

SKRIPSI

**IMPLEMENTASI *WEB SERVICE* REST API SEBAGAI
MEDIA KOMUNIKASI DATA DAN PENYIMPANAN
DATASET GAMBAR**

Disusun dan diajukan oleh

A. AMALIA DWI AYU SARJANI. M

H071171507



**PROGRAM STUDI SISTEM INFORMASI
DEPARTEMEN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS HASANUDDIN
MAKASSAR
2021**

**IMPLEMENTASI *WEB SERVICE* REST API SEBAGAI
MEDIA KOMUNIKASI DATA DAN PENYIMPANAN
DATASET GAMBAR**

SKRIPSI

**Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana
Komputer pada Program Studi Sistem Informasi Departemen Matematika
Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin**

A. AMALIA DWI AYU SARJANI. M

H071171507

PROGRAM STUDI SISTEM INFORMASI DEPARTEMEN MATEMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS HASANUDDIN

MAKASSAR

2021

PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini:

Nama : A. Amalia Dwi Ayu Sarjani. M

NIM : H071171507

Program Studi : Sistem Informasi

Jenjang : S1

Menyatakan dengan ini bahwa karya tulisan saya berjudul

**Implementasi *Web Service* REST API Sebagai Media Komunikasi Data dan
Penyimpanan *Dataset* Gambar**

adalah karya tulisan saya sendiri dan bukan merupakan pengambilan alihan tulisan orang lain dan belum pernah dipublikasikan dalam bentuk apapun.

Apabila dikemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan skripsi ini hasil karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Makassar, 12 Oktober 2021

Yang menyatakan,



A. Amalia Dwi Ayu Sarjani. M

NIM. H071171507

Implementasi *Web Service* REST API Sebagai Media Komunikasi Data dan Penyimpanan *Dataset* Gambar

Disusun dan diajukan oleh

A. AMALIA DWI AYU SARJANI. M
H071171507

Telah dipertahankan di hadapan Panitia Ujian yang dibentuk dalam rangka Penyelesaian Studi Program Sarjana Program Studi Sistem Informasi Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin pada tanggal 12 Oktober 2021 dan dinyatakan telah memenuhi syarat kelulusan.

Menyetujui,

Pembimbing Utama



Dr. Hendra, S.Si., M.Kom.
NIP. 19760102 200212 1 001

Pembimbing Pertama



Supri Bin Hj. Amir, S.Si., M.Eng.
NIP. 19880504 201903 1 012

Ketua Program Studi,



Dr. Muhammad Hasbi, M.Sc.
NIP. 19630720 198903 1 003



HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :

Nama : A. Amalia Dwi Ayu Sarjani. M
NIM : H071171507
Program Studi : Sistem Informasi
Judul Skripsi : Implementasi *Web Service* REST API Sebagai Media Komunikasi Data dan Penyimpanan *Dataset* Gambar

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Komputer pada Program Studi Sistem Informasi Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin.

DEWAN PENGUJI

		Tanda tangan
Ketua	: Dr. Hendra, S.Si., M.Kom.	(.....)
Sekretaris	: Supri Bin Hj. Amir, S.Si., M.Eng.	(.....)
Anggota	: A. Muh. Amil Siddik, S.Si., M.Si.	(.....)
Anggota	: Edy Saputra, S.Si., M.Si.	(.....)

Ditetapkan di : Makassar
Tanggal : 12 Oktober 2021



KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Allah *Subbhanahu Wa Ta'ala* karena atas berkat dan rahmat-Nya penulis dapat menyelesaikan pendidikan jenjang Strata 1 pada Program Studi Sistem Informasi, Universitas Hasanuddin. Alhamdulillah, skripsi dengan judul “Implementasi *Web Service* REST API Sebagai Media Komunikasi Data dan Penyimpanan *Dataset* Gambar” yang disusun guna memenuhi salah satu syarat untuk mencapai gelar Sarjana Komputer (S.Kom.) pada Program Studi Sistem Informasi, Departemen Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin ini dapat dirampungkan.

Penulis menyadari bahwa telah mendapatkan banyak bantuan serta bimbingan dari berbagai pihak, dari masa perkuliahan hingga pada tahap akhir penyusunan skripsi ini. Oleh karena itu, pada kesempatan ini, dengan segala kerendahan hati penulis menyampaikan terima kasih yang sebesar-besarnya kepada:

1. Rektor Universitas Hasanuddin, Ibu **Prof. Dr. Dwia Aries Tina Pulubuhu, M.A.** beserta jajarannya; Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam, Bapak **Dr. Eng. Amiruddin, S.Si.** beserta jajarannya; Ketua Departemen Matematika, Bapak **Prof. Dr. Nurdin, S.Si., M.Si.** beserta jajarannya; Ketua Program Studi Sistem Informasi, Bapak **Dr. Muhammad Hasbi, M.Sc.** beserta jajarannya; serta Bapak/Ibu dosen Departemen Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Hasanuddin atas ilmu-ilmu dan bantuannya dalam berbagai bentuk.
2. Ayahanda **A. Mahmud, S.Sos.**, Ibunda **Hj. A. Hafsa S.Pd.**, kakak tersayang **A. Rini Khaerunnisa, S.K.M.**, adik tersayang **Raqeesha Shanin Maezura** dan keluarga lainnya atas segala do'a, motivasi dan bantuannya hingga penulis dapat sampai di titik yang luar biasa ini.
3. Bapak **Dr. Hendra, S.Si., M.Kom.** sebagai pembimbing utama dan Bapak **Supri Bin Hj Amir, S.Si., M.Eng.** sebagai pembimbing pertama sekaligus pendamping akademik atas kesediaan, kesabaran, dan waktu yang telah diluangkan untuk membimbing penulis selama proses penyusunan tugas akhir.

4. Bapak **A. Muh. Amil Siddik, S.Si., M.Si.** dan Bapak **Edy Saputra, S.Si., M.Si.** atas waktu dan kesediaannya sebagai penguji untuk tugas akhir. Terima kasih pula untuk ilmu-ilmu yang telah diberikan selama proses perkuliahan.
5. **Muhammad Fitrah** sebagai teman sekaligus guru atas segala bimbingan, waktu, ilmu, serta petuahnya yang sangat berguna dan **Siti Rabiatul Adawiyah** sebagai teman yang senantiasa mengingatkan, membantu, dan menemani sejak awal perkuliahan sampai penuntasan skripsi ini.
6. Seluruh teman-teman **Integral, Triana Rahayu, Tasya Febrina Utami, Hasanah Nur Aprilia Palupi, Rizky Ridhayani Basihu, Muhammad Rusdi, Dalila Zati Amani, A. Nila Gading** dengan segala dukungan, motivasi, waktu, hiburan dan penyemangat penulis sejak SMA.
7. Teman-teman **Popal** yaitu **Farhan, Denny, Ayu, Yustika, Reski, Irham, Dani, Fandy, Nunu** serta **Ardiansyah** teman yang selalu menghibur atau membantu tugas selama proses perkuliahan dan saudara-saudara sesama mahasiswa **Sistem Informasi** khususnya **Angkatan 2017** atas kebersamaan dan segala suka dukanya selama menjadi mahasiswa Universitas Hasanuddin.
8. Teman-teman **KM FMIPA, Himatika FMIPA Unhas** dan **DISKRIT 2017** yang sudah menemani penulis dalam berproses sama-sama dalam organisasi kemahasiswaan.
9. Teman-teman **KKN UNHAS 104 MAROS 6** khususnya **Muh. Aidil Dwi Zulhaq** dan **Qina Amalia Takhir** atas segala bantuannya selama proses kkn berlangsung dan segala dukungan dan motivasi kepada penulis selama ini.
10. Seluruh pihak yang tidak dapat disebutkan satu per satu atas segala bentuk kontribusi, partisipasi, serta motivasi yang diberikan kepada penulis selama ini. Semoga Allah SWT membalas segala kebaikan dari semua pihak yang telah membantu penulis. Akhir kata, penulis memohon maaf apabila terdapat kesalahan dalam penyusunan tugas akhir ini dan semoga membawa manfaat bagi pembaca.

Makassar, 12 Oktober 2021



A. Amalia Dwi Ayu Sarjani. M

**PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR
UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademik Universitas Hasanuddin, saya yang bertanda tangan di bawah ini:

Nama : A. Amalia Dwi Ayu Sarjani. M
NIM : H071171507
Program Studi : Sistem Informasi
Departemen : Matematika
Fakultas : Matematika dan Ilmu Pengetahuan Alam
Jenis karya : Skripsi

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Hasanuddin **Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty-Free Right*)** atas karya ilmiah saya yang berjudul:

**Implementasi *Web Service* REST API Sebagai Media Komunikasi Data dan
Penyimpanan *Dataset* Gambar**

beserta perangkat yang ada (jika diperlukan). Terkait dengan hal di atas, maka pihak universitas berhak menyimpan, mengalih-media/format-kan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di Makassar pada tanggal 12 Oktober 2021

Yang menyatakan



(A. Amalia Dwi Ayu Sarjani. M)

ABSTRAK

Mengambil data secara manual untuk sebuah penelitian biasanya membutuhkan waktu yang cukup lama khususnya data gambar. Maka dari itu tersedia beberapa *website* penyedia *dataset* gambar yang dapat di akses secara mudah. Tetapi terkadang *dataset* yang dibutuhkan belum tersedia atau belum sesuai dengan keinginan dan kebutuhan pengguna. Oleh karena itu, penelitian ini dilakukan untuk meneliti tentang merancang dan membangun suatu integrasi sistem layanan media pembuatan sekaligus tempat memperoleh *dataset* gambar menggunakan *web service* dalam mendistribusikan datanya ke pengguna dengan berbagai kebutuhan melalui lingkungan *mobile* berbasis android dan *website*. Implementasi teknologi *web service* dengan menggunakan metode REST API dan *framework* flask menghasilkan aplikasi yang mampu menjadi media pembuatan dan penyedia *dataset* gambar yang dapat mengintegrasikan aplikasi *website* dengan aplikasi *mobile* berbasis android.

Kata kunci: *Web Service*, *Dataset* Gambar, *Framework* Flask, Aplikasi *Mobile*, *Website*, REST API

ABSTRACT

Manual data retrieval for research usually takes time, especially for image data. Therefore, there are websites that are easily accessible as providers of image datasets. However, sometimes the required dataset is not available or not suitable with the wishes and needs of the user. Therefore, this research was conducted to study the design and development of an integrated media creation service system as well as a place of obtaining image datasets using web services for distributing data to users with various needs through an Android-based mobile environment and website. The implementation of web service technology using the REST API method with flask framework produces applications that can be a medium for creating and providing image datasets that integrate web based with Android-based mobile applications.

Keyword: Web Service, Image Dataset, Flask Framework, Aplikasi Mobile, Website, REST API

DAFTAR ISI

PERNYATAAN KEASLIAN	iii
HALAMAN PENGESAHAN	v
KATA PENGANTAR	vi
PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR	viii
ABSTRAK.....	ix
ABSTRACT.....	x
DAFTAR GAMBAR	xiv
DAFTAR TABEL.....	xvi
DAFTAR LAMPIRAN.....	xvii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan Penelitian.....	3
1.4 Batasan Masalah.....	3
1.5 Manfaat penelitian	3
BAB II TINJAUAN PUSTAKA	4
2.1 <i>Web Service</i>	4
2.1.1 <i>Arsitektur Web Service</i>	5
2.1.2 <i>REST (Representational State Transfer)</i>	6
2.1.3 <i>Format Pesan Pertukaran</i>	7
2.1.4 <i>API (Application Programming Interface)</i>	8
2.1.5 <i>Flask Framework</i>	8
2.1.6 <i>JSON Web Token</i>	8
2.2 <i>Python</i>	8

2.2.1	<i>Open Source Computer Vision (OpenCV)</i>	9
2.3	<i>Web</i>	9
2.3.1	PHP	9
2.3.2	<i>Hypertext Markup Language (HTML)</i>	10
2.3.3	<i>Bootstrap</i>	10
2.4	<i>Aplikasi Mobile</i>	10
2.4.1	<i>Retrofit</i>	11
2.5	Insomnia	11
2.6	MySQL.....	11
2.7	<i>Unified Modelling Language (UML)</i>	11
2.7.1	<i>Use Case Diagram</i>	12
2.7.2	<i>Activity Diagram</i>	13
2.8	<i>Preprocessing</i>	14
2.9	Pengujian <i>Black Box</i>	16
BAB III METODE PENELITIAN		17
3.1	Waktu dan Lokasi Penelitian.....	17
3.2	Tahapan Penelitian	17
3.3	Rancangan Sistem	18
3.3.1	<i>Use case Diagram</i>	18
3.3.2	Struktur <i>Web Service REST API</i>	20
3.4	Arsitektur Sistem.....	20
3.4.1	Rancangan Tampilan	21
3.5	Instrumen Penelitian.....	27
BAB IV HASIL DAN PEMBAHASAN		28
4.1	Rancangan Sistem	28
4.1.1	<i>Use Case Diagram</i>	28

4.1.2	<i>ER Diagram</i>	29
4.1.3	<i>CRUD Matrix</i>	29
4.1.4	<i>Sequence Diagram</i>	30
4.1.5	<i>Activity Diagram</i>	31
4.2	Implementasi JSON Web Token (JWT)	34
4.3	Implementasi REST <i>Web Service</i>	36
4.4	Aplikasi <i>Mobile</i>	37
4.4.1	Visualisasi Aplikasi <i>Mobile</i>	37
4.5	Proses <i>Preprocessing</i>	38
4.5.1	<i>Resize</i>	39
4.5.2	<i>Denoise</i>	39
4.5.3	<i>Image Augmentation</i>	40
4.6	Aplikasi <i>Website</i>	41
4.6.1	Visualisasi Aplikasi <i>Website</i>	41
4.7	Pengujian Sistem	44
4.7.1	Pengujian Terhadap Fungsi Login	44
4.7.2	Pengujian Terhadap Fungsi Registrasi Pada <i>Website</i>	46
4.7.3	Pengujian Terhadap Fungsi Tambah <i>Dataset</i> Pada <i>Website</i>	47
4.7.4	Pengujian Terhadap Fungsi Tambah Gambar.....	48
4.8	Hasil <i>Output Dataset</i>	49
BAB V KESIMPULAN DAN SARAN		51
5.1	Kesimpulan.....	51
5.2	Saran.....	51
DAFTAR PUSTAKA		52
LAMPIRAN.....		54

DAFTAR GAMBAR

Gambar 2.1. Antarmuka <i>Web Service</i> terhadap Sistem Lainnya (Newcomer, 2002).	5
Gambar 2.2. Arsitektur <i>Web Service</i> (Brittenham, 2002).	5
Gambar 2.3. Alur Proses Augmentasi.....	15
Gambar 3.1. Tahapan Penelitian	18
Gambar 3.2. <i>Use Case Diagram</i>	19
Gambar 3.3. Struktur <i>Web Service</i> REST API	20
Gambar 3.4. Arsitektur Sistem.....	20
Gambar. 3.5 Rancangan Tampilan Halaman <i>Login User</i> Aplikasi <i>Mobile</i> dan Halaman <i>Dataset</i>	21
Gambar 3.6. Rancangan Tampilan Halaman Sub-Kategori <i>Dataset</i>	22
Gambar 3.7. Rancangan Tampilan Halaman Menambah Gambar	23
Gambar 3.8. Rancangan Tampilan Halaman Awal dan <i>Login User</i>	24
Gambar 3.9. Rancangan Tampilan Halaman <i>Register</i>	24
Gambar 3.10. Rancangan Tampilan Halaman <i>User</i> yang tidak melakukan <i>login</i> (<i>guest</i>)	25
Gambar 3.11. Rancangan Tampilan Halaman Tampilan <i>Dataset</i> Admin	26
Gambar 3.12. Rancangan Tampilan Halaman Tampilan <i>Dataset</i> <i>User</i>	27
Gambar 4.1. <i>Use Case Diagram</i>	28
Gambar 4.2. <i>ER Diagram</i>	29
Gambar 4.3. <i>Sequence Diagram</i> <i>User</i> Mengunggah Gambar	31
Gambar 4.4. <i>Activity Diagram</i> Admin Menambah Kategori dan Sub-kategori <i>Dataset</i>	32
Gambar 4.5. <i>Activity Diagram</i> <i>User</i> Menambah Gambar	33
Gambar 4.6. <i>Activity Diagram</i> <i>Guest</i> Melihat Data.....	34
Gambar 4.7. Mekanisme Autentikasi dengan Token JWT	35
Gambar 4.8. Contoh Implementasi JWT	36
Gambar 4.9. Contoh Salah Satu Implementasi REST API Menggunakan <i>Method</i> POST	37

Gambar 4.10. Dari kiri: Halaman login dan Halaman <i>Dataset</i> Pada Aplikasi <i>Mobile</i>	38
Gambar 4.11. Dari kiri: Tampilan Halaman Gambar Asli dan Gambar Hasil <i>Preprocessing</i> Pada Aplikasi <i>Mobile</i>	38
Gambar 4.12. <i>Resize</i>	39
Gambar 4.13. <i>Denoise</i>	40
Gambar 4.14. <i>Image Augmentation</i>	41
Gambar 4.15. Tampilan Halaman <i>Login User</i> dan Admin Pada <i>Website</i>	41
Gambar 4.16. Tampilan Halaman <i>Register</i> Pada <i>Website</i>	42
Gambar 4.17. Tampilan Halaman <i>Guest</i> Pada <i>Website</i>	42
Gambar 4.18. Tampilan Halaman <i>User</i> Pada <i>Website</i>	43
Gambar 4.19. Tampilan Halaman Admin Pada <i>Website</i>	43
Gambar 4.20. Tampilan Isi Sub-Folder <i>Dataset</i> Pada <i>Website</i>	44
Gambar 4.21. Hasil <i>Output Dataset</i> Dalam Format URL.....	49
Gambar 4.22. Hasil <i>Output Dataset</i> Dalam Format ZIP.....	50

DAFTAR TABEL

Tabel 2.1 Simbol <i>Use Case Diagram</i>	13
Tabel 2.2. Simbol <i>Activity Diagram</i>	14
Tabel 4.1 Tabel <i>CRUD Matrix</i>	30
Tabel 4.2 Tabel Pengujian Fungsi Login Aplikasi <i>Mobile</i>	44
Tabel 4.3 Tabel Pengujian Fungsi Login <i>Website</i>	45
Tabel 4.4 Tabel Pengujian Fungsi Registrasi Pada <i>Website</i>	46
Tabel 4.5 Tabel Pengujian Fungsi Tambah <i>Dataset</i> Pada <i>Website</i>	47
Tabel 4.6 Tabel Pengujian Fungsi Tambah Gambar Terhadap Aplikasi <i>Mobile</i> ..	48
Tabel 4.7 Tabel Pengujian Fungsi Tambah Gambar Terhadap <i>Website</i>	48

DAFTAR LAMPIRAN

Lampiran 1. Sintaks API.....	54
Lampiran 2. Sintaks <i>Preprocessing</i>	70
Lampiran 3. Contoh sintaks Aplikasi <i>Website</i>	71
Lampiran 4. Contoh sintaks Aplikasi <i>Mobile</i>	84

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kumpulan data atau *dataset* merupakan objek yang merepresentasikan data dengan relasinya di dalam suatu *memory* dan data yang mencantumkan nilai setiap variabel, seperti tinggi dan berat objek untuk setiap anggota kumpulan data. Selain itu, *dataset* merupakan suatu kumpulan data atau satu data statistik di mana setiap atribut data dapat mewakili suatu variabel dan setiap *instance* memiliki deskripsi sendiri (Ian dkk, 2011). Pengumpulan data merupakan proses mengumpulkan dan mengukur informasi tentang variabel yang diminati, dalam cara sistematis yang memungkinkan seseorang untuk menjawab pertanyaan penelitian yang dinyatakan, menguji hipotesis, dan mengevaluasi hasil. Untuk mendapatkan *dataset* terutama *dataset* gambar itu sendiri, dapat dilakukan melalui dua cara yaitu mengambil data secara manual atau mengunjungi beberapa *website* seperti *UCI Machine Learning Repository* dan *kaggle*.

Mengambil data secara manual untuk sebuah penelitian biasanya membutuhkan waktu yang cukup lama. Maka dari itu tersedia beberapa *website* penyedia *dataset* yang dapat di akses secara mudah. Tetapi terkadang *dataset* yang dibutuhkan belum tersedia atau belum sesuai dengan keinginan dan kebutuhan pengguna. Oleh karena itu, maka perlu dilakukan suatu penelitian untuk meneliti tentang merancang dan membangun suatu integrasi sistem layanan media pembuatan sekaligus tempat memperoleh *dataset* gambar menggunakan *web service* dalam mendistribusikan datanya ke pengguna dengan berbagai kebutuhan melalui lingkungan *mobile* berbasis android dan *website*.

Implementasi *web service* pada penelitian ini terdapat proses *preprocessing* untuk menghasilkan *dataset* yang siap diproses lebih lanjut. Proses *preprocessing* merupakan cara untuk meningkatkan kualitas *dataset* dan menghapus *noise* data.

Selain itu tahap *preprocessing* adalah proses awal yang akan mentransformasikan data masukan menjadi format yang sesuai dan siap untuk diproses.

Saat ini aplikasi *mobile* atau sering juga disingkat dengan istilah *Mobile Apps* telah banyak digunakan. Aplikasi *mobile* merupakan aplikasi dari sebuah perangkat lunak yang dalam pengoperasiannya dapat berjalan di perangkat *mobile*. Aplikasi *mobile* dapat membantu pengguna untuk lebih mudah mengakses layanan internet menggunakan perangkat *mobile* mereka (Wang dkk, 2013).

Untuk mendukung *scalability* maka digunakan teknologi *web service* sebagai solusi pertukaran data atau media komunikasi data. Dengan memanfaatkan teknologi *web service* maka setiap *platform* baik aplikasi *smartphone*, *web*, maupun aplikasi *desktop* bisa memanggil *method/fungsi* yang diinginkan.

Pada penelitian ini akan menerapkan teknologi *web service* sebagai media komunikasi data untuk membuat dan memperoleh *dataset* gambar. Metode yang digunakan pada *web service* adalah REST dan aplikasi *mobile* berbasis *android*.

Selain itu, penelitian ini dilatar belakangi pada penelitian – penelitian terdahulu. Berdasarkan penelitian yang dilakukan (Tyas dkk, 2016) pada penelitian ini menyimpulkan penerapan teknologi *web service* membuat sistem dengan struktur data dan *platform* yang berbeda dapat saling berkomunikasi dalam mendistribusikan informasi.

Selanjutnya penelitian yang dilakukan (Putra dkk, 2019) yang menyimpulkan bahwa untuk *web service* yang berbasis Flask dengan menggunakan SOAP dan REST API dari grafik tersebut, REST memiliki performa yang lebih bagus dibandingkan dengan SOAP untuk pengetesan *request* dan respon untuk *web service*.

1.2 Rumusan Masalah

Berdasarkan uraian latar belakang masalah diatas, permasalahan yang dihadapi adalah:

1. Bagaimana implementasi *web service* dalam merancang dan membangun aplikasi yang mampu menjadi media pembuatan dan penyedia *dataset*

gambar yang mempunyai kemampuan untuk mengintegrasikan aplikasi berbasis *website* dengan aplikasi *mobile* berbasis android?

2. Bagaimana implementasi *web service* sebagai media *preprocessing* dalam pengolahan *dataset* gambar?

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah, maka tujuan dari penelitian ini adalah:

1. Membuat media pembuatan *dataset* gambar dengan implementasi *web service*.
2. Penyedia *dataset* gambar yang telah di *preprocessing*.

1.4 Batasan Masalah

Batasan masalah pada penelitian ini adalah:

1. Metode yang digunakan pada *web service* adalah REST.
2. *Dataset* yang digunakan adalah *dataset* yang dapat dikategorikan.
3. Jenis *dataset* sebagai inputan merupakan *dataset* berupa gambar.
4. Format *dataset* inputan berupa jpeg/jpg.
5. Batasan ukuran inputan gambar sebesar 4 mb.

1.5 Manfaat penelitian

Hasil penelitian ini diharapkan dapat bermanfaat untuk memudahkan pembuatan *dataset* gambar serta menyediakan tempat untuk memperoleh *dataset* gambar itu sendiri.

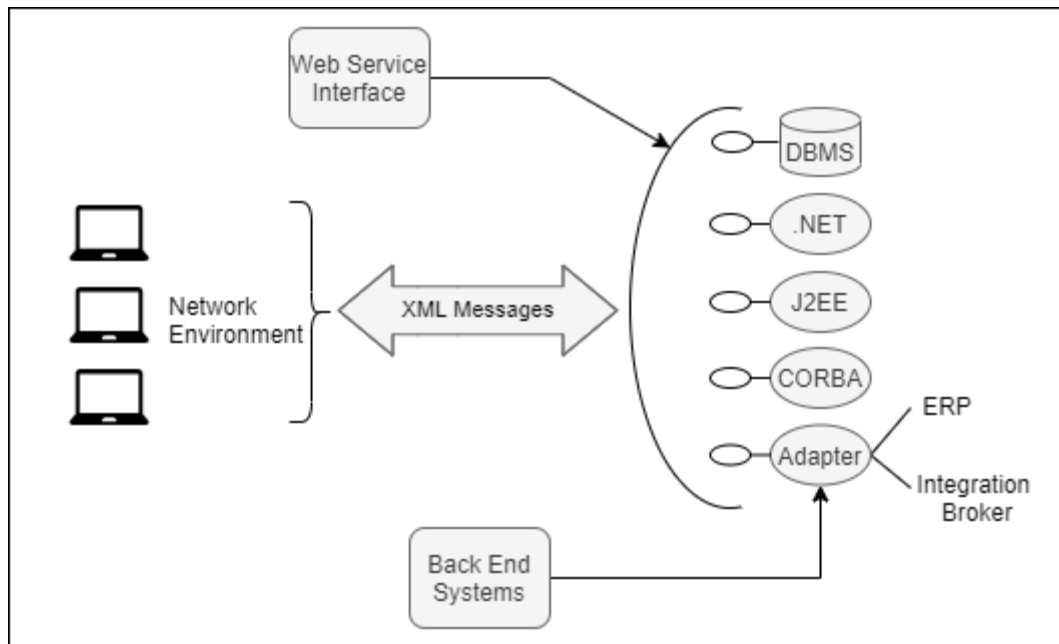
BAB II

TINJAUAN PUSTAKA

2.1 *Web Service*

Web service merupakan tahapan ketiga dari tahapan evolusi (*Application Service Provider*) dimana pada tahapan pertama ditekankan pada penyedia aplikasi *desktop* sedangkan pada tahapan kedua ditekankan pada penyediaan aplikasi berbasis *client-server*. Pada tahapan ketiga ini, komponen-komponen atau *building blocks software* disediakan sebagai *service* dan disebarluaskan lewat jaringan internet untuk diintegrasikan dengan aplikasi-aplikasi lainnya (Microsoft, 2000).

World Wide Web Consortium (W3C) mendefinisikan *Web Service* sebagai suatu sistem perangkat lunak yang dirancang untuk mendukung interoperabilitas dan interaksi antar mesin/sistem pada suatu jaringan (W3C, 2004). *Web service* digunakan sebagai suatu fasilitas yang disediakan oleh suatu website untuk menyediakan layanan (dalam bentuk informasi) kepada sistem lain, sehingga sistem lain dapat berinteraksi dengan sistem tersebut melalui layanan – layanan (*services*) yang disediakan oleh suatu sistem yang menyediakan *web service*. *Web service* menyimpan data informasi dalam format pesan universal (misal: XML dan JSON), sehingga data ini dapat diakses oleh sistem lain walaupun berbeda platform, sistem operasi, maupun bahasa *compiler*. *Web service* memiliki layanan terbuka untuk kepentingan integrasi data dan kolaborasi informasi yang bisa diakses melalui internet oleh berbagai pihak menggunakan teknologi yang dimiliki oleh masing - masing pengguna (Sutanta dkk, 2012). Ilustrasi mengenai posisi *web service* terhadap aplikasi lainnya dijelaskan pada Gambar 2.1.



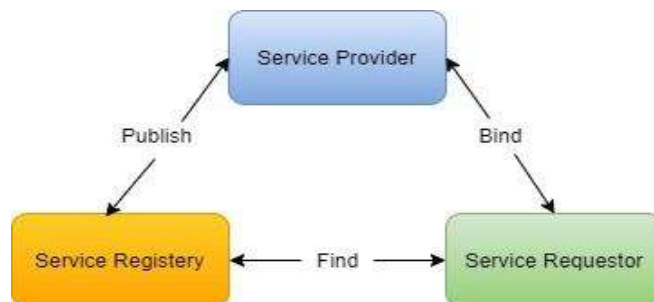
Gambar 2.1. Antarmuka *Web Service* terhadap Sistem Lainnya (Newcomer, 2002).

2.1.1 Arsitektur *Web Service*

Web service memiliki tiga entitas dalam arsitekturnya, yaitu :

1. ***Service Requester*** (Peminta Layanan)
2. ***Service Provider*** (Penyedia Layanan)
3. ***Service Registry*** (Daftar Layanan)

Arsitektur *Web Service* secara umum dapat dilihat pada gambar 2.2



Gambar 2.2. Arsitektur *Web Service* (Brittenham, 2002).

Pada Gambar 2.2 terdapat tiga komponen utama dari *web service* yaitu:

- a. *Service provider* merupakan penyedia *web service* yang berfungsi menyediakan kumpulan *services* yang dapat diakses oleh pengguna.
- b. *Service requestor* merupakan aplikasi yang bertindak sebagai pengguna yang melakukan permintaan layanan (berupa *web service*) ke *service provider*.
- c. *Service registry* merupakan tempat dimana *service provider* mempublikasikan layanannya. Pada arsitektur *Web service*, *Service registry* bersifat opsional.

2.1.2 REST (*Representational State Transfer*)

Salah satu metode yang dapat digunakan untuk pengembangan teknologi *web service* adalah REST. REST merupakan singkatan dari *Representational State Transfer*. Metode REST *web service* menerapkan konsep perpindahan antar *state*. *State* yang dimaksud disini dapat digambarkan apabila *browser* melakukan permintaan suatu *web*, maka *server* akan melakukan pengiriman *state* halaman *web* yang sekarang ke *browser*. Ide dasar dari metode REST adalah bagaimana menghubungkan jalur komunikasi antar mesin atau aplikasi melalui HTTP sederhana. Istilah REST atau RESTful pertama kali diperkenalkan oleh Roy Fielding pada disertasinya di tahun 2000. REST merupakan interaksi antara *client* dan *server* difasilitasi oleh sejumlah tipe operasional yang unik bagi setiap sumberdaya. Tipe operasional tersebut dapat berupa POST, GET, PUT dan DELETE dll. Istilah yang biasa digunakan untuk menyatakan prinsip *uniform interface* pada REST adalah CRUD (Create, Read, Update, Delete). Berikut ini ulasan detail mengenai *method-method* tersebut :

- a. **GET**, operasi *read-only* yang digunakan untuk meminta informasi spesifik pada server dalam bentuk *query*. Karakteristik dari operasi GET adalah *idempotent* dan *safe*.
- b. **PUT**, operasi untuk meminta kepada *server* agar membuat sebuah *resource* baru.
- c. **DELETE**, operasi untuk menghapus *resource* tertentu.
- d. **POST**, operasi untuk membuat *resource* baru atau memodifikasi *resource* yang telah ada.

- e. **HEAD**, operasi yang mirip dengan GET, namun *response message* yang dikembalikan hanyalah berupa *response code* dan *message header*.
- f. **OPTIONS**, operasi yang mengembalikan informasi mengenai berbagai HTTP *methods* yang didukung oleh suatu *server*, berguna untuk mengecek fungsionalitas sebuah *server* sebelum melakukan operasi sesungguhnya.

Metode REST didasari oleh empat prinsip :

1. **Identifikasi *resource* melalui *Uniform Resource Identifier (URI)***, REST *web service* menyediakan sejumlah *resource* yang mengidentifikasi tujuan dari interaksi dengan para *client*. *Resource* diidentifikasi oleh URI yang dapat mengakomodasi ruang pengalamatan untuk *resource* dan *service discovery*.
2. **Keseragaman di dalam *resource***, Setiap *resource* di dalam REST dimanipulasi oleh empat macam operasi atau *method*, yaitu GET, POST, PUT dan DELETE yang berfungsi untuk membaca, memperbarui, membuat, dan menghapus.
3. **Penggunaan *hyperlink* untuk interaksi *stateful***, Interaksi *stateful* adalah salah satu dimana adanya transfer *state* secara eksplisit. Salah satu cara yang digunakan untuk melakukan pertukaran *state* adalah menulis ulang URI.
4. **Message dengan *Self Description***, *resource* pada REST tidak terikat sehingga dapat melakukan akses terhadap berbagai macam format seperti PDF, HTML, XML, JPEG, JSON, dan lainnya.

2.1.3 Format Pesan Pertukaran

Saat ini terdapat 2 jenis format pesan yang dipertukarkan (*data interchange format*) yang dapat digunakan pada *web service*, yaitu XML dan JSON. XML adalah singkatan dari *Extensible Markup Language* yang merupakan turunan dari format *Standard Generalized Markup Language (SGML)* (Bosak, 1997). Sedangkan JSON singkatan dari *JavaScript Object Notation* yang merupakan objek asli bawaan JavaScript (JSON). JSON didesain sebagai format pesan pertukaran yang *human readable* serta mudah dibaca (*parsing*) oleh program

komputer. Format pesan pertukaran yang digunakan pada penelitian ini adalah format pesan *JavaScript Object Notation* (JSON).

2.1.4 API (*Application Programming Interface*)

Application Programming Interface atau biasa disingkat API adalah suatu penghubung yang memungkinkan suatu aplikasi untuk berinteraksi dengan aplikasi lainnya dan berbagi data (Ahmadi, 2017). API dirancang dengan konsep *Representational State Transfer* (REST). REST memungkinkan *client* dapat melakukan *request* melalui protokol HTTP dengan mudah menggunakan *Uniform Resource Identifier* (URI) (Kurniawan dkk, 2013).

2.1.5 Flask Framework

Dalam penelitian ini menggunakan *framework flask*. Dalam sejarah perkembangannya, *flask* pertama kali dirilis pada April 2010 dibuat oleh Armin Ronacher. *Flask* adalah sebuah *web framework* yang ditulis dengan bahasa *python* yang menyediakan *library* yang dapat digunakan untuk membangun sebuah *website*. *Flask* tersedia di *repository* publik *python* yang semua orang dapat mengunduh dan menggunakannya. *Flask* juga memiliki fungsi yang bisa dikembangkan sesuai kebutuhan (Samudera dkk, 2015).

2.1.6 JSON Web Token

JSON Web Token atau disingkat JWT adalah *access* token dengan format *JavaScript Object Nation* (Bradley, 2015). Token tersebut digunakan sebagai autentikasi token pada *client-server*. Token JWT didesain ringkas dan *url-safe*.

2.2 Python

Python adalah bahasa paling populer karena lebih mudah untuk membuat kode dan memahaminya (Malik, 2019). *Python* merupakan bahasa pemrograman dalam bentuk skrip yang dapat digunakan untuk berbagai pengembangan perangkat lunak dan secara lisensi dapat diakses secara terbuka di berbagai *platform* sistem operasi (Triasanti, 2001). Bahasa pemrograman ini dioptimalisasikan untuk *software quality*, *developer productivity*, *program portability*, dan *component integration* (Lutz, 2010).

Bahasa pemrograman *Python* memiliki beberapa fitur yang dapat digunakan oleh pengembangan perangkat lunak. Berikut adalah beberapa fitur yang ada pada bahasa pemrograman *Python* (Lutz, 2010) :

1. *Multi Paradigm Design*
2. *Open Source*
3. *Simplicity*
4. *Portability*
5. *Extendable*
6. *Extendable*
7. *Scalability*

2.2.1 *Open Source Computer Vision (OpenCV)*

OpenCV (*Open Computer Vision*) adalah *library open source* yang memiliki tujuan yang dikhususkan untuk melakukan pengolahan citra serta dapat ditulis dengan berbagai bahasa pemrograman termasuk *Python*. Penggunaan OpenCV yang berguna sebagai *library* atau pustaka untuk memproses sebuah citra. Kode *library* ini memiliki lebih dari 2500 algoritma yang dapat dioptimalkan.

Modul – modul yang terdapat pada OpenCV antara lain: *cv* – fungsi utama OpenCV, *cvaux* – fungsi penolong OpenCV, *cxcore* – pendukung struktur data dan aljabar linear, *highgui* – fungsi GUI (Sogen dkk, 2015).

2.3 *Web*

Web merupakan ruang informasi yang berisikan sumber-sumber daya yang berguna mengidentifikasi *Uniform Resource Identifier* (URL). Mengakses *Web* dibutuhkan *browser web* yang akan memecah bagian nama *server* dari URLnya ke dalam alamat protokol internet menggunakan basis data terdistribusi yang dikenal dengan istilah *Domain Name System* (DNS) (Tyas dkk, 2016)

2.3.1 *PHP*

PHP merupakan bahasa pemrograman *scripting* untuk membuat halaman web yang dinamis. Cara kerja php adalah dengan menyelipkan diantara kode html.

Website yang dibuat menggunakan php memerlukan *software* bernama *webserver* tempat pemrosesan kode php dilakukan (Zaky, 2008).

2.3.2 Hypertext Markup Language (HTML)

HTML adalah bahasa yang disisipkan (*embedded language*) pada dokumen dengan memberi tanda tertentu yang disebut *tag* (Mulyanto, 2008). *Tag* merupakan aturan penulisan kode yang ditulis dengan diawali tanda lebih kecil dan di akhiri dengan tanda lebih besar (<*tag*>). *Browser* akan menentukan tampilan teks atau dokumen berdasarkan *tag* yang digunakan. *Tag* tersebut membuat hirarki yang menyusun konten hingga menjadi bagian, paragraf, *heading*, dan *block* konten lainnya.

2.3.3 Bootstrap

Bootstrap merupakan *front-end framework* yang mengedapankan tampilan untuk *mobile device* (*Handphone, smartphone* dll.) guna mempercepat dan mempermudah pengembangan *website*. Bootstrap merupakan *framework* untuk membangun desain *web* secara responsif. Artinya, tampilan *web* yang dibuat oleh bootstrap akan menyesuaikan ukuran layar dari *browser* yang kita gunakan baik di desktop, tablet ataupun *mobile device* (Suprayogi dkk, 2019).

2.4 Aplikasi Mobile

Aplikasi *mobile* adalah perangkat lunak yang berjalan pada perangkat *mobile* seperti *smartphone* atau *tablet PC* (Irsan, 2015). Aplikasi *mobile* dapat membantu manusia melakukan berbagai macam aktivitas mulai dari hiburan, belajar, mengerjakan pekerjaan kantor dan lain sebagainya. Untuk mendapatkan aplikasi *mobile* itu sendiri dapat diunduh melalui situs tertentu sesuai dengan sistem operasi yang dimiliki. Google Play dan iTunes merupakan beberapa contoh dari situs yang menyediakan beragam aplikasi bagi pengguna Android dan iOS untuk mengunduh aplikasi yang diinginkan (Irsan, 2015).

Pada penelitian ini pengembangan aplikasi menggunakan bahasa pemrograman kotlin. Kotlin adalah bahasa pemrograman berbasis Java Virtual Machine (JVM) yang dikembangkan oleh JetBrains (Leiva, 2017).

2.4.1 Retrofit

Retrofit merupakan sebuah *library* yang digunakan untuk mempermudah proses pertukaran data antara aplikasi android dengan *server* melalui REST API. Retrofit mengambil data yang ada didalam basis data dan mengkonversikannya ke dalam GSON (Sutrisno, 2017).

2.5 Insomnia

Insomnia adalah aplikasi *cross-platform* untuk mengatur, menjalankan, dan melakukan debugging HTTP *request*. Keunggulan dari insomnia adalah :

1. Mudah untuk bertukar atau berganti *platform* dengan variabel yang terpisah.
2. Mampu untuk mereferensikan nilai respon dari permintaan lain.
3. Dapat memproses kode program ke hampir semua bahasa yang diinginkan.
4. Instalasi tak terbatas dengan paket gratis dan tersedia untuk MAC, Windows, Linux, dan ekstensi Google Chrome.
5. Merupakan aplikasi *Open source*. (Damrongchai, 2017).

2.6 MySQL

MySQL adalah salah satu jenis *database server* yang sangat terkenal dan banyak digunakan untuk membangun aplikasi *web* yang menggunakan *database* sebagai sumber dan pengolahan datanya (Rudianto, 2011). MySQL merupakan *software database open source*, MySQL menjadi pilihan utama bagi banyak pengembang *software* dikarenakan kelebihan MySQL didukung program-program umum seperti C, C++, Java, PHP, dan *Python*.

Proses menggunakan MySQL pada dasarnya adalah mengelola data dan informasi agar data dan informasi tersimpan dengan tertata dan rapi, proses-proses yang sering terjadi biasanya adalah membuat *database*, membuat sebuah *table*, memodifikasi struktur sebuah *table*, mengisikan data dalam sebuah *table*, menghapus data dalam sebuah *table*, memodifikasi (merubah atau mengedit) data dalam sebuah *table* dan mencari data dalam sebuah *table* (Saputra, 2012).

2.7 Unified Modelling Language (UML)

Unified Modelling Language (UML) adalah “bahasa” yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem perangkat lunak (Dharwiyanti dkk, 2003). UML menawarkan sebuah

standar untuk merancang model sebuah sistem. UML tidak hanya digunakan dalam pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan. Adapun tujuan dari penggunaan UML yaitu sebagai berikut :

- Dapat menyatukan praktek-praktek terbaik yang ada dalam pemodelan.
- Dapat memberikan bahasa pemodelan visual kepada pengguna dari berbagai macam pemograman maupun proses rekayasa.
- Dapat menciptakan suatu bahasa pemodelan yang nantinya dapat dipergunakan oleh manusia maupun oleh mesin.
- Dapat memodelkan sistem yang berkonsep berorientasi objek, jadi tidak hanya digunakan untuk memodelkan perangkat lunak saja.
- Berguna sebagai *blue print*, karena sangat lengkap dan detail dalam perancangannya yang nantinya akan diketahui informasi yang detail mengenai koding suatu program.
- Dapat memberikan model yang siap untuk digunakan, merupakan bahasa pemodelan visual yang ekspresif untuk mengembangkan sistem dan untuk saling menukar model secara mudah.

2.7.1 *Use Case Diagram*





Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Sebuah *use case diagram* diharapkan dapat mempresentasikan sebuah interaksi antara aktor dengan sistem. *Use case diagram* juga mempunyai beberapa simbol yang dapat dilihat di tabel 2.1.

Adapun karakteristik dari *use case diagram* yaitu sebagai berikut :

- *Use case* merupakan interaksi atau dialog antara sistem dan aktor, termasuk pertukaran pesan dan tindakan yang dilakukan oleh sistem.
- *Use case* diprakarsai oleh aktor dan mungkin melibatkan peran aktor lain. *Use case* harus menyediakan nilai minimal kepada satu aktor.
- *Use case* bisa memiliki perluasan yang mendefinisikan tindakan khusus dalam interaksi atau *use case* lain mungkin disisipkan.
- *Use case class* memiliki objek *use case* yang disebut skenario. Skenario

menyatakan urutan pesan dan tindakan tunggal.



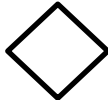



Tabel 2.1 Simbol *Use Case Diagram*

SIMBOL	NAMA SIMBOL	FUNGSI
	Aktor	Pihak yang mengakses <i>use case</i>
	<i>Use Case</i>	Mewakili apa yang sistem bisa lakukan
	<i>Association</i>	Merelasikan aktor dengan <i>use case</i>
	<i>System Boundary</i>	Menggambarkan batasan sistem terhadap lingkungannya

2.7.2 *Activity Diagram*

Activity diagram menggambarkan berbagai alur aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alur berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* juga mempunyai beberapa simbol yang dapat dilihat di tabel 2.2.

Tabel 2.2. Simbol *Activity Diagram*

SIMBOL	NAMA SIMBOL	FUNGSI
	Status Awal	Status awal sistem
	Aktivitas	Aktivitas yang dilakukan oleh sistem
	<i>Decision</i>	Percabangan jika ada pilihan aktivitas yang lebih dari satu
	<i>Control Flow</i>	Memperlihatkan urutan eksekusi
	<i>Join</i>	Penggabungan lebih dari satu aktivitas digabungkan menjadi satu
	Status Akhir	Status akhir yang dilakukan oleh sistem

2.8 *Preprocessing*

Preprocessing merupakan suatu proses untuk menghilangkan bagian – bagian yang tidak diperlukan pada gambar input. Proses tahap *preprocessing* yang dilakukan pada sistem adalah sebagai berikut :

a. *Resize*

Pertama, proses *resize* dimana besar ukuran piksel citra akan diubah dan dibuat sama antara satu citra dengan citra lainnya. Besar ukuran piksel citra tidak ditentukan karena nantinya pengguna akan melakukan *resizing* ulang

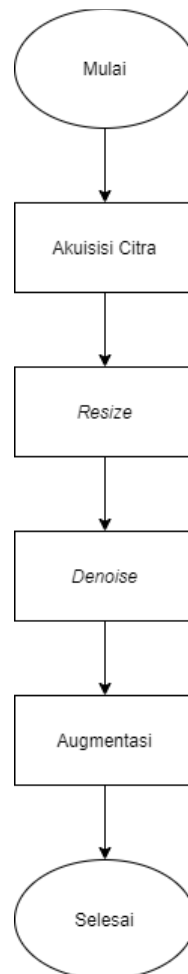
sesuai kebutuhan algoritma yang digunakan.

b. *Denoise*

Denoising merupakan teknik penghilangan *noise* yang terdapat pada citra dan mempertahankan informasi yang penting. *Noise* adalah suatu piksel yang mengganggu kualitas citra. *Noise* merupakan penyebab utama penurunan kualitas citra.

c. *Image Augmentation*

Data *augmentation* adalah sebuah teknik manipulasi sebuah data tanpa kehilangan inti atau esensi dari data tersebut. Untuk data berupa gambar beberapa teknik yang dilakukan adalah *rotate*, *flip*, *crop*, dll. Tujuan dilakukannya data augmentasi adalah untuk memperbanyak variasi data. Alur dari proses augmentasi dapat dilihat pada Gambar 2.3.



Gambar 2.3. Alur Proses Augmentasi

2.9 Pengujian *Black Box*

Pengujian pada sistem menggunakan metode *Black Box*, tujuannya untuk mengetahui bahwa bagian-bagian dalam sistem aplikasi telah benar menampilkan pesan-pesan kesalahan jika terjadi kesalahan dalam pemasukan data (Ferdinandus dkk, 2012). *Black Box Testing* sendiri merupakan pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak (Krismadi dkk, 2019). Pengujian black box ini menitik beratkan pada fungsi sistem (Syaban dkk, 2015).