

**PENGEMBANGAN APLIKASI PEMBELAJARAN PENGELOLAAN
SAMPAH DAUR ULANG DENGAN MENGGUNAKAN IBM
COGNITIVE SERVICE DAN YOUTUBE API BERBASIS ANDROID**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Andri Suranta Ginting
NIM: 155150200111229



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG

2019

PENGESAHAN

PENGEMBANGAN APLIKASI PEMBELAJARAN PENGELOLAAN SAMPAH DAUR
ULANG DENGAN MENGGUNAKAN IBM COGNITIVE SERVICE DAN YOUTUBE DATA
API BERBASIS ANDROID

SKRIPSI

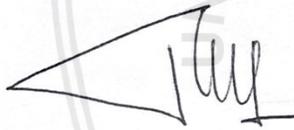
Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Andri Suranta Ginting
NIM: 155150200111229

Skrripsi ini telah diuji dan dinyatakan lulus pada
02 Januari 2019

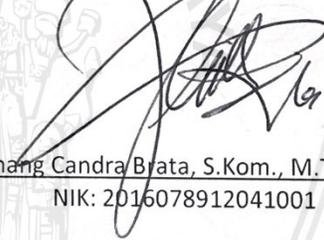
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I



Adam Hendra Brata, S.Kom., M.T., M.Sc.
NIK: 2016079001051001

Dosen Pembimbing II



Komang Candra Brata, S.Kom., M.T., M.Sc.
NIK: 2016078912041001

Mengetahui

Ketua Jurusan Teknik Informatika



Asli Rizki Kurniawan, S.T, M.T, Ph.D
NIP. 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 17 Desember 2018



Andri Suranta Ginting
NIM:15515020011229

KATA PENGANTAR

Segala puji bagi Allah SWT atas limpahan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “PENGEMBANGAN APLIKASI PEMBELAJARAN PENGELOLAAN SAMPAH DAUR ULANG MENGGUNAKAN IBM COGNITIVE SERVICE DAN YOUTUBE API BERBASIS ANDROID”.

Untuk kesempatan ini penulis juga menyampaikan rasa terima kasih kepada pihak – pihak yang telah membantu penulis selama penyusunan skripsi, diantaranya:

1. Allah SWT yang telah memberi kemudahan dalam semua proses penulisan skripsi ini.
2. Kepada kedua orang tua penulis, yaitu Bapak Junedi Ginting dan Ibu Dosmerita Purba beserta keluarga besar yang selalu memberikan segala masukan, doa, motivasi dan semangat yang tidak terputus.
3. Bapak Adam Hendra Brata, S.Kom., M.T., M.Sc.. selaku dosen pembimbing I yang telah meluangkan waktu untuk memberikan masukan, ilmu, serta saran yang bermanfaat dalam proses penyelesaian skripsi ini.
4. Bapak Komang Candra Brata, S.Kom., M.T., M.Sc selaku dosen pembimbing II yang juga telah meluangkan waktu untuk memberikan masukan, ilmu, serta saran yang bermanfaat dalam proses penyelesaian skripsi ini.
5. Bapak Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D, Bapak Ir. Heru Nurwasito, M.Kom, Bapak Suprpto,S.T, M.T, dan Bapak Edy Santoso, S.Si, M.Kom selaku Dekan, Wakil Dekan 1, Wakil Dekan 2, dan Wakil Dekan 3 Fakultas Ilmu Komputer, Universitas Brawijaya.
6. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D dan Bapak Agus Wahyu Widowo, S.T, M.Cs selaku Ketua Jurusan Teknik Informatika dan Ketua Program Studi Teknik Informatika Fakultas Ilmu Komputer, Universitas Brawijaya.
7. Seluruh Dosen Fakultas Ilmu Komputer, Universitas Brawijaya atas kesediaannya dalam mengajarkan dan membagikan ilmu yang bermanfaat

bagi penulis.

8. Team Spesialku (Fajri dan Ajeng), rekan-rekan Laboratorium Siskombot, Student Employee MGM, wisemansay, dan teman kontrakan engas yang membantu memberikan ilmu dalam implementasi dari penelitian ini.
9. Semua pihak yang tidak semuanya bisa dituliskan disini yang terlibat secara langsung maupun tidak langsung dalam proses pengerjaan skripsi maupun sebagai pemberi semangat dan motivasi.

Penulis menyadari masih banyak kekurangan dalam penyusunan skripsi ini baik dalam teknik penyajian materi maupun pembahasan. Demi kesempurnaan penelitian skripsi ini, saran dan kritik yang sifatnya membangun sangat penulis harapkan. Semoga karya tulis ini bermanfaat dan dapat memberikan sumbangan yang berarti bagi pihak yang membutuhkan.

Malang, 17 Desember 2018

Penulis

andris.ginting@gmail.com

ABSTRAK

Andri Suranta Ginting, Pengembangan Aplikasi Pembelajaran Pengelolaan Sampah Daur Ulang Menggunakan IBM Cognitive Service dan Youtube API Berbasis Android

Pembimbing: Adam Hendra Brata, S.Kom., M.T., M.Sc. dan Komang Candra Brata, S.Kom., M.T., M.Sc.

Pada tahun 2018, Malang menghasilkan 659,88 ton sampah dalam sehari dan jumlah tersebut akan terus bertambah seiring bertambahnya populasi penduduk. Indonesia sendiri menempati peringkat kedua sebagai penghasil sampah plastik terbesar ke laut setelah Tiongkok. Penanganan sampah plastik biasanya dilakukan dengan pembakaran namun hal ini dapat menghasilkan zat-zat beracun yang berbahaya bagi makhluk hidup. Dalam penelitian ini penulis mengembangkan sebuah aplikasi yang berbasis android dengan menggunakan IBM Cognitive Service yang mampu menganalisis objek untuk membantu memilah sampah yang dapat didaur ulang dengan teknologi *image recognition*. Teknologi IBM Cognitive Service memiliki kemampuan *image captioning* dengan tingkat akurasi 89%. Aplikasi ini juga menggunakan youtube api untuk membantu pengguna menampilkan cara pengelolaan sampah yang dapat didaur ulang dan juga sebagai pendukung fitur *search content dan playlist* yang diintegrasikan dengan IBM Watson Visual Recognition untuk menganalisis objek.

Pengujian yang dilakukan dalam pengembangan aplikasi ini menggunakan SUS (*System Usability Scale*) dengan menggunakan 10 pertanyaan sebagai tolak ukur penilaian, metode *whitebox testing* dan *blackbox testing*. Hasil yang dihasilkan dengan *System Usability Scale (SUS)* adalah sebesar 79.5. Hal ini menunjukkan bahwa aplikasi ini dikategorikan *acceptable* dalam *acceptable ranges*, dikategorikan *grade B* dalam *grade scales*, dan dikategorikan *good* dalam *adjective ratings*. Hasil pengujian dengan metode *whitebox testing* dan *blackbox testing* dikatakan valid karena sudah memenuhi kebutuhan.

Kata kunci: Android, IBM Cognitive Service, image recognition, Youtube API, Dalang, Usability

ABSTRACT

Andri Suranta Ginting, Learning Application Development Recycling Waste Using IBM Cognitive Service and Youtube API Based on Android

Supervisor: Adam Hendra Brata, S.Kom., M.T., M.Sc. dan Komang Candra Brata, S.Kom., M.T., M.Sc.

In the year 2018, the Malang region has produced 659,88 tonnes of waste on a daily basis and the rate gradually increases directly proportional to the number of the local population. Indonesia is ranked the second biggest contributor to the amount of plastic waste produced worldwide, second to China. Plastic waste is usually managed by means of burning, but this leads to fumes that are harmful to humans and wildlife. This research aims to apply a IBM Cognitive Service that able to analyze objects to manage recyclable waste using image recognitions technology. IBM Cognitive Service has ability to caption images with an accuracy rate of 89%. This application uses YouTube API to show users on how to manage recycable waste and to support a content search feature and a playlist that is integrated with IBM Watson Visual Recognitions which has the ability to analyze objects.

The testing methods used by this application development is called SUS (System Usability Scale) which uses 10 questions to measure the score, whitebox testing method, and blackbox testing method. The results from System Usability Scale (SUS) measure at 79.5. This shows that the application is categorized as acceptable in acceptable ranges, categorized with a B grade in grade scales, and categorized as good in an adjective rating. The results from the whitebox and blackbox testing methods are deemed valid, because they have fulfilled the necessary requirements.

Keywords: Android, IBM Cognitive Service, Image Recognition, Youtube API, Dalang, Usability

DAFTAR ISI

<i>PENGESAHAN</i>	<i>ii</i>
<i>PERSETUJUAN</i>	<i>iii</i>
<i>PERNYATAAN ORISINALITAS</i>	<i>iv</i>
<i>KATA PENGANTAR</i>	<i>v</i>
<i>ABSTRAK</i>	<i>vii</i>
<i>ABSTRACT</i>	<i>viii</i>
<i>DAFTAR ISI</i>	<i>ix</i>
<i>DAFTAR TABEL</i>	<i>xii</i>
<i>DAFTAR GAMBAR</i>	<i>xiii</i>
<i>DAFTAR LAMPIRAN</i>	<i>xiv</i>
BAB 1 <i>Pendahuluan</i>	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	3
1.3. Tujuan	3
1.4. Manfaat	3
1.5. Batasan Masalah	4
1.6. Sistematika Pembahasan	4
BAB 2 <i>Landasan Kepustakaan</i>	5
2.1 Tinjauan Pustaka	5
2.2 Model Pengembangan Perangkat Lunak	5
2.2.1 Software Development Lifecycle: Prototyping	5
2.3 UML (Unified Modelling Language)	7
2.3.1 Use Case Diagram	7
2.4 Pemrograman Android dengan Kotlin	9
2.5 Android	10
2.5.1 Android Activity Lifecycle	10
2.5.2 Android Fragment Lifecycle	11
2.6 Youtube Data API	13
2.7 IBM Watson Visual Recognition	14
2.8 Android Framework Camera API	14



2.9	Firestore	15
2.10	JSON (JavaScript Object Notation)	17
2.11	Teori Pengujian	18
2.11.1	Blackbox Testing	18
2.11.2	WhiteBox Testing	19
2.11.3	System Usability Scale (SUS)	19
2.12	Purposive Sampling	19
BAB 3 Metodologi Penelitian		20
3.1	Strategi dan Rancangan Penelitian	20
3.1.1	Studi Literatur	21
3.1.2	Pengumpulan Data	21
3.1.3	Analisis Kebutuhan	21
3.1.4	Perancangan	22
3.1.5	Implementasi	22
3.1.6	Pengujian dan Analisis Hasil	23
3.1.7	Penarikan Kesimpulan dan Saran	24
BAB 4 Analisis Kebutuhan		25
4.1	Gambaran Umum Aplikasi	25
4.2	Identifikasi Aktor	25
4.3	Analisis Kebutuhan Iterasi Pertama	25
4.3.1	Kebutuhan Fungsional	25
4.3.2	Diagram Use Case	26
4.3.3	Skenario Use Case	27
4.4	Analisis Kebutuhan Iterasi Kedua	30
4.4.1	Kebutuhan Fungsional	30
4.4.2	Use Case Diagram	30
4.4.3	Skenario Use Case	31
4.5	Analisis Data	32
BAB 5 Perancangan dan Implementasi		33
5.1	Perancangan	33
5.1.1	Perancangan Arsitektur Sistem	33
5.1.2	Class Diagram	34
5.1.3	Sequence Diagram	38
5.1.4	Perancangan Komponen	40
5.1.5	Perancangan Basis Data	41
5.1.6	Perancangan Antarmuka	44
5.2	Implementasi Sistem	47
5.2.1	Spesifikasi Sistem	48
5.2.2	Implementasi Data	49
5.2.3	Implementasi Kode Program	50



5.2.4 Implementasi Antarmuka	53
BAB 6 Pengujian	56
6.1 Pengujian Unit	56
6.1.1 Pengujian Unit Klas AsyncTask operasi doInBackground(file)..	56
6.1.2 Pengujian unit klas VideoPlayerActivity operasi initializeYoutubeFrag()	58
6.1.3 Pengujian unit klas ImpResultPresenter operasi getListVideo().	60
6.2 Pengujian Validasi	61
6.2.1 Pengujian Validasi	62
6.2.2 Hasil Pengujian Validasi	64
6.3 Pengujian Usabilitas	66
6.3.1 Prosedur Pengujian Usabilitas	67
6.3.2 Analisis Data dan Hasil Pengujian Usabilitas	68
6.4 Analisis Hasil Pengujian	70
6.4.1 Pengujian Unit	71
6.4.2 Pengujian Validasi	71
6.4.3 Pengujian Usabilitas	71
BAB 7 Penutup	72
7.1 Penarikan Kesimpulan	72
7.2 Saran	72
Daftar Pustaka	73
LAMPIRAN	75

DAFTAR TABEL

Tabel 2.1 Daur Hidup <i>Activity</i>	11
Tabel 2.2 Fragment Lifecycle	12
Tabel 2.3 YouTube Data API.....	13
Tabel 2.4 Tipe Intent Android Camera	15
Tabel 4.1 Penjelasan Aturan Penomoran Kebutuhan	26
Tabel 4.2 Kebutuhan Fungsional Iterasi Pertama	26
Tabel 4.3 Skenario Use Case Mengambil Objek Menggunakan Kamera Iterasi Pertama	27
Tabel 4.4 Skenario Use Case Memilih Objek Menggunakan Gallery Iterasi Pertama	28
Tabel 4.5 Skenario Use Case Melihat Hasil Analisis Objek Iterasi Pertama	28
Tabel 4.6 Skenario Use Case Melihat Daftar Video Iterasi Pertama.....	29
Tabel 4.7 Skenario Use Case Melihat Harga Sampah Iterasi Pertama.....	29
Tabel 4.8 Skenario <i>Use Case</i> Memutar Video Iterasi Pertama.....	30
Tabel 4.9 Kebutuhan Fungsional Iterasi Kedua.....	30
Tabel 4.10 Scenario Use Case Riwayat Pencarian.....	31
Tabel 5.1 Perancangan Basis Data IBM Visual Recognition.....	41
Tabel 5.2 Penjelasan Skema Basis Data IBM Visual Recognition.....	42
Tabel 5.3 Perancangan Basis Data Youtube Data API	43
Tabel 5.4 Penjelasan Perancangan Antarmuka Memutar Video Iterasi Pertama	45
Tabel 5.5 Penjelasan Perancangan Antarmuka Tampilan Awal Iterasi Pertama..	45
Tabel 5.6 Perancangan Antarmuka Melihat Daftar Video.....	46
Tabel 5.7 Penjelasan Perancangan Antarmuka Tampilan Awal	47
Tabel 5.8 Spesifikasi Perangkat Keras	48
Tabel 5.9 Spesifikasi Perangkat Lunak.....	48
Tabel 5.10 Spesifikasi Sistem Operasi	49
Tabel 5.11 Implementasi Basis Data Youtube Data API	49
Tabel 6.1 Tabel Kasus Uji Coba Mengambil Objek Menggunakan Kamera.....	62
Tabel 6.2 Kasus Uji Coba Mengambil Objek Menggunakan <i>Gallery</i>	62
Tabel 6.3 Kasus Uji Coba Memutar Video	63
Tabel 6.4 Kasus Uji Coba Menampilkan Daftar Video.....	63
Tabel 6.5 Kasus Uji Coba Menampilkan Harga Sampah.....	63
Tabel 6.6 Kasus Uji Menganalisis Objek Lain	64
Tabel 6.7 Tabel Hasil Pengujian Validasi.....	64
Tabel 6.8 Parameter Bobot Nilai SUS	67
Tabel 6.9 Daftar Pertanyaan SUS	68
Tabel 6.10 Hasil Skor Pertanyaan SUS.....	69
Tabel 6.11 Hasil Konversi Nilai SUS	70



DAFTAR GAMBAR

Gambar 2.1 Aliran SDLC Prototype sumber: Presman 2010	6
Gambar 2.2 Use Case Sumber: Pressman 2010.....	8
Gambar 2.3 Simbol Aktor	8
Gambar 2.4 Contoh Association Relationship	8
Gambar 2.5 Contoh Extends Relationship.....	9
Gambar 2.6 Contoh Depends On	9
Gambar 2.7 Activity <i>Life Cycle</i> Sumber: Google.....	10
Gambar 2.8 Fragment Life Cycle Sumber: Google.....	12
Gambar 2.9 IBM Watson Visual Recognition.....	14
Gambar 2.10 JSON Object	17
Gambar 2.11 JSON Array	18
Gambar 2.12 JSON Value.....	18
Gambar 3.1 Metodologi Penelitian	20
Gambar 4.1 Diagram Use Case Sistem	27
Gambar 4.2 <i>Use Case</i> Diagram Iterasi Kedua	31
Gambar 5.1 Arsitektur MVP Android	33
Gambar 5.2 Diagram Arsitektur MVP.....	34
Gambar 5.3 Class Diagram.....	35
Gambar 5.4 Informasi Class Activity.....	36
Gambar 5.5 Informasi Kelas Model.....	37
Gambar 5.6 Informasi Kelas Presenter.....	38
Gambar 5.7 Sequence Diagram Mengambil Foto Dari Gallery.....	38
Gambar 5.8 Sequence Diagram Menganalisis Objek Dari kamera	39
Gambar 5.9 Sequence Diagram Memutar Video	40
Gambar 5.10 Rancangan Antarmuka Halaman Memutar Video Iterasi Pertama	44
Gambar 5.11 Perancangan Antarmuka Tampilan Awal Iterasi Pertama.....	45
Gambar 5.12 Rancangan Antarmuka Melihat Daftar Video.....	46
Gambar 5.13 Perancangan Antarmuka Tampilan Awal	47
Gambar 5.14 Implementasi Antarmuka Halaman Mengambil Foto.....	54
Gambar 5.15 Implementasi Antarmuka Halaman Melihat Daftar Video.....	55
Gambar 5.16 Implementasi Antarmuka Video Player.....	55
Gambar 6.1 Rating dan Skala Konversi SUS Sumber: measuringu.com	67

DAFTAR LAMPIRAN

Lampiran 1 Lampiran Kuisisioner SUS	78
Lampiran 2 Evaluasi Prototyping.....	79
Lampiran 3 Pertanyaan Wawancara	81
Lampiran 4 Daftar Harga Sampah	83



BAB 1 PENDAHULUAN

1.1. Latar Belakang

Menurut data Dinas Lingkungan Hidup (DLH) kota Malang tahun 2018, kota Malang menghasilkan 659,88 ton sampah dalam sehari dan jumlah itu akan terus bertambah seiring bertambahnya populasi penduduk. Berdasarkan jenisnya, jumlah sampah anorganik mencapai 35% atau sekitar 230,96 ton dan sampah organik mencapai 65% dengan total 428,92 ton (Wasto, 2018). Menurut Agoes (2018) berkurangnya volume sampah yang terbuang di TPA merupakan berkat upaya pengelolaan yang selama ini dilakukan. Salah satu diantaranya dengan pemberdayaan masyarakat melalui daur ulang sampah. Cara ini dianggap tak hanya mengurangi sampah tetapi juga membantu perekonomian warga setempat khususnya kota Malang. Banyak sampah yang dihasilkan masyarakat Malang dapat dimanfaatkan kembali dalam bentuk lain. Beberapa di antaranya menjadi tas plastik, dompet dari bungkus permen atau bungkus mi dan sebagainya.

Saat ini, Indonesia masuk dalam peringkat kedua di dunia penghasil sampah plastik ke laut setelah Tiongkok. Penanganan sampah plastik dilakukan dengan cara mendaur ulang, pembakaran (dengan menggunakan *incinerator*) dan menguburkan (*landfill*). Pembakaran sampah plastik menghasilkan zat-zat beracun yang berbahaya bagi makhluk hidup, sementara cara penguburan tidak efektif karena plastik sangat sulit terdegradasi. Namun, kebutuhan dalam negeri akan bahan baku industri plastik daur ulang sebesar 789 ribu ton/tahun. Cara daur ulang merupakan alternatif terbaik untuk menangani sampah plastik. Jika sampah plastik terkelola dengan baik, Indonesia tidak memerlukan impor bahkan mendorong pertumbuhan industri daur ulang biji plastik (idntimes.com, 2017).

Menurut Dirjen Pengelolaan Sampah Limbah dan Bahan Beracun Berbahaya (PSLB3), Tuti hendrawati Mintarsih, pemerintah mendorong pengelolaan bak sampah khususnya dengan melibatkan generasi muda sebagai generasi kreatif yang diharapkan mampu menyulap sampah anorganik menjadi benda-benda bernilai jual. Karena potensi timbulan sampah plastik sebesar 8.960.000 ton/tahun juga menjadi potensi besar untuk bahan baku industri daur ulang dan industri kreatif. Dengan begitu bukan hanya kota telah mengurangi jumlah sampah dibuang ke tempat pembuangan akhir tetapi dapat meningkatkan pendapatan masyarakat (Tuti, 2017).

Salah satu solusi yang ditawarkan oleh Pemerintah Kota Malang ialah menerapkan sistem Bank Sampah Masyarakat (BSM) untuk meminimalisir jumlah sampah tersebut. Dengan adanya BSM, sampah yang ada dimasyarakat kurang

lebih sekitar 72 jenis sampah dapat bernilai ekonomi dan kapasitas pengolahan 5 ton mampu direduksi setiap harinya. Artinya, 2,1% sampah dapat direduksi melalui proses daur ulang (Wasto,2018). Berdasarkan hasil survey yang dilakukan peneliti terhadap mahasiswa Fakultas Ilmu Komputer, Universitas Brawijaya, 54,7% responden belum pernah mendaur ulang sampahnya. Dari 75 responden yang belum pernah mendaur ulang sampah sebanyak 75,6% tidak mengetahui ide atau cara mendaur ulang sampah yang baik. Dengan begitu dibutuhkan sebuah aplikasi pembelajaran untuk memenuhi kebutuhan responden. Setelah aplikasi selesai dibangun, maka akan diuji dengan pengujian usabilitas. Pengujian usabilitas berdasarkan *International Organization for Standarization* yang digunakan untuk mengetahui apakah aplikasi yang dibangun sudah sesuai dengan nilai efektifitas, efisien dalam penggunaan dan juga kepuasan dalam menggunakan aplikasi Dalang.

YouTube adalah sebuah platform *video sharing* dimana penggunanya dapat memuat, menonton dan berbagi video secara gratis. YouTube menyediakan *API (Application Programming Interface)* yang membantu developer mengembangkan aplikasi berbagi video berkualitas tinggi. YouTube terdiri dari banyak fitur yang dapat dipadukan sesuai dengan kebutuhan anda. Dalam pengembangan aplikasi pembelajaran ini fitur yang mungkin digunakan adalah *Search Content* dan *Playlist*. Fitur *search content* digunakan untuk mencari daftar video yang tersedia dan *playlist* digunakan untuk menampilkan daftar video sesuai dengan keinginan pengguna. *Search Content* akan diintegrasikan dengan *IBM Watson* yang memiliki kemampuan *image captioning* dengan tingkat akurasi hingga 89%. (*Watson Multimodal Algorithms and Engines Group IBM T.J. Watson Research Center, 2018*). IBM Watson merupakan sebuah layanan yang dapat digunakan untuk menganalisis gambar atau objek ataupun wajah. Pengembang yang ingin menggunakan IBM Watson juga dapat mengidentifikasi buah segar, sayuran ataupun makanan.

Aplikasi ini dibangun untuk dapat mengetahui sampah yang dapat di daur ulang dan akan memberikan arahan cara mendaur ulang sampah tersebut. Oleh karena uraian diatas, penelitian yang dilakukan ini bermaksud untuk mengembangkan aplikasi perangkat bergerak yang dapat membantu memilah sampah yang dapat di daur ulang dengan memanfaatkan *IBM Watson Service* yang merupakan sebuah teknologi *image recognition* untuk mengenali objek dan akan mengirimkan data pengolahan data ke *Youtube Data API* yang nantinya akan menampilkan cara pengelolaan sampah yang dapat di daur ulang. Dengan dibuatnya aplikasi ini harapannya dapat memudahkan daur ulang sampah dengan arahan yang baik, semakain efektif, efisien dan informatif. Dengan demikian aplikasi ini dapat dimaksimalkan dalam proses pembelajaran.

1.2. Rumusan Masalah

Berdasarkan latar belakang tersebut, rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana hasil analisis kebutuhan yang digunakan dalam pengembangan aplikasi pembelajaran daur ulang sampah dengan IBM Watson dan YouTube Data API?
2. Bagaimana hasil rancangan dan implementasi aplikasi pengembangan aplikasi pembelajaran daur ulang sampah dengan IBM Watson dan YouTube Data API berdasarkan analisis kebutuhan yang telah dilakukan sebelumnya?
3. Bagaimana penerapan IBM Watson dan YouTube Data API dalam pengembangan aplikasi pembelajaran daur ulang sampah?
4. Bagaimana aplikasi dapat memudahkan pengguna dengan melakukan pengujian dapat memenuhi kebutuhan dan perancangan sistem yang dibangun?

1.3. Tujuan

Adapun tujuan dalam penelitian ini adalah sebagai berikut :

1. Mengetahui kebutuhan yang digunakan dalam pembangunan aplikasi pembelajaran daur ulang sampah dengan IBM Watson dan YouTube Data API.
2. Mengetahui perancangan dan implementasi berdasarkan analisis kebutuhan yang telah dilakukan sebelumnya
3. Mengetahui penerapan IBM Watson dan YouTube Data API pada pengembangan aplikasi pembelajaran daur ulang sampah
4. Mengetahui kemudahan dalam penggunaan aplikasi untuk memenuhi kebutuhan pengguna

1.4. Manfaat

Adapun manfaat yang didapatkan dari penelitian ini adalah:

1. Bagi peneliti: Memperdalam pengetahuan dan keahlian dalam mengembangkan aplikasi perangkat bergerak yang memanfaatkan *IBM Watson Service* dan *Youtube Data API*
2. Bagi Pengguna: Membantu pengguna untuk memilah sampah dan mendaur ulang sampahnya sendiri serta dapat meningkatkan pendapatan pengguna.

1.5. Batasan Masalah

Adapun batasan pada penelitian ini adalah sebagai berikut:

1. Peneliti menggunakan *IBM Watson Visual Recognition* untuk mengenali objek tanpa turut serta meneliti secara detail tentang *visual recognition*.
2. Pengujian pengenalan objek masih menggunakan botol plastik, kantong plastik dan kaleng minuman
3. Ruang lingkup penelitian ini dilakukan di Fakultas Ilmu Komputer (Filkom), Universitas Brawijaya
4. Aplikasi hanya dapat berjalan pada platform android dengan versi minimal 16

1.6. Sistematika Pembahasan

Penyusunan dokumen skripsi ini terdiri dari beberapa bab yang terdiri dari:

1. BAB 1 – PENDAHULUAN

Memuat latar belakang penelitian, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan penelitian dan sistematika pembahasan.

2. BAB 2 – LANDASAN KEPUSTAKAAN

Memuat kajian-kajian kepastakaan terkait yang digunakan sebagai referensi dalam melakukan penelitian.

3. BAB 3 – METODOLOGI PENELITIAN

Memuat alur kerja peneliti dalam proses penyelesaian permasalahan penelitian.

4. BAB 4 – ANALISIS KEBUTUHAN & PERANCANGAN

Memuat hal-hal yang terkait seputar proses penggalian kebutuhan dalam proses pengembangan sistem serta memuat hal-hal yang berkaitan dengan perancangan dan pemodelan sistem berdasarkan data yang telah didapat di tahap analisis kebutuhan.

5. BAB 5 – IMPLEMENTASI

Memuat hal-hal yang berkaitan dengan implementasi pengembangan sistem berdasarkan pemodelan yang telah dilakukan sebelumnya.

6. BAB 6 – PENGUJIAN

Memuat hal-hal yang berkaitan dengan pengujian sistem yang dilakukan oleh responden dan menganalisis hasil yang telah didapat.

7. BAB 7 – PENUTUP

Memuat hal-hal yang berisi kesimpulan dari penelitian yang telah dilakukan sebelumnya dan berisi saran dalam pengembangan penelitian selanjutnya.

BAB 2 LANDASAN KEPUSTAKAAN

Pada bab ini membahas mengenai teori yang mendasari penelitian ini, diantaranya adalah metode yang dipakai yaitu *Visual Recognition* dengan *IBM Watson Visual Recognition*, metode pengembangan perangkat lunak, teori tentang pengembangan aplikasi berbasis android, teori tentang Rekayasa Perangkat Lunak (RPL), Sistem Pembelajaran, serta konsep dari pengujian.

2.1 Tinjauan Pustaka

Penelitian ini dilakukan berdasarkan penelitian sebelumnya yang berjudul "*Self-critical Sequence Training for Image Captioning*" yang membahas tentang memberi deskripsi dari sebuah gambar. Pada penelitian tersebut, masukan yang dikirim ke sistem berupa gambar dan keluaran yang dihasilkan berupa kata-kata baik dengan menggunakan *Watson Visual Recognition*. (Watson Multimodal Algorithms and Engines Group IBM T.J. Watson Research Center).

Penelitian kedua dilakukan oleh Hui Wu, Michele Merler, Rosario Uceda-Sosa dan John Smith yang berjudul "*Learning to make better mistakes: semantics-aware visual food recognition*" pada tahun 2016. Dengan fitur *Watson Visual Recognition* yang dapat mengenali 2000+ jenis makanan, membuat sebuah model makanan untuk mempercepat waktu prediksi dan solusi khusus pada domain permasalahan yang berbeda. Dengan menggunakan pemodelan makanan, *Watson Visual Recognition* dapat melakukan pengenalan makanan lebih akurat

Penelitian ini dilakukan berdasarkan hasil survei yang dilakukan peneliti terkait kebiasaan masyarakat ketika menangani sampah. Dari 69 responden, 54 responden menjawab membuang sampah botol atau kertas ketika sudah habis dan tidak dipakai lagi. 45 responden menjawab belum pernah mendaur ulang sampah mereka sendiri. Pada penelitian ini sistem dikembangkan dalam bentuk aplikasi perangkat bergerak berbasis android. Hal ini didasarkan pada data pengguna rata-rata meluangkan waktu 60 menit setiap hari untuk berinteraksi dengan aplikasi.

2.2 Model Pengembangan Perangkat Lunak

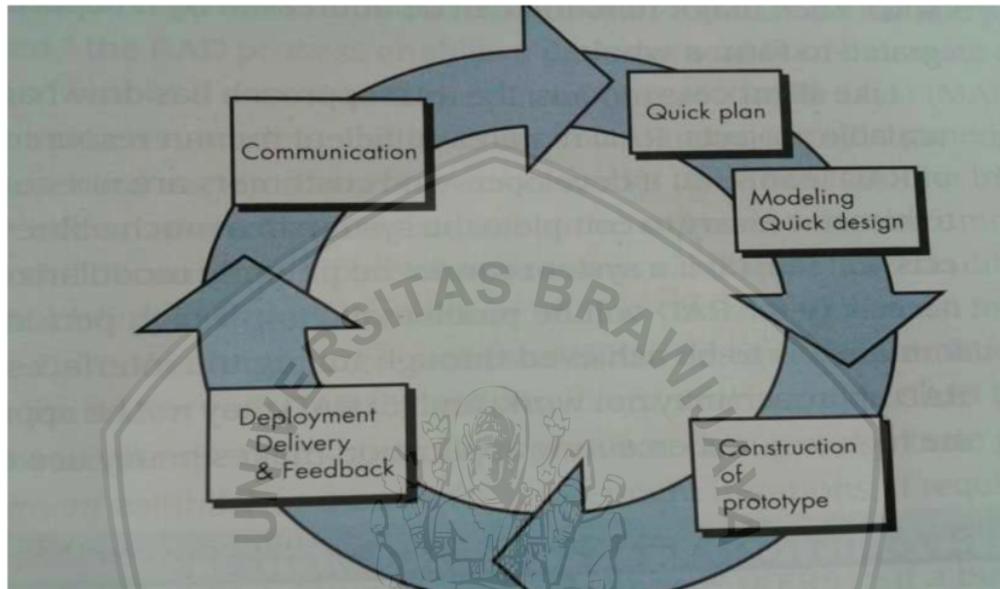
Dalam proses pengembangan perangkat lunak, terdapat beberapa model pengembangan yang ditawarkan., salah satunya adalah model yang dipakai dalam penelitian ini yaitu *prototyping*. Selain model *prototyping* ada juga model lainnya seperti *incremental*, *rapid development*, dan *extreme programming*.

2.2.1 Software Development Lifecycle: Prototyping

Dalam proses pengembangan perangkat lunak, pemilihan model disesuaikan dengan hasil analisis yang didapatkan dari masalah. Pemilihan model ini karena seringkali pengguna mendefinisikan secara umum, kebutuhan

pengguna yang kadang berubah-ubah, pengguna juga dapat terlibat dalam proses pengembangan yang pada akhirnya dapat terlihat lebih awal seiring berkembangnya kebutuhan untuk pengembangan aplikasi pembelajaran daur ulang sampah ini. Pada model *prototyping*, pengembangan dilakukan iterasi sebanyak dua kali dan hanya sampai proses pengujian, tidak sampai proses *deployment* karena dalam pengembangan ini tidak digunakan semua tahap pada model *prototyping*.

Berikut penjelasan mengenai tahapan-tahapan pada model *prototyping*



Gambar 2.1 Aliran SDLC Prototype

sumber: Presman 2010

- 1. Pengumpulan Kebutuhan**
Pengguna dan pengembang mendefinisikan format dalam seluruh perangkat lunak, mengidentifikasi semua kebutuhan yang diinginkan, dan garis besar sistem yang nantinya akan dibuat.
- 2. Membangun Prototyping**
Membangun prototyping dengan cara membuat perancangan sementara yang berfokus pada penyajian kepada pengguna (misalnya dengan membuat input dan output).
- 3. Evaluasi Prototyping**
Evaluasi ini dilakukan oleh pengguna untuk melihat apakah prototyping yang dibangun sudah sesuai dengan keinginan pengguna. Jika sudah sesuai maka langkah 4 akan diambil. Jika tidak prototype akan direvisi dengan mengulang langkah 1,2, dan 3.
- 4. Mengkodekan Sistem**
Dalam tahap ini *prototyping* yang sudah disepakati diterjemahkan ke dalam Bahasa pemrograman yang sesuai.

5. Menguji Sistem

Setelah sistem sudah menjadi suatu perangkat lunak yang siap pakai, harus dites dahulu sebelum digunakan. Pengujian ini dilakukan dengan White box, Black Box, Basis Path, pengujian arsitektur dan lain-lain.

6. Evaluasi Sistem

Pengujian ini dilakukan dengan White Box, Black Box, basis Path, pengujian arsitektur dan lain-lain. Pengguna mengevaluasi apakah sistem yang sudah jadi sudah sesuai dengan yang diharapkan. Jika ya, langkah 7 dilakukan, jika tidak ulangi langkah 4 dan 5.

7. Menggunakan Sistem

Perangkat lunak yang telah diuji dan diterima pengguna siap untuk digunakan.

2.3 UML (Unified Modelling Language)

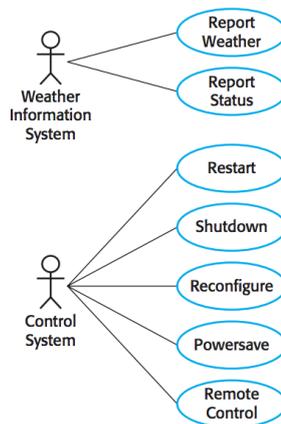
Unified Modelling Language (UML) adalah sebuah bahasa standar untuk memodelkan perangkat lunak. UML dapat digunakan untuk menggambarkan, membangun, dan mendokumentasikan artefak yang diperlukan dalam proses pengembangan sistem (Pressman, 2010). Tujuan dari UML adalah untuk menggambarkan sebuah mekanisme pemodelan yang mudah dipahami dalam pembangunan suatu sistem. UML memiliki berbagai jenis diagram diantaranya *use case diagram*, *activity diagram*, *class diagram*, dan *sequence diagram*.

2.3.1 Use Case Diagram

Use case adalah sebuah diagram untuk menggambarkan sebuah tingkah laku sistem yang dibangun beserta fungsionalitas yang ada pada sistem (Booch, et al., 2007). Keuntungan menggunakan *use case diagram* adalah mudah dipahami oleh pihak pemangku kepentingan serta sebagai alat komunikasi kebutuhan yang diinginkan pengguna dan pengembangan untuk memastikan sistem yang dibuat sesuai kebutuhan pengguna.

Berikut ini beberapa penjelasan terkait dengan *use case diagram*:

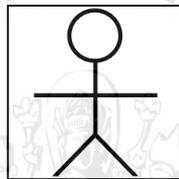
1. *Use Case* adalah urutan langkah-langkah yang secara tindakan saling terkait untuk melengkapi suatu tugas bisnis tunggal.



Gambar 2.2 Use Case

Sumber: Pressman 2010

2. Aktor merepresentasikan seseorang atau sesuatu (seperti perangkat atau sistem lain) yang berinteraksi dengan sistem. Aktor hanya berinteraksi dengan *use case* tetapi tidak memiliki kontrol akses terhadap *use case*



Gambar 2.3 Simbol Aktor

Sumber: Pressman 2010

3. *Relationship* pada diagram use case digambarkan sebagai sebuah garis antara dua simbol. Beberapa tipe hubungan yang terdapat pada diagram use case, yaitu sebagai berikut:
 - 1) *Association* adalah hubungan antara pelaku dengan use case, dimana terjadi interaksi antara mereka

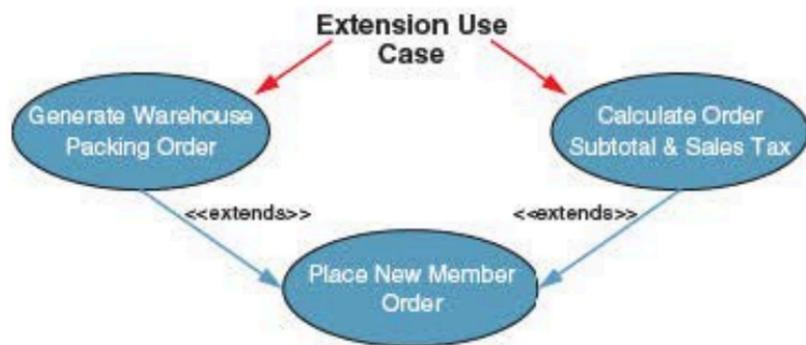


Gambar 2.4 Contoh Association Relationship

Sumber: Pressman 2010

- 2) *Extends* terdiri dari langkah yang diekstraksi dari use case yang lebih kompleks untuk menyederhanakan masalah orisinal dan arena itu memperluas fungsinya.

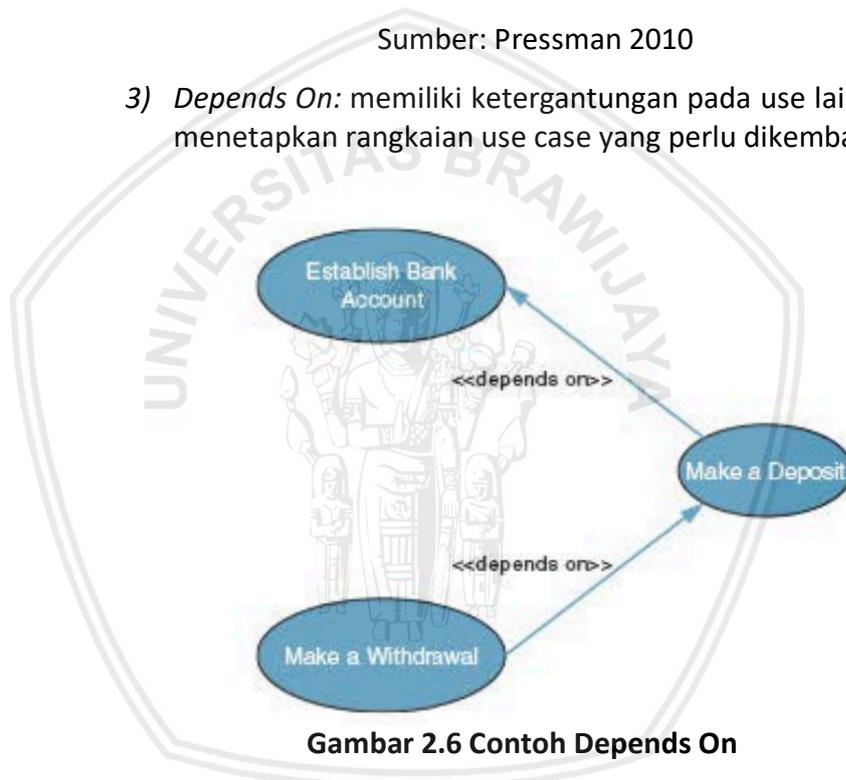




Gambar 2.5 Contoh Extends Relationship

Sumber: Pressman 2010

- 3) *Depends On*: memiliki ketergantungan pada use lain untuk menetapkan rangkaian use case yang perlu dikembangkan.



Gambar 2.6 Contoh Depends On

Sumber: Pressman 2010

2.4 Pemrograman Android dengan Kotlin

Kotlin adalah bahasa pemrograman baru untuk pengembangan aplikasi android maupun androidTV. Kotlin bersifat *statically-typed* yang dapat dijalankan diatas platform *java virtual machine(JVM)* dan dapat di kompilasi kedalam bentuk *JavaScript* (JetBrains, 2017). Berikut beberapa fitur dari Kotlin:

1) Concise

Mampu mengurangi *boilerplate of code* atau dengan kata lain mengurangi tingkat kerumitan dari kode yangditulis ketika menggunakan java

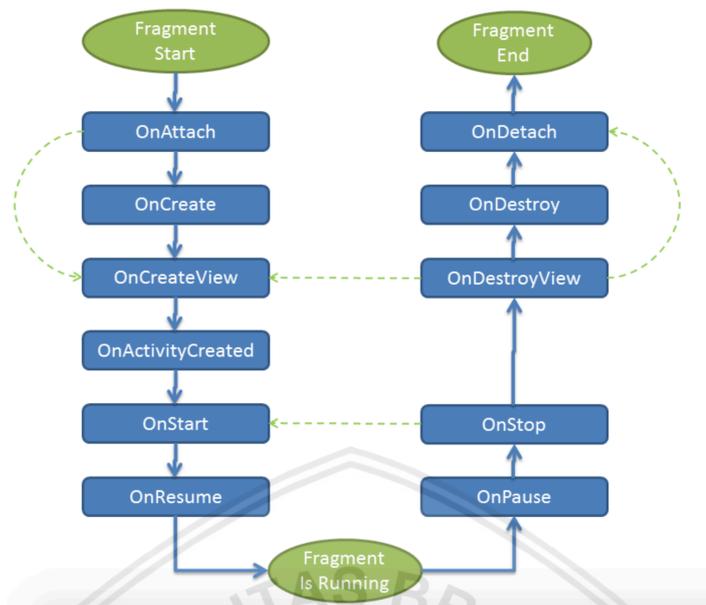
Berikut ini tabel yang menjelaskan daur hidup *activity*

Tabel 2.1 Daur Hidup Activity

Daur Hidup	Penjelasan
<code>onCreate()</code>	Pada method ini <i>activity</i> sudah dimulai tetapi belum terlihat oleh pengguna. Inisialisasi sebagian besar belum dimulai, misalnya <code>setContentView</code>
<code>onStart()</code>	Method ini dipakai untuk mendaftarkan sebuah <i>broadcastReceiver</i> untuk mengamati perubahan yang dapat memengaruhi <i>user interface</i>
<code>onResume()</code>	Pada method ini, <i>activity</i> sudah terlihat dan pengguna dapat berinteraksi
<code>onPause()</code>	Pada method ini, <i>activity</i> sudah akan bersiap untuk meninggalkan layar dan tidak lagi berada di <i>foreground</i>
<code>onStop()</code>	Method ini dipanggil ketika <i>activity</i> tidak lagi terlihat oleh pengguna
<code>onDestroy()</code>	Method ini dipanggil ketika method <code>finish()</code> dipanggil atau karena sistem membutuhkan memori lebih dan method ini dipanggil sebelum <i>activity</i> dihancurkan. Method ini akan membersihkan proses-proses yang berada pada <i>background</i> seperti proses unduh
<code>onRestart()</code>	Method ini dipakai ketika <i>activity</i> sudah melalui <code>onStop</code> tetapi akan diaktifkan kembali.

2.5.2 Android Fragment Lifecycle

Merupakan komponen tampilan yang lebih fleksibel dan dapat disisipkan pada sebuah tampilan dari *activity*. *Fragment* digunakan untuk memecah komponen pada aplikasi agar bias lebih fleksibel dan mudah untuk digunakan. *Fragment* memiliki daur hidup sendiri sehingga tidak tergantung pada *activity* yang disisipkan. Berikut ini penjelasan mengenai daur hidup *fragment*.



Gambar 2.8 Fragment Life Cycle
Sumber: Google

Tabel 2.2 Fragment Lifecycle

Daur Hidup	Penjelasan
<code>onAttach()</code>	Method ini dipanggil ketika sebuah fragment terhubung ke <i>activity</i>
<code>onCreate()</code>	Method ini dipanggil ketika sebuah <i>fragment</i> dibuat
<code>onCreateView()</code>	Dipanggil ketika <i>fragment</i> akan menjalankan sebuah tampilan
<code>onViewCreated()</code>	Method ini dipanggil setelah <i>onCreateView</i> dan akan memastikan tampilan yang akan ditampilkan tidak kosong (<i>non-null</i>)
<code>onActivityCreated()</code>	Method ini dipanggil setelah <i>activity</i> sudah menyelesaikan <code>onCreate()</code>
<code>onStart()</code>	Method ini dipanggil setelah <i>fragment</i> akan menjalankan tampilan pada antarmuka
<code>onResume()</code>	Method ini dipakai untuk melakukan pembacaan data yang lebih rumit seperti lokasi dan sensor
<code>onPause()</code>	Method ini dipanggil ketika terdapat data yang rumit dan ingin melakukan <i>commit</i>
<code>onDestroyView()</code>	Dipanggil ketika sebuah layout akan dihapus dari memori



onDestroy()	Method ini dipanggil ketika <i>fragment</i> sudah tidak dipakai
onDetach()	Method ini dipanggil ketika <i>fragment</i> tidak lagi terhubung ke sebuah <i>activity</i>

2.6 Youtube Data API

YouTube Data API memungkinkan pengguna menggabungkan fungsi yang biasanya dijalankan di situs web YouTube ke dalam situs web atau aplikasi pengguna sendiri. Pengguna membutuhkan akun *Google* untuk mengakses *Google Developers Console*, meminta kunci API dan mendaftarkan aplikasi pengguna. Daftar di bawah ini mengidentifikasi berbagai jenis sumber daya yang dapat pengguna ambil menggunakan *API (Application Programming Interface)*.

Tabel 2.3 YouTube Data API

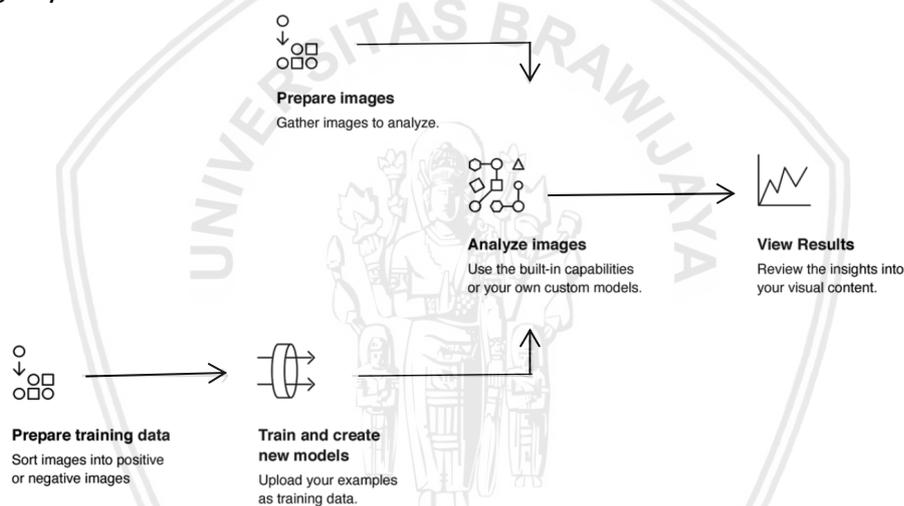
Sumber Daya	
Activity	Berisi informasi tentang tindakan ayang diambil pengguna tertentu di situs YouTube
channel	Berisi informasi tentang satu saluran YouTube
channelBanner	Mengidentifikasi URL yang akan digunakan untuk menetapkan gambar yang baru giunggah sebagai gambar untuk <i>channel</i>
channelSection	Berisi informasi tentang sekumpulan video yang telah dipilih untuk ditampilkan.
guideCategory	Mengidentifikasi kategori yang dikatikan YouTube dengan <i>channel</i> berdasarkan konten atau indicator lainnya, seperti popularitas.
l18nLanguage	Mengidentifikasi bahasa aplikasi yang didukung situs web YouTube
l18nRegion	Mengidentifikasi area geografis yang dapat dipilih pengguna YouTube sebagai wilayah konten pilihan
playlist	Kumpulan video yang dapat dilihat secara berurutan dan dibagikan dengan pengguna lain
playlistItem	Berisi detail yang menjelaskan bagaimana sumber daya yang disertakan digunakan dalam <i>playlist</i>
Search result	Berisi informasi tentang video YouTube, channel atau daftar putar yang cocok dengan parameter yang ditentukan dalam permintaan API
subscription	Berisi informasi tentang langganan pengguna YouTube
thumbnail	Mengidentifikasi gambar kecil yang terkait dengan sumber daya
video	Mewakili satu video YouTube

videoCategory	Mengidentifikasi kategori yang telah atau dapat dikaitkan dengan video yang diunggah
watermark	Mengidentifikasi gambar yang ditampilkan selama pemutaran video channel tertentu

2.7 IBM Watson Visual Recognition

IBM Watson Visual Recognition merupakan layanan yang memiliki kemampuan untuk memahami isi gambar. Layanan ini dapat digunakan untuk menganalisis gambar untuk menemukan adegan, objek, wajah, dan konten lainnya. Pengembang juga dapat membuat konsep klasifikasi tertentu dan juga melatih kemampuan dari layanan ini.

Pengembang dapat mengembangkan produk yang dapat mengidentifikasi pakaian dalam toko, mengidentifikasi buah segar dalam lemari persediaan dan lain sebagainya.



Gambar 2.9 IBM Watson Visual Recognition

2.8 Android Framework Camera API

Android Framework Camera API mendukung untuk berbagai tipe kamera dan fitur kamera yang tersedia oleh perangkat android yang memungkinkan pengguna untuk mengambil gambar atau video dari aplikasi. Ada dua cara untuk menggunakan kamera pada aplikasi

1. Menggunakan aplikasi kamera android yang ada di aplikasi pengguna
2. Langsung menggunakan API kamera yang disediakan android

Berikut ini tipe-tipe *intent* yang disediakan oleh android:



Tabel 2.4 Tipe Intent Android Camera

Tipe Intent	Deskripsi
ACTION_IMAGE_CAPTURE_SECURE	Akan menampilkan gambar yang dihasilkan oleh kamera
ACTION_VIDEO_CAPTURE	Akan memanggil aplikasi video yang tersedia di android untuk merekam video
EXTRA_SCREEN_ORIENTED	Digunakan untuk menentukan orientasi dari layar baik berorientasi <i>vertical</i> ataupun <i>landscape</i>
EXTRA_FULL_SCREEN	Digunakan untuk mengatur tampilan pada ViewImage
INTENT_ACTION_VIDEO_CAMERA	Digunakan untuk mengaktifkan kamera ketika merekam
EXTRA_SIZE_LIMIT	Digunakan untuk menentukan ukuran dari foto dan video yang dihasilkan

2.9 Firebase

Firebase merupakan sebuah layanan dari Google yang menyediakan layanan basis data secara *realtime*. Basis data ini disimpan dalam format JSON serta melakukan sinkronisasi secara berkala untuk mendapatkan keterhubungan dengan klien (Khedkar & Thube, 2017). Firebase merupakan layanan basis data yang dapat digunakan dari berbagai macam *platform*, antara lain android, iOS, serta aplikasi web. Saat aplikasi *cross-platform* dibuat dengan SDK Android, iOS dan JavaScript, semua klien akan berbagi sebuah *instance realtime database* dan akan menerima update secara otomatis. Saat API diimplementasikan pada aplikasi cross-platform, Firebase dapat memberikan fitur dengan kode sederhana. Fitur-fitur yang ditawarkan yaitu:

1. *Authentication*

Authentication ini merupakan layanan yang diberikan oleh Firebase untuk memberikan izin kepada pengguna tertentu yang sudah diberikan izin untuk mengakses aplikasi. Firebase menyediakan fitur *login* dan *register* melalui *email*, nomor ponsel, hingga mendukung beberapa media social seperti Gmail, twitter, dan Facebook untuk mempermudah pengguna menggunakan aplikasi dengan bantuan Firebase.

2. *Hosting*

Hosting merupakan konten web yang ditujukan pada pengembang aplikasi, layanan ini dapat dengan dan efektif mengirim aplikasi web dan

konten statis ke *Content Delivert Network (CDN)* dengan satu perintah. *Hosting* ini berisi dukungan domain khusus, sertifikat *SSL Global CDN* dan *Auto Provisioned*.

3. *Analytics*

Analytics merupakan fitur yang disediakan oleh Firebase untuk membantu pengembangan mengetahui seberapa jauh aplikasi mendapatkan interaksi dari pengguna dan dapat menganalisis perilaku dari pengguna.

4. *Messaging*

Fitur ini merupakan salah satu layanan yang disediakan oleh Firebase untuk mengirim pesan tanpa menggunakan biaya. Dengan Firebase Cloud Messaging pengembang dapat melakukan aktivitas berkirim pesan pada pengguna.

5. *Storage*

Storage merupakan fitur yang disediakan oleh Firebase yang berfungsi sebagai media penyimpanan secara *real-time*. *Storage* ini dapat menyimpan berkas seperti gambar, audio, video, sertaa berkas lainnya yang memungkinkan pengguna untuk melakukan aktifitas unggah dan unduh secara langsung.

6. *Real-time Database*

Fitur ini adalah database yang diposisikan pada *cloud*. Datanya disimpan dalam bentuk JSON dan disinkronkan secara *real-time* tiap kali pengguna aplikasi terhubung. Ketika pengembang membuat aplikasi lintas platform dengan SDK Android, iOS, dan JavaScript, semua klien akan berbagi sebuah *instance Realtime Database* dan menerima *update* data terbaru secara otomatis.

7. *Crash Reporting*

Firestore Crash Reporting membuat detail laporan tentang terjadinya *error* pada aplikasi yang dibuat oleh pengembang. *Error* yang terjadi akan dikelompokkan berdasarkan *log error* yang serupa atau sejenis serta diurutkan berdasarkan tingkat terparah terjadinya *error* pada aplikasi. Selain otomatis, pengembang juga dapat membuat *log* untuk mempermudah menemukan masalah yang menyebabkan aplikasi tidak bekerja.

8. *App Indexing*

App Indexing sangat berguna bagi pengembang untuk memasukkan aplikasi yang telah dikembangkan dalam penelusuran Google. Layanan ini juga menawarkan pelengkap otomatis kueri untuk membantu pengguna menemukan hal yang diinginkan lebih cepat.

9. AdMob

Layanan *AdMob* ini digunakan untuk mendapatkan pendapatan dari aplikasi yang telah dibuat oleh pengembang. Layanan ini dapat ditambah kemampuannya untuk lebih analitis pada data pengguna aplikasi jika menggunakan layanan ini dengan Google Analytics.

10. ML Kit

Layanan ini dihadirkan pada Mei 2018 pada acara Google I/O. Dengan *ML Kit*, pengembang dapat menggunakan *machine learning* untuk membuat fitur menarik di android, iOS. ML Kit memberikan anda API dalam bentuk SDK ataupun *cloud API*. Dengan layanan ML Kit, pengembang dapat melakukan *text recognition*, *face detection*, *barcode scanning*, *image labelling*, dan *landmark recognition*.

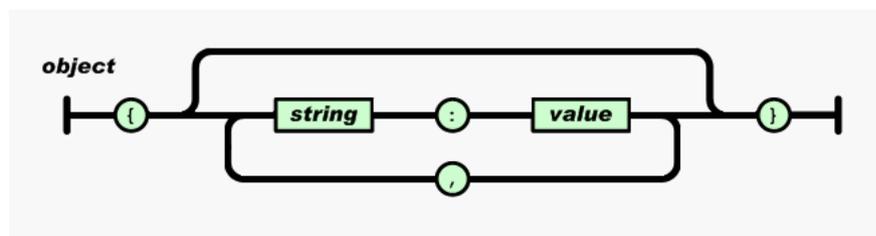
2.10 JSON (JavaScript Object Notation)

JSON (*JavaScript Object Notation*) adalah format penukaran data yang sangat ringan, lebih mudah dibaca oleh manusia serta mudah untuk diterjemahkan oleh komputer. JSON merupakan format teks yang tidak bergantung pada Bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran data.

Json terbuat dari dua struktur:

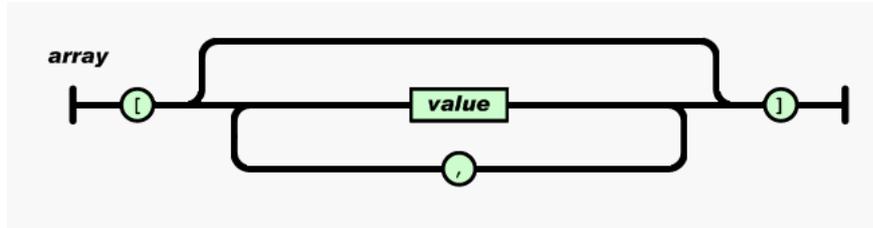
1. Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek, rekaman(*record*), struktur(*structure*), vektor(*vector*), kamus(*dictionary*), *table hash(hash table)*, daftar berkunci (*keyed list*) atau *associative array*.
2. Daftar nilai terurutkan. Pada kebanyakan bahasa,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

Struktur data ini disebut sebagai struktur data universal. Pada dasarnya, semua Bahasa pemrograman modern mendukung struktur data dalam bentuk yang sama maupun berlainan. Hal ini pantas disebut demikian karena format data mudah dipertukarkan dengan Bahasa pemrograman yang juga berdasarkan pada struktur data ini. JSON menggunakan bentuk sebagai berikut:



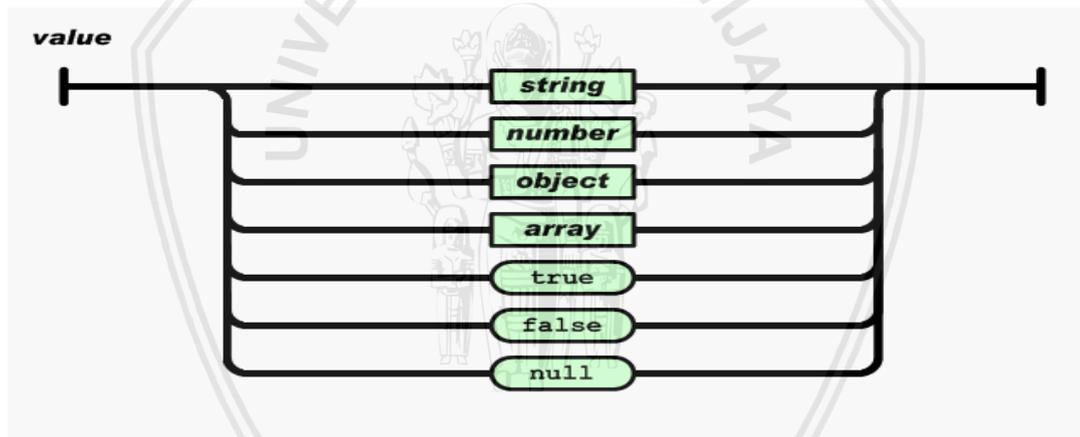
Gambar 2.10 JSON Object

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



Gambar 2.11 JSON Array

Array merupakan kumpulan nilai yang terurut. Array dimulai dengan tanda kurung siku ([), diakhiri dengan kurung siku (]) dan setiap nilai dipisahkan oleh koma.



Gambar 2.12 JSON Value

Value dapat berupa sebuah *string* dalam tanda kutip ganda atau angka atau *true* atau *false* atau null atau sebuah objek atau sebuah array yang dapat disusun secara bertingkat.

2.11 Teori Pengujian

2.11.1 Blackbox Testing

Pengujian yang didasarkan pada detail aplikasi seperti tampilan aplikasi, fungsi-fungsi yang ada pada aplikasi, dan kesesuaian alur fungsi dengan bisnis yang diinginkan oleh pengguna (Pressman, 2010). Pengujian ini tidak melihat dan menguji *source code* program. Pengujian ini bertujuan untuk menemukan kesalahan fungsi yang tidak benar atau belum ada, kesalahan antrmuka, kesalahan

struktur data atau akses database, kesalahan kinerja, kesalahan inisialisasi dan kesalahan terminasi.

2.11.2 WhiteBox Testing

Pengujian ini didasarkan pada detail prosedur dan alur logika kode program. Pada pengujian ini, tester melihat kode program dan menemukan kesalahan dari kode program yang diuji (Pressman, 2010). Pada *white-box testing* pengujian didasarkan pada kasus uji yang diperoleh melalui pencarian jalur independen kode program yang dibangun. Perancangan *Flow Graph* dapat diketahui melalui jalur independen kode program yang terdiri dari *node* yang direpresntasikan dalam data.

2.11.3 System Usability Scale (SUS)

System Usability Scale (SUS) merupakan pengujian pada sisi pengguna yang menyediakan alat ukur yang "*quick and dirty*" dan dapat diandalkan (Brooke, 1986). Metode ini pertama kali digunakan untuk mengevaluasi berbagai jenis produk ataupun servis, termasuk software perangkat mobile bahkan website. Pada SUS metode ini menggunakan 10 pernyataan berbentuk kuisisioner yang dijawab dengan 5 opsi jawaban untuk setiap pernyataannya, mulai sangat setuju hingga sangat tidak setuju dengan nilai 1 sampai 5.

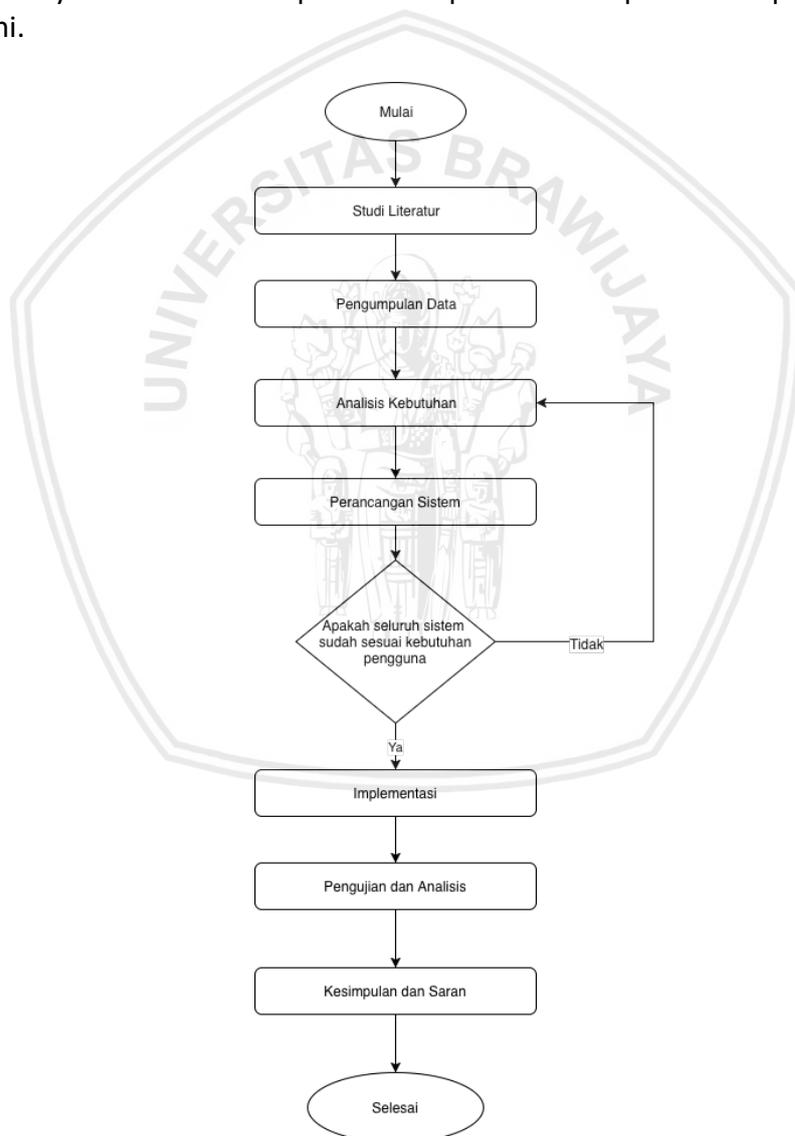
2.12 Purposive Sampling

Purposive sampling merupakan pengambilan sampel "*with purpose in mind*" atau secara sengaja sesuai dengan persyaratan sampel yang diperlukan. Dalam Bahasa sederhananya, purposive sampling itu dapat dikatakan sebagai *judgemental sampling* atau pengambilan sampel berdasarkan penilaian peneliti mengenai siapa-siapa saja yang pantas untuk dijadikan sampel. Oleh karenanya agar tidak sangat subjektif, peneliti harus punya latar belakang mengenai sampai yang dimaksud agar benar-benar bias mendapatkan sampel yang sesuai dengan persyaratan atau tujuan penelitian.

BAB 3 METODOLOGI PENELITIAN

3.1 Strategi dan Rancangan Penelitian

Metode penelitian menjelaskan mengenai langkah-langkah yang dilakukan dalam penelitian. Dalam penelitian tentang pengembangan aplikasi pembelajaran pengelolaan sampah daur ulang dengan menggunakan *IBM Cognitive Service* dan *YouTube Data API* berbasis android, langkah-langkah yang dilakukan adalah studi literatur, analisis kebutuhan, perancangan, implementasi, pengujian dan analisis serta diakhiri dengan kesimpulan dan saran. Dalam penelitian ini, peneliti akan menggunakan model *Software Development Life Cycle (SDLC) Prototyping* dengan iterasi sebanyak dua kali. Tahapan dalam penelitian dapat dilihat pada gambar berikut ini.



Gambar 3.1 Metodologi Penelitian

3.1.1 Studi Literatur

Studi literatur dilakukan untuk mengumpulkan data dan informasi yang diperlukan dalam penelitian dengan cara membaca serta membandingkan sumber-sumber data informasi. Sumber data dan informasi yang dimaksud adalah buku-buku referensi maupun sumber penelitian lain yang terkait sebagai acuan untuk melakukan penelitian ini. Teori pendukung tersebut diantaranya:

1. Rekayasa Perangkat Lunak
2. IBM Watson Visual Recognition
3. YouTube Data API
4. Pemrograman Android dengan Kotlin
5. Android Framework Camera API
6. Pengembangan Perangkat Lunak
7. Firebase
8. JSON (JavaScript Object Notation)

3.1.2 Pengumpulan Data

Pada penelitian ini data diperoleh melalui proses survei dengan menggunakan teknik survei terstruktur yang merupakan salah satu teknik survei dengan cara pengumpulan data yang dilakukan dengan cara membuat daftar pertanyaan yang telah ditentukan dan alternatif jawaban yang sudah tersedia. Survey dilakukan terhadap mahasiswa Fakultas Ilmu Komputer Universitas Brawijaya terhadap 75 responden untuk mengetahui penanganan sampah yang dilakukan oleh responden. Pada survei dilakukan dengan memberikan satu set pertanyaan kepada responden sehingga mudah untuk dianalisis dan potensi bias rendah.

3.1.3 Analisis Kebutuhan

Analisis kebutuhan dilakukan dengan mengumpulkan data dan informasi terkait aplikasi pembelajaran pengelolaan sampah daur ulang. Data dan informasi yang didapatkan kemudian dianalisis untuk mendapatkan kebutuhan fungsional dan non-fungsional dari aplikasi yang akan dibangun. Analisis kebutuhan dilakukan agar tujuan penelitian dapat tercapai, yaitu untuk memudahkan masyarakat untuk memilah dan membedakan jenis sampah daur ulang. Pada penelitian ini, aplikasi dibangun dengan menggunakan pendekatan berorientasi objek. Analisis kebutuhan pada pengembangan aplikasi pembelajaran daur ulang sampah ini meliputi:

1. Elisitasi Kebutuhan

Pada tahap ini elisitasi kebutuhan dilakukan proses kuisisioner untuk mengetahui apa saja kebutuhan Sistem yang akan dibangun. Kuisisioner akan disebar kepada mahasiswa Fakultas Ilmu Komputer.

2. Spesifikasi Kebutuhan

Pada proses ini kebutuhan yang sudah didapatkan lebih di spesifikkan. Dalam hal ini juga dilakukan identifikasi actor yang akan menggunakan aplikasi dan juga kebutuhan yang didapatkan kedalam dua jenis kebutuhan yaitu kebutuhan fungsional dan nonfungsional. Serta dilakukan pemodelan kebutuhan yang berupa *use case diagram* dan *use case scenario*.

3. Manajemen Kebutuhan

Pada proses ini dilakukan kontrol terhadap kebutuhan yang sudah didefinisikan dan memberikan kode pada kebutuhan dalam menulis kebutuhan sistem

3.1.4 Perancangan

Pada tahap perancangan aplikasi, perancangan dilakukan berdasarkan hasil dari tahap analisis kebutuhan yang dilakukan sebelumnya dan akan dibuat sebagai acuan dalam implementasi dan pengujian yang akan dilakukan setelah pembangunan sistem. Tahap-tahap perancangan dilakukan dengan tahapan berikut ini:

1. Perancangan Arsitektur

Dalam perancangan arsitektur ini akan dilakukan pemodelan dengan menggunakan diagram UML seperti *sequence diagram* dan *class diagram*.

2. Perancangan Komponen

Dalam perancangan komponen ini akan dituliskan beberapa sampel algoritma utama yang diambil dari setiap kelas pada controller. Algoritme ini akan dituliskan dalam bentuk *pseudocode*

3. Perancangan Data

Dalam perancangan data memuat rancangan tabel database yang menggunakan JSON skema. Perancangan data ini akan menjadi dasar dalam implementasi database pada sistem.

4. Perancangan Antarmuka

Perancangan antarmuka dari sistem yang akan dibangun ini terdiri dari tata letak komponen yang harus disediakan oleh sistem berdasarkan kebutuhan sistem. Dalam perancangan antarmuka ini akan dituliskan beberapa sampel antarmuka utama berdasarkan level pengguna. Perancangan antarmuka ini akan menjadi dasar implementasi antarmuka sistem.

3.1.5 Implementasi

Tahapan implementasi dilakukan dengan menterjemahkan hasil perancangan ke dalam kode program dengan hasil akhir berupa aplikasi sistem pembelajaran pengelolaan sampah daur ulang yang utuh. Implementasi dilakukan

menggunakan *Android Studio* sebagai *tools* untuk membuat aplikasi android dan menggunakan Bahasa pemrograman kotlin, *IBM Watson Visual Recognition*, *YouTube Data API* dan *Framework Android Camera*. Proses yang ada dalam implementasi sistem ini antara lain:

1. Implementasi sistem ini dilakukan dengan mengacu pada spesifikasi sistem yang digunakan untuk membangun aplikasi ini.
2. Implementasi kode program mengacu pada perancangan komponen yang berisi algoritma dan logika program pada perancangan sistem dengan menggunakan bahasa pemrograman kotlin. Logika pemrograman dapat diterapkan pada layer model ataupun pada layer *activity*
3. Implementasi data dilakukan berdasarkan kebutuhan yang dilakukan pada analisis kebutuhan, data pengenalan objek diimplementasikan dengan *IBM Cognitive Service* dan *YouTube Data API*.
4. Implementasi antarmuka, implementasi ini dibuat berdasarkan perancangan antarmuka pada perancangan sistem dengan menggunakan bahasa pemrograman kotlin.

3.1.6 Pengujian dan Analisis Hasil

Pada tahap pengujian dan analisis hasil dilakukan pengujian pada aplikasi untuk memastikan aplikasi berjalan sesuai fungsionalitas dan menganalisis hasil dari pengujian terhadap aplikasi yang telah dibangun. Pengujian dilakukan menggunakan metode *White Box Testing* dan *Black Box Testing*. Pengujian *White Box* dilakukan untuk mengecek apakah komponen terkecil pada aplikasi yaitu kelas dan method di dalamnya sudah berjalan sebagaimana mestinya. Sedangkan pengujian *Black Box* dilakukan untuk memastikan bahwa fungsionalitas dari aplikasi sudah berjalan sesuai kebutuhan yang telah didefinisikan sebelumnya. Pengujian sistem yang akan dibangun mempunyai beberapa tahapan yang harus dilakukan sebagai berikut:

1. *Unit Testing*

Pengujian yang dilakukan dengan menguji setiap unit seperti *class* atau objek dari perangkat lunak yang telah didefinisikan. Jika terdapat unit atau modul yang dieksekusi dan hasilnya tidak sesuai dengan proses bisnis maka baris program variabel dan fungsi dalam unit akan dicek lalu diperbaiki dan dijalankan kembali. Karena dalam sebuah perangkat lunak banyak memiliki unit-unit kecil maka untuk mengujinya biasanya dibuat program kecil atau *main program* untuk menguji unit-unit perangkat lunak. Unit-unit kecil dapat berupa sekumpulan fungsi yang ada dalam kumpulan kelas.

2. *Validation Testing*

Pengujian ini dilakukan seri uji coba *Black Box Testing* untuk menunjukkan sesuai dengan kebutuhan. Dalam proses pengujiannya ini berusaha untuk mencari kesalahan seperti fungsi yang tidak benar pada pembuatan *interface*, struktur data dan kinerja.

3. *Usability Testing*

Pengujian ini dilakukan untuk memastikan bahwa sistem sudah sesuai dengan kebutuhan dan mudah dipahami oleh user ketika dioperasikan nantinya. Pengujian ini dilakukan ketika sistem sudah siap digunakan oleh pengguna. *Usability testing* merupakan salah satu cara untuk mengetahui apakah pengguna dapat dengan mudah menggunakan aplikasi, seberapa efisien dan efektif sebuah aplikasi dapat membantu pengguna mencapai tujuannya dan apakah pengguna puas dengan aplikasi yang digunakan

3.1.7 Penarikan Kesimpulan dan Saran

Kesimpulan diambil setelah proses pengujian dan analisis hasil telah selesai dilakukan. Kesimpulan yang diambil sesuai dengan hasil dari pengujian dan analisis dimana hal ini dilakukan untuk mengetahui inti dari penelitian ini. Dengan kata lain, kesimpulan yang diambil akan menjawab permasalahan yang telah dirumuskan. Saran untuk penelitian ini dibuat berdasarkan hasil dari pengujian, hasil analisis dan kesimpulan yang ada. Saran dapat berupa kekurangan dari penelitian untuk dikembangkan pada penelitian selanjutnya.



BAB 4 ANALISIS KEBUTUHAN

4.1 Gambaran Umum Aplikasi

Dalam penelitian yang dilakukan peneliti, dikembangkan sebuah aplikasi pembelajaran pengelolaan sampah daur ulang sampah yang interaktif berbasis android. Aplikasi Dalang (Daur Ulang) ini merupakan aplikasi yang dibuat untuk memberikan informasi mengenai cara mendaur ulang sampah melalui media video. Pengguna dapat melakukan analisis objek menggunakan *IBM Cognitive service* yang tersedia pada aplikasi Dalang ini untuk memudahkan pencarian video dan juga *YouTube Data API* yang digunakan untuk menjembatani pertukaran data antara *IBM Cognitive Service* dan aplikasi yang dikembangkan peneliti. Aplikasi pembelajaran ini dibangun untuk memudahkan mahasiswa untuk meningkatkan minat mahasiswa dalam mengelola sampah yang dapat di daur ulang sehingga dapat semakin efektif, efisien dan informasi yang didapat lebih spesifik mengenai cara pengelolaan sampah yang baik dan benar. Aplikasi pembelajaran pengelolaan sampah daur ulang ini dikembangkan dalam *platform* android, menggunakan bahasa pemrograman Kotlin, dan menggunakan IDE (*Integrated Development Environment*) Android Studio.

4.2 Identifikasi Aktor

Aktor menggambarkan seseorang atau sesuatu yang berinteraksi dengan sistem. Aktor hanya berinteraksi dengan *use case* tetapi tidak memiliki kontrol atas *use case*.

Tabel 4.1 Identifikasi Aktor

Aktor	Deskripsi
Mahasiswa	Merupakan pengguna yang akan mengakses dan menjalankan aplikasi

4.3 Analisis Kebutuhan Iterasi Pertama

4.3.1 Kebutuhan Fungsional

Kebutuhan fungsional adalah kebutuhan yang berisi proses-proses apa saja yang nantinya harus disediakan oleh sistem mencakup bagaimana sistem harus merespon input dan bagaimana respon sistem pada situasi tertentu. Setiap kebutuhan memiliki kode sebagai penunjuk kebutuhan.

DLG – F/NF – XX

Tabel 4.1 Penjelasan Aturan Penomoran Kebutuhan

Penomoran	Deskripsi
DLG	DLG merupakan singkatan dari nama aplikasi yang dikembangkan, yaitu DALANG (Daur Ulang)
F/NF	Merupakan kode untuk menyatakan jenis kebutuhan. F menyatakan kebutuhan fungsional dan NF menyatakan kebutuhan non fungsional
XX	Merupakan kode untuk menyatakan nomor urut kebutuhan

Tabel 4.2 Kebutuhan Fungsional Iterasi Pertama

No	Kode Kebutuhan Sistem	Deskripsi Kebutuhan
1	DLG-F-1	Sistem dapat mengambil objek menggunakan kamera
2	DLG-F-2	Sistem dapat memilih objek yang diambil dari <i>gallery</i>
3	DLG-F-3	Sistem dapat menampilkan keluaran berupa hasil analisis objek
4	DLG-F-4	Sistem dapat menampilkan keluaran berupa daftar video berdasarkan hasil analisis objek
5	DLG-F-5	Sistem dapat menampilkan keluaran berupa kisaran harga sampah yang dianalisis
6	DLG-F-6	Sistem dapat memutar video yang tersedia pada daftar video berdasarkan hasil analisis objek
7	DLG-F-7	Sistem akan menganalisis objek yang diambil oleh aktor
8	DLG-F-8	Sistem dapat melakukan pencarian video berdasarkan hasil analisis dari objek yang diambil oleh aktor

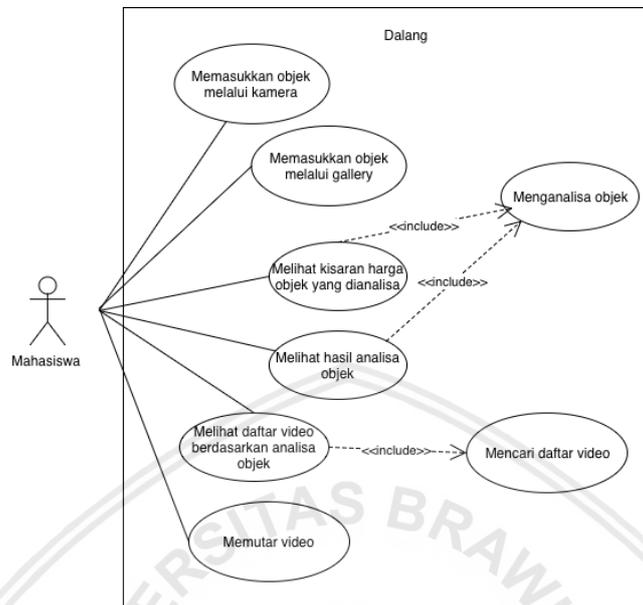
Tabel 4.3 Kebutuhan Non Fungsional Sistem

No	Kode Kebutuhan Sistem	Deskripsi Kebutuhan
1	DLG-NF-1	Kemudahan dalam penggunaan aplikasi yang akan digunakan oleh mahasiswa filkom

4.3.2 Diagram Use Case

Use case diagram merupakan gambaran dari semua aktor, *use case* dan interaksi yang diantaranya memperkenalkan suatu sistem. *Use case diagram* memberikan gambaran singkat mengenai hubungan antara *use case*, aktor dan

sistem. Pemodelan didasarkan pada daftar kebutuhan fungsional yang diperoleh dan digambarkan.



Gambar 4.1 Diagram Use Case Sistem

4.3.3 Skenario Use Case

Use Case Scenario merupakan penjelasan yang lebih detail dari setiap *use case* yang digambarkan pada use case diagram. *Use case scenario* memiliki beberapa komponen seperti nama use case, kode kebutuhan terkait, actor, tujuan use case, *precondition*, *main flow*, *post condition*, dan *alternative flow*. Bagian *precondition* merupakan kondisi yang harus dicapai sebelum menjalankan *main flow*. Bagian *main flow* merupakan bagian yang mendefinisikan alur kondisi secara berurutan. Bagian *alternative flow* merupakan bagian yang berisi alur kondisi tertentu yang tidak dideklarasikan di main flow. Bagian *post condition* merupakan bagian yang berisi kondisi akhir setelah aktor menjalankan proses pada main flow.

Tabel 4.3 Skenario Use Case Mengambil Objek Menggunakan Kamera Iterasi Pertama

Nama Use Case	Mengambil objek menggunakan kamera
Kode Kebutuhan Terkait	DLG-F-1
Aktor	Mahasiswa
Tujuan	Mengambil objek dengan kamera
Pre Condition	Aktor telah berada pada menu utama

<i>Main flow</i>	<ol style="list-style-type: none"> 1. Aktor menekan tombol kamera 2. Sistem akan menampilkan fitur kamera <i>default</i> dari android 3. Aktor mengambil foto objek yang akan dianalisis 4. Sistem akan menampilkan foto sementara 5. Aktor menekan tombol OK 6. Sistem akan melakukan analisis objek
<i>Post Condition</i>	Sistem akan melakukan analisis terhadap objek yang diambil oleh aktor
<i>Alternative Flow 1</i>	Jika aktor menekan tombol <i>retry</i> , maka aktor akan mengambil foto kembali

Tabel 4.4 Skenario Use Case Memilih Objek Menggunakan Gallery Iterasi Pertama

Nama <i>Use Case</i>	Memilih objek menggunakan <i>gallery</i>
Kode Kebutuhan Terkait	DLG-F-2
Aktor	Mahasiswa
Tujuan	Memilih objek dari <i>gallery</i>
<i>Pre Condition</i>	Aktor telah berada pada menu utama
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Aktor menekan tombol <i>gallery</i> 7. Sistem akan menampilkan fitur <i>gallery default</i> dari android 8. Aktor memilih foto objek yang akan dianalisis 9. Sistem akan menampilkan foto sementara 10. Sistem akan melakukan analisis objek
<i>Post Condition</i>	Sistem akan melakukan analisis terhadap objek yang diambil oleh aktor
<i>Alternative Flow</i>	-

Tabel 4.5 Skenario Use Case Melihat Hasil Analisis Objek Iterasi Pertama

Nama <i>Use Case</i>	Melihat Hasil Analisis Objek
Kode Kebutuhan Terkait	DLG-F-3
Aktor	Mahasiswa
Tujuan	Melihat hasil analisis objek
<i>Pre Condition</i>	Sistem telah menganalisis objek

<i>Main flow</i>	<ol style="list-style-type: none"> 1. Sistem akan menentukan nama objek dengan melakukan analisis menggunakan IBM Cognitive Service 2. Sistem menampilkan hasil analisis objek
<i>Post Condition</i>	Sistem akan melakukan analisis terhadap objek yang diambil oleh actor dan menampilkan hasil analisis
<i>Alternative Flow</i>	-

Tabel 4.6 Skenario Use Case Melihat Daftar Video Iterasi Pertama

Nama <i>Use Case</i>	Melihat Daftar Video
Kode Kebutuhan Terkait	DLG-F-4
Aktor	Mahasiswa
Tujuan	Melihat daftar video berdasarkan hasil analisis
<i>Pre Condition</i>	Sistem telah menganalisis objek
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Sistem akan menentukan nama objek dengan melakukan analisis menggunakan <i>IBM Cognitive Service</i> 2. Sistem menampilkan hasil analisis objek 3. Sistem akan menampilkan daftar video berdasarkan hasil analisis objek
<i>Post Condition</i>	Sistem akan melakukan analisis terhadap objek yang diambil oleh aktor dan menampilkan daftar video berdasarkan hasil analisis
<i>Alternative Flow</i>	-

Tabel 4.7 Skenario Use Case Melihat Harga Sampah Iterasi Pertama

Nama <i>Use Case</i>	Melihat Harga Objek
Kode Kebutuhan Terkait	DLG-F-5
Aktor	Mahasiswa
Tujuan	Melihat hasil harga berdasarkan analisis objek
<i>Pre Condition</i>	Aktor telah berada pada menu analisis
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Sistem akan menentukan nama objek dengan melakukan analisis menggunakan IBM Cognitive Service 2. Sistem menampilkan hasil analisis objek



	3. Sistem menampilkan harga sampah berdasarkan hasil analisis objek
<i>Post Condition</i>	Sistem akan melakukan analisis terhadap objek yang diambil oleh aktor dan menampilkan harga sampah berdasarkan hasil analisis
<i>Alternative Flow</i>	-

Tabel 4.8 Skenario Use Case Memutar Video Iterasi Pertama

Nama Use Case	Memutar Video
Kode Kebutuhan Terkait	DLG-F-6
Aktor	Mahasiswa
Tujuan	Memutar salah satu video yang tersedia
<i>Pre Condition</i>	Aktor telah berada pada menu analisis
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih salah satu video yang tersedia pada daftar video 2. Sistem akan memutar video yang dipilih oleh aktor
<i>Post Condition</i>	Sistem akan memutar video yang dipilih aktor
<i>Alternative Flow</i>	-

4.4 Analisis Kebutuhan Iterasi Kedua

4.4.1 Kebutuhan Fungsional

Pada Iterasi pertama, dilakukan survey terhadap keinginan calon pengguna. Hasil dari evaluasi tersebut adalah adanya penambahan beberapa fitur untuk melihat daftar riwayat pencarian yang dilakukan oleh pengguna.

Tabel 4.9 Kebutuhan Fungsional Iterasi Kedua

No	Kode Kebutuhan Sistem	Deskripsi Kebutuhan
9	DLG-F-9	Sistem dapat menampilkan riwayat pengambil data oleh pengguna

4.4.2 Use Case Diagram

Setelah melakukan iterasi yang pertama, adanya perubahan pada *use case diagram*. Adapun perubahan yang dilakukan dapat dilihat pada gambar berikut ini:



Gambar 4.2 Use Case Diagram Iterasi Kedua

Setelah adanya perubahan pada iterasi pertama, ada satu fitur yang mengalami penambahan yaitu riwayat pencarian yang bertujuan agar pengguna dapat melihat riwayat pencarian objek yang sudah dianalisis.

4.4.3 Skenario Use Case

Dari hasil itreasi pertama, didapatkan perubahan berupa penambahan kebutuhan fungsional yaitu, melihat riwayat pencarian. Adapun penambahan *use case scenario* adalah sebagai berikut:

Tabel 4.10 Scenario Use Case Riwayat Pencarian

Nama <i>Use Case</i>	Riwayat Pencarian
Kode Kebutuhan Terkait	DLG-F-9
Aktor	Mahasiswa
Tujuan	Melihat riwayat pencarian objek yang sudah dianalisis



<i>Pre Condition</i>	Aktor telah berada pada menu utama
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Aktor menekan <i>tab history</i> 2. Sistem akan menampilkan daftar riwayat pencarian
<i>Post Condition</i>	Sistem akan menampilkan daftar riwayat pencarian yang telah dilakukan oleh aktor
<i>Alternative Flow 1</i>	Jika aktor belum pernah melakukan pencarian, maka sistem tidak akan menampilkan daftar riwayat pencarian pada <i>tab history</i>

Pada Tabel 4.10 berisi *use case* untuk melihat daftar riwayat pencarian yang dilakukan oleh aktor. Alurnya adalah ketika aktor sudah berada pada menu *history*, aktor akan melihat riwayat pencarian yang pernah dilakukan. Tetapi jika aktor belum pernah melakukan pencarian, maka pada menu *history*, daftar riwayat tidak akan muncul atau ditampilkan.

4.5 Analisis Data

Dari hasil analisis kebutuhan fungsional yang telah dibuat, kemudian dapat dilakukan analisis data yang berguna untuk menentukan data-data apa saja yang akan digunakan dalam sistem agar bisa memenuhi seluruh kebutuhan fungsional. Hasil dari analisis data adalah sebagai berikut:

1. Data Youtube untuk mengambil data-data Youtube Data API
2. Data IBM Visual Recognition untuk mengambil data-data dari respons IBM Visual Recognition.
3. Data harga untuk menyimpan daftar harga sampah yang tersedia.

BAB 5 PERANCANGAN DAN IMPLEMENTASI

Pada bab ini akan dijelaskan mengenai tahap perancangan dan implementasi. Sebelum melakukan implementasi sistem, dilakukan perancangan sistem yang digunakan sebagai dasar untuk mengimplementasikan sistem, dalam tahap ini perancangan yang akan dilakukan berdasarkan analisis kebutuhan yang dilakukan. Sedangkan implementasi dilakukan dengan menerapkan hasil rancang sistem untuk membangun sistem secara keseluruhan yang terbagi dalam implementasi kode program dan implementasi antarmuka sistem.

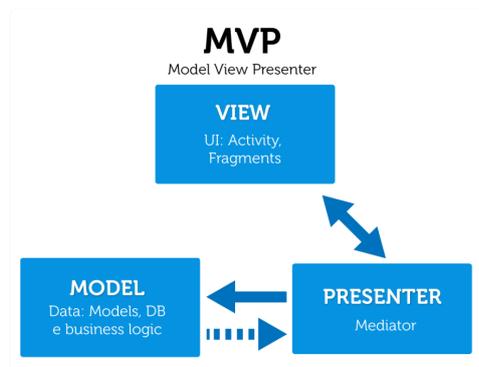
5.1 Perancangan

Perancangan dilakukan setelah tahap analisis kebutuhan selesai dilakukan. Hasil dari proses analisis kebutuhan berupa kebutuhan fungsional dan hasil dari pemodelan kebutuhan berupa *use case* dan *use case scenario* dijadikan dasar dalam merancang sistem. Proses perancangan Dalang ini dibagi menjadi tiga tahap, yaitu perancangan arsitektur, perancangan komponen, dan perancangan antarmuka.

5.1.1 Perancangan Arsitektur Sistem

Aplikasi Dalang ini dibangun dengan menggunakan arsitektur MVP atau *Model View Presenter*. MVP ini merupakan sebuah arsitektur yang berasal dari konsep MVC (*Model View Controller*). Arsitektur ini bekerja sebagai abstraksi antara lapisan antar muka dan lapisan *business logic*. Pemilihan MVP didasari untuk menghindari kopling dan menghindari duplikat kode. Penggunaan MVP meningkatkan kemampuan uji coba pada kode dengan *unit test*.

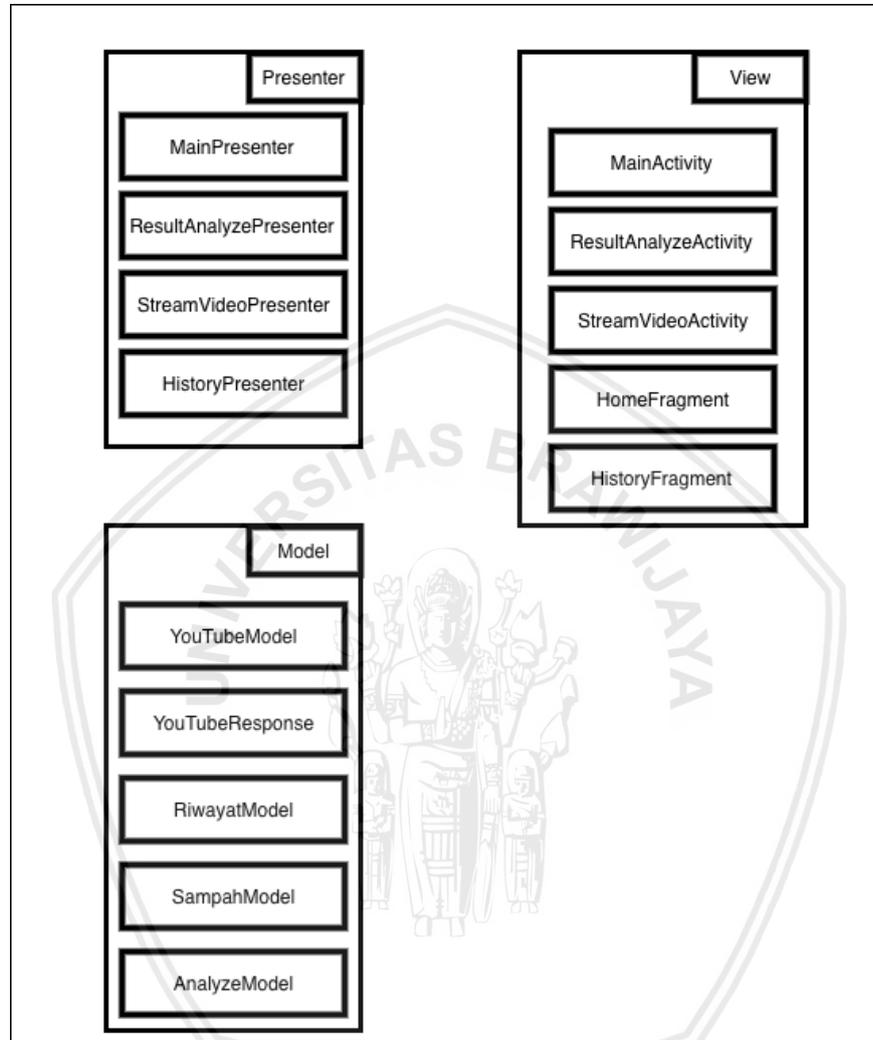
Pada dasarnya, arsitektur pada android menggunakan MVC (*Model View Controller*) dengan *Activity/Fragment* sebagai *controller*. Tetapi dengan menggunakan arsitektur MVP, *activity/fragment* memiliki tanggungjawab untuk mengolah data sekaligus untuk memperbaharui tampilan. *Activity/fragment* memiliki tanggung jawab lebih karena bekerja sebagai *View* pada waktu bersamaan. Hal ini mengakibatkan *activity/fragment* semakin besar dan sulit untuk dilakukan pengujian.



Gambar 5.1 Arsitektur MVP Android

Sumber: Google

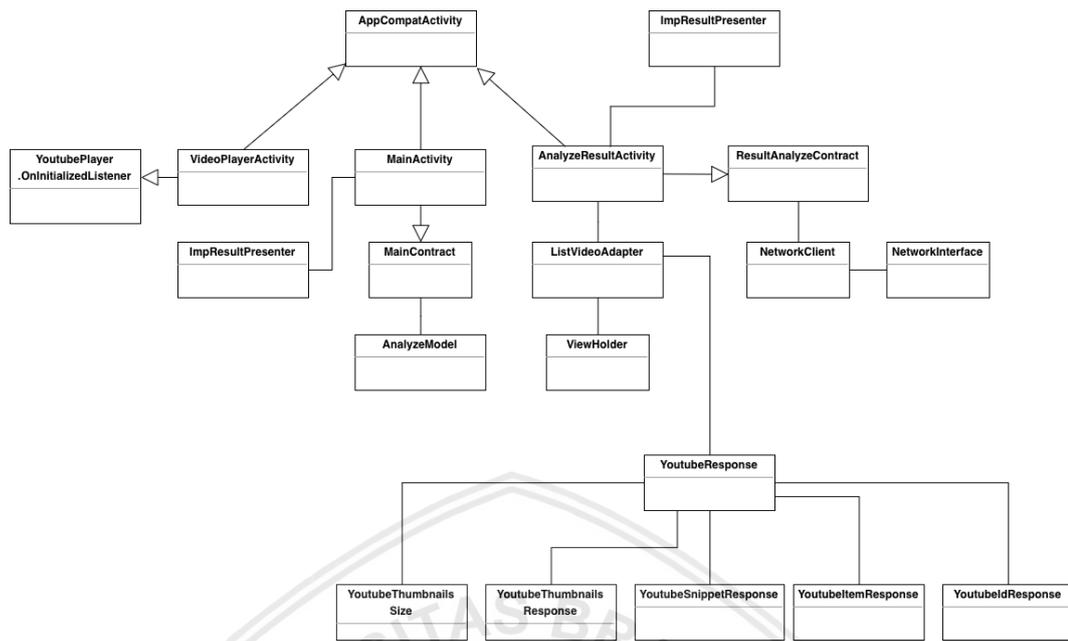
Setelah proses analisis kebutuhan selesai dilakukan, tahap selanjutnya adalah perancangan. Proses perancangan dilakukan menjadi tiga tahapan yaitu perancangan arsitektur, perancangan komponen, dan perancangan antarmuka.



Gambar 5.2 Diagram Arsitektur MVP

5.1.2 Class Diagram

Class diagram merupakan sebuah diagram yang menunjukkan seluruh objek yang ada pada sistem yang dinyatakan pada kelas-kelas serta hubungan antar objek kelas-kelas.



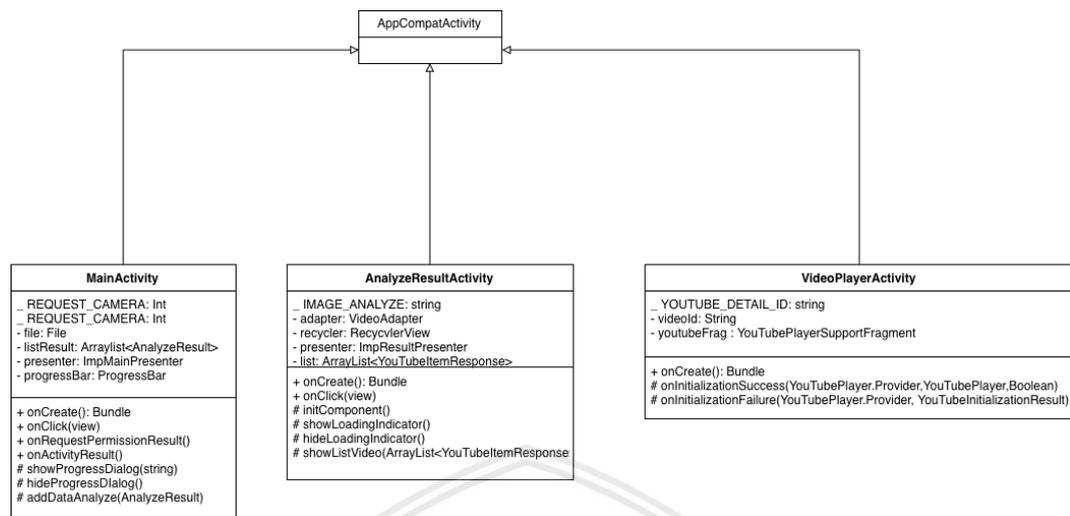
Gambar 5.3 Class Diagram

Pada gambar 5.3 diatas, kelas kelas dibagi menjadi beberapa kelas, yaitu

1. *Activity*, merupakan kelas yang berfungsi untuk menampilkan halaman-halaman antarmuka aplikasi kepada pengguna
2. *Presenter*, merupakan kelas yang berfungsi
3. *Model*, merupakan kelas yang digunakan untuk merepresentasikan objek-objek yang diperlukan pada sistem. Kelas model terbagi menjadi menjadi *package model* dan *package response*
4. *Adapter*, merupakan kelas yang digunakan untuk menjembatani antara *model* dan *view* ataupun antar *view*
5. *Network*, merupakan kelas yang digunakan untuk berkomunikasi dengan Youtube Data API

5.1.2.2 Informasi Class Activity

Kelas-kelas yang terdapat pada gambar 5.3 dapat dilihat pada gambar 5.4 dibawah ini.



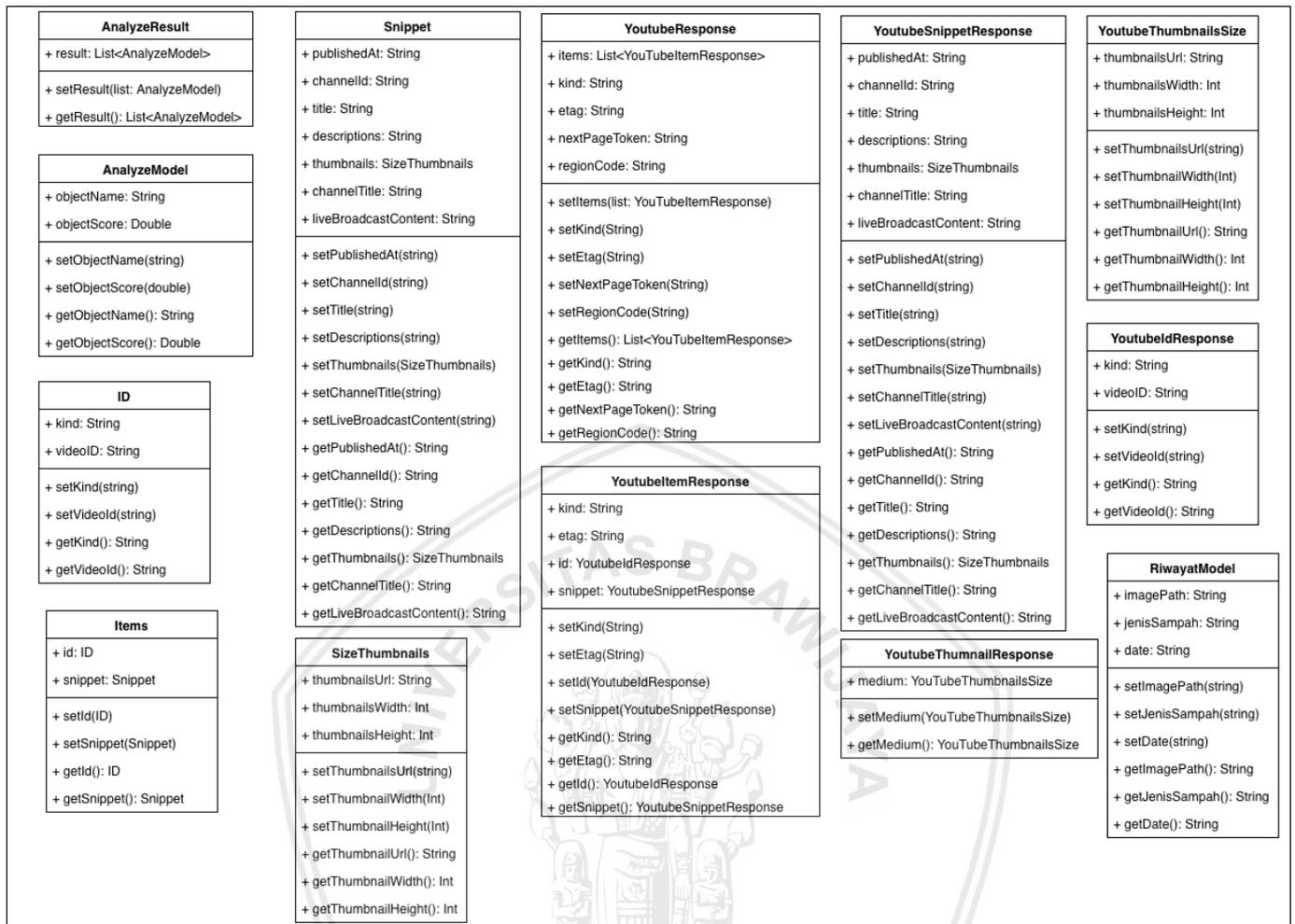
Gambar 5.4 Informasi Class Activity

Pada gambar 5.4 diatas, terdapat tiga kelas *activity* dalam mengembangkan aplikasi Dalang. Keseluruhan *activity* merupakan kelas turunan dari kelas *AppCompatActivity*. Kelas *MainActivity* digunakan untuk menampilkan halaman utama pada aplikasi Dalang. Kelas *AnalyzeResultActivity* digunakan untuk menampilkan halaman hasil analisis objek dan juga daftar video daur ulang. Kelas *VideoPlayerActivity* digunakan untuk menampilkan dan memutar video yang dipilih pengguna pada kelas *AnalyzeResultActivity*.

5.1.2.3 Informasi Kelas Model

Informasi kelas model dan kelas respons yang terdapat pada Gambar 5.3 dapat dilihat pada Gambar 5.5 dibawah ini.



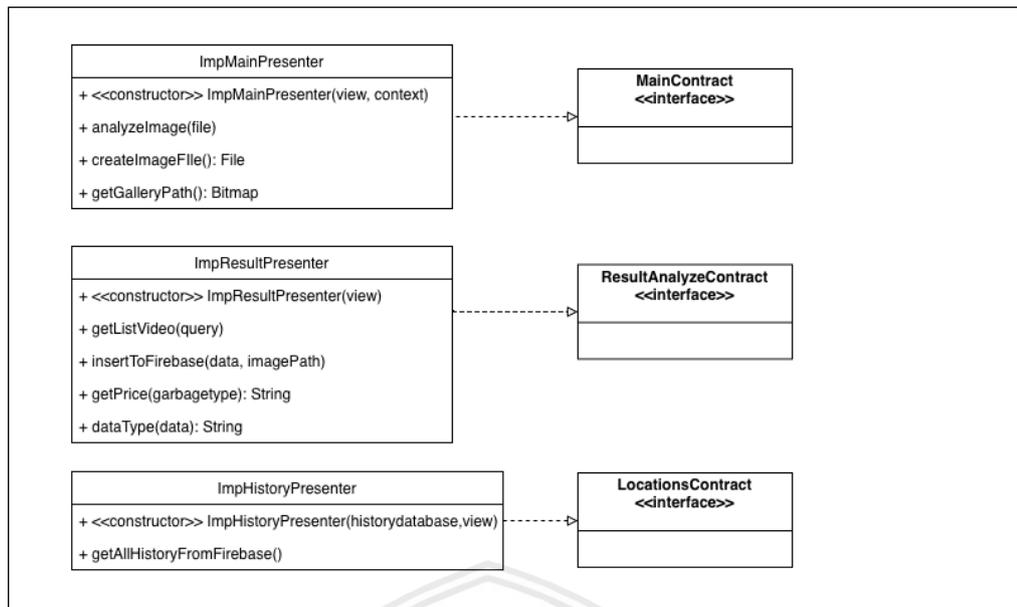


Gambar 5.5 Informasi Kelas Model

Pada gambar 5.5 terdapat tujuh kelas model dan enam kelas *response*, yaitu *AnalyzeResult*, *AnalyzeModel*, *ID*, *Items*, *Snippet*, dan *SizeThumbnails*. Model ini digunakan untuk menyimpan data hasil analisis objek. Selanjutnya kelas *response* terdapat kelas *YoutubeResponse*, *YoutubeItemResponse*, *YoutubeSnippetResponse*, *YoutubeThumbnailsSize*, *YoutubeIdResponse*.

5.1.2.4 Informasi Kelas Presenter

Informasi kelas Presenter yang terdapat pada Gambar 5.3 dapat dilihat pada Gambar 5.6 dibawah ini.



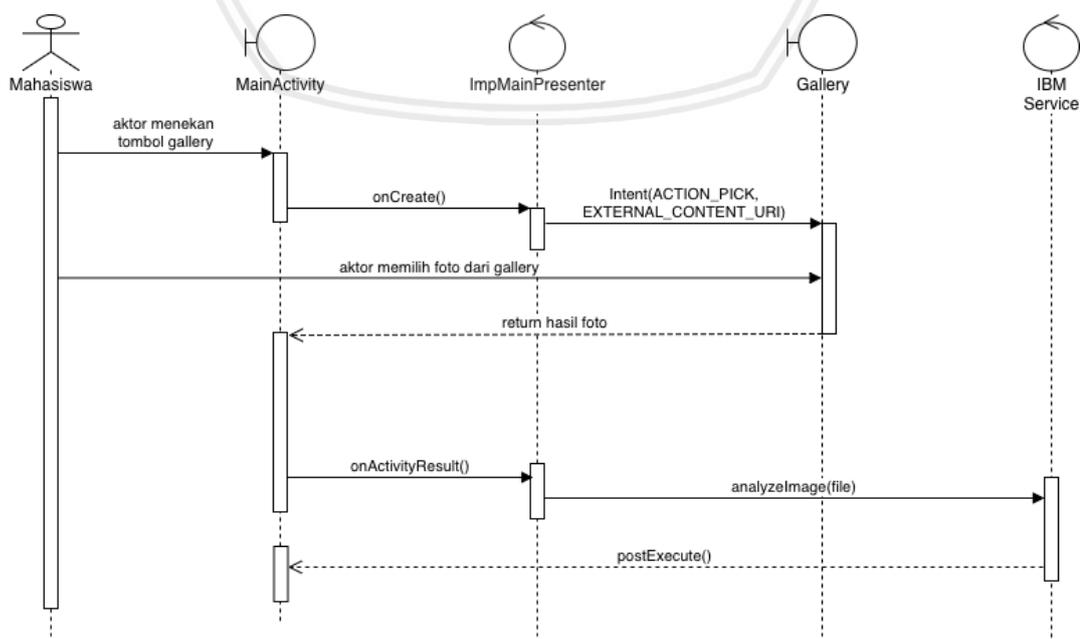
Gambar 5.6 Informasi Kelas Presenter

Pada Gambar 5.6, terdapat dua kelas presenter yaitu ImpMainPresenter dan ImpResultPresenter. Kedua kelas tersebut merupakan turunan dari kelas MainContract untuk kelas ImpMainPresenter dan ResultAnalyzeContract untuk kelas ResultAnalyzeContract.

5.1.3 Sequence Diagram

Pada *sequence diagram* dijelaskan bahwa urutan proses yang terjadi untuk mencapai suatu kebutuhan sistem. Semua objek pada *sequence diagram* merupakan hasil identifikasi dari spesifikasi kebutuhan dan *use case scenario* yang ada pada tahap analisis kebutuhan sebelumnya.

5.1.3.1 Sequence Diagram Mengambil Foto Dengan Gallery

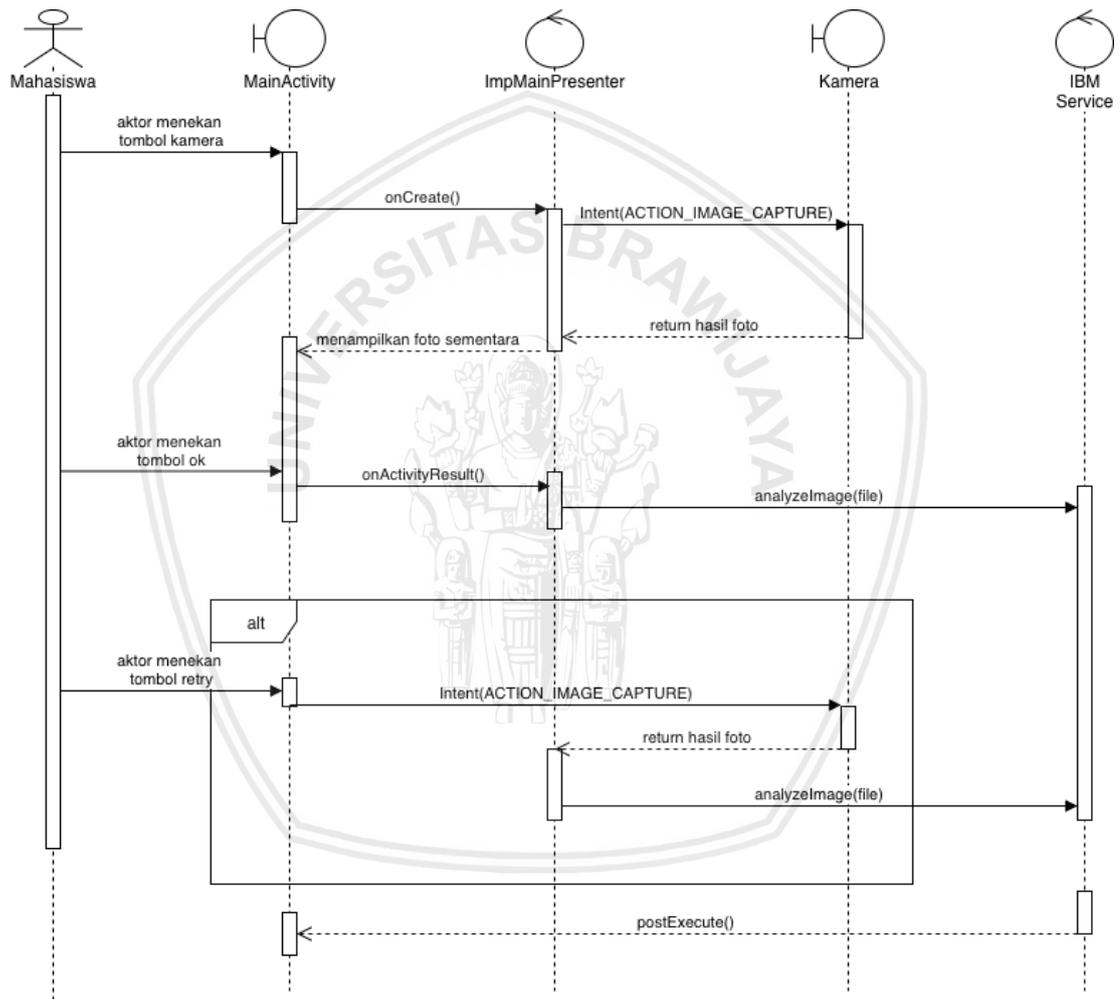


Gambar 5.7 Sequence Diagram Mengambil Foto Dari Gallery



Dalam gambar 5.7 *Sequence diagram* mengambil foto dari *gallery*, mahasiswa cukup menekan tombol *gallery* yang ada pada halaman utama. Setelah itu sistem akan memanggil method `onCreate()` dan akan memanggil method `Intent(ACTION_PICK, EXTERNAL_CONTENT_URI)`. Method tersebut akan memanggil *gallery default* yang ada pada android. Kemudian mahasiswa akan memilih foto untuk dianalisis oleh *IBM Visual Recognition*.

5.1.3.2 Sequence Diagram Menganalisis Objek dari Kamera

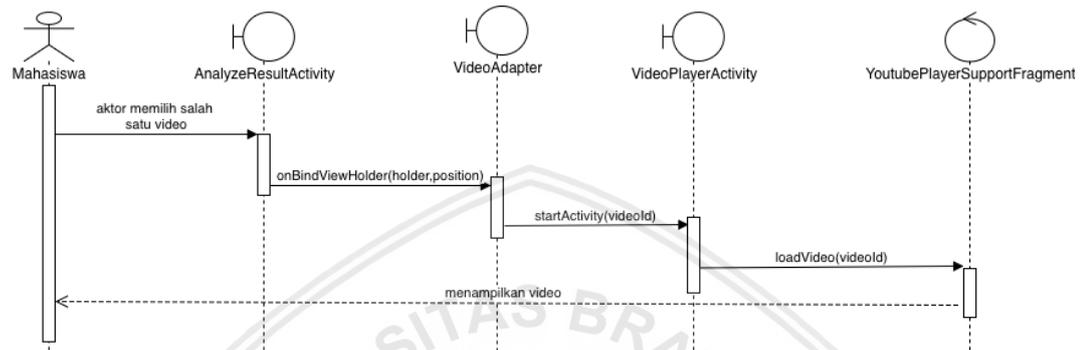


Gambar 5.8 Sequence Diagram Menganalisis Objek Dari kamera

Dari Gambar 5.8 *sequence diagram* menganalisis objek dari kamera, mahasiswa cukup menekan tombol kamera kemudian *MainActivity* akan memanggil method `onCreate()`. Setelah itu *presenter* akan memanggil `Intent(ACTION_IMAGE_CAPTURE)` untuk menampilkan kamera *default* yang tersedia pada android. Kemudian fitur kamera akan menampilkan foto yang diambil dan akan menampilkan pilihan dua tombol. Jika mahasiswa menekan

tombol ok, maka sistem akan memanggil method `onActivityResult()` kemudian pada `ImpMainPresenter` akan memanggil `analyzeImage()` untuk melakukan analisis objek. Jika mahasiswa menekan tombol `retry`, maka sistem akan kembali ke kamera untuk mengambil ulang foto untuk dianalisis.

5.1.3.3 Sequence Diagram Memutar Video



Gambar 5.9 Sequence Diagram Memutar Video

Pada Gambar 5.9 sequence diagram memutar video, mahasiswa cukup memilih salah satu video pada daftar video yang tersedia pada `AnalyzeResultActivity`, kemudian akan memanggil method `onBindViewHolder()` pada `VideoAdapter` untuk mengirimkan data berupa `videoId` ke `VideoPlayerActivity`. Setelah `VideoPlayerActivity` mendapatkan `videoId`, maka akan memanggil method `loadVideo()` dengan `videoId` sebagai parameternya dan akan menampilkan video sesuai dengan yang dipilih.

5.1.4 Perancangan Komponen

Perancangan komponen menggambarkan rincian sub-sistem dari setiap komponen perangkat lunak. Pada perancangan komponen pada pengembangan aplikasi ini dipilih satu algoritma method utama dari masing-masing komponen.

5.1.4.1 Algoritme Menganalisis Objek

Nama kelas : `ImpMainPresenter`
 Nama Operasi : `analyzeImage(file)`
 Algoritma :

```

if image != null then
    inisialisasi VisualRecognition
    set API key
    inisialisasi ClassifyImageOptions
    memanggil fungsi execute
    mendapat respon dari VisualRecognition
endif
    
```

5.1.4.2 Algoritme Mengirim *Request* Video

Nama kelas : `ImpResultPresenter`

Nama Operasi : `getListVideo(query)`

Algoritma :

```
Inisialisasi networkInstance
Memanggil fungsi getListVideo()
Memanggil fungsi showLoadingIndicator()
If respon == 200 OK then
    Mendapatkan respon dari API
    Memanggil fungsi hideLoadingIndicator()
Else
    Return exception
endif
```

5.1.4.3 Algoritme Memutar Video

Nama Kelas : `VideoPlayerActivity`

Nama operasi : `loadVideo()`

Algoritma :

```
Inisialisasi YoutubePlayerSupportFragment
If videoId != null
    Memanggil loadVideo(videoId)
else
    memanggil method onDetach()
endif
```

5.1.5 Perancangan Basis Data

Perancangan basis data pada aplikasi Dalang menggunakan *Firestore* dan *API response* dari *Youtube Data API* dan *IBM Visual Recognition* sehingga database tidak memiliki relasi. Untuk gambar perancangan basis data aplikasi Dalang dapat dilihat pada gambar berikut:

5.1.5.1 Perancangan Basis Data IBM Visual Recognition Response

Berikut merupakan skema basis data *IBM Visual Recognition Response* yang dapat dilihat pada Gambar 5.10 berikut ini:

Tabel 5.1 Perancangan Basis Data IBM Visual Recognition

```
{
  "custom_classes": int,
  "images": [
    {
      "classifiers": [
```

```

        {
            "classes": [
                {
                    "class": string,
                    "score": long,
                    "type_hierarchy": string
                }
            ],
            "classifier_id": string,
            "name": string
        }
    ],
    "image": string
}
],
"images_processed": int
}
    
```

Berikut pada Tabel 5.2 merupakan keterangan skema basis data yang ada pada Tabel 5.1

Tabel 5.2 Penjelasan Skema Basis Data IBM Visual Recognition

No	Nama Field	Tipe data	Keterangan
1	Images	List	Untuk menyimpan daftar keterangan gambar yang dianalisis
2	Classifiers	List	Untuk menyimpan daftar keterangan klasifikasi gambar yang dianalisis
3	Classifier_id	String	Untuk menyimpan id gambar
4	name	String	Untuk menyimpan tipe gambar
5	Classes	List	Untuk menyimpan daftar hasil analisis sistem
6	Class	String	Untuk menyimpan hasil analisis gambar
7	Score	Long	Untuk menyimpan nilai hasil analisis gambar
8	Image	String	Untuk menyimpan nama gambar yang di unggah

9	Images_processed	Int	Untuk menyimpan jumlah gambar yang dianalisis
10	Custom_classes	int	Untuk menyimpan jumlah gambar khusus yang dianalisis

5.1.5.2 Perancangan Basis Data Youtube Data API Response

Berikut ini merupakan perancangan basis data Youtube Data API response yang digunakan untuk menampilkan daftar video

Tabel 5.3 Perancangan Basis Data Youtube Data API

```

{
  "kind": string,
  "etag": string,
  "nextPageToken": string,
  "regionCode": string,
  "pageInfo": {
    "totalResults": int,
    "resultsPerPage": int
  },
  "items": [
    {
      "kind": string,
      "etag": string,
      "id": {
        "kind": string,
        "videoId": string
      },
      "snippet": {
        "publishedAt": string,
        "channelId": string,
        "title": string,
        "description": string,
        "thumbnails": {
          "default": {
            "url": string,
            "width": int,
            "height": int
          },
          "medium": {
            "url": string,
            "width": int,
            "height": int
          },
          "high": {
            "url": string,

```

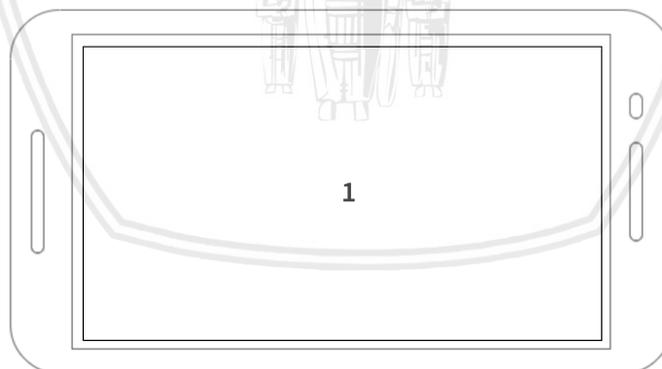
```
        "width": int,  
        "height": int  
    },  
    },  
    "channelTitle": string,  
    "liveBroadcastContent": string  
  }  
  ]  
}
```

5.1.6 Perancangan Antarmuka

5.1.6.1 Perancangan Antarmuka Iterasi Pertama

Pada perancangan antarmuka berisi halaman-halaman yang menyusun aplikasi dalam bentuk *low level design*. Berikut ini adalah hasil rancangan antarmuka aplikasi Dalang. Untuk memudahkan dalam menjelaskan setiap bagian dalam rancangan antarmuka diberikan penomoran pada masing-masing halaman antarmuka yang dibahas.

(a) Perancangan Antarmuka Halaman Video Player



Gambar 5.10 Rancangan Antarmuka Halaman Memutar Video Iterasi Pertama

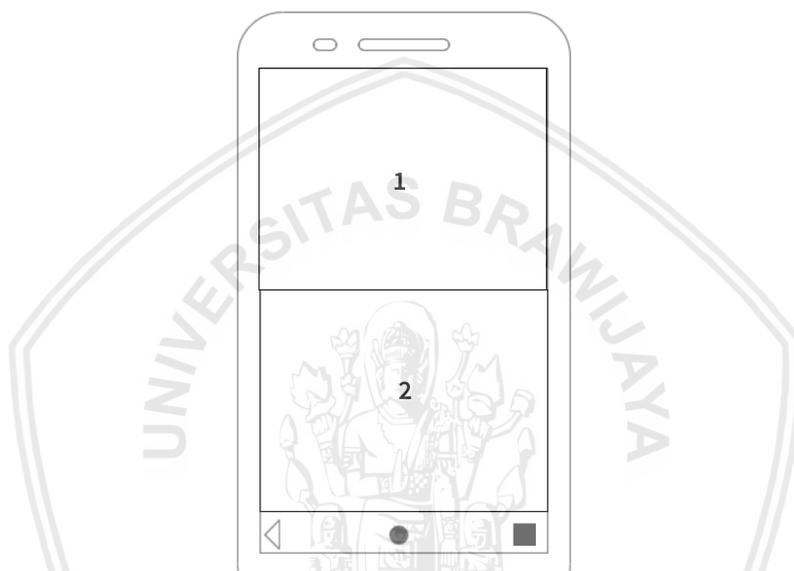
Perancangan antarmuka pada Gambar 5.10 akan dijelaskan pada Tabel 5.4 dibawah ini

Tabel 5.4 Penjelasan Perancangan Antarmuka Memutar Video Iterasi Pertama

No	Nama Objek	Tipe	Keterangan
1	Stream video player	Video Player	Video player untuk menampilkan video yang dipilih oleh actor.

(b) Perancangan Antarmuka Mengambil Foto

Perancangan antarmuka pada Gambar 5.11 akan dijelaskan pada Tabel 5.5 dibawah ini



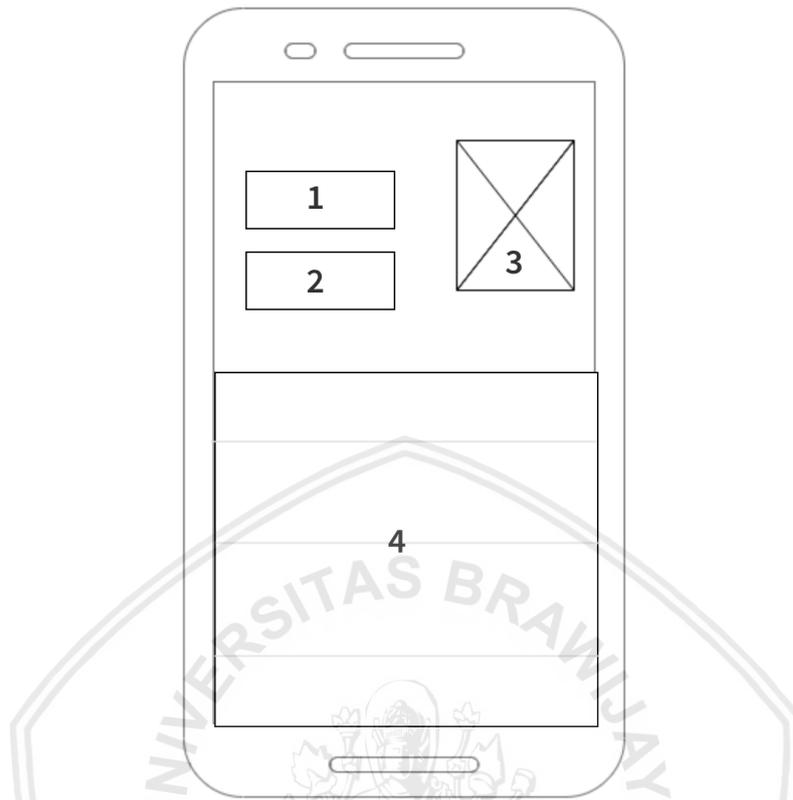
Gambar 5.11 Perancangan Antarmuka Tampilan Awal Iterasi Pertama

Tabel 5.5 Penjelasan Perancangan Antarmuka Tampilan Awal Iterasi Pertama

No	Nama Objek	Tipe	Keterangan
1	Kamera	Logo Button	Button untuk menampilkan fitur kamera <i>default</i> yang ada pada <i>android</i>
2	Gallery	Logo Button	Button untuk menampilkan fitur <i>gallery default</i> yang ada pada <i>android</i>

(c) Perancangan Antarmuka Halaman Melihat Daftar Video

Perancangan antarmuka pada Gambar 5.12 akan dijelaskan pada tabel 5.6 dibawah ini



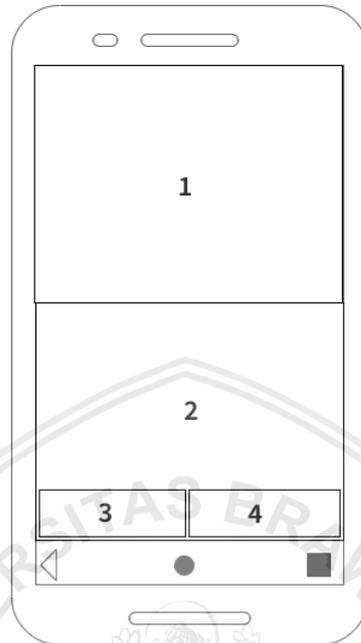
Gambar 5.12 Rancangan Antarmuka Melihat Daftar Video

Tabel 5.6 Perancangan Antarmuka Melihat Daftar Video

No	Nama Objek	Tipe	Keterangan
1	Nama objek yang dianalisis	Text	Text untuk menampilkan hasil analisis object yang dilakukan oleh <i>IBM Visual Recognition</i>
2	Kisaran harga objek yang dianalisis	Text	Text untuk menampilkan kisaran harga dari hasil analisis objek
3	Gambar hasil analisis	Gambar	Gambar untuk menampilkan objek yang sudah dianalisis sebelumnya
4	<i>List Video</i>	<i>List View</i>	<i>List View</i> akan menampilkan daftar video berdasarkan hasil analisis

5.1.6.2 Perancangan Antarmuka Iterasi Kedua

Perancangan antarmuka pada Gambar 5.13 akan dijelaskan pada Tabel 5.2 dibawah ini



Gambar 5.13 Perancangan Antarmuka Tampilan Awal

Tabel 5.7 Penjelasan Perancangan Antarmuka Tampilan Awal

No	Nama Objek	Tipe	Keterangan
1	Kamera	Logo Button	Button untuk menampilkan fitur kamera <i>default</i> yang ada pada <i>android</i>
2	Gallery	Logo Button	Button untuk menampilkan fitur <i>gallery default</i> yang ada pada <i>android</i>
3	Tab Bar <i>Home</i>	Logo Button	Tab baru untuk menampilkan tombol aksi
4	Tab Bar <i>History</i>	Logo Button	Tab baru untuk menampilkan riwayat pencarian

5.2 Implementasi Sistem

Pada bagian implementasi sistem dijelaskan proses pembangunan sistem yang mengacu pada perancangan yang telah dibuat. Perancangan dilakukan berdasarkan hasil analisis kebutuhan sebelumnya. Tahapan implementasi dimulai dengan mendefinisikan spesifikasi perangkat keras dan perangkat lunak yang digunakan dalam membangun sistem, selanjutnya implementasi kode program dan implementasi antarmuka.

5.2.1 Spesifikasi Sistem

5.2.1.1 Spesifikasi Sistem Perangkat Keras

Perangkat keras yang digunakan dalam proses pengembangan aplikasi Dalang dijelaskan pada tabel 5.4 dibawah ini.

Tabel 5.8 Spesifikasi Perangkat Keras

Nama Komponen	Spesifikasi
Laptop	<ol style="list-style-type: none">1. MacBook Pro (13-inch, 2017, Two Thunderbolt 3 ports)2. Processor 2,3 GHz Intel Core i53. Memory 8 GB 2133 MHz LPDDR34. Graphic Intel Iris Graphics 640 1536 MB

5.2.1.2 Spesifikasi Sistem Perangkat Lunak

Perangkat lunak yang digunakan dalam proses pengembangan aplikasi Dalang dijelaskan pada tabel 5.5 dibawah ini

Tabel 5.9 Spesifikasi Perangkat Lunak

Nama Komponen	Spesifikasi
<i>Editor</i> Dokumentasi	Microsoft Office Word 2016
<i>Editor</i> Perancangan	<i>Draw.io</i> dan <i>wireframepro</i>
<i>Editor</i> Pemrograman	Android Studio 3.2
Youtube Data API	Youtube Data API v2
Visual Recognition	IBM Visual Recognition 6.8.0
Bahasa Pemrograman	Kotlin 1.2.51
DBMS	Firebase Database
<i>Emulator</i>	<i>Samsung A8 5.6"</i>

5.2.1.3 Spesifikasi Sistem Operasi

Tabel 5.10 Spesifikasi Sistem Operasi

Nama Komponen	Spesifikasi
Sistem Operasi	macOS High Sierra 10.14

5.2.2 Implementasi Data

Implementasi data dilakukan berdasarkan analisis data yang sebelumnya dan telah dilakukan sesuai dengan kebutuhan sistem. Database yang diterapkan pada pengembangan aplikasi Dalang ini menggunakan *Firestore*. *Firestore* memiliki fitur *real-time database* dan diintegrasikan dengan Google. *Firestore* menggunakan *NoSQL* sehingga basis data yang digunakan dapat diubah menjadi JSON.

Tabel 5.11 Implementasi Basis Data Youtube Data API

```
{
  "kind": "youtube#searchListResponse",
  "etag": "\"XI7nbFXulyBIpL0ayR_gDh3eu1k/TErokAgMAoO1LyNA8rl2LRz1254\"",
  "nextPageToken": "CBQQAA",
  "regionCode": "ID",
  "pageInfo": {
    "totalResults": 11656,
    "resultsPerPage": 20
  },
  "items": [
    {
      "kind": "youtube#searchResult",
      "etag": "\"XI7nbFXulyBIpL0ayR_gDh3eu1k/w2bD-QUL4bPhmt1Bm9ccan5_p9A\"",
      "id": {
        "kind": "youtube#video",
        "videoId": "uiBehMzH11U"
      },
      "snippet": {
        "publishedAt": "2016-08-09T23:59:52.000Z",
        "channelId": "UCtCPfGNFqWIVhSgioWTtQqQ",
        "title": "10 Cara Kreatif DIY untuk Reuse / Botol Plastik Daur Ulang bagian 2",
        "description": "",
        "thumbnails": {
          "default": {
```

```

        "url":
"https://i.ytimg.com/vi/uiBehMzH11U/default.jpg",
        "width": 120,
        "height": 90
    },
    "medium": {
        "url":
"https://i.ytimg.com/vi/uiBehMzH11U/mqdefault.jpg",
        "width": 320,
        "height": 180
    },
    "high": {
        "url":
"https://i.ytimg.com/vi/uiBehMzH11U/hqdefault.jpg",
        "width": 480,
        "height": 360
    }
    },
    "channelTitle": "Jihan Talita Ulfa",
    "liveBroadcastContent": "none"
}
}
]

```

5.2.3 Implementasi Kode Program

Setelah proses analisis kebutuhan selesai dilakukan, tahap selanjutnya adalah implementasi kode program seperti dibawah ini:

5.2.3.1 Implementasi Kode Program Menganalisis Objek

Kode program yang terdapat dibawah ini merupakan kode program untuk menganalisis objek. Terdapat satu parameter yaitu params dengan tipe data string. Pada potongan kode program ini terdapat method untuk mengirimkan file ke *IBM Visual Service* yang nantinya akan diteruskan ke aplikasi dalam bentuk *JSON respons*.

```

override fun doInBackground(vararg params: String?):
String {
    var imageStream: InputStream? = null

    try {
        imageStream = FileInputStream(file)
    } catch (e: FileNotFoundException) {
        e.printStackTrace()
    }

    val options = ClassifyOptions.Builder()

```

```

        .imagesFile(imageStream)
        .imagesFilename(file.name)
        .build()

visualRecognition.classify(options).enqueueWithDetails(
    object :
    ServiceCallbackWithDetails<ClassifiedImages>{
        override fun onFailure(e: Exception?) {
            try {
                Log.e("failure-ibm",
e?.localizedMessage + " " + e?.cause)
                view.hideProgressDialog()
            } catch (e: IOException) {
                Log.e("failure-ibm", e.message)
                view.hideProgressDialog()
            }
        }

        override fun onResponse(response:
Response<ClassifiedImages>?) {
            Log.d("data-rec", "response -> " +
response?.result?.images?.get(0)?.classifiers?.get(0)
?.classes)
            val result =
response?.result?.images?.get(0)?.classifiers?.get(0)
?.classes?.let {
                filterData(it)
            }.toString()
            Log.d("data-list", "list 2 -> $result")
        }
    })

    return result
}

```

5.2.3.2 Implementasi Kode Program Mengirim Request Video

Potongan kode program ini untuk mengirimkan *request video* ke YouTube Data API. Inisialisasi *networkInstance* dilakukan untuk melakukan *request* ke server Youtube dan akan dikembalikan ke aplikasi dalam bentuk *JSON respons*.

```

// Inisialisasi networkInstance

private var request =
NetworkClient().getInstance().create(NetworkInterface
::class.java)

```

```

// Memanggil fungsi getListVideo()
override fun getListVideo(query: String){

//Memanggil fungsi showLoadingIndicator()
    view.showLoadingIndicator()

request.getListVideo("snippet", 10,
    query, BuildConfig.YOUTUBE_API_KEY)
    .subscribeOn(Schedulers.io())

.observeOn(AndroidSchedulers.mainThread())
    .subscribe({ res ->
        when {
            // If respon == 200 OK then
            res.isSuccessful -> {
                Log.d("data-youtube",
res.toString())

                // Mendapatkan respon dari API
view.showListVideo(res.body()!!.items!!)

                // Memanggil fungsi
hideLoadingIndicator()
                view.hideLoadingIndicator()
            }else -> {
                view.hideLoadingIndicator()
            }
        }
    },
//Return exception
{ error ->
    Log.e("error-
video",error.localizedMessage.toString())
})
}

```

5.2.3.3 Implementasi Kode Memutar Video

Potongan kode program memutar video ini digunakan untuk melakukan pemutaran video berdasarkan Id video yang diberikan. Terdapat inisialisasi `YoutubePlayerSupportFragment` dan akan memanggil dua *callback* untuk mengatasi `onInitializationSuccess` dan `onInitializationFailure`

```
//inisialisasi YoutubePlayerSupportFragment
youtubeFrag =
supportFragmentManager.findFragmentById(R.id.youtube_
player_streaming) as YouTubePlayerSupportFragment
youtubeFrag.initialize(BuildConfig.YOUTUBE_API_KEY,
this)

// If videoId != null
override fun onInitializationSuccess(provider:
YouTubePlayer.Provider?, youtubePlayer:
YouTubePlayer?, b: Boolean) {
    if (videoId.isNullOrEmpty()) {

// Memanggil loadVideo(videoId)
        youtubePlayer?.loadVideo(videoId)
youtubePlayer?.setFullscreen(true)

    }else{
        // memanggil method onDetach()
        youtubeFrag.onDetach()
    }
}

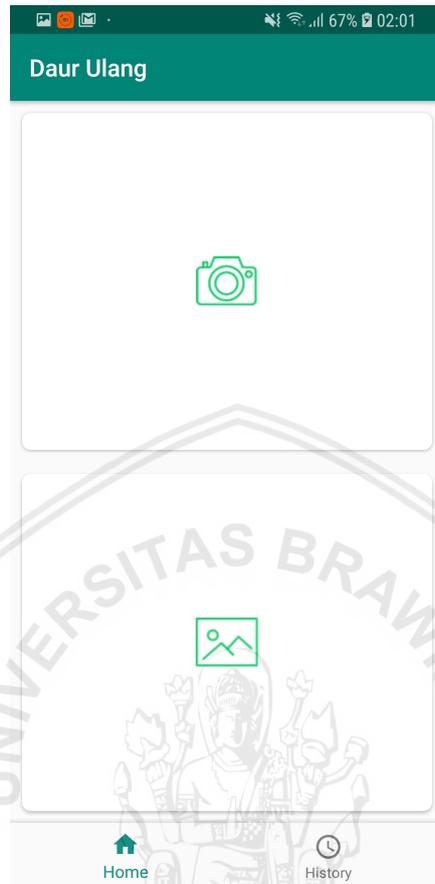
override fun onInitializationFailure(provider:
YouTubePlayer.Provider?, youtubeInitializationResult:
YouTubeInitializationResult?) {

}
```

5.2.4 Implementasi Antarmuka

Proses Implementasi antarmuka ini dilakukan dengan mengacu pada perancangan antarmuka yang sudah dirancang sebelumnya.

5.2.4.1 Implementasi Perancangan Antarmuka Mengambil Foto

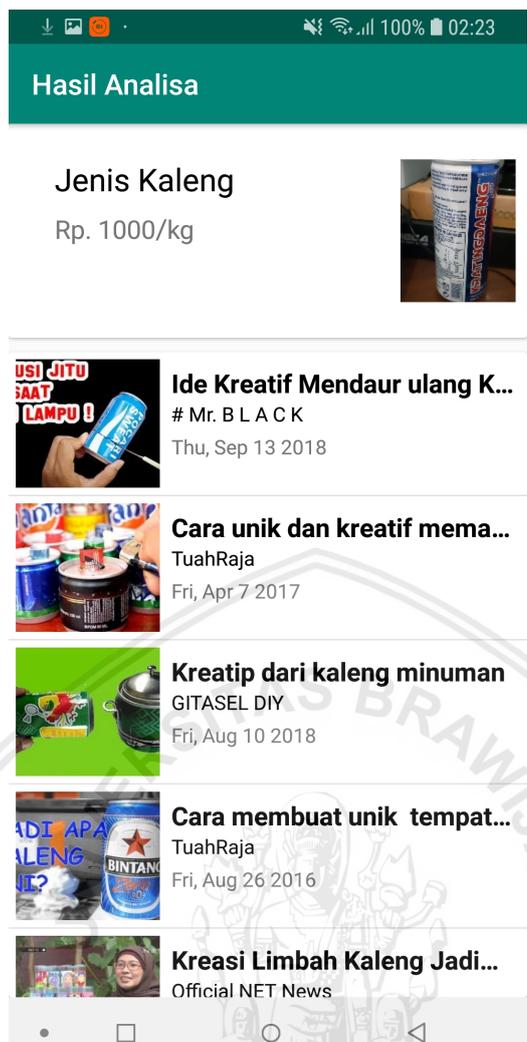


Gambar 5.14 Implementasi Antarmuka Halaman Mengambil Foto

Gambar 5.13 diatas menunjukkan halaman Main menunjukkan dua *button*. *Button* dengan logo kamera berfungsi untuk mengambil foto dengan menggunakan kamera *default* android. *Button* dengan logo *gallery* berfungsi untuk mengambil foto dari *gallery default* android.

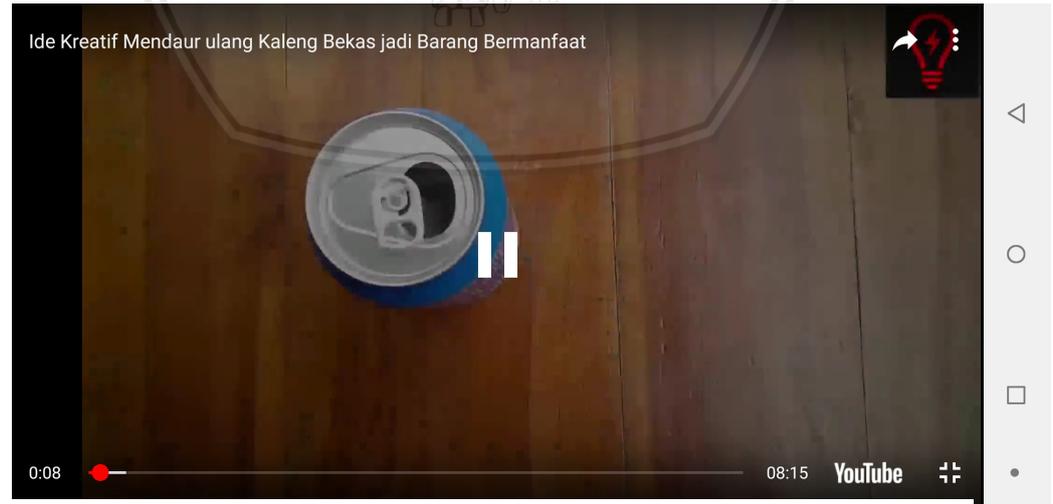
5.2.4.2 Implementasi Antarmuka Melihat Daftar Video

Implementasi antarmuka melihat daftar video terdiri dari 4 komponen, yaitu dua teks yang menampilkan jenis atau kategori objek yang sudah dianalisis dan menampilkan harga sampah berdasarkan objek yang dianalisis oleh sistem. Terdapat gambar pada bagian kanan atas yang merupakan objek yang dianalisis oleh sistem. Pada bagian bawah antarmuka terdapat daftar video yang disediakan oleh Youtube Data berdasarkan objek yang dianalisis oleh sistem.



Gambar 5.15 Implementasi Antarmuka Halaman Melihat Daftar Video

5.2.4.3 Implementasi Antarmuka Video Player



Gambar 5.16 Implementasi Antarmuka Video Player

Pada Gambar 5.17 diatas terdapat sebuah *video player* yang menampilkan dan memutar video yang dipilih oleh pengguna. *Video player* yang digunakan pada aplikasi Dalang ini menggunakan *YouTube*.

BAB 6 PENGUJIAN

Tahapan pengujian dilakukan setelah melakukan implementasi sistem. Pengujian bertujuan untuk memastikan apakah implementasi dan seluruh fungsionalitas sistem sudah sesuai dengan analisis kebutuhan dan perancangan sistem atau tidak. Tahap pengujian yang akan dilakukan adalah pengujian unit, pengujian integrasi dan pengujian validasi.

6.1 Pengujian Unit

Pengujian unit merupakan pengujian yang dilakukan pada komponen terkecil dari sebuah sistem yaitu kelas. Pengujian unit dilakukan untuk memastikan bahwa komponen yang telah diimplementasikan berjalan sesuai dengan fungsionalitasnya. Pengujian unit dilakukan dengan menggunakan metode pengujian *whitebox* dengan jenis pengujian *basis path*. Metode ini memungkinkan perancang *test case* mendapatkan ukuran kompleksitas logika dari perancangan prosedural dan menggunakan ukuran ini sebagai petunjuk untuk mendefinisikan basis set dari jalur pengerjaan. *Test case* perintah minimal satu kali selama uji coba. Langkah-langkah dalam pembuatan *test case* pada *basis path testing* adalah sebagai berikut:

1. Pembuatan diagram alir dari perancangan procedural atau *source code*
2. Menentukan *cyclomatic complexity* dari diagram alir
3. Menentukan *independent path* dari diagram alir

6.1.1 Pengujian Unit Kelas AsyncTask operasi `doInBackground(file)`

1. Pseudocode

```

var inputStream: InputStream? = null
var result: String? = null
try {
    inputStream = FileInputStream(file)
} catch (e: FileNotFoundException) {
    e.printStackTrace()
}

val options = ClassifyOptions.Builder()
    .imagesFile(inputStream)
    .imagesFilename(file.name)
    .build()
visualRecognition.classify(options).enqueue(object :
ServiceCallback<ClassifiedImages> {
    override fun onResponse(response: ClassifiedImages) {
        Log.d("data-rec", "response image -> " +
response.images.toString())
        Log.d("data-rec", "response custom class -> " +
response.customClasses)
        Log.d("data-rec", "response image processed -> " +
response.imagesProcessed)
        Log.d("data-rec", "response image warning -> " +
response.warnings)
        Log.d("data-rec", "response -> " +

```

```

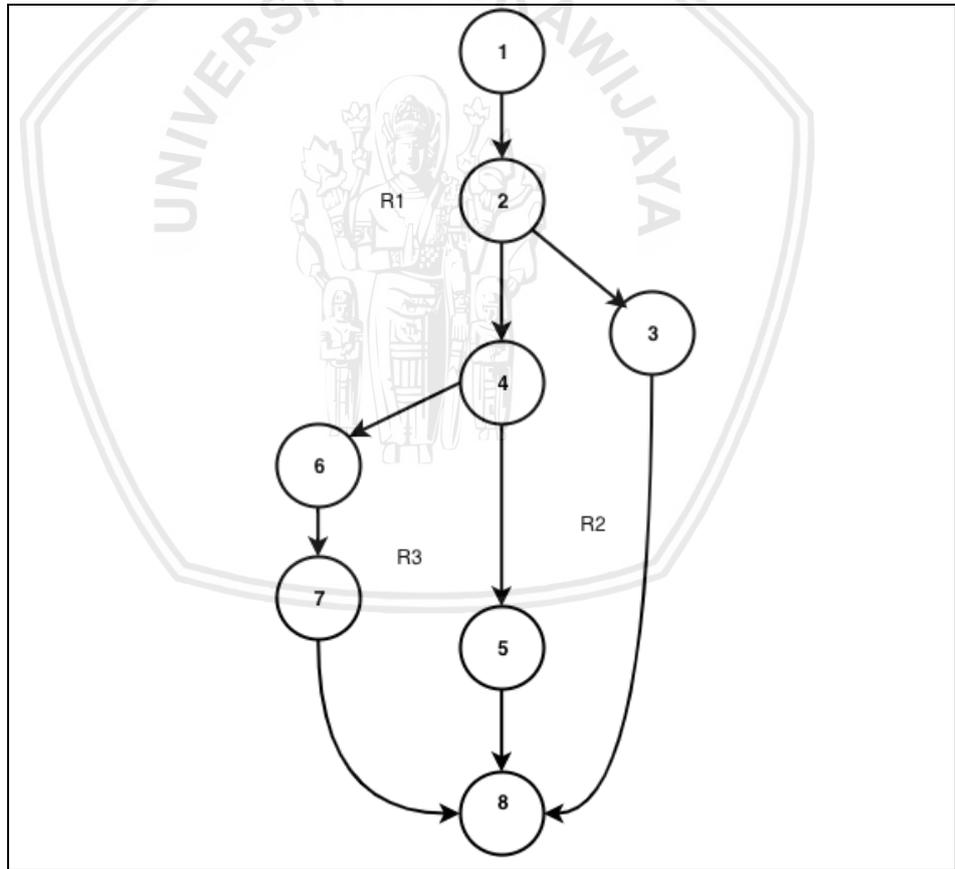
response.toString()

    }

    override fun onFailure(e: Exception) {
        try {
            Log.e("failure-ibm", e.localizedMessage + " " +
                e.cause)
            view.hideProgressDialog()
        } catch (e: IOException) {
            Log.e("failure-ibm", e.message)
            view.hideProgressDialog()
        }
    }
})
return result
    
```

2. Basis Path Testing

a. Flow Graph



b. Cyclomatic Complexity

- a) $V(G) = 3$, ada 3 region R1, R2, R3
- b) $V(G) = 9 \text{ edges} - 8 \text{ node} + 2 = 3$
- c) $V(G) = 2 \text{ predicate nodes} + 1 = 3$

c. Independent Path

- a) Jalur 1 = 1 – 2 – 3 – 8
- b) Jalur 2 = 1 – 2 – 4 – 5 – 8
- c) Jalur 3 = 1 – 2 – 4 – 6 – 7 – 8

Test case dan hasil akan dijelaskan pada Tabel 6.1 berikut ini

Table 6.1 Hasil Pengujian unit kelas AsyncTask operasi doInBackground()

No	No. Jalur	Prosedur Uji	Expected Result	Result	Status
1.	1	Memanggil operasi doInBackground dengan parameter null	Tidak berhasil dianalisis oleh sistem dan menampilkan pesan error	Tidak berhasil dianalisis oleh sistem dan menampilkan pesan error	Valid
2.	2	Memanggil operasi doInBackground dengan parameter file	File foto berhasil dianalisis oleh sistem dan menuju ke halaman hasil analisis	File foto berhasil dianalisis oleh sistem dan menuju ke halaman hasil analisis	Valid
3	3	Memanggil operasi doInBackground dengan parameter file	File foto tidak berhasil dianalisis oleh sistem	File foto tidak berhasil dianalisis oleh sistem	Valid

6.1.2 Pengujian unit klas MediaPlayerActivity operasi initializeYoutubeFrag ()

1) Pseudocode

```

youtubeFrag.initialize (BuildConfig.YOUTUBE_API_KEY, object
YouTubePlayer.OnInitializedListener {
    override fun onInitializationSuccess (provider:
YouTubePlayer.Provider?, youtubePlayer: YouTubePlaye 2 ):
Boolean) {
        youtubePlayer?.loadVideo (videoId) 3
        youtubePlayer?.setFullscreen (true)
    }

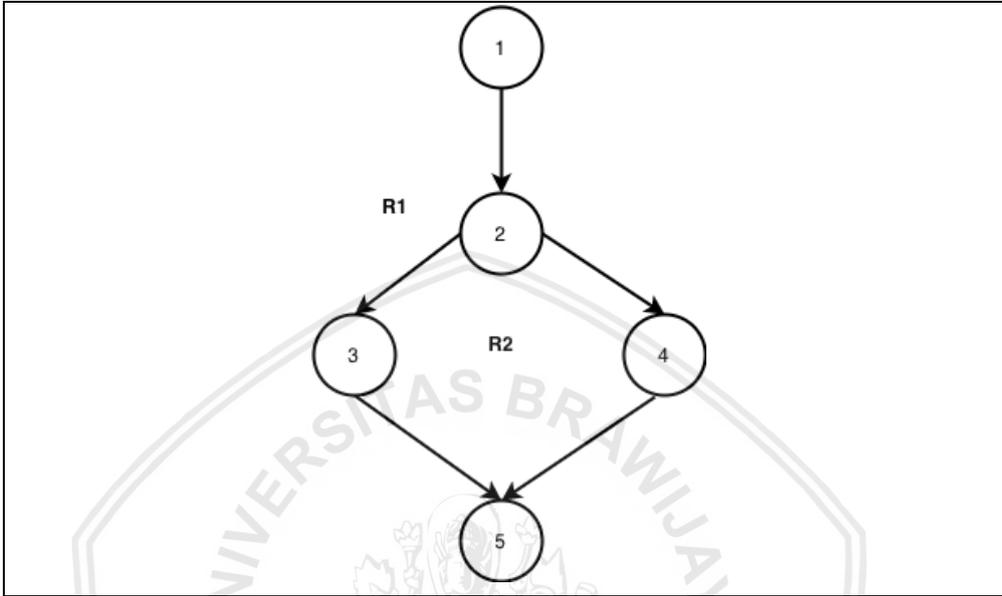
    override fun onInitializationFailure (provider:
YouTubePlayer.Provider?, youTubeInitializationResult:
YouTubeInitializationResult?) {

youtubeInitializationResult?.getErrorDialog (this@MediaPlayer
Activity,0)?.show () 5
    }
}
    
```

```
})
```

2) Basis Path Testing

a. Flow Graph



b. Cyclomatic Complexity

- a) $V(G) = \text{ada 2 region, R1, R2}$
- b) $V(G) = 5 \text{ edges} - 5 \text{ node} + 2 = 2$
- c) $V(G) = 1 \text{ Predicate} + 1 = 2$

c. Independent Path

- a) Jalur 1 = 1 – 2 – 3 – 5
- b) Jalur 2 = 1 – 4 – 5

Test case dan hasil akan dijelaskan pada tabel 6.2 berikut ini

No	No Jalur	Prosedur Uji	Expected Result	Result	Status
1	1	Memasukkan parameter berupa API_KEY yang didapat dari YouTube Data API	Sistem dapat memutar video yang dipilih oleh pengguna	Sistem dapat memutar video yang dipilih oleh pengguna	valid

2	2	Memasukkan parameter berupa API_KEY yang salah	Sistem tidak dapat memutar video yang dipilih oleh pengguna	Sistem tidak dapat memutar video yang dipilih oleh pengguna	valid
---	---	--	---	---	-------

6.1.3 Pengujian unit klas ImpResultPresenter operasi getListVideo ()

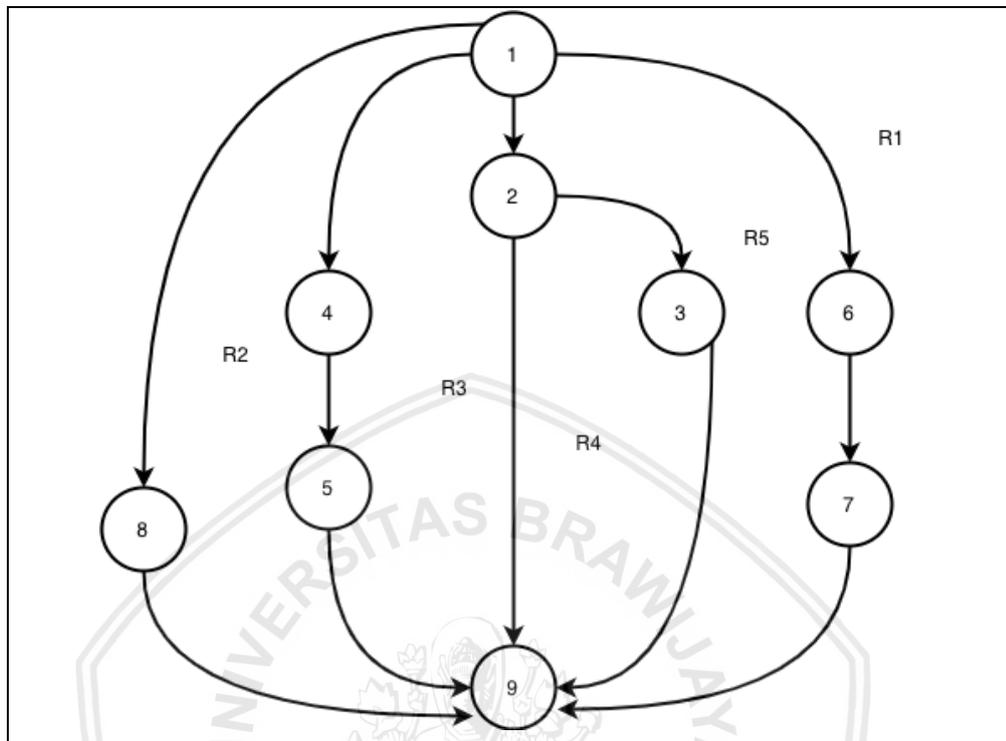
1) Pseudocode

```

view.showLoadingIndicator()
request.getListVideo("snippet", 15,
    query, BuildConfig.YOUTUBE_API_KEY)
    .subscribeOn(Schedulers.io())
    .observeOn(AndroidSchedulers.mainThread())
    .subscribe({ res ->
        when {
            res.isSuccessful -> {
                Log.d("data-youtub
res.toString()
view.showListVideo(res.body()!!.items!!)
    view.hideLoadingIndicator()
    }
    res.code() == 500 -> {
        Log.e("youtube-failur
"internal server error")
        view.hideLoadingIndicator()
    }
    else -> {
        Log.e("youtube-failure",
"${res.code()} ${res.errorBody()}")
        view.hideLoadingIndicator()
    }
    }, { error ->
        Log.e("error-video",
error.localizedMessage.toString()
    })
    
```

2) Basis Path Testing

a. Flow Graph



b. Cyclomatic Complexity

- a) $V(G) = \text{ada } 5 \text{ region, } R1, R2, R3, R4, R5$
- b) $V(G) = 12 \text{ edges} - 9 \text{ node} + 2 = 5$
- c) $V(G) = 4 \text{ Predicate} + 1 = 5$

c. Independent Path

- a) Jalur 1 = 1 – 2 – 9
- b) Jalur 2 = 1 – 2 – 3 – 9
- c) Jalur 3 = 1 – 6 – 7 – 9
- d) Jalur 4 = 1 – 4 – 5 – 9
- e) Jalur 5 = 1 – 8 – 9

6.2 Pengujian Validasi

Pada pengujian sistem terdapat pengujian sistem dari sisi kebutuhan fungsional. Pada pengujian sistem dari sisi kebutuhan fungsional, penelitian ini menggunakan pengujian validasi. Pengujian validasi berfungsi untuk menguji perangkat lunak yang telah dibuat apakah sesuai dengan kebutuhan fungsional dan *Use Case Scenario* yang telah dirumuskan. Metode pengujian dilakukan

dengan teknik *blackbox testing* yang akan menguji input dan output yang diberikan sesuai dengan kebutuhan.

6.2.1 Pengujian Validasi

6.2.1.1 Kasus Uji Coba Mengambil Objek Menggunakan Kamera

Tabel 6.1 Tabel Kasus Uji Coba Mengambil Objek Menggunakan Kamera

Nama Kasus Uji	Mengambil Objek menggunakan kamera
Tujuan Pengujian	Untuk membuktikan bahwa pengguna dapat mengambil objek menggunakan kamera
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Memilih tombol kamera untuk mengambil objek 2. Mengambil objek menggunakan kamera 3. Menekan tombol <i>OK</i> jika objek yang diambil sudah sesuai
Hasil yang Diharapkan	Sistem dapat menganalisis objek yang diambil menggunakan kamera
Status	Valid

6.2.1.2 Kasus Uji Coba Mengambil Objek Menggunakan *gallery*

Tabel 6.2 Kasus Uji Coba Mengambil Objek Menggunakan *Gallery*

Nama Kasus Uji	Mengambil Objek menggunakan <i>gallery</i>
Tujuan Pengujian	Untuk membuktikan bahwa pengguna dapat mengambil objek menggunakan <i>gallery</i>
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Memilih tombol <i>gallery</i> untuk mengambil objek 2. Memilih objek yang tersedia pada daftar gambar 3. Menekan tombol <i>OK</i> jika objek yang diambil sudah sesuai
Hasil yang Diharapkan	Sistem dapat menganalisis objek yang dipilih menggunakan <i>gallery</i>
Status	Valid

6.2.1.3 Kasus Uji Coba Memutar Video

Tabel 6.3 Kasus Uji Coba Memutar Video

Nama Kasus Uji	Memutar video
Tujuan Pengujian	Untuk menampilkan video yang diputar oleh sistem
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Memilih salah satu video yang tersedia di daftar video 2. Melihat video yang diputar oleh sistem berdasarkan video yang dipilih oleh pengguna
Hasil yang diharapkan	Sistem dapat menampilkan video yang dipilih oleh pengguna
Status	Valid

6.2.1.4 Kasus Uji Coba Menampilkan Daftar Video

Tabel 6.4 Kasus Uji Coba Menampilkan Daftar Video

Nama Kasus Uji	Kasus uji coba menampilkan daftar video
Tujuan Pengujian	Untuk menampilkan daftar video berdasarkan objek yang dianalisis
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Pengguna mengambil objek untuk dianalisis 2. Sistem menganalisis objek dengan bantuan <i>IBM Visual Recognition</i>. 3. Melihat daftar video
Hasil yang diharapkan	Sistem dapat menampilkan daftar video berdasarkan objek yang dianalisis
Status	Valid

6.2.1.5 Kasus Uji Coba Menampilkan Harga Sampah

Tabel 6.5 Kasus Uji Coba Menampilkan Harga Sampah

Nama Kasus Uji	Kasus uji coba menampilkan harga sampah
Tujuan Pengujian	Untuk menampilkan harga sampah berdasarkan hasil analisis.
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Pengguna mengambil objek untuk dianalisis

	<ol style="list-style-type: none"> 2. Sistem menganalisis objek dengan bantuan <i>IBM Visual Recognition</i> 3. Menampilkan harga sampah
Hasil yang diharapkan	Sistem dapat menampilkan harga sampah berdasarkan objek yang dianalisis
Status	Valid

6.2.1.6 Kasus Uji Menampilkan Informasi Objek Belum mampu dianalisis

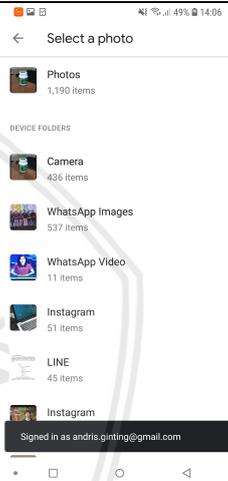
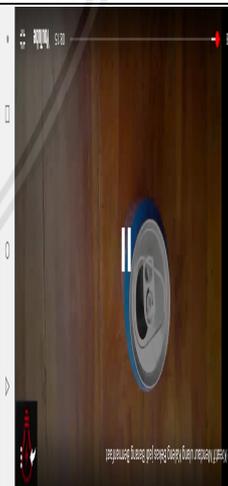
Tabel 6.6 Kasus Uji Menganalisis Objek Lain

Nama Kasus Uji	Kasus uji coba menampilkan objek belum mampu dianalisis
Tujuan Pengujian	Untuk menampilkan informasi objek belum mampu dianalisis
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Pengguna mengambil objek untuk dianalisis 2. Sistem menganalisis objek dengan bantuan <i>IBM Visual Recognition</i> 3. Menampilkan informasi objek belum mampu dianalisis
Hasil yang diharapkan	Sistem dapat menampilkan informasi objek belum mampu dianalisis
Status	Valid

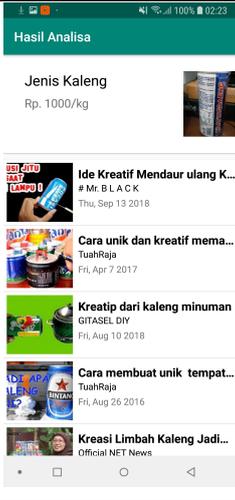
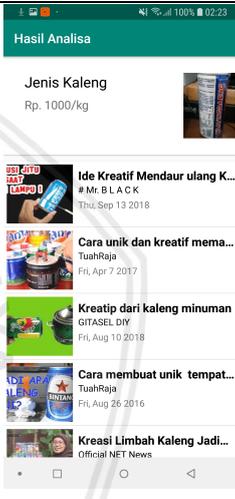
6.2.2 Hasil Pengujian Validasi

Tabel 6.7 Tabel Hasil Pengujian Validasi

No	Nama Kasus Uji	Hasil yang Diharapkan	Hasil yang didapatkan	Screenshot validasi	Status validasi
----	----------------	-----------------------	-----------------------	---------------------	-----------------

1	Mengambil objek menggunakan kamera	Pengguna dapat mengambil objek menggunakan kamera	Pengguna dapat mengambil objek menggunakan kamera		valid
2	Memilih objek menggunakan gallery	Pengguna dapat memilih objek dari gallery	Pengguna dapat memilih objek dari gallery		valid
3	Memutar video yang tersedia pada daftar video	Pengguna dapat melihat video yang ditampilkan sistem	Pengguna dapat melihat video yang ditampilkan sistem		valid



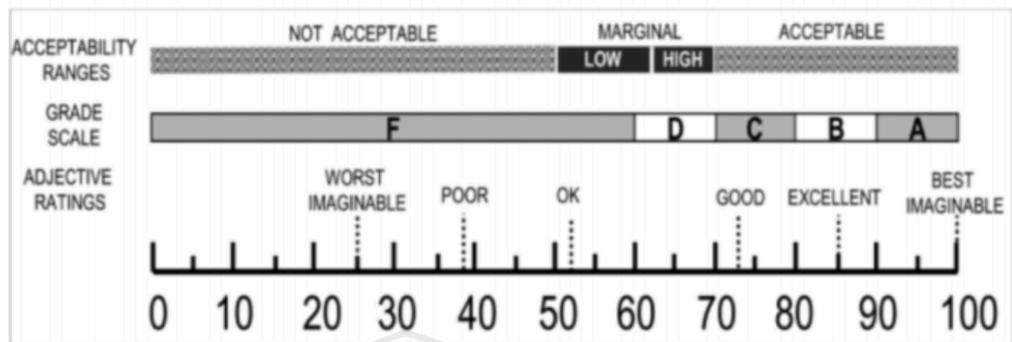
4	Berhasil melihat harga berdasarkan objek yang dianalisis	Pengguna dapat melihat harga berdasarkan objek yang dianalisis	Pengguna dapat melihat harga berdasarkan objek yang dianalisis		valid
5	Berhasil melihat daftar video berdasarkan objek yang dianalisis	Pengguna dapat melihat daftar video yang ditampilkan oleh sistem berdasarkan objek yang dianalisis	Pengguna dapat melihat daftar video yang ditampilkan oleh sistem berdasarkan objek yang dianalisis		valid
6	Berhasil menampilkan informasi objek belum mampu dianalisis	Pengguna dapat melihat informasi objek belum dapat dianalisis	Pengguna dapat melihat informasi objek belum dapat dianalisis		valid

6.3 Pengujian Usabilitas

Pengujian usabilitas dilakukan untuk melakukan pengujian sistem dengan cara menguji sistem secara langsung kepada sejumlah responden untuk mengetahui tingkat kemudahan penggunaan sebuah sistem. Dalam pengujian ini



usability diukur menggunakan pengujian *post task study* yang digunakan untuk mengukur kemudahan yang dirasakan oleh pengguna setelah menyelesaikan semua scenario yang diberikan (Sauro & Lewis, 2012).



Gambar 6.1 Rating dan Skala Konversi SUS

Sumber: measuringu.com

Tabel 6.8 Parameter Bobot Nilai SUS

Skor	Kualifikasi	Hasil
>80%	Sangat Baik(SB)	Berhasil
68-80%	Baik (B)	Berhasil
68%	Cukup (C)	Berhasil
51-68%	Buruk (D)	Tidak Berhasil
<51%	Kurang (K)	Tidak Berhasil

Sedangkan kinerja *usability* diukur dengan menggunakan pengujian *post study* dengan metode *System Usability Scale (SUS)* (Broke,2013). *SUS* merupakan pengujian yang yang efektif serta handal untuk digunakan pada produk dan aplikasi (Bangor, Kortum, & Miller, 2009). *SUS* terdiri dari 10 pertanyaan. Pertanyaan dengan nomor ganjil merupakan pertanyaan dengan nada positif (1, 3, 5, 7, 9) sementara pertanyaan dengan nomor genap merupakan pertanyaan dengan tipe negatif (2, 4, 6, 8, 10). Seperti yang ditunjukkan pada Tabel 6.7, setiap pertanyaan diberi nilai atau bobot 0-4. Untuk setiap pertanyaan bernomor ganjil , kurangi 1 dari skor (x-1) dan untuk pertanyaan bernomor genap, skornya kurangi nilainya dari 5 (5-x). Berdasarkan *acceptability ranges* yang dijelaskan pada Gambar 6.1, skala konversi dibagi menjadi tiga bagian yaitu *not acceptable* (skor 0 – 50.9), *marginal* (skor 51-70.9) dan *acceptable* (skor 71-100).

6.3.1 Prosedur Pengujian Usabilitas

Pengujian usabilitas pada aplikasi Dalang akan dicoba oleh mahasiswa Fakultas Ilmu Komputer untuk menganalisis objek menggunakan Dalang dan melihat daftar video cara daur ulang berdasarkan objek yang dianalisis oleh sistem. Responden akan diberikan arahan cara penggunaan aplikasi Dalang. Kemudian responden akan mulai menggunakan aplikasi Dalang sesuai *task* yang

ada pada Tabel 6.8 dan akan melakukan skenario penggunaan aplikasi Dalang. Setelah menjalankan aplikasi Dalang, maka responden diminta untuk mengisi angket yang tersedia pada pengujian *SUS*.

Tabel 6.9 Daftar Pertanyaan SUS

No.	Pertanyaan
1	Saya mampu menjalankan aplikasi “Dalang” dengan mudah
2	Mengambil objek dengan kamera cukup sulit
3	Informasi yang ditampilkan sangat saya butuhkan sebagai mahasiswa yang ingin mendaur ulang sampah
4	Saya ingin dan perlu membandingkan dengan orang yang paham tentang daur ulang sampah
5	Petunjuk yang diberikan mudah dipahami
6	Saya merasa huruf pada aplikasi “Dalang” sangat kecil
7	Saya suka warna yang ditampilkan pada aplikasi “Dalang”
8	Menurut saya tombolnya kecil, sehingga sulit untuk menekan tombolnya
9	Jika ingin mendaur ulang sampah, maka saya akan menggunakan aplikasi “Dalang”
10	Saya tidak akan pernah menggunakan aplikasi “Dalang” ini

6.3.2 Analisis Data dan Hasil Pengujian Usabilitas

Setelah melakukan tahap pengujian, maka selanjutnya dilakukan perhitungan data serta analisis hasil yang diperoleh dari responden yang menjawab *SUS* yang ada pada Tabel 6.10

Tabel 6.10 Hasil Skor Pertanyaan SUS

No.	Nama				
	Wahyu	Rizqi	Arifin	Anita	Akmal
1	4	4	3	2	4
2	1	1	0	1	1
3	5	4	4	3	3
4	0	0	1	2	1
5	4	4	3	4	3
6	1	2	0	1	2
7	4	3	3	3	3
8	0	2	0	3	2
9	3	4	3	3	4
10	0	0	3	0	1

Pada Tabel 6.11 dibawah ini terdapat nilai dari kuisisioner yang diajukan kepada 5 responden. Seluruh jawaban dari responden akan ditotal dari setiap jawaban yang sudah diberikan sebelumnya.

Tabel 6.11 Hasil Konversi Nilai SUS

No.	Nama				
	Wahyu	Rizqi	Arifin	Anita	Akmal
1	3	3	2	1	2
2	4	4	5	4	4
3	4	3	3	2	2
4	5	5	4	3	4
5	3	3	2	3	2
6	4	3	5	4	3
7	3	2	0	2	2
8	5	3	5	2	3
9	2	3	2	2	3
10	5	5	2	5	4
Jumlah	38	34	30	28	29
Dikali 2.5	95	85	75	70	72,5
Rata-rata					79,5

Dari Tabel 6.13 diatas, didapatkan perhitungan dan juga penyelesaian sebagai berikut ini, berdasarkan hasil konversi dari lima responden aplikasi "Dalang", skor likert yang didapatkan menggunakan pengujian *post study* adalah 79,5 yang berarti masuk dalam kategori *acceptable* sehingga dapat disimpulkan aplikasi "Dalang" sesuai dengan kebutuhan pengguna.

6.4 Analisis Hasil Pengujian

Setelah pengujian telah selesai dilakukan, hasil dari semua pengujian akan disimpulkan bahwa semua pengujian telah memenuhi kebutuhan yang telah dijelaskan pada analisis kebutuhan.

6.4.1 Pengujian Unit

Dari pengujian unit yang dilakukan pada method `getListVideo()`, `initializeYoutubeFrag()` dan `doInBackground(file)`, dengan menggunakan metode *whitebox testing* dihasilkan *basis path* yang memiliki semua kemungkinan yang dapat terjadi sudah diuji dan mendapatkan hasil valid pada setiap pengujiannya dan sudah sesuai dengan kebutuhan yang telah ditentukan pada analisis kebutuhan.

6.4.2 Pengujian Validasi

Dalam pengujian validasi semua fungsi telah diuji berdasarkan pengujian yang dilakukan dengan mencoba aplikasi yang dijalankan dan menjalankan aplikasi sesuai dengan prosedur pengujian yang sudah ditentukan dan hasil yang diperoleh adalah valid karena telah berhasil melakukan implementasi sesuai perancangan kebutuhan yang diinginkan. Dari hasil pengujian tersebut dapat disimpulkan bahwa aplikasi Dalang sudah memenuhi kebutuhan fungsional

6.4.3 Pengujian Usabilitas

Pengujian usabilitas dilakukan dengan menguji aplikasi kepada responden, yaitu mahasiswa Fakultas Ilmu Komputer yang mencoba aplikasi “Dalang” setelah itu responden diberikan kuisisioner untuk diisi untuk menentukan seberapa baikkah aplikasi sudah sesuai dengan kebutuhan yang telah ditentukan pada analisis kebutuhan. Hasil dari pengujian kali ini usability diukur menggunakan pengujian post study dengan metode *SUS (System usability scale)* adalah 79.5 yang berarti masuk kedalam kategori *acceptable* sehingga aplikasi sudah sesuai dengan kebutuhan pengguna. Hal ini menunjukkan bahwa aplikasi ini dikategorikan *acceptable* dalam *acceptable ranges*, dikategorikan *grade B* dalam *grade scales*, dan dikategorikan *good* dalam *adjective ratings*.

BAB 7 PENUTUP

7.1 Penarikan Kesimpulan

Berdasarkan hasil analisis kebutuhan, perancangan, implementasi dan pengujian yang dilakukan, maka dapat disimpulkan sebagai berikut:

1. Berdasarkan hasil analisis kebutuhan yang dilakukan, aplikasi Dalang mempunyai 9 kebutuhan fungsional dan 1 kebutuhan non fungsional. Penentuan kebutuhan fungsional diperoleh berdasarkan permasalahan yang ada yaitu, kurangnya pengetahuan responden mengenai daur ulang sampah dan tidak mengetahui cara mendaur ulang sampah dengan baik. Selain itu diperoleh pemodelan kebutuhan yang diperlukan untuk memudahkan pengembang dalam memahami sistem, seperti *use case diagram*, *class diagram* dan *sequence diagram*.
2. Perancangan yang dilakukan berdasarkan hasil analisis kebutuhan dan diperoleh perancangan data, perancangan komponen, dan perancangan antarmuka. Pada perancangan data, diperoleh rancangan data respons dari IBM Visual Recognition dan YouTube Data API berupa JSON skema. Pada perancangan komponen terdapat algoritme-algoritme yang digunakan dan di implementasikan kedalam sistem. Pada aplikasi Dalang juga terdapat perancangan arsitektur yang disusun berdasarkan *class-class* dan dijelaskan secara rinci di tiap klasnya. Perancangan antarmuka berisi *mock-up* dari sistem yang dibangun pada aplikasi Dalang.
3. Penerapan IBM Visual Recognition dan YouTube Data API digunakan untuk menganalisis objek yang diambil atau dipilih oleh pengguna menggunakan kamera atau *gallery* dan juga menampilkan video berdasarkan objek yang dianalisis. IBM Visual Recognition digunakan untuk menganalisis objek dan akan mengembalikan respons berupa JSON skema. YouTube Data API digunakan untuk menampilkan video dan memutar video yang dipilih oleh pengguna berdasarkan objek yang dianalisis.
4. Dengan hasil *usability test* yang dilakukan dalam hal kemudahan penggunaan aplikasi “Dalang” dapat disimpulkan bahwa kemudahan dalam menggunakan aplikasi ini sebesar 79.5 dan sudah memenuhi kebutuhan dalam hal kemudahan dan dikategorikan *acceptable*.

7.2 Saran

Saran yang diberikan untuk proses pengembangan aplikasi “Dalang” selanjutnya adalah sebagai berikut”

1. Dapat menonton video tutorial secara *offline*.
2. Perbanyak objek yang dapat dianalisis

DAFTAR PUSTAKA

- Whitten, Bentley. 2007. System analysis & Design Methods, McGraw-Hill/Irwin, New York.
- Sommerville, I., 2011. Software Engineering. 9th ed. United States of America: Pearson Education, Inc.
- Imaduddin, A., Permana, S., 2017. Menjadi Android Developer Expert. Bandung: PT.Presentologics.
- Imaduddin, A., Apriolio, Y., Arumsari, M., 2018. Kotlin Android Developer Expert. Bandung: PT.Presentologics.
- Google Developers. 2018. "Camera API". Diakses 2 Agustus 2018. <https://developer.android.com/guide/topics/media/camera>.
- IBM Watson, 2018. "IBM Watson Visual Recognition". Diakses 19 Agustus 2018. <https://www.ibm.com/watson/developercloud/visual-recognition/api/v3/curl.html?curl#introduction>
- Rennie, S. J., Marcheret, E., Mrouech, Y., Ross, J., Goel, V., (2017). Self-critical Sequence Training for Image Capturing. <http://doi.ieeecomputersociety.org/10.1109/CVPR.2017.131>
- Liputan 6 (2016). "Pengguna mobile lebih suka pake aplikasi daripada browser". Diakses 2 Agustus 2018. <http://tekno.liputan6.com/read/2477796/pengguna-mobile-lebih-suka-pakai-aplikasi-dibanding-browser>
- IDN Times, 2017 "Volume Sampah 2018 Diprediksi Mencapai 66,5 Juta Ton". Diakses 2 Agustus 2018. <https://www.idntimes.com/news/indonesia/indianamalia/volume-sampah-2018-diprediksi-mencapai-665-juta-ton-1/full>
- CNN Indonesia, 2018. "24 Persen Sampah Indonesia Masih Tak Terkelola". Diakses 2 Agustus 2018. <https://www.cnnindonesia.com/gaya-hidup/20180425101643-282-293362/riset-24-persen-sampah-di-indonesia-masih-tak-terkelola>.
- Wu, H., Merler, M., Uceda-sosa, R., Smith, J. R., (2016) Learning to Make Better Mistakes: Semantics: Aware visual Food Recognition. <https://dl.acm.org/citation.cfm?id=2967205>
- YouTube, 2018. "YouTube Data API". Diakses 19 Agustus 2018. <https://developers.google.com/youtube/v3/getting-started>.
- Republika, 2018. "Kota Malang Hasilkan 600 ton Sampah per hari". Diakses 9 September 2018.

<https://www.republika.co.id/berita/nasional/daerah/18/03/06/p55zrl359-kota-malang-hasilkan-600-ton-sampah-per-hari>

Ibm,2018. IBM and Google Combined. Diakses 16 September 2018.

https://www.ibm.com/developerworks/community/blogs/ibmandgoogle/entry/Training_Watson_and_TensorFlow_with_your_own_images?lang=en

Pressman, R.S. 2010, Software Engineering: A practitioner's Approach, McGraw-Hill, New York

Brooke John. 2011. Measuring Usability With The System Usability Scale (SUS). Diakses 10 Desember 2018. <http://www.measuringu.com/sus.php>.

