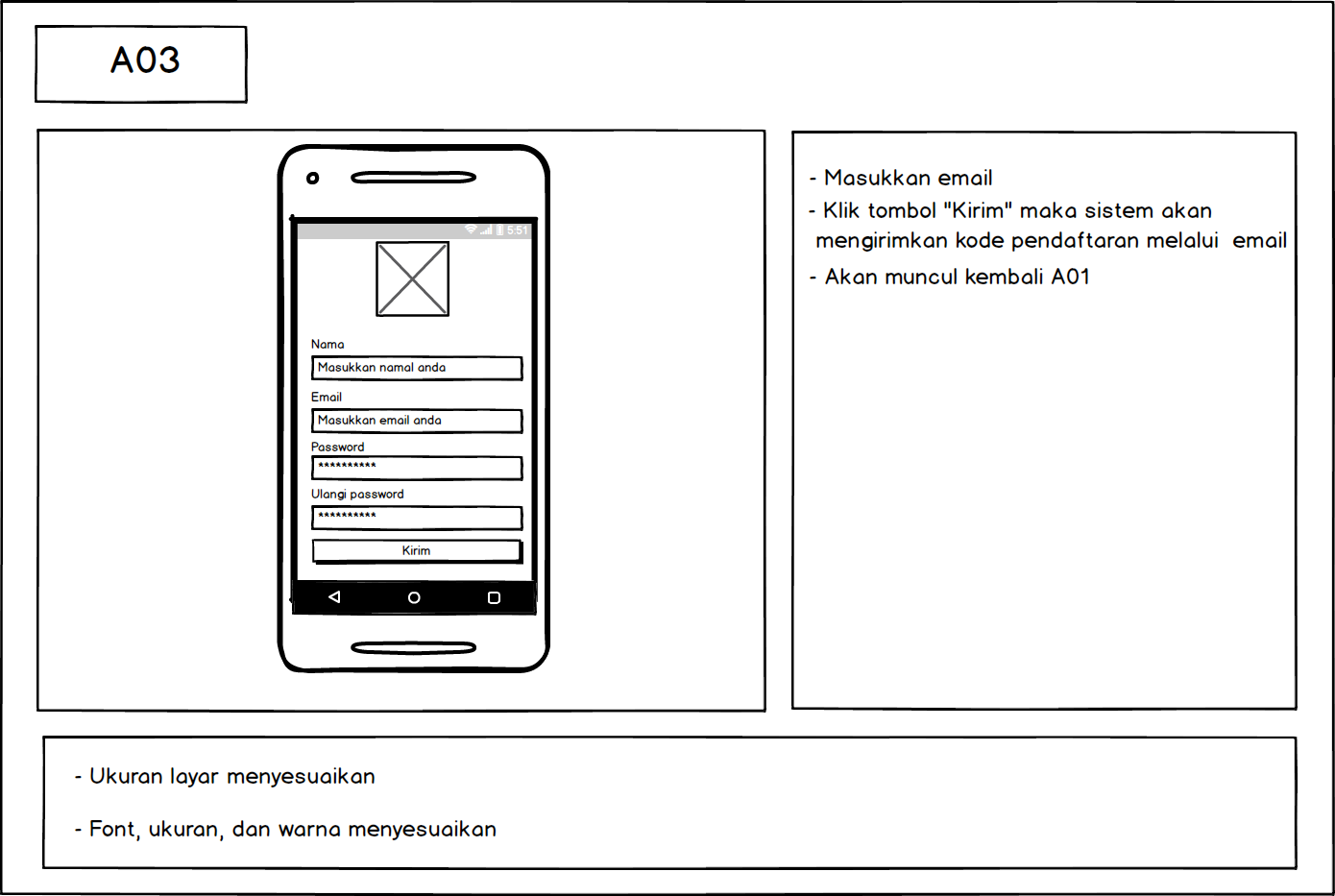
Perancangan antarmuka lupa *password* merupakan rancangan yang akan muncul saat pengguna ingin melakukan set ulang pada *password* akun yang hendak dikirim melalui email konfirmasi. Adapun perancangan antarmuka lupa *password* dapat dilihat pada Gambar 45.



Gambar Perancangan Antarmuka Lupa *password*

1. Perancangan Antarmuka Daftar

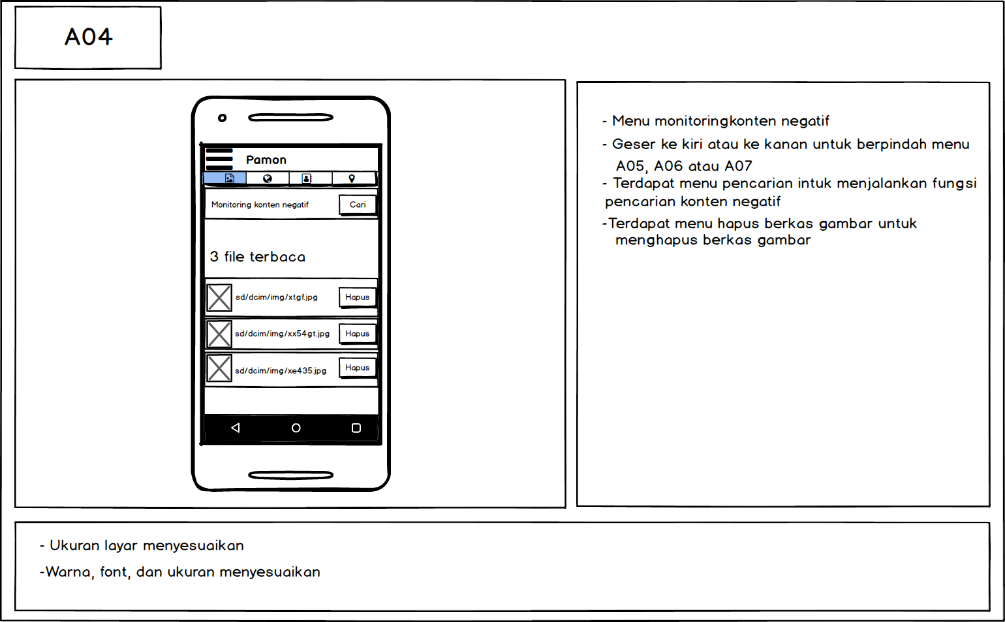
Perancangan antarmuka daftar merupakan rancangan yang akan muncul saat pengguna ingin melakukan pendaftaran atau membuat akun untuk dapat masuk ke dalam aplikasi. Adapun perancangan antarmuka daftar dapat dilihat pada Gambar 46.



Gambar Perancangan Antarmuka Daftar

1. Perancangan Antarmuka *Monitoring* Konten Negatif

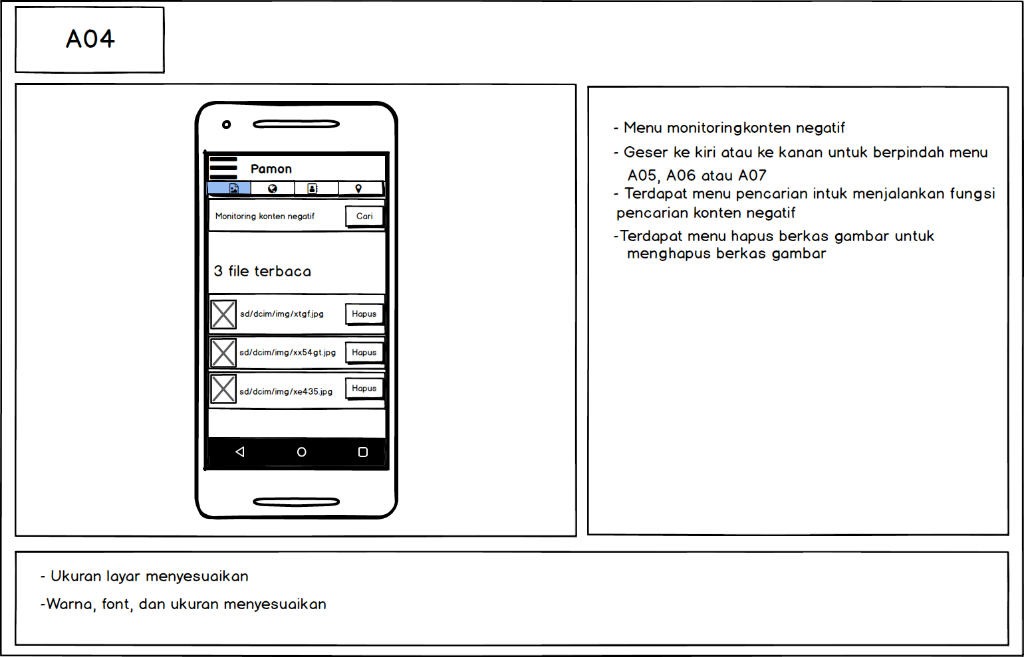
Perancangan antarmuka *Monitoring* Konten Negatif merupakan rancangan yang akan muncul saat pengguna memilih menu *Monitoring* Konten Negatif untuk dapat melihat hasil pencarian konten negatif yang nungkin terdapat di *smartphone* anak. Adapun perancangan antarmuka *Monitoring* Konten Negatif dapat dilihat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



Gambar Perancangan Antarmuka *Monitoring* Konten Negatif

1. Perancangan Antarmuka Penghapusan berkas gambar

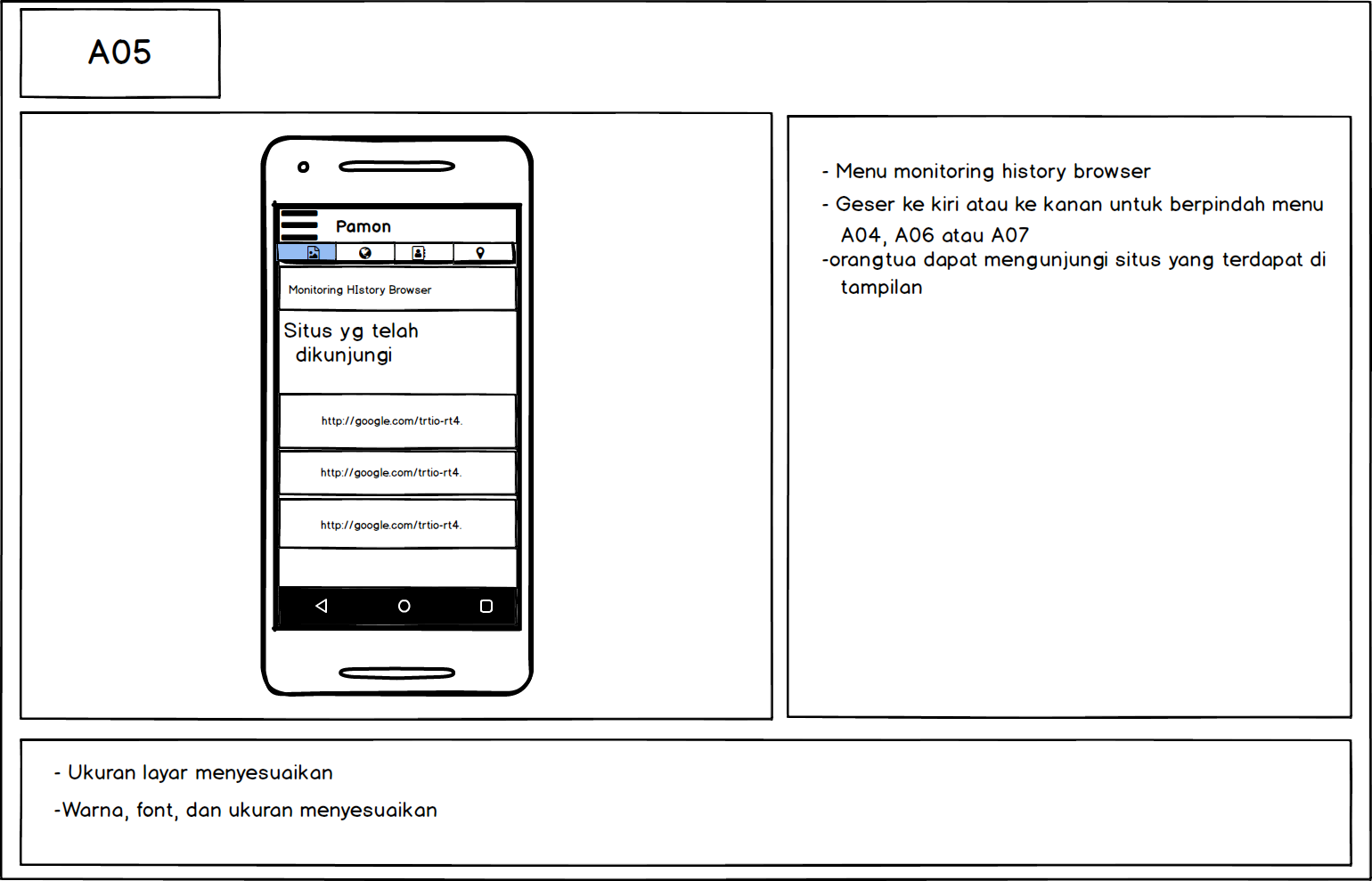
Perancangan antarmuka penghapusan berkas gambar merupakan rancangan yang akan muncul saat pengguna memilih menu penghapusan berkas gambar untuk dapat melakukan fungsi penghapusan berkas gambar pada *smartphone* anak. Menu penghapusan berkas gambar berada pada menu *monitoring konten negatif.* Adapun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



Gambar Perancangan Antarmuka Penghapusan berkas gambar

1. 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 Perancangan Antarmuka *Monitoring History Browser*

1. Perancangan Antarmuka *Monitoring* Lokasi

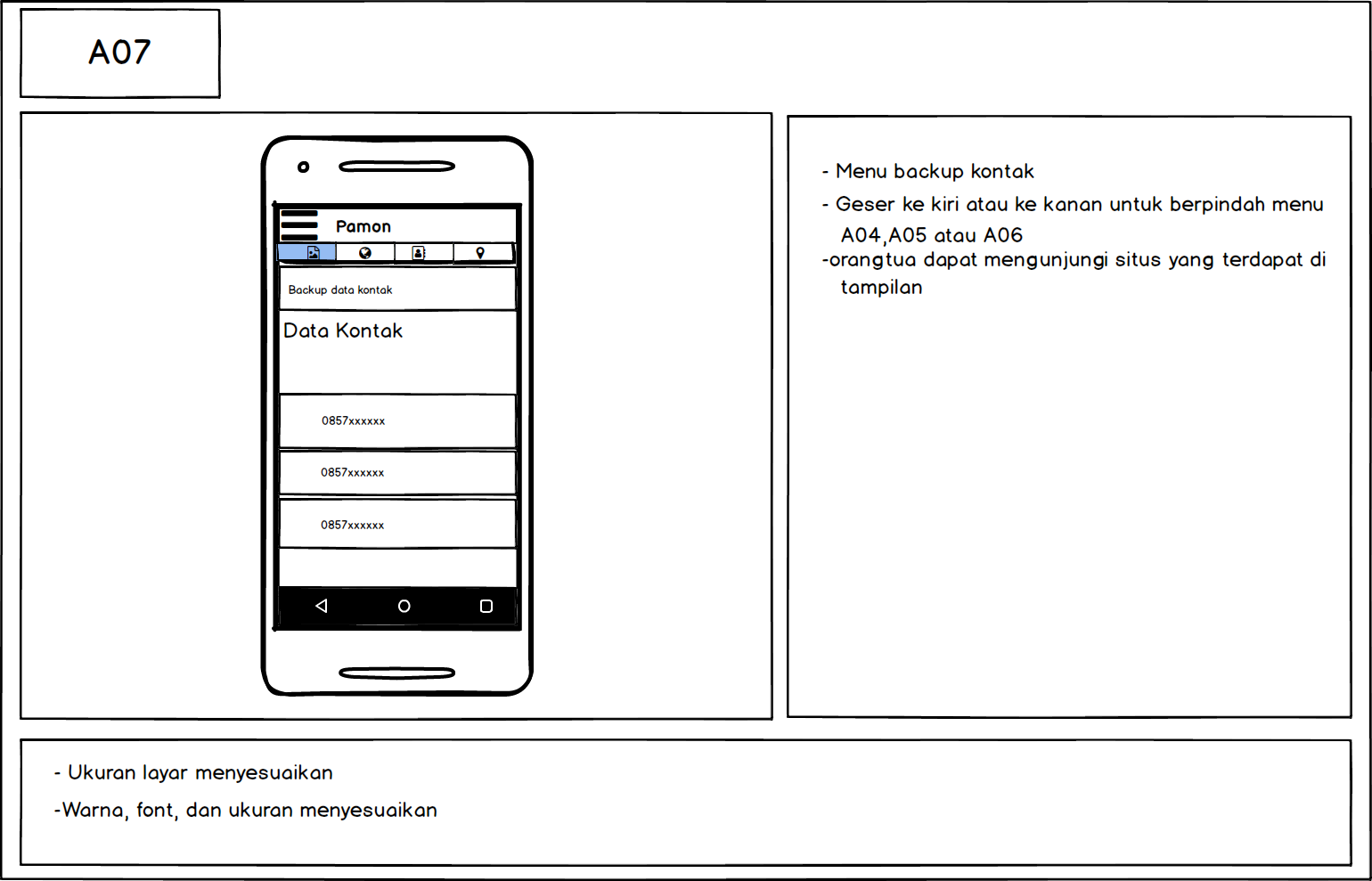
Perancangan antarmuka *Monitoring* Lokasi merupakan rancangan yang akan muncul saat pengguna memilih menu *Monitoring* Lokasi untuk dapat melihat posisi anak. Adapun perancangan antarmuka *Monitoring* Lokasi dapat dilihat pada Gambar 69 berikut.



Gambar Perancangan Antarmuka *Monitoring* Lokasi

1. Perancangan Antarmuka *Backup* Kontak

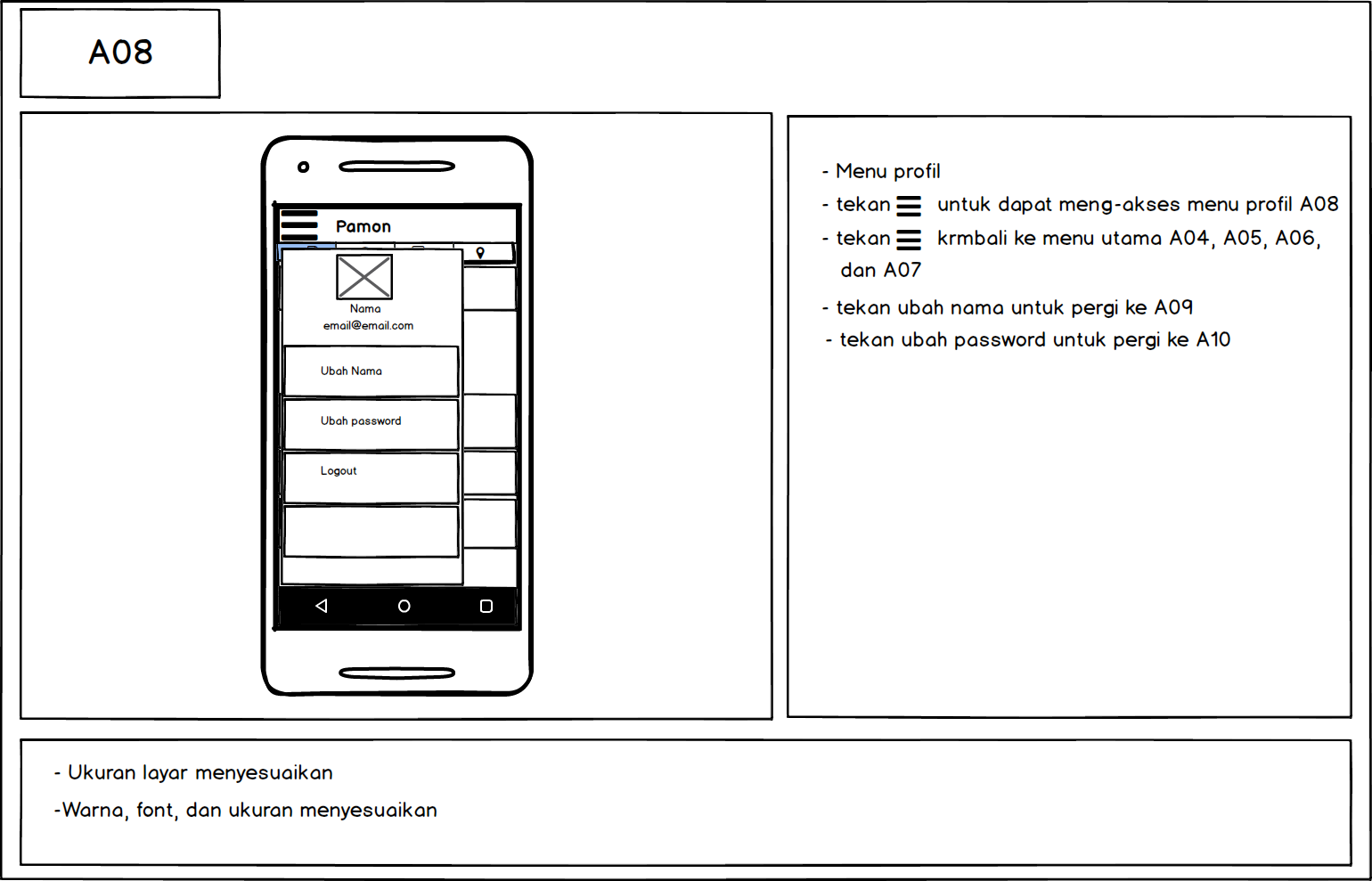
Perancangan antarmuka *backup* kontak merupakan rancangan yang akan muncul saat pengguna memilih menu *backup* kontak untuk dapat melihat kontak pada *smartphone* anak. Adapun perancangan antarmuka *backup* kontak dapat dilihat pada Gambar 70 berikut.



Gambar Perancangan Antar Muka *Backup* Kontak

1. 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 Perancangan Antarmuka Profil

1. Perancangan Antarmuka Ubah nama

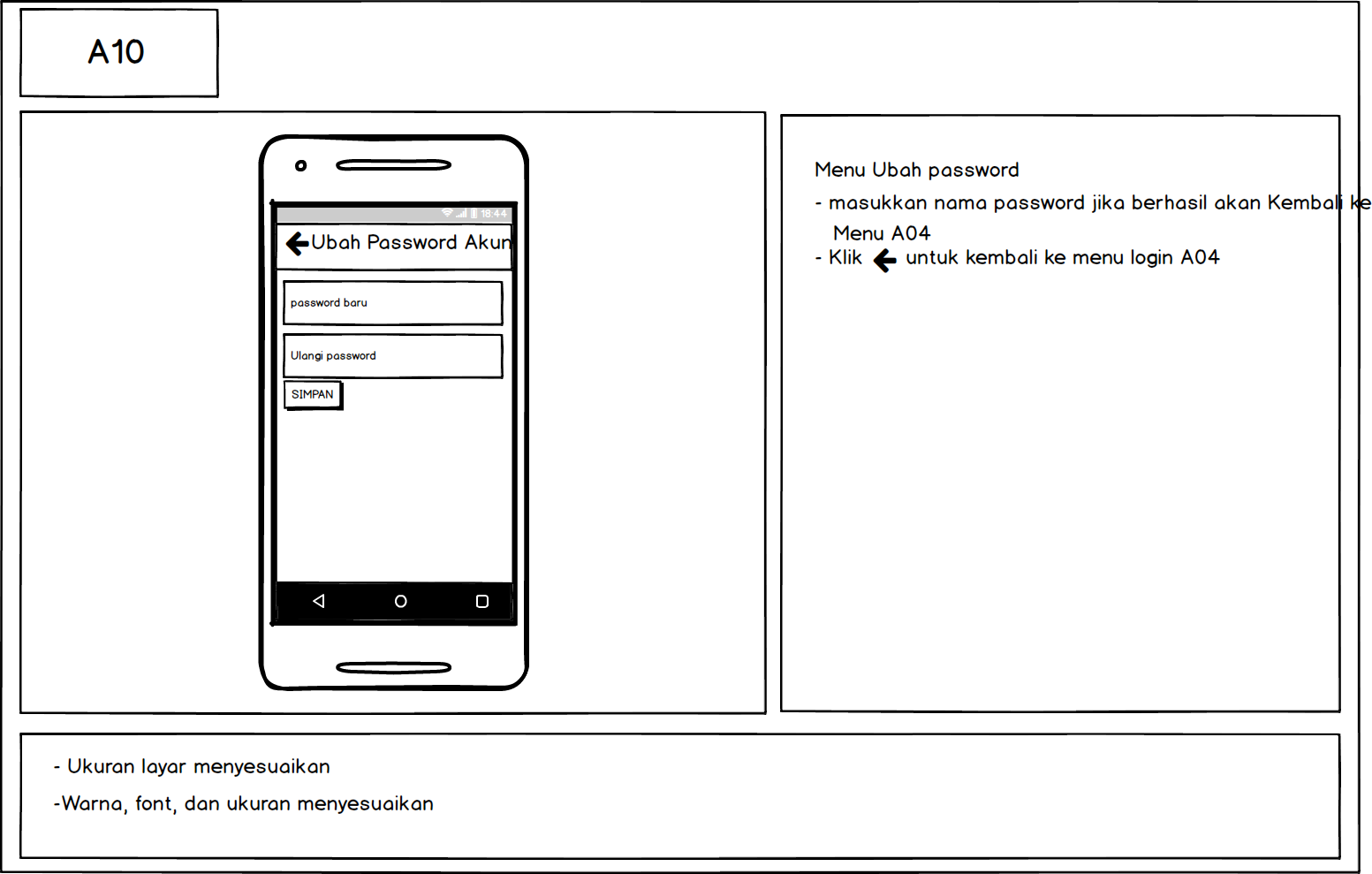
Perancangan antarmuka Ubah nama merupakan rancangan yang akan muncul saat pengguna memilih menu profil untuk dapat melihat profil dari akun kemudian memilih ubah nama untuk menyunting nama dari akun. Adapun perancangan antarmuka Ubah nama dapat dilihat pada Gambar 72.



Gambar Perancangan Antar Muka Ubah Nama

1. 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 Perancangan Antarmuka Ubah *Password*

### 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 . Perancangan Pesan

|  |  |
| --- | --- |
| **Kode** | **Pesan** |
| P01 | Masukkan *email* |
| P02 | Masukkan *password* |
| P03 | *Email* dan *password* yang anda gunakan salah |
| P04 | Masukkan No Telepon |
| P05 | Masukkan *Email* |
| P06 | Masukkan Nama |
| P07 | Masukkan *Password* |
| P08 | …*File* 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 |

### Perancangan Jaringan Semantik

Jaringan semantik merupakan gambaran tentang pengetahuan grafis yang menunjukan 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 Jaringan Semantik

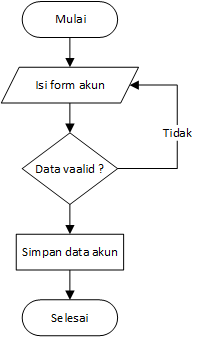
### Perancangan Prosedural

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 Prosedural Daftar

1. Prosedural Lupa *Password*

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



Gambar Prosedural Lupa *Password*

1. Prosedural *Login*

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



Gambar Prosedural *Login*

1. Prosedural *Log Out*

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



Gambar Prosedural *Log Out*

1. Prosedural Profil

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



Gambar Prosedural Profil

1. Prosedural Ubah Nama

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



Gambar Prosedural Ubah Nama

1. Prosedural Ubah Password

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



Gambar Prosedural Ubah *Password*

1. Prosedural *Monitoring* Konten Negatif

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



Gambar Prosedural *Monitoring* Konten Negatif

1. Prosedural Penghapusan Berkas Gambar

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



Gambar Prosedural Penghapusan Berkas Gambar

1. Prosedural *Monitoring* Lokasi Anak

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



Gambar Prosedural *Monitoring* Lokasi Anak

1. Prosedural Daftar *Monitoring History Browser*

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



Gambar Prosedural *Monitoring History Browser*

1. Prosedural *Backup Data* Kontak

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



Gambar Prosedural *Backup Data* Kontak

1. Prosedural Notifikasi

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



Gambar Prosedural Notifikasi

1. Prosedural Notifikasi *Monitoring* konten Negatif

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



Gambar Prosedural Notifikasi *Monitoring* Konten Negatif

1. Prosedural Notifikasi *Monitoring History Browser*

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



Gambar Prosedural Notifikasi *Monitoring History Browser*

1. Prosedural Notifikasi Lokasi Anak

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



Gambar 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 Controling 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. Mukhlasin, "Mengenal Restful Web Service?," 8 December 2015. [Online]. Available: http://www.hafidmukhlasin.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]. |