

**PENGEMBANGAN APLIKASI MOBILE UNTUK
PENILAIAN DESAIN ARSITEKTUR MENGGUNAKAN
METODE EXTREME PROGRAMMING
(STUDI KASUS ARSITEKTUR ITERA)**

TUGAS AKHIR

Diajukan sebagai syarat menyelesaikan jenjang strata Satu (S-1)
di Program Studi Teknik Informatika, Jurusan Teknologi,
Produksi dan Industri, Institut Teknologi Sumatera

Oleh:

MUHAMMAD FIRDAUS SATI

119140150



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI, PRODUKSI DAN INDUSTRI
INSTITUT TEKNOLOGI SUMATERA
LAMPUNG SELATAN**

2022

LEMBAR PENGESAHAN

Tugas Akhir dengan judul “**Tulis Judul Disini**” adalah benar dibuat oleh saya sendiri dan belum pernah dibuat dan diserahkan sebelumnya, baik sebagian ataupun seluruhnya, baik oleh saya ataupun orang lain, baik di Institut Teknologi Sumatera maupun di institusi pendidikan lainnya.

Lampung Selatan, **DD-MM-YYYY**
Penulis,

PHOTO
BERWARNA

Nama Mahasiswa
NIM. XXXXXX

Diperiksa dan disetujui oleh,

Pembimbing

Tanda Tangan

1. Nama Pembimbing 1 + Gelar
NIP. XXXXXX

.....

2. Nama Pembimbing 2 + Gelar
NIP. XXXXXX

.....

Penguji

Tanda Tangan

1. Nama Penguji 1 + Gelar
NIP. XXXXXXXXXXXXX

.....

2. Nama Penguji 2+ Gelar
NIP. XXXXXXXXXXXXX

.....

Disahkan oleh,
Koordinator Program Studi Teknik Informatika
Jurusan Teknologi, Produksi dan Industri
Institut Teknologi Sumatera

Nama Kaprodi + Gelar
NIP. XXXXXXXXXXXXXXXX

HALAMAN PERNYATAAN ORISINALITAS

Tugas Akhir dengan judul “**TULIS JUDUL DISINI**” adalah karya saya sendiri, dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan benar.

Nama :

NIM :

Tanda Tangan :

Tanggal :

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai civitas akademik Institut Teknologi Sumatera, saya yang bertanda tangan di bawah ini:

Nama :
NIM :
Program Studi : Teknik Informatika
Jurusan : Jurusan Teknologi, Produksi dan Industri
Jenis Karya : Tugas Akhir

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

TULIS JUDUL DISINI

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini, Institut Teknologi Sumatera berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan skripsi saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di Lampung Selatan
Pada tanggal **DD Bulan YYYY**

Yang menyatakan,

Nama Mahasiswa

KATA PENGANTAR

Puji syukur kehadiran Allah SWT atas limpahan rahmat, karunia, serta petunjuk-Nya sehingga penyusunan tugas akhir ini telah terselesaikan dengan baik. Dalam penyusunan tugas akhir ini penulis telah banyak mendapatkan arahan, bantuan, serta dukungan dari berbagai pihak. Oleh karena itu pada kesempatan ini penulis mengucapkan terima kasih kepada:

1. <isi dengan nama Rektor ITERA>
2. <isi dengan nama Kajur JTPI>
3. <isi dengan nama Kaprodi IF>
4. <isi dengan nama Sesprodi IF>
5. <isi dengan nama Koordinator TA>
6. <isi dengan nama Dosen Pembimbing>
7. Kedua Orang Tua, kakak dan adik yang selalu memberikan arahan selama belajar dan menyelesaikan tugas akhir ini.
8. <isi dengan nama orang lainnya>

Akhir kata penulis berharap semoga tugas akhir ini dapat memberikan manfaat bagi kita semua, amin. [Contoh]

RINGKASAN

Judul TA

Nama Mahasiswa

Halaman Ringkasan berisi uraian singkat tentang latar belakang masalah, rumusan masalah, tujuan, metodologi penelitian, hasil dan analisis data, serta kesimpulan dan saran. Isi ringkasan tidak lebih dari 1500 kata (sekitar 3 halaman).

ABSTRAK

Judul TA

Nama Mahasiswa

Halaman ABSTRAK berisi uraian tentang latar belakang, tujuan, metodologi penelitian, hasil / kesimpulan. Ditulis dalam BAHASA INDONESIA tidak lebih dari 250 kata, dengan jarak antar baris satu spasi.

Pada akhir abstrak ditulis kata “Kata Kunci” yang dicetak tebal, diikuti tanda titik dua dan kata kunci yang tidak lebih dari 5 kata. Kata kunci terdiri dari kata-kata yang khusus menunjukkan dan berkaitan dengan bahan yang diteliti, metode/instrumen yang digunakan, topik penelitian. Kata kunci diketik pada jarak dua spasi dari baris akhir isi abstrak.

Kata Kunci : Penambangan Data, Kecerdasan Buatan, Lampung Selatan

ABSTRACT

Judul TA (Bahasa Inggris)

Nama Mahasiswa

Halaman ABSTRACT berisi uraian tentang latar belakang, tujuan, metodologi penelitian, hasil / kesimpulan. Ditulis dalam BAHASA INGGRIS tidak lebih dari 250 kata, dengan jarak antar baris satu spasi. Secara khusus, kata dan kalimat pada halaman ini tidak perlu ditulis dengan huruf miring meskipun menggunakan Bahasa Inggris, kecuali terdapat huruf asing lain yang ditulis dengan huruf miring (misalnya huruf Latin atau Greek, dll).

Pada akhir abstract ditulis kata “Keywords” yang dicetak tebal, diikuti tanda titik dua dan kata kunci yang tidak lebih dari 5 kata. Keywords terdiri dari kata-kata yang khusus menunjukkan dan berkaitan dengan bahan yang diteliti, metode/instrumen yang digunakan, topik penelitian. Keywords diketik pada jarak dua spasi dari baris akhir isi abstrak.

Keywords : Data Mining, Artificial Intelligence, Lampung Selatan

DAFTAR ISI

| | |
|--|------|
| LEMBAR PENGESAHAN | ii |
| HALAMAN PERNYATAAN ORISINALITAS | iii |
| HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS | iv |
| KATA PENGANTAR | v |
| RINGKASAN | vi |
| ABSTRAK | vii |
| ABSTRACT | viii |
| DAFTAR ISI | ix |
| DAFTAR TABEL | xii |
| DAFTAR GAMBAR | xiii |
| DAFTAR LAMPIRAN | xiv |
| BAB I PENDAHULUAN | 1 |
| 1.1 Latar Belakang Masalah | 1 |
| 1.2 Rumusan Masalah | 5 |
| 1.3 Tujuan Penelitian | 5 |
| 1.4 Batasan Masalah | 5 |
| 1.5 Manfaat Penelitian | 6 |
| 1.6 Sistematika Penulisan | 6 |
| 1.6.1 Bab I Pendahuluan | 6 |
| 1.6.2 Bab II Tinjauan Pustaka | 6 |
| 1.6.3 Bab III Metode Penelitian | 6 |
| 1.6.4 Bab IV HASIL DAN PEMBAHASAN | 6 |
| 1.6.5 Bab V KESIMPULAN DAN SARAN | 7 |
| BAB II TINJAUAN PUSTAKA | 8 |
| 2.1 Tinjauan Pustaka | 8 |
| 2.2 Dasar Teori | 15 |
| 2.2.1 Extreme Programming (XP) | 15 |
| 2.2.2 Use Case Diagram | 17 |
| 2.2.3 Activity Diagram | 18 |
| 2.2.4 Unit Testing | 19 |

| | | |
|---------------------------------|--|----|
| 2.2.5 | Acceptance Testing | 20 |
| 2.2.6 | TypeScript | 21 |
| 2.2.7 | Express | 22 |
| 2.2.8 | Android..... | 22 |
| 2.2.9 | React Native | 23 |
| 2.2.10 | Machine Learning..... | 24 |
| BAB III METODE PENELITIAN | | 25 |
| 3.1 | Alur Penelitian | 25 |
| 3.2 | Penjabaran Langkah Penelitian..... | 26 |
| 3.2.1 | Identifikasi Masalah | 26 |
| 3.2.2 | Studi Literatur..... | 26 |
| 3.2.3 | Penerapan eXtreme Programming (XP) | 26 |
| 3.2.4 | Pengujian | 27 |
| 3.3 | Alat dan Bahan Tugas Akhir | 27 |
| 3.3.1 | Alat | 27 |
| 3.3.2 | Bahan..... | 28 |
| 3.4 | Metode Pengembangan..... | 29 |
| 3.4.1 | Perencanaan (Planning) | 30 |
| 3.4.1.1 | Wawancara | 30 |
| 3.4.1.2 | User Story..... | 31 |
| 3.4.1.3 | Kebutuhan Fungsional..... | 32 |
| 3.4.1.4 | Kebutuhan Non-Fungsional..... | 32 |
| 3.4.2 | Perancangan (Design)..... | 33 |
| 3.4.2.1 | Architecture Diagram | 34 |
| 3.4.2.2 | Use Case Diagram | 35 |
| 3.4.2.3 | Activity Diagram | 37 |
| 3.4.2.4 | Rancangan Alur Proses Penggunaan | 38 |
| 3.4.2.5 | Rancangan Antarmuka (Interface) | 40 |
| 3.4.3 | Pengembangan (Coding) | 42 |
| 3.4.4 | Validasi..... | 43 |
| 3.4.5 | Perbaikan (Adjustment)..... | 43 |
| 3.4.6 | Peningkatan (Incremental)..... | 43 |
| 3.4.7 | Pengerahan (Deployment) | 44 |
| 3.5 | Rancangan Pengujian..... | 44 |

| | | |
|-----------------------------------|---------------------------------|----|
| 3.5.1 | Unit Testing | 44 |
| 3.5.1.1 | Server..... | 44 |
| 3.5.1.2 | Mobile..... | 45 |
| 3.5.2 | Acceptance Testing | 46 |
| 3.5.3 | Pengujian Efektivitas | 47 |
| 3.5.3.1 | Waktu Penilaian..... | 47 |
| 3.5.3.2 | Hasil Penilaian..... | 48 |
| BAB IV HASIL DAN PEMBAHASAN | | 50 |
| 4.1 | Hasil Penelitian | 50 |
| 4.2 | Hasil Pengujian | 50 |
| 4.3 | Analisis Hasil Penelitian | 50 |
| 4.3.1 | Analisis Hasil Data 1 | 50 |
| 4.3.2 | Analisis Hasil Data 2 | 50 |
| 4.4 | Pembahasan | 51 |
| 4.5 | Pengujian | 51 |
| BAB V KESIMPULAN DAN SARAN | | 52 |
| 5.1 | Kesimpulan | 52 |
| 5.2 | Saran | 52 |
| DAFTAR PUSTAKA | | 53 |
| LAMPIRAN..... | | 58 |

DAFTAR TABEL

| | |
|---|----|
| Tabel 2. 1 Tabel Perbandingan Referensi | 9 |
| Tabel 2. 2 Komponen Use Case Diagram | 18 |
| Tabel 2. 3 Komponen Activity Diagram | 19 |
| Tabel 3. 1 Hasil Wawancara | 30 |
| Tabel 3. 2 User Story | 31 |
| Tabel 3. 3 Kebutuhan Fungsional | 32 |
| Tabel 3. 4 Kebutuhan Non-Fungsional | 32 |
| Tabel 3. 5 Server User Testing | 44 |
| Tabel 3. 6 Mobile User Testing | 45 |
| Tabel 3. 7 Acceptance Testing | 47 |
| Tabel 3. 8 Perbandingan Waktu Penilaian | 47 |
| Tabel 3. 9 Perbandingan Hasil Penilaian | 49 |

DAFTAR GAMBAR

| | |
|---|----|
| Gambar 2. 1 Extreme Programming Life Cycle [13] | 16 |
| Gambar 3. 1 Diagram Alir Penelitian | 25 |
| Gambar 3. 2 Alur Pengembangan Extreme Programming (XP) | 29 |
| Gambar 3. 3 Architecture Diagram..... | 34 |
| Gambar 3. 4 Use Case Diagram..... | 36 |
| Gambar 3. 5 Activity Diagram | 37 |
| Gambar 3. 6 Alur Proses Penggunaan | 39 |
| Gambar 3. 7 Layar Utama Aplikasi | 40 |
| Gambar 3. 8 Layar Pemotongan (<i>Cropping</i>) Desain | 41 |
| Gambar 3. 9 Daftar Desain Fasad Bangunan yang Dinilai | 42 |
| Gambar 3. 10 Hasil Penilaian Desain Fasad Bangunan..... | 42 |

DAFTAR LAMPIRAN

| | |
|-------------------------------|----|
| LAMPIRAN 1 Isi Lampiran | 58 |
|-------------------------------|----|

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Pada tahun 2020, Indonesia mengalami pandemic Covid-19 yang berdampak pada beberapa sektor seperti politik, ekonomi, sosial, budaya, pertahanan, dan keamanan [1]. Sebagai dampak dari Covid-19, metode pembelajaran di Institut Teknologi Sumatera berubah dari yang semula berupa luring (luar jaringan) menjadi daring (dalam jaringan). Pembelajaran daring ini menyebabkan proses pengumpulan tugas mahasiswa yang semulanya berupa *hardcopy* menjadi *softcopy*. Opu et al. [2] menyebutkan bahwa proses pendidikan yang menggunakan pensil dan kertas mulai digantikan dengan *stylus* dan *touchscreen*.

Program studi Arsitektur merupakan program studi yang membutuhkan tingkat ketelitian yang tinggi. Program studi ini memiliki mata kuliah bernama Seni Dalam Arsitektur dengan kode mata kuliah AR4153. Merujuk pada wawancara yang telah dilakukan dengan ibu Adelia Enjelina Matondang, S.T., M.T. selaku dosen arsitektur ITERA, mata kuliah Seni Dalam Arsitektur (AR4153) memiliki beberapa penilaian desain dengan penilaian desain fasad depan bangunan menjadi salah satunya. Berdasarkan wawancara yang telah dilakukan desain fasad bangunan merupakan salah satu hal yang penting untuk dipahami seorang insinyur dalam merancang suatu bangunan. Hal ini dikarenakan ilmu fasad depan bangunan dapat menunjukkan identitas suatu bangunan sebagaimana yang disebutkan oleh Anggraini et al. [3] dalam penelitiannya. Selain itu, pada program studi arsitektur ITERA fasad depan bangunan akan digunakan pada tugas akhir sehingga pemahaman mahasiswa perlu memahami bagaimana merancang sebuah fasad depan bangunan.

Berdasarkan wawancara yang telah dilakukan diketahui bahwa pada proses penilaian desain fasad depan bangunan terdapat beberapa parameter penilaian seperti penggunaan tekstur, penggunaan bayangan, dan komposisi pola pada rancangan yang terdiri dari ritme, irama, aksent, kombinasi pola, serta pemanfaatan gelap-terang pada rancangan. Melalui wawancara yang telah dilakukan diketahui bahwa proses penilaian dapat dipengaruhi oleh kecenderungan seorang dosen terhadap suatu desain sehingga

tingkat objektivitas penilaian menjadi berkurang. Dampak lain dengan adanya kecenderungan tersebut ialah waktu yang dibutuhkan oleh dosen dalam melakukan penilaian. Demikian, dosen satu bisa saja menilai suatu desain yang sama dengan dosen lainnya namun membutuhkan waktu yang lebih lama ataupun lebih singkat dikarenakan adanya *bias* tersebut. Hal ini didukung dengan pernyataan Sethu et al. pada penelitiannya [4] bahwa *human errors* tersebut dapat terjadi disebabkan oleh kebiasaan, motivasi, pengetahuan, kecenderungan, dan kondisi fisik maupun psikologis. Oleh sebab itu, proses penilaian dapat menjadi tidak efektif dan dapat mempengaruhi kelulusan mahasiswa yang mengambil mata kuliah Seni Dalam Arsitektur.

Berdasarkan wawancara yang telah dilakukan tersebut diperoleh informasi bahwa jumlah mahasiswa dan dosen yang tidak berimbang. Hal ini dapat dilihat pada Sistem Akademik (Siakad) ITERA bahwa terdapat lima kelas pada tahun ajaran 2021/2022 semester ganjil. Pada kelima kelas tersebut dapat dilihat bahwa terdapat sejumlah 64 mahasiswa yang mengambil mata kuliah Seni Dalam Arsitektur (AR4153). Merujuk pada Surat Edaran Menristekdikti Nomor 105/M/VI/2015 tanggal 5 Juni 2015 perihal Pangkalan Data Pendidikan Tinggi (PDDikti) disebutkan bahwa rasio dosen terhadap mahasiswa ialah 1:20 untuk program studi eksakta dan 1:30 untuk program studi ilmu sosial. Jika menggunakan surat edaran tersebut sebagai acuan maka jika jumlah mahasiswa dibagi dengan jumlah dosen maka setidaknya 1 dosen berhak mengajar sejumlah $\pm 10,67$ mahasiswa. Namun, hal tersebut tidak dapat terjadi dikarenakan persebaran jumlah mahasiswa yang tidak merata sehingga terdapat kemungkinan bahwa terdapat dosen yang mengajar dengan jumlah yang melebihi dari yang seharusnya. Oleh sebab itu, dosen tersebut bisa saja melakukan *human errors* dalam melakukan penilaian sebagaimana yang dijelaskan oleh Sethu et al. [4].

Berdasarkan permasalahan tersebut dibutuhkan sebuah *media* yang dapat mempermudah proses penilaian sehingga permasalahan seperti *human errors* dapat dihindari dan waktu yang dibutuhkan dalam proses penilaian dapat dikurangi. Merujuk pada penelitian Opu et al. [2], Opu et al. mengembangkan sebuah aplikasi *mobile android* yang digunakan untuk mengajarkan pengguna cara menulis. Pada penelitian tersebut, Opu et al. menggunakan sebuah *model machine learning* untuk melakukan penilaian otomatis pada masukan pengguna. Alves et al. [5] dan Garcia-Moreno et al. [6] melakukan penelitian yang serupa dimana mengembangkan sebuah aplikasi *mobile*

android yang digunakan untuk melakukan penilaian/pengklasifikasian otomatis menggunakan *model machine learning*. Mohammed et al. [7] mengembangkan hal yang serupa dengan penelitian yang disebutkan sebelumnya namun menerapkan *model machine learning* pada sebuah *web*. Merujuk pada penelitian-penelitian tersebut maka *media* yang akan dikembangkan untuk menjadi solusi atas permasalahan tersebut adalah sebuah aplikasi *mobile* berbasis *android* yang menggunakan *model machine learning* untuk melakukan penilaian otomatis. Sistem operasi *android* dipilih dikarenakan sebesar 74.95% pengguna sistem operasi *mobile* di dunia menggunakan *android* [8] dan sebesar 91.53% pangsa pasar sistem operasi di Indonesia di dominasi oleh *android* [9].

Pada penelitian tugas akhir ini, penelitian dilakukan menggunakan metode pengembangan berupa *Extreme Programming* (XP) yang merupakan bagian dari *Software Development Life Cycle* (SDLC) *Agile* [10]. Pemilihan penggunaan metode SDLC *Agile* dikarenakan Islam et al. [11] menyebutkan bahwa *Agile* merupakan kerangka kerja yang berfokus pada tahap iteratif pengembangan yang berpusat pada kebutuhan pengguna sehingga memungkinkan terjadinya perubahan pada saat pengembangan. Selain itu, Islam et al. [11] dan Al-Saqqa et al. [10] telah melakukan penelitian yang membandingkan metode-metode pengembangan *Agile* dimana pada penelitian tersebut diperoleh kelebihan dan kekurangan dari masing-masing metode pengembangan *Agile* yang ada.

Berdasarkan penelitian Islam et al. [11] dan Al-Saqqa et al. [10] pada metode pengembangan *Agile* setidaknya terdapat lima metode yakni *Feature Driven Development* (FDD), *Test Driven Development* (TDD), *Dynamic System Development Method* (DSDM), *Scrum*, dan *Extreme Programming* (XP). *Feature Driven Development* (FDD) memiliki kekurangan berupa tidak adanya panduan dalam tahap pengambilan informasi dan FDD tidak terlalu memperhatikan permasalahan-permasalahan yang ada pada sebuah proyek [10]. Oleh sebab itu, FDD tidak mungkin digunakan karena mengabaikan beberapa permasalahan yang ada. *Test Driven Development* (TDD) memiliki kekurangan berupa sang pengembang haruslah mengetahui bagaimana cara menuliskan sebuah pengujian dan jika pengujian tersebut tidak dituliskan dengan baik maka proses perbaikan kode akan memakan waktu yang sangat lama [10]. Selain itu, TDD pada tahap pengembangan bergantung pada

pengujian yang ada sebagai alat pendokumentasian dan penulisan dokumentasi hanya akan dilakukan pada tahap pemeliharaan [11]. Berdasarkan kekurangan yang dimiliki oleh TDD ini, TDD menjadi tidak memungkinkan untuk digunakan dikarenakan proses pendokumentasian yang buruk jika hanya bergantung pada pengujian yang ada pada tahap pengembangan. Al-Saqqa et al. [10] menyebutkan bahwa *Dynamic System Development Method* (DSDM) merupakan metode *Agile* yang memiliki banyak kelebihan namun juga memiliki kekurangan yang dapat menghambat proses-proses yang ada. Kekurangan ini berupa terdapat banyaknya peran (*role*) pada tahap pengembangan sehingga mempersulit proses administrasi yang ada. Selain itu, DSDM kurang memperhatikan hal yang penting/kritikal pada suatu proyek sehingga jika terjadi suatu permasalahan pada tahap pengembangan dapat diabaikan begitu saja [10]. Al-Saqqa et al. [10] juga menyebutkan bahwa pada DSDM tidak ada panduan yang spesifik untuk mengatasi permasalahan seperti ukuran dari tim yang ada dan waktu yang dibutuhkan pada tiap iterasi. Berdasarkan permasalahan tersebut DSDM tidak digunakan dikarenakan dapat mengabaikan permasalahan yang ada. *Scrum* memiliki kelebihan yang menekankan pada bagaimana mengelola sebuah proyek namun terdapat kekurangan yang mengakibatkan metode ini tidak digunakan pada penelitian ini. *Scrum* memungkinkan terjadinya pelanggaran pada tanggung jawab pada tiap anggota dikarenakan tidak adanya definisi yang presisi terkait tanggung jawab masing-masing anggota [11]. Permasalahan ini dapat diabaikan dikarenakan pengembangan dilakukan secara individu namun metode ini tidak dipilih dikarenakan tidak adanya panduan yang spesifik pada proses pengembangannya sehingga penggunaanya dapat menimbulkan kebingungan ketika tahap pengembangan berlangsung [10]. *Extreme Programming* (XP) memiliki kekurangan berupa kurang adanya dukungan pada tim yang tersebar, XP bergantung pada dokumentasi informal, dan keterlibatan *consumer* secara langsung merupakan hal yang efektif namun memakan cukup banyak waktu [10]. Kekurangan yang ada pada XP yang bersifat tim dapat diabaikan dikarenakan pengembangan dilakukan secara individu dan permasalahan dokumentasi informal dapat diatasi dengan menggunakan *tools* proyek manajemen sehingga proses dokumentasi dapat dilakukan dengan baik.

Berdasarkan studi literatur yang dilakukan diperoleh metode *Extreme Programming* (XP) yang memungkinkan untuk digunakan pada penelitian dalam mengembangkan sebuah aplikasi *mobile*. Sebagai penelitian yang menggunakan

metode *Extreme Programming* (XP) dibutuhkan sebuah referensi pengembangan berupa penelitian terdahulu yang menggunakan metode yang sama. Penelitian Bhiantara et al. [12] dan Sucipto et al. [13] akan digunakan sebagai referensi dikarenakan kedua penelitain tersebut sama-sama mengembangkan sebuah aplikasi *mobile* menggunakan metode pengembangan *Extreme Programming* (XP). Metode pengembangan ini akan melalui beberapa tahapan pengujian berupa *unit test* dan *acceptance test* sehingga sebuah fitur yang dikembangkan sesuai dengan kebutuhan pengguna atau tidak. Selain itu, pengujian pada *eXtreme Programming* ditujukan untuk memastikan tidak adanya *bugs* maupun *errors* sehingga bila terjadinya perubahan pada kebutuhan maupun pada kode sehingga dapat dipastikan fitur tersebut masih berfungsi seperti yang diharapkan [14].

1.2 Rumusan Masalah

Untuk menyelesaikan permasalahan dalam proses penilaian desain arsitektur secara otomatis, terdapat rumusan masalah sebagai berikut:

1. Bagaimana menerapkan metode *extreme programming* dalam mengembangkan aplikasi *mobile* penilaian otomatis?
2. Bagaimana mengukur efektifitas penggunaan aplikasi *mobile* penilaian otomatis dalam melakukan penilaian?

1.3 Tujuan Penelitian

Berdasarkan latar belakang dan rumusan masalah di atas, terdapat tujuan dari pelaksanaan penelitian tugas akhir ini sebagai berikut:

1. Mengetahui penerapan metode *extreme programming* dalam mengembangkan aplikasi *mobile* penilaian otomatis.
2. Mengetahui efektifitas penggunaan aplikasi *mobile* penilaian otomatis yang telah dikembangkan.

1.4 Batasan Masalah

Adapun batasan-batasan dalam penelitian tugas akhir ini sebagai berikut:

1. Aplikasi *mobile* akan dikembangkan untuk sistem operasi *android* menggunakan *framework React Native* dan bahasa pemrograman *TypeScript*.
2. Desain fasad depan bangunan yang diterima adalah desain yang berformat JPG.

3. *Model machine learning* yang digunakan adalah *model* yang telah dikembangkan (*pre-trained model*).

1.5 Manfaat Penelitian

Adapun manfaat dari pelaksanaan penelitian tugas akhir ini sebagai berikut:

1. Aplikasi *mobile* dapat mempermudah proses penilaian desain fasad depan bangunan.
2. Faktor *human error* dalam proses penilaian dapat dikurangi.

1.6 Sistematika Penulisan

1.6.1 Bab I Pendahuluan

Bab ini berisikan latar belakang dari penelitian “Pengembangan Aplikasi *Mobile* Untuk Penilaian Desain Arsitektur Menggunakan Metode *Extreme Programming* (Studi Kasus Arsitektur ITERA)”, rumusan masalah, dan tujuan penelitian dilaksanakan. Serta membahas batasan masalah penelitian, manfaat dari penelitian, dan sistematika penulisan penelitian.

1.6.2 Bab II Tinjauan Pustaka

Bab ini berisikan studi literatur mengenai penelitian terdahulu terkait penelitian yang dilaksanakan dan teori-teori dasar pendukung sehingga penelitian tugas akhir dapat berjalan dengan lancar.

1.6.3 Bab III Metode Penelitian

Bab ini berisikan alur dari bagaimana penelitian dilaksanakan, alat dan bahan yang digunakan untuk menunjang penelitian, metode pengembangan, dan rancangan pengujian yang akan dilakukan pada aplikasi yang dikembangkan.

1.6.4 Bab IV HASIL DAN PEMBAHASAN

Bab ini berisikan hasil beserta pembahasan dari hasil penelitian tugas akhir yang telah dilaksanakan.

1.6.5 Bab V KESIMPULAN DAN SARAN

Bab ini berisikan kesimpulan dari hasil penelitian yang telah dilakukan. Selain itu, bab ini juga memuat saran terkait penelitian lebih lanjut.

BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

Pada penelitian tugas akhir ini terdapat beberapa referensi yang digunakan sebagai acuan dasar dan pendukung dalam tahap pelaksanaan penelitian. Referensi-referensi yang digunakan merupakan penelitian terdahulu yang mengembangkan perangkat lunak baik aplikasi *mobile* maupun *website* dan pengembangan *model machine learning*. Penelitian-penelitian terdahulu dapat dilihat pada Tabel 2. 1 berikut.

Pengembangan aplikasi *mobile* pada penelitian tugas akhir ini akan menggunakan metode SDLC *eXtreme Programming* (XP) sebagaimana yang telah dilakukan oleh Bhiantara et al. [12] dan Sucipto et al. [13] dalam mengembangkan aplikasi *mobile*. Selain itu, metode pengujian pada SDLC yang digunakan akan menggunakan tahapan pengujian yang disebutkan oleh Al-Saqqa et al. [10] berupa *unit test* dan *acceptance test* yang berbeda dengan yang dilakukan oleh Noerlina et al. [15], *black-box testing* dan *acceptance testing*. Selain itu, pada penelitian terdahulu tersebut terdapat beberapa metode implementasi penerapan *model machine learning* yang dapat digunakan pada penelitian tugas akhir ini. Merujuk pada penelitian Opu et al. [2] dan Alves et al. [5] terdapat perbedaan waktu yang signifikan antara perangkat yang memiliki spesifikasi rendah dengan perangkat dengan spesifikasi tinggi sehingga dapat memunculkan ke tidak nyamanan dalam penggunaan aplikasi *mobile* dalam proses penilaian. Oleh sebab itu, *model machine learning* akan di implementasikan menggunakan pendekatan pada penelitian Mohammed et al. [7] dimana *model machine learning* akan di implementasikan pada suatu *server* tersendiri sehingga penggunaan aplikasi *mobile* diharapkan tidak memiliki perbedaan waktu yang signifikan seperti yang ada pada penelitian Alves et al. [5].

Tabel 2. 1 Tabel Perbandingan Referensi

| No | Penelitian | Permasalahan | Metode | Hasil |
|----|---|---|---|---|
| 1. | Learn2Write: Augmented Reality and Machine Learning-Based Mobile App to Learn Writing (Opu et al., 2022) | Proses belajar menulis yang sebelumnya dilakukan menggunakan sebuah pensil dan kertas mulai digantikan dengan konten-konten digital. Proses tersebut juga menggantikan peran instruktur dalam membimbing proses belajar menulis dengan proses <i>tracing</i> suatu pola tertentu dalam belajar menulis. | - <i>Augmented Reality</i> - <i>Convolutional Neural Network</i> | - Aplikasi membutuhkan waktu sebesar 457.56 ms dan 5.69 ms untuk memuat <i>model</i> dan mengeksekusi masukan pengguna. - Aplikasi membutuhkan penggunaan RAM sebesar 349 – 398 MB dengan penggunaan terbesar berasal dari komponen AR untuk melakukan visualisasi 3D <i>model</i> . |
| 2. | Automatic Focus Assessment on Dermoscopic Images Acquired with Smartphones (Alves et al., 2019) | Melanoma maligna merupakan penyebab utama kanker antara pria dan wanita. Hal ini dapat ditangani jika pasien dapat di diagnosis dengan lebih cepat. Menggunakan <i>smartphone</i> gambar kulit dapat di tangkap | - <i>Decision Tree Classifier</i> | - Model memiliki akurasi sebesar 77.9% untuk mengidentifikasi kulit yang berkemungkinan kanker. - Dibutuhkan waktu sebanyak 1200 ms dalam melakukan penilaian pada perangkat dengan |

| No | Penelitian | Permasalahan | Metode | Hasil |
|----|--|--|---|--|
| | | melalui kamera dan dapat dianalisa lebih lanjut menggunakan <i>machine learning</i> . | | spesifikasi rendah dan 123 – 126 ms pada perangkat dengan spesifikasi tinggi. |
| 3. | A Machine Learning Approach for Semi-Automatic Assessment of IADL Dependence in Older Adults with Wearable Sensors (Garcia-Moreno et al., 2022) | Penilaian ketergantungan pada orang dewasa yang lebih tua dilakukan dengan mengumpulkan data manual melalui sebuah kuesioner. Proses pengambilan data tersebut dapat mengganggu kehidupan sehari-hari sang pasien dan memakan waktu yang cukup lama bagi dokter untuk menganalisa dan memahami data hasil kuesioner. | <ul style="list-style-type: none"> - <i>k-Nearest Neighbors</i> (k-NN) - <i>Random Forest</i> (RF) - <i>Support Vector Machine</i> (SVM) - <i>5-Fold Stratified CV</i> (5-FSCV) | <ul style="list-style-type: none"> - <i>Model</i> yang dikembangkan menggunakan k-NN memiliki F1-score sebesar 97%. - Sensor yang dirancang dapat mengirimkan hasil pengukuran pada aplikasi <i>mobile</i>. - Aplikasi <i>mobile</i> dapat memprediksi dan menampilkan hasil prediksi yang dihasilkan oleh <i>model machine learning</i>. |
| 4. | Deploying Machine Learning Models Using | Implementasi <i>machine learning</i> pada sebuah <i>website</i> | - <i>Progressive Web Apps</i> (PWA) | - <i>Model machine learning</i> dapat memprediksi kemungkinan |

| No | Penelitian | Permasalahan | Metode | Hasil |
|----|--|--|-------------------|--|
| | Progressive Web Applications: Implementation using Neural Network Prediction Model for Pneumonia Related Child Mortality in The Gambia (Mohammed et al., 2022) | memiliki keterbatasan berupa koneksi internet yang berkemungkinan tidak memadai. Sedangkan implementasi <i>machine learning</i> pada aplikasi <i>native</i> memiliki kecenderungan untuk mengalami sebuah perubahan pada <i>platform</i> tertentu. | | seseorang meninggal dikarenakan oleh Pneumonia. - <i>Website</i> yang dikembangkan dapat di <i>install</i> layaknya sebuah aplikasi <i>mobile</i> . - <i>Website</i> yang telah di <i>install</i> dapat menjalankan prediksi tanpa membutuhkan koneksi internet. |
| 5. | Agile Software Development: Methodologies and Trends (Al-Saqqa et al., 2020) | Terdapat banyak <i>software development method</i> yang mengalami permasalahan ketika terjadinya perubahan pada kebutuhan pengguna. Menanggapi masalah tersebut <i>agile</i> dikembangkan, namun kurang adanya literatur yang melakukan perbandingan antara metode-metode tersebut. | - Studi literatur | - Setiap metode <i>agile</i> memiliki keunggulan dan kelemahan masing-masing sehingga tidak ada <i>SDLC</i> yang sempurna untuk semua tipe proyek. - <i>eXtreme Programming</i> (XP) merupakan salah satu metode <i>agile</i> yang sering digunakan. |

| No | Penelitian | Permasalahan | Metode | Hasil |
|----|--|--|---|---|
| | | | | - <i>Scrum</i> merupakan metode <i>agile</i> yang menekankan pada proses mengelola proyek. |
| 6. | Pengembangan Sistem Informasi Pariwisata Terintegrasikan <i>E-Ticket Mobile</i> dengan Metode <i>Extreme Programming</i> (Studi Kasus Dinas Pariwisata Karangasem) (Bhiantara et al., 2021) | Perkembangan pariwisata di Indonesia menjadi salah satu sektor pemasukan dalam penghasilan suatu daerah maupun pendapatan devisa negara. Kabupaten Karangasem di Bali memiliki banyak tempat wisata yang indah yang dapat dikunjungi oleh para wisatawan. Meskipun memiliki tempat wisata yang banyak, proses pendataan dan pembelian tiket masih dilakukan secara manual dan tidak efisien. | - <i>eXtreme Programming</i> (XP) - <i>Black-box testing</i> | - Baik sistem informasi maupun <i>E-Ticket Mobile</i> dapat memenuhi kebutuhan pengguna yang telah didefinisikan. - Rata-rata <i>running time</i> sebesar 479 ms dibutuhkan oleh sistem informasi maupun <i>E-Ticket Mobile</i> dalam melakukan sebuah proses. |

| No | Penelitian | Permasalahan | Metode | Hasil |
|----|---|---|---|--|
| 7. | Sistem Pemeliharaan Menara BTS (Base Transceiver Station) Berbasis Mobile (Sucipto et al., 2022) | <i>Base Transceiver Station</i> merupakan menara yang terbuat dari rangkaian besi yang digunakan untuk pemancar maupun menerima gelombang telekomunikasi. Sebagai sebuah pemancar, perawatan terhadap menara BTS diperlukan namun dibutuhkannya sebuah cara/metode untuk memperoleh informasi apakah suatu menara membutuhkan perawatan atau tidak. | - <i>eXtreme Programming</i> (XP) | - Hasil pengujian pada aspek fungsionalitas bernilai 100%. - Hasil pengujian aspek <i>usability</i> bernilai 89%. |
| 8. | Sales Revenue Sharing Model using Dynamics NAV Modifications in Health Industries | <i>Enterprise Resource Planning</i> (ERP) merupakan sebuah sistem yang terintegrasi dengan semua kebutuhan fungsional perusahaan menjadi data-data | - <i>eXtreme Programming</i> (XP) - <i>Black-box testing</i> | - Proses rekapitulasi pada rumah sakit menjadi lebih mudah dan hingga 50% lebih cepat. |

| No | Penelitian | Permasalahan | Metode | Hasil |
|----|-------------------------|---|--------|---|
| | (Noerlina et al., 2020) | kecil yang dapat digunakan untuk membuat keputusan. ERP dikembangkan menggunakan sebuah <i>best practice</i> yang tidak menjamin dengan kebutuhan suatu perusahaan. Demikian, sebuah modifikasi terhadap ERP dibutuhkan sehingga proses-proses tersebut sesuai dengan kebutuhan suatu perusahaan. | | - Proses pembuatan <i>invoice</i> manual digantikan dengan sebuah automasi. |

2.2 Dasar Teori

Pada penelitian tugas akhir ini diperlukan beberapa dasar teori pendukung yang akan digunakan untuk memperkuat pemahaman dan sebagai acuan dalam pelaksanaan penelitian. Adapun dasar teori dari penelitian tugas akhir ini sebagai berikut:

2.2.1 Extreme Programming (XP)

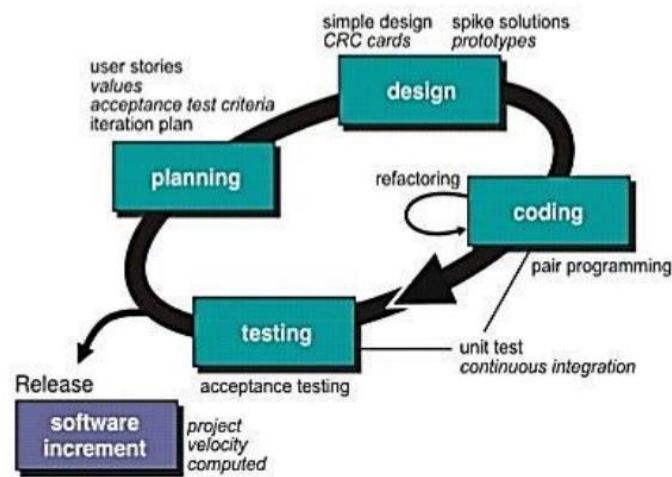
eXtreme Programming (XP) merupakan salah satu bagian dari *Software Development Life Cycle* (SDLC) *Agile*. *Agile* merupakan SDLC yang merepresentasikan sebuah filosofi untuk melakukan kolaborasi antara beberapa orang dengan tujuan untuk mencapai kepuasan *customer* melalui iterasi yang pendek dan cepat [10]. Anharudin et al. [16] menyebutkan bahwa XP cenderung menggunakan pendekatan berorientasi objek dan dihadapkan dengan *requirement* yang tidak jelas maupun terjadinya perubahan-perubahan *requirement* yang sangat cepat. Al-Saqqa et al. [10] dan Anharudin et al. [16] menyebutkan bahwa metode XP dipelopori oleh Kent Beck, Ron Jeffries, dan Ward Cunningham dengan tujuan untuk mengurangi biaya perubahan *requirements* dengan menggantikan siklus pengembangan yang panjang menjadi siklus yang lebih pendek.

Al-Saqqa et al. [10] menyebutkan terdapat beberapa kelebihan dalam penggunaan *eXtreme Programming* (XP) sebagai berikut:

1. Terdapatnya dukungan terhadap pengembangan secara bertahap pada perilsan sistem yang kecil dan dilakukan secara berkala.
2. *Feedback* yang diperoleh dengan cepat pada tingkat “*extreme*” dapat meningkatkan produktivitas dan banyak versi yang dapat dirilis hanya jika telah melewati serangkaian pengujian.
3. Mempertahankan kesederhanaan kode melalui proses *refactoring* yang terjadi secara konstan.
4. Meningkatkan kualitas pengembangan dengan adanya automasi pengujian sebelum fitur di implementasikan pada sistem.

Selain menyebutkan kelebihan Al-Saqqa et al. [10] juga menyebutkan kelemahan dari penggunaan *eXtreme Programming* (XP) sebagai berikut:

1. Kurang adanya dukungan pada tim yang terdistribusi dikarenakan fokus pada komunitas.
2. Pendekatan menggunakan *Test-Driven Development* (TDD) membutuhkan sebuah pelatihan untuk tiap anggota tim.
3. XP bergantung pada dokumentasi yang dilakukan secara *informal*.
4. Keterlibatan *consumer* secara langsung merupakan hal yang efektif namun memakan cukup banyak waktu.



Gambar 2. 1 Extreme Programming Life Cycle [15]

Melalui Gambar 2. 1, Noerlina et al. [15] menyebutkan terdapat beberapa fase dalam pengembangan sebuah perangkat lunak menggunakan *eXtreme Programming* (XP) sebagai berikut:

1. *Planning Phase* merupakan fase yang digunakan untuk melakukan analisa kebutuhan pengguna (*user requirements*). Pada tahapan ini umumnya akan diperoleh sebuah *user stories* yang berisikan kebutuhan pengguna yang belum terkategori menjadi fungsional maupun non-fungsional.
2. *Design Phase* merupakan fase dimana kebutuhan-kebutuhan pengguna tersebut dirancang ke dalam bentuk rancangan seperti *use case diagram* maupun *activity diagram*.
3. *Coding Phase* merupakan fase pengembangan fitur perangkat lunak berdasarkan dari kebutuhan pengguna yang telah diperoleh pada *planning phase*.

4. *Testing Phase* merupakan tahapan pengujian yang diperuntukan untuk menemukan *bugs* maupun *errors* pada sistem yang dikembangkan. Fase ini diulang kembali kedalam *coding phase* jika menemukan *bugs*, *errors*, maupun perubahan kebutuhan pengguna.

Al-Saqqa et al. [10] menyebutkan terdapat dua macam pengujian yang dapat dilakukan pada proses pengembangan yang menggunakan *eXtreme Programming* (XP). Pengujian tersebut diantaranya *acceptance test* dan *unit test*. Hal ini berbeda dengan yang di implementasikan oleh Noerlina et al. [15] yang mana menggunakan *black-box testing*. Pada penelitian tugas akhir ini, proses pengujian akan menggunakan pendekatan yang disebutkan oleh Al-Saqqa et al. [10] dikarenakan proses pengujian yang akan dilakukan berulang kali dan akan lebih baik menggunakan *unit test* sehingga proses pengujian dapat di otomatisasi.

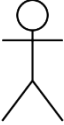
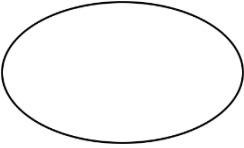
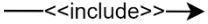

Penelitian tugas akhir ini menggunakan SDLC *eXtreme Programming* (XP) dikarenakan kebutuhan pengguna yang dapat berubah pada saat pengembangan dan fitur yang akan di implementasikan dapat dipastikan terlebih dahulu apakah sesuai dengan kebutuhan, apakah ada *bugs* maupun *errors* menggunakan *unit test* yang telah dikembangkan. Selain itu, merujuk pada kelebihan dan kekurangan yang disebutkan Al-Saqqa et al. [10], terdapat kekurangan yang dapat diabaikan seperti “kurang adanya dukungan pada tim yang terdistribusi dikarenakan fokus pada komunitas”. Kekurangan tersebut dapat diabaikan dikarenakan pengembangan akan dilakukan secara individu dan permasalahan dokumentasi informal dapat diatasi menggunakan *project management tools* sehingga proses dokumentasi dapat dilakukan dengan baik.

2.2.2 Use Case Diagram

Use case diagram merupakan diagram yang menggambarkan sebuah tingkah laku pada suatu sistem [17]. Diagram ini ditujukan untuk menggambarkan kebutuhan fungsional suatu perangkat lunak. Selain itu, *use case diagram* juga dapat digunakan untuk menggambarkan bagaimana sebuah sistem bekerja. Pressman [18] menyebutkan *use case diagram* merupakan salah satu metode pemodelan suatu sistem yang merujuk pada suatu skenario tertentu. Pressman [18] dan Sommerville [19] menyebutkan pada

sebuah *use case diagram* terdapat beberapa komponen yang dapat dilihat pada Tabel 2. 2 berikut.

Tabel 2. 2 Komponen Use Case Diagram

| Simbol | Nama | Deskripsi |
|---|----------|---|
|  | Actors | Merupakan orang/perangkat yang menggunakan sistem/produk yang menggunakan fungsi/bertindak sesuai dengan yang di deskripsikan. |
|  | Use Case | Merepresentasikan dialog antara aktor dengan sistem. <i>Use case</i> dapat merepresentasikan dialog antara satu atau lebih aktor. |
|  | Include | Menunjukkan bahwa suatu <i>use case</i> merupakan sebuah fungsionalitas dari <i>use case</i> lainnya. |
|  | Extend | Menunjukkan bahwa suatu <i>use case</i> merupakan sebuah tambahan fungsional dari <i>use case</i> lainnya. |




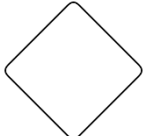
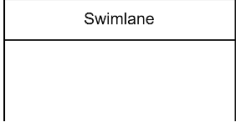
Pada penelitian tugas akhir ini, metode pengembangan yang digunakan berupa *eXtreme Programming* (XP) yang memiliki sebuah tahapan berupa perencanaan (*planning*). Pada tahap perencanaan (*planning*), *use case diagram* akan diperoleh setelah memperoleh kebutuhan fungsional dan non-fungsional pengguna sehingga skenario kebutuhan pengguna dapat tergambarkan dengan baik. Selain itu, pada tahap pengembangan *use case diagram* akan menjadi sebuah rujukan apakah aplikasi telah memenuhi semua skenario yang ada atau tidak.

2.2.3 Activity Diagram

Activity diagram merupakan sebuah diagram yang bertujuan untuk mendeskripsikan tingkah laku pada sebuah sistem dan logika internal pada sebuah operasi yang kompleks [20]. Al-Fedaghi [20] menyebutkan *activity diagram* merupakan sebuah alat yang sangat membantu dalam proses validasi suatu kebutuhan pengguna. Pressman

[18] menyebutkan *activity diagram* merupakan diagram yang mendukung *use case* dengan memberikan sebuah gambaran alur dari suatu skenario yang ada. Adapun komponen-komponen pada suatu *activity diagram* sebagaimana yang telah dijelaskan oleh Pressman [18] dan Sommerville [19] yang dapat dilihat pada Tabel 2. 3 berikut.

Tabel 2. 3 Komponen Activity Diagram

| Simbol | Nama | Deskripsi |
|---|----------|---|
|  | Start | Melambangkan status awal dari aktivitas sistem. |
|  | End | Melambangkan status akhir dari aktivitas sistem. |
|  | Activity | Melambangkan sebuah aktivitas yang dilakukan oleh sebuah sistem. |
|  | Decision | Melambangkan sebuah percabangan dimana terdapat pilihan aktivitas yang lebih dari satu. |
|  | Swimlane | Melambangkan pemisahan organisasi bisnis yang bertanggung jawab pada sebuah aktivitas. |

Pada penelitian tugas akhir ini, *activity diagram* akan diperoleh setelah memperoleh *use case diagram* pada tahap perencanaan (*planning*) *eXtreme Programming* (XP). *Activity diagram* yang diperoleh tersebut sebagaimana yang telah disebutkan oleh Pressman [18] akan digunakan untuk memberikan gambaran alur sebuah skenario pada *use case diagram*. Selain itu, *activity diagram* tersebut dapat digunakan sebagai rujukan alur bagaimana suatu proses berjalan pada aplikasi *mobile*.

2.2.4 Unit Testing

Unit Testing merupakan *framework* yang telah didefinisikan terlebih dahulu yang membandingkan luaran suatu kode program dengan luaran yang diharapkan [21]. Berbeda dengan definisi Lucas et al. [21], Guo et al. [22] menyebutkan *unit testing* merupakan salah satu metode pengujian perangkat lunak dengan melakukan pengujian

suatu unit/komponen tertentu. Berdasarkan penjelasan tersebut, dapat ditarik kesimpulan bahwa *unit test* merupakan sebuah metode pengujian yang dilakukan dengan membandingkan masukan dengan keluaran dari unit kode program tertentu. Yang et al. [14] menyebutkan bahwa sebuah *unit test* dapat membantu pengembang untuk memahami kebutuhan pengembangan yang spesifik dan dapat meningkatkan kepercayaan diri sang pengembang dalam melakukan perubahan. Keuntungan lain dari penggunaan *unit test* ialah proses *refactoring* kode yang lebih cepat [14].

Pada penelitian tugas akhir ini, pengujian akan dilakukan menggunakan *unit test* mengikuti SDLC yang digunakan yakni *eXtreme Programming* (XP) dengan pendekatan yang disebutkan Al-Saqqah et al. [10]. Sebagaimana yang disebutkan oleh Noerlina et al. [15] pengujian akan dilakukan berulang kali ketika menemukan sebuah *bugs* atau *errors* maupun terdapat perubahan *user requirements*. Menggunakan *unit test*, pengujian tersebut dapat di otomasi dan kode program yang telah ditulis dapat dipastikan sesuai dengan kebutuhan jika terdapat sebuah perubahan sehingga fitur yang ditambahkan ke dalam sistem diharapkan tidak memiliki *bugs* maupun *errors*.

2.2.5 Acceptance Testing

Acceptance Testing merupakan metode pengujian yang ditulis sebelum proses pengembangan dilakukan dan ditulis menggunakan perspektif pengguna perangkat lunak [23]. Tatale dan Chandra [23] menyebutkan bahwa *acceptance test* ditulis menggunakan informasi yang diperoleh dari *user requirements* yang kemudian diubah ke dalam sebuah skenario-skenario tertentu. Berdasarkan penjelasan tersebut, implementasi *acceptance testing* menggunakan SDLC *eXtreme Programming* (XP) akan dilakukan pada *design phase*. Merujuk pada Noerlina et al. [15] hasil dari *planning phase* umumnya merupakan sebuah *user stories* yang kemudian dapat digunakan untuk membuat berbagai macam skenario *acceptance test*.

Pada penelitian tugas akhir ini, *acceptance test* akan digunakan untuk memastikan apakah fitur-fitur yang telah dikembangkan sesuai dengan kebutuhan pengguna (*user requirements*). Merujuk pada fase *eXtreme Programming* (XP) yang disebutkan oleh Noerlina et al. [15] pengujian akan dilakukan setiap terjadi perubahan *user requirements*. Oleh sebab itu, pengujian menggunakan *acceptance test* akan dilakukan

untuk memastikan apakah fitur yang dikembangkan memenuhi semua kebutuhan pengguna yang sudah ada maupun yang baru saja ditambahkan.

2.2.6 TypeScript

TypeScript merupakan sebuah *superset* dari *JavaScript*. Bahasa pemrograman ini merupakan bahasa yang dikembangkan oleh *Microsoft* dan dirilis pada bulan Oktober 2012 [24]. Bahasa pemrograman ini ditujukan untuk menerapkan semua aturan panduan *Ecma Script* (ES) [25] yang telah dikeluarkan untuk bahasa pemrograman *JavaScript*. Berbeda dengan *JavaScript* yang bersifat *gradual typing* [26] dimana memungkinkannya penggunaan kode program yang memiliki tipe data dengan kode program yang tidak memiliki tipe data [27], Bogner dan Merkel menyebutkan bahwa *TypeScript* bersifat *static typing* dimana tipe data suatu variabel tidak dapat berubah pada saat proses eksekusi kode program [24]. Sebagai *superset* dari bahasa pemrograman *JavaScript*, semua kode program *JavaScript* merupakan kode program yang *valid* pada *TypeScript* namun tidak sebaliknya.

TypeScript sebagai bahasa yang dikembangkan sebagai *superset* dari *JavaScript* memiliki semua karakteristik yang dimiliki *JavaScript* namun dengan beberapa penambahan fitur tertentu. *TypeScript* yang bersifat *static typing* membuat proses *debugging* menjadi lebih mudah dikarenakan adanya ke konsistenan tipe data pada proses eksekusi kode program. Berbeda dengan *JavaScript* yang memungkinkan terjadinya perubahan tipe data secara dinamis [25] sehingga proses *debugging* menjadi lebih sulit dikarenakan tidak adanya ke konsistenan tipe data sebuah variabel.

Bahasa pemrograman *TypeScript* ini akan digunakan baik pada proses pengembangan aplikasi *mobile* maupun pengembangan *server* dalam menjalankan *model machine learning*. Semua kode program yang dikembangkan menggunakan *TypeScript* akan di *transpile* menjadi *JavaScript* menggunakan *TypeScript Compiler* (tsc) [24] sehingga proses eksekusi akan menjadi lebih cepat pada saat penggunaan aplikasi *mobile* dan *server* dikarenakan tidak adanya proses pengerjaan tipe data yang umumnya dilakukan pada *TypeScript*. Pada sisi *mobile TypeScript* akan digunakan dalam pengembangan menggunakan *React Native* dan pada sisi *server TypeScript* akan digunakan dalam menggunakan *model machine learning* dengan *Tensorflow Js*.

2.2.7 Express

Express merupakan sebuah *framework* yang digunakan untuk mempermudah proses pengembangan sebuah *RESTful API* pada *Node Js* menggunakan bahasa pemrograman *JavaScript* [28]. *RESTful* (REST) merupakan sebuah model arsitektur untuk menghubungkan *client* dan *server* menggunakan *protocol Hyper Text Transfer Protocol* (HTTP) [28]. Sedangkan *Application Protocol Interface* (API) merupakan sebuah *protocol* yang digunakan untuk melakukan integrasi antara beberapa aplikasi/perangkat lunak seperti *server* dengan aplikasi *mobile* [28]. Pada *Express*, *Node Js* digunakan sebagai media/alat untuk melakukan eksekusi pada bahasa pemrograman *JavaScript* [29].

Pada penelitian tugas akhir ini, *framework Express* akan disandingkan dengan *Tensorflow Js* untuk membuat sebuah *endpoint API* untuk melakukan penilaian otomatis. Diyasa et al. [28] menyebutkan terdapat beberapa kode *response* yang umum digunakan seperti 200 (*OK*), 400 (*Bad Request*), 401 (*Unauthorized*), 403 (*Forbidden*), 404 (*Not Found*), 429 (*Too Many Request*), dan 500 (*Internal Server Error*). *Response* dari *endpoint API* yang telah disediakan tersebut akan dikirimkan melalui aplikasi *mobile* dan hasil dari *response* tersebut dapat ditampilkan pada aplikasi *mobile*.

2.2.8 Android

Android merupakan sistem operasi yang pertama kali rilis pada tahun 2007 yang menggunakan *UNIX Kernel* [30]. Sistem operasi ini merupakan salah satu sistem operasi yang paling banyak digunakan di dunia [31] dan memiliki daya tarik berupa sistem operasi yang bersifat *open-source* [32]. Melalui penelitian Awanda et al. [8] dari tahun 2018 – 2022 terdapat 74.95% penduduk di dunia menggunakan sistem operasi *android*. Berbeda dengan Awanda et al. [8], Dewi et al. [9] menyempitkan penelitiannya dengan memusatkan pada Indonesia dan diperoleh bahwa di Indonesia sebesar 91.35% pangsa pasar di Indonesia didominasi oleh sistem operasi *android*.

Pada penelitian tugas akhir ini, *media* yang dikembangkan dalam mengatasi permasalahan yang ada adalah mengembangkan sebuah aplikasi *mobile* berbasis *android*. Sistem operasi ini dipilih dikarenakan penelitian terdahulu yang digunakan sebagai acuan mengembangkan aplikasi *mobile* pada sistem operasi *android*. Selain

itu, merujuk pada penelitian Dewi et al. [9] pengembangan sistem operasi *android* meningkatkan kemungkinan bahwa aplikasi ini dapat digunakan oleh dosen arsitektur ITERA.

2.2.9 React Native

React Native merupakan sebuah *framework* yang dikembangkan oleh *Facebook* [33] dan diperkenalkan pertama kali pada tahun 2015 dan mengalami pengembangan yang sangat cepat [34]. *Framework* ini dikembangkan diatas *framework* lain bernama *React Js* [34]. *Framework React Js* merupakan sebuah *framework* yang ditujukan untuk mengembangkan *Progressive Web App* (PWA) [35] yang mengusung konsep *components* sehingga penulisan kode program dapat dipecah menjadi beberapa komponen [36]. Sebagai *framework* yang dikembangkan diatas *framework React Js* maka semua konsep yang ada pada *React Js* seperti komponen maupun *hooks* juga dapat diterapkan pada *framework React Native*.

Framework ini merupakan *framework cross platform app* yang memungkinkannya pengembangan aplikasi *mobile* menggunakan kode program yang sama untuk *platform Android* maupun *iOS* [33]. Biørn-Hansen et al. menyebutkan dalam penelitiannya [37] bahwa penggunaan *code base* yang sama sehingga proses pengembangan aplikasi *mobile* menjadi lebih cepat dan tidak diperlukannya perubahan kode program pada saat aplikasi memasuki tahapan peluncuran. Zohud dan Zein pada penelitiannya [38] menyebutkan bahwa dukungan komunitas *React Native* dapat mempermudah pengembangan dikarenakan telah menyediakan berbagai macam *library/package* yang siap digunakan.

Merujuk pada pernyataan Biørn-Hansen et al. [37] dengan menggunakan *code base* yang sama maka pengembangan aplikasi pada *platform iOS* tidak diperlukan dikarenakan penggunaan *code base* yang sama. Selain itu, jika terjadi ketidaksesuaian pada sisi *Android* maka dapat dilakukan perbaikan dengan cepat dan perubahan tersebut juga langsung diterapkan pada sisi *iOS* dikarenakan penggunaan *code base* yang sama. Oleh sebab itu, pengembangan menggunakan *framework React Native* menjadi pilihan yang tepat dikarenakan ketika aplikasi yang telah dikembangkan akan

digunakan pada perangkat *iOS* cukup melakukan *build* menggunakan *Xcode* yang dapat digunakan pada *MacOS*.

2.2.10 Machine Learning

Machine Learning merupakan sebuah program komputer yang performanya meningkat dengan pengalaman dalam tugas tertentu. Oleh sebab itu, *machine learning* ditujukan pada proses automasi pembuatan model analitik dalam melakukan tugas kognitif seperti deteksi objek maupun penerjemahan bahasa alami [39]. Berdasarkan definisi yang disebutkan oleh Janiesch et al. dapat ditarik kesimpulan bahwa *machine learning* digunakan untuk tugas-tugas tertentu yang berhubungan dengan kemampuan kognitif yang dapat di automasi.

Opu et al. menggunakan *machine learning* pada penelitiannya [2] untuk melakukan klasifikasi pada sebuah karakter alfabet yang ditulis tangan. Pada penelitian tersebut dapat dilihat bahwa *machine learning* yang dikembangkan digunakan untuk proses klasifikasi sebuah objek yang bersumber dari suatu gambar. Berbeda dengan penelitian Opu et al. yang menggunakan *Augmented Reality* untuk memproses gambar, Li et al. menggunakan *machine learning* dalam melakukan klasifikasi jagung dengan kondisi baik atau tidak dengan melakukan pengolahan gambar terlebih dahulu menggunakan *computer vision* [40].

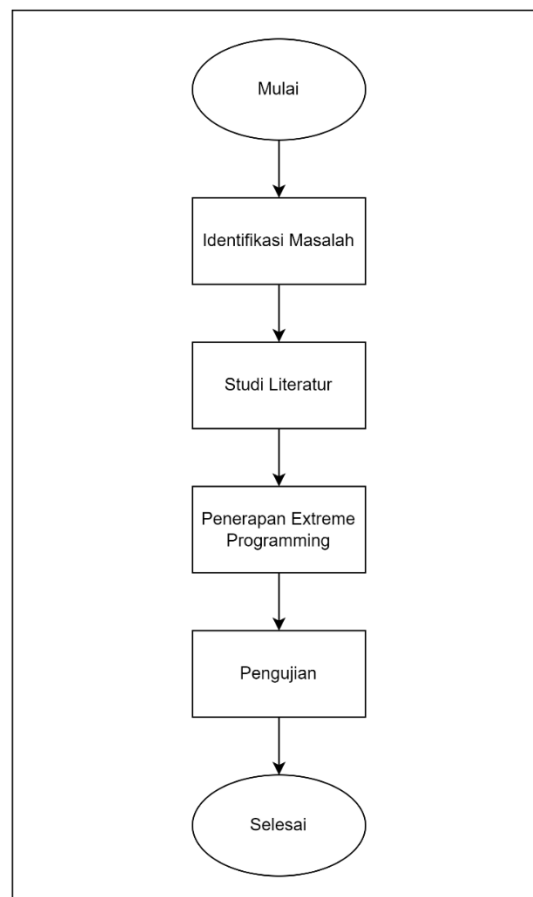
Pada tahap penggunaan *model machine learning*, *model* akan dijalankan menggunakan *library Tensorflow Js* yang memungkinkan penggunaan *model machine learning* yang dikembangkan menggunakan *Tensorflow* pada bahasa pemrograman *JavaScript* [7]. Pada penelitian yang dilakukan oleh Mohammed et al. [7] *model* yang dikembangkan merupakan *model* yang dikembangkan dengan *Tensorflow* pada bahasa pemrograman *R* dan dijalankan menggunakan *Tensorflow Js*. Merujuk pada penelitian tersebut, dapat ditarik kesimpulan bahwa terdapat sebuah peluang dimana *model* dapat di implementasikan pada *server Node Js*.

BAB III

METODE PENELITIAN

3.1 Alur Penelitian

Pada penelitian tugas akhir ini, terdapat alur penelitian yang digambarkan dalam sebuah diagram alir (*flowchart*). Diagram alir penelitian tugas akhir dapat dilihat pada Gambar 3. 1 yang dimulai dengan melakukan identifikasi masalah, studi literatur, menerapkan SDLC *eXtreme Programming* (XP) dan diakhiri dengan melakukan pengujian efisiensi penggunaan aplikasi yang telah dikembangkan. Proses pengembangan akan terjadi pada tahap penerapan SDLC *eXtreme Programming* (XP) dan pengujian efektivitas aplikasi dilakukan pada pengujian setelah melewati SDLC *eXtreme Programming* (XP).



Gambar 3. 1 Diagram Alir Penelitian

3.2 Penjabaran Langkah Penelitian

Berdasarkan diagram alir yang terdapat pada Gambar 3. 1, terdapat empat langkah penelitian utama berupa identifikasi masalah, studi literatur, penerapan SDLC *eXtreme Programming* (XP) dan pengujian efisiensi penggunaan aplikasi. Adapun penjabaran lebih lanjut dari langkah-langkah tersebut sebagai berikut:

3.2.1 Identifikasi Masalah

Penelitian tugas akhir diawali dengan proses pengidentifikasian masalah. Tahapan ini dilakukan dengan tujuan untuk mendapatkan permasalahan apa saja yang memungkinkan untuk dapat diselesaikan dengan salah satu bidang keilmuan teknik informatika. Melalui tahapan ini diperoleh sebuah permasalahan yang akan diteliti pada tugas akhir ini menggunakan salah satu bidang keilmuan teknik informatika.

3.2.2 Studi Literatur

Pemahaman secara teoritis sangatlah dibutuhkan dalam mendukung keberlangsungan penelitian tugas akhir ini. Pemahaman teoritis tersebut dapat diperoleh melalui buku maupun jurnal penelitian yang membahas hal yang serupa. Penelitian terdahulu tersebut akan digunakan sebagai acuan dasar dalam melakukan penelitian sehingga dapat diperoleh informasi seperti metode apa saja yang pernah digunakan. Selain itu, penelitian terdahulu tersebut juga dapat digunakan untuk menentukan apakah penelitian pada tugas akhir ini memungkinkan untuk dilakukan atau tidak.

3.2.3 Penerapan eXtreme Programming (XP)

Berdasarkan permasalahan yang diperoleh akan dikembangkan sebuah aplikasi *mobile* yang diharapkan dapat menjadi solusi atas permasalahan yang ada. Aplikasi *mobile* yang dikembangkan akan menggunakan metode SDLC berupa *eXtreme Programming* (XP) yang menekankan pada pengembangan kode program yang teruji menggunakan *unit test* maupun *acceptance test*. Oleh karena itu, pengembangan aplikasi *mobile* dapat dipastikan teruji dengan baik sehingga dapat memenuhi semua kebutuhan yang ada.

Merujuk pada Gambar 3. 1, XP yang digunakan pada penelitian terdiri dari beberapa tahapan berupa perencanaan, perancangan, pengembangan, validasi, perbaikan, dan

pengerahan. Pada tahap validasi, terdapat dua metode validasi yang akan digunakan yakni *unit test* dan *acceptance test*. Kode program yang telah dikembangkan akan diuji menggunakan kedua metode pengujian tersebut dan jika tidak bisa melewati pengujian yang ada maka kode program akan diperbaiki dan pengujian akan diulang kembali. Jika kode program dapat melewati semua pengujian yang telah didefinisikan maka akan dilakukan pengecekan apakah harus melakukan peningkatan (*incremental*) pada pengembangan atau tidak. Jika harus melakukan peningkatan maka pada tahap pengembangan selanjutnya akan berfokus pada fitur yang berbeda dan akan melakukan prosedur yang sama. Jika semua fitur telah selesai dikembangkan maka akan masuk ke tahap pengerahan aplikasi yang siap untuk dilakukan pengujian lebih lanjut.

3.2.4 Pengujian

Aplikasi *mobile* yang telah dikembangkan akan diujikan sehingga diperoleh apakah dengan adanya aplikasi *mobile* yang dikembangkan proses penilaian menjadi lebih efektif atau tidak. Pengujian akan dilakukan dengan melakukan sebuah perbandingan antara waktu yang dibutuhkan oleh dosen arsitektur dalam menilai fasad depan bangunan dengan waktu yang dibutuhkan oleh aplikasi dalam melakukan penilaian. Selain membandingkan waktu yang dibutuhkan, hasil dari penilaian yang dilakukan oleh dosen dan aplikasi juga akan dibandingkan. Demikian, dapat ditarik kesimpulan apakah aplikasi *mobile* dapat mempercepat proses penilaian dan memberikan nilai yang menyerupai dosen atau tidak.

3.3 Alat dan Bahan Tugas Akhir

Pada penelitian tugas akhir ini diperlukan beberapa alat dan bahan dalam menunjang proses penelitian. Adapun alat dan bahan yang dibutuhkan penelitian tugas akhir ini sebagai berikut:

3.3.1 Alat

Pada penelitian tugas akhir ini dibutuhkan beberapa alat pendukung untuk mendukung berjalannya penelitian. Adapun alat-alat pendukung penelitian yang dimaksud adalah sebagai berikut:

1. *Notebook* dengan spesifikasi minimum, sistem operasi Windows 10 / GNU/Linux, *processor* AMD Ryzen 3250U @ 2,0 GHz, memori 8GB DDR4 2400MHz, *hard drive* 512GB, grafis AMD Radeon Vega Mobile Gfx. Pada penelitian ini digunakan Windows 11 *dual boot* dengan Zorin OS (GNU/Linux), AMD Ryzen 3500U @ 2,1 GHz, memori 20GB DDR4 2400MHz, SSD 2,5TB, grafis AMD Radeon Vega Mobile Gfx.
2. *Smartphone* dengan spesifikasi minimum, sistem operasi Android v9 (*Pie*), CPU Octa-core $\pm 1,8$ GHz, GPU Adreno 610, Internal 32 GB, RAM 4 GB. Pada penelitian ini digunakan perangkat dengan sistem operasi Android v10 (Q), CPU Octa-core 2 x 2,0 GHz & 6 x 1,8 GHz, GPU Mali-G52 MC2, Internal 64 GB, RAM 6 GB.
3. *Run timer JavaScript* berupa *Node JS* dengan versi minimum 14.20.0 (*Fermium*). Pada penelitian ini digunakan *Node JS* dengan versi 16.16.0 (*Gallium*).
4. *Android Studio* untuk mengubah kode program menjadi kode program yang dapat dipahami oleh *platform android* dengan versi minimum 4.2.1. Pada penelitian ini, *Android Studio* yang digunakan adalah versi 4.2.1 pada sistem operasi Windows 11 dan versi 4.3.1 pada Zorin OS (GNU/Linux).
5. *Visual Studio Code* sebagai *Code Editor* dengan versi minimum 1.50.1. Pada penelitian digunakan *Visual Studio Code* dengan versi 1.69.2 (dapat berubah seiring adanya pembaruan *Visual Studio Code*).

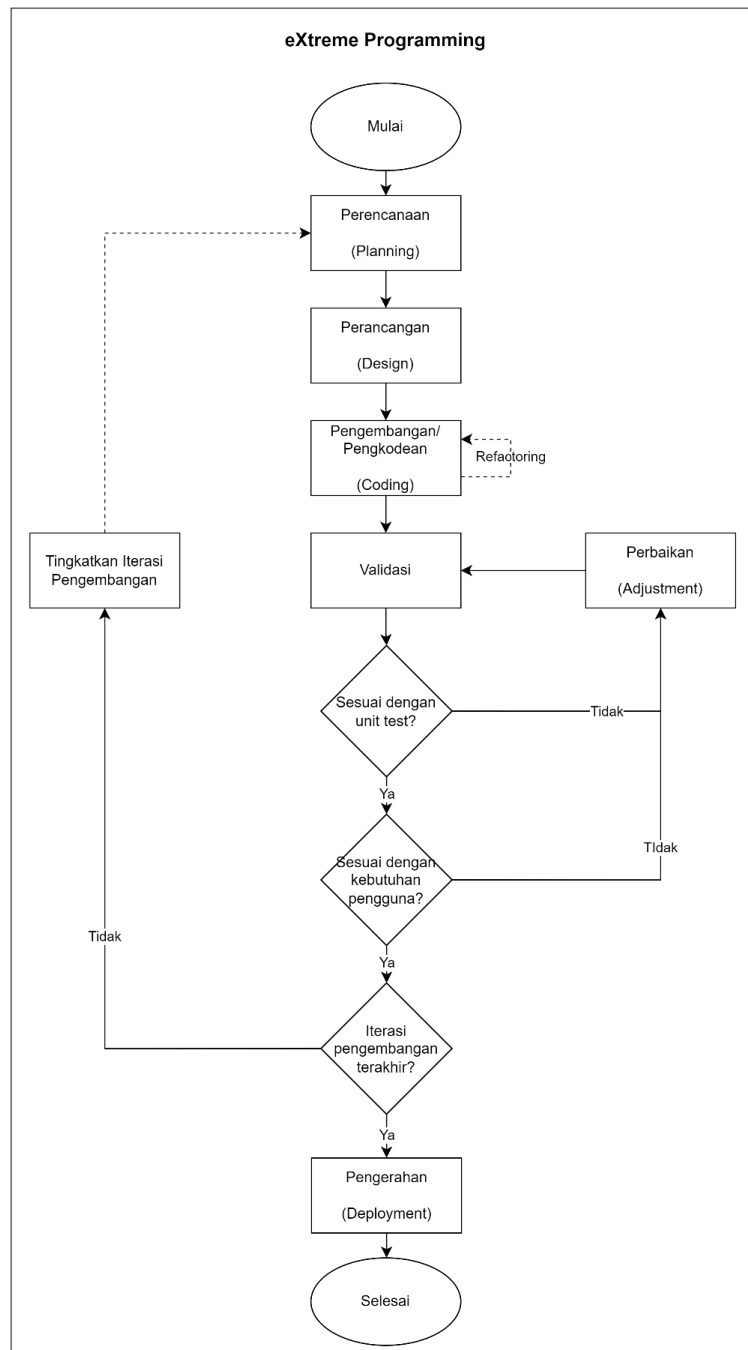
3.3.2 Bahan

Selain membutuhkan alat, penelitian tugas akhir ini juga membutuhkan beberapa bahan yang diperlukan untuk menunjang keberlangsungan penelitian. Bahan-bahan yang dibutuhkan pada penelitian ini adalah sebagai berikut:

1. Hasil wawancara kebutuhan aplikasi *mobile* yang dilakukan dengan dosen program studi arsitektur.

3.4 Metode Pengembangan

Pada penelitian tugas akhir ini, metode pengembangan yang digunakan ialah *Extreme Programming* (XP) yang merupakan bagian dari SDLC *Agile*. Alur dari pengembangan menggunakan *Extreme Programming* dapat dilihat pada Gambar 3. 2 berikut.



Gambar 3. 2 Alur Pengembangan Extreme Programming (XP)

Pada Gambar 3. 2 dapat dilihat bahwa *Extreme Programming* memiliki beberapa tahapan berupa perencanaan, perancangan, pengembangan, validasi, perbaikan, peningkatan iterasi pengembangan, dan pengerahan. Adapun penjabaran tahapan-tahapan dari metode *Extreme Programming* sebagai berikut:

3.4.1 Perencanaan (Planning)

Tahap perencanaan merupakan tahapan yang pertama kali dilewati ketika menggunakan metode pengembangan *eXtreme Programming* (XP). Tahapan ini digunakan untuk mendapatkan semua kebutuhan pengguna dan hasil analisa dari kebutuhan pengguna tersebut umumnya berupa *user story*. Adapun beberapa tahapan dan hasil dari tahapan ini yang dapat dilihat sebagai berikut:

3.4.1.1 Wawancara

Pada penelitian tugas akhir ini, informasi-informasi yang dibutuhkan diperoleh dengan cara melakukan wawancara dengan ibu Adelia Enjelina Matondang, S.T., M.T. selaku dosen arsitektur ITERA. Pertanyaan dan jawaban dari wawancara yang dilakukan dapat dilihat pada Tabel 3. 1 berikut.

Tabel 3. 1 Hasil Wawancara

| No | Pertanyaan | Jawaban |
|----|--|---|
| 1. | Apakah terdapat permasalahan dalam proses penilaian suatu mata kuliah pada program studi arsitektur? | Terdapat permasalahan berupa jumlah dosen dan jumlah mahasiswa yang tidak berimbang sehingga memiliki kemungkinan untuk terjadinya kesalahan dalam penilaian. Salah satu masalah ini dapat dilihat pada mata kuliah Seni Dalam Arsitektur yang memerlukan mahasiswa untuk membuat beberapa macam desain yang salah satunya berupa desain tampak depan fasad bangunan. |
| 2. | Apa saja pengaruh dari perbedaan jumlah dosen dan mahasiswa yang tidak berimbang? | Perbedaan jumlah dosen dengan jumlah mahasiswa yang signifikan menyebabkan terjadinya sebuah <i>human errors</i> dikarenakan keterbatasan tenaga dosen. Keterbatasan tersebut menyebabkan |

| No | Pertanyaan | Jawaban |
|----|---|---|
| | | menurunnya tingkat objektivitas dosen dalam melakukan penilaian. |
| 3. | Apakah terdapat suatu perbedaan standar dalam penilaian antara dosen satu dengan lainnya? | Terdapat suatu perbedaan standar antara dosen satu dengan lainnya sehingga waktu yang dibutuhkan untuk melakukan penilaian pun berbeda. |
| 4. | Apakah waktu yang lama yang dibutuhkan oleh dosen dalam melakukan penilaian mempengaruhi hasil dari penilaian yang ada? | Waktu yang lama umumnya dikarenakan penilaian dilakukan dengan ketelitian yang tinggi serta adanya perbedaan standar antara dosen dengan lainnya. Namun, proses penilaian tersebut jika dilakukan berulang kali dapat menghabiskan banyak tenaga seorang dosen sehingga performa dalam penilaian berkurang. |
| 5. | Berdasarkan dari permasalahan yang disampaikan, apakah ada ketentuan khusus pada media solusi yang dibutuhkan? Ketentuan ini dapat berupa mudah digunakan, bersifat fleksibel dan lain sebagainya. | Media solusi yang dibutuhkan diharapkan dapat diakses dengan cepat dan digunakan dengan mudah. Media solusi setidaknya juga dapat melakukan pemotongan/ <i>cropping</i> sehingga media penilaian dapat lebih fokus pada desain yang akan dinilai. |

3.4.1.2 User Story

User story merupakan hasil analisis dari kebutuhan fungsional maupun kebutuhan non-fungsional yang belum diklasifikasikan. *User story* ini nantinya yang akan dikelompokkan kedalam kebutuhan fungsional dan kebutuhan non-fungsional. *User story* dari hasil wawancara yang dilakukan dapat dilihat pada Tabel 3. 2 berikut.

Tabel 3. 2 User Story

| No | Role | Goal/Objective | Benefit/Result/Reason |
|----|-------|--|---|
| 1. | Dosen | Mengambil gambar desain fasad bangunan. | Melakukan pemotongan (<i>cropping</i>) pada gambar desain fasad bangunan. |
| 2. | Dosen | Mengambil gambar desain fasad bangunan. | Mendapatkan hasil penilaian desain fasad bangunan. |
| 3. | Dosen | Memperoleh hasil penilaian gambar desain fasad bangunan. | Mendapatkan hasil penilaian yang konsisten pada desain yang sama. |
| 4. | Dosen | Melakukan penilaian dengan cara yang lebih mudah. | Penilaian dapat dilakukan kapan saja dan dimana saja. |

3.4.1.3 Kebutuhan Fungsional

Berdasarkan dari *user story* yang diperoleh, diperolehlah beberapa kebutuhan fungsional dari aplikasi *mobile* yang hendak dikembangkan. Kebutuhan fungsional aplikasi mobile dapat dilihat pada Tabel 3. 3 berikut.

Tabel 3. 3 Kebutuhan Fungsional

| ID | Deskripsi |
|------|--|
| F-01 | Aplikasi <i>mobile</i> dapat menerima masukan berupa gambar desain fasad depan bangunan dengan format JPG. |
| F-02 | Aplikasi <i>mobile</i> dapat melakukan pemotongan (<i>cropping</i>) pada masukan gambar desain fasad depan bangunan. |
| F-03 | Aplikasi <i>mobile</i> dapat melakukan penilaian otomatis terhadap masukan gambar desain fasad depan bangunan. |

3.4.1.4 Kebutuhan Non-Fungsional

Berdasarkan dari *user story* yang diperoleh, diperolehlah beberapa kebutuhan non-fungsional dari aplikasi *mobile* yang hendak dikembangkan. Kebutuhan non-fungsional aplikasi mobile dapat dilihat pada Tabel 3. 4 berikut.

Tabel 3. 4 Kebutuhan Non-Fungsional

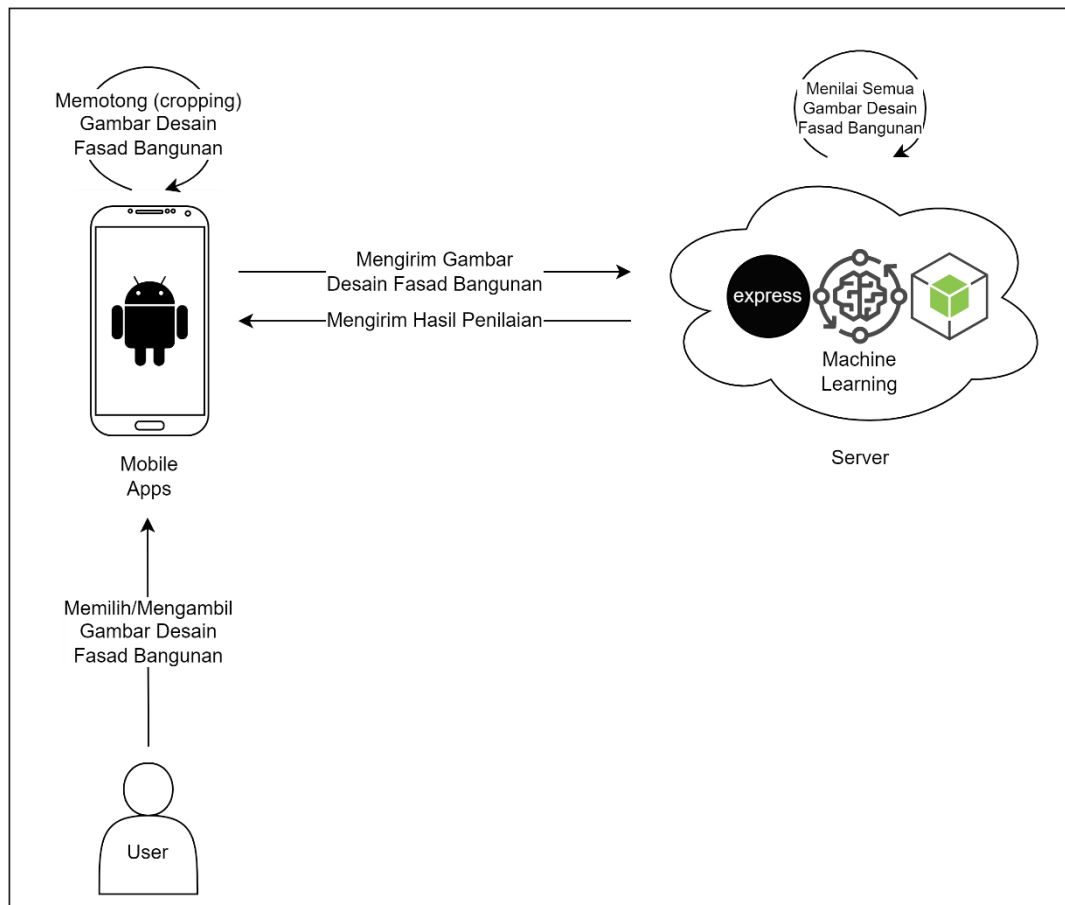
| ID | Parameter | Deskripsi |
|-------|--------------------|---|
| NF-01 | <i>Portability</i> | Aplikasi <i>mobile</i> dapat dijalankan pada perangkat <i>mobile</i> dengan sistem operasi <i>android</i> . |
| NF-02 | <i>Reliability</i> | Aplikasi dapat memberikan hasil penilaian yang konsisten pada gambar desain arsitektur yang sama. |

3.4.2 Perancangan (Design)

Tahap perancangan merupakan tahapan dimana dilakukannya berbagai perancangan yang akan digunakan dikemudian hari sebagai acuan dalam proses pengembangan. Tahapan ini akan menghasilkan beberapa diagram seperti *architecture diagram*, *use case diagram*, dan *activity diagram*. Selain itu, tahapan ini juga akan menghasilkan rancangan alur proses penggunaan serta rancangan antarmuka aplikasi *mobile*.

3.4.2.1 Architecture Diagram

Architecture diagram merupakan sebuah diagram yang ditujukan untuk dapat memberikan gambaran umum proses yang akan terjadi pada aplikasi *mobile* yang dikembangkan. Menggunakan diagram arsitektur, dapat dilihat komponen apa saja penyusun sistem yang akan dikembangkan pada penelitian serta proses apa saja yang dapat terjadi. Selain itu, diagram tersebut juga dapat digunakan sebagai referensi komponen apa saja yang perlu dilengkapi pada tahap pengembangan.



Gambar 3. 3 Architecture Diagram

Server yang ada pada Gambar 3. 3 merupakan *server Node Js* yang dikembangkan menggunakan *framework Express*. *Server* tersebut hanya akan menerima permintaan melalui *RESTful API* dan akan mengembalikan hasil permintaan dalam bentuk *JSON*. *Server* pada penelitian tugas akhir ini berfungsi sebagai penghubung antara aplikasi *mobile* dengan *model machine learning* sehingga proses penilaian otomatis dapat dilakukan.

Merujuk pada Gambar 3. 3 aplikasi *mobile* akan dikembangkan menggunakan *framework React Native* untuk mempercepat proses pengembangan kedalam bentuk komponen-komponen dan bahasa pemrograman yang digunakan ialah *TypeScript* sehingga proses *debugging* menjadi lebih mudah. Pada penggunaannya, pengguna aplikasi *mobile* dapat mengambil gambar desain fasad bangunan melalui kamera perangkat maupun melalui gambar desain yang telah tersimpan pada memori perangkat. Semua gambar yang telah diambil/dipilih tersebut akan dipotong

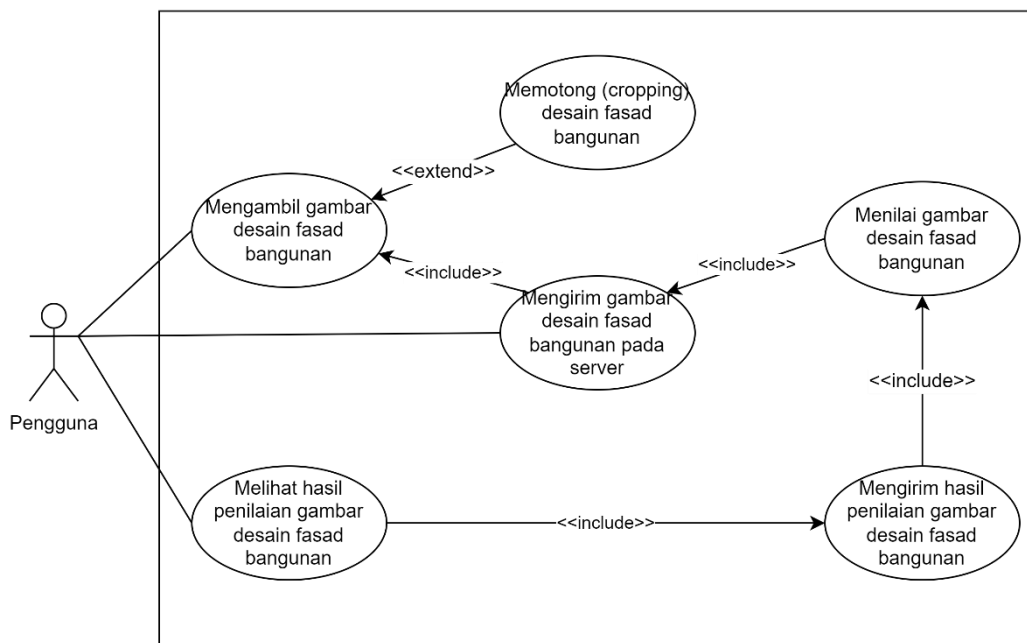
(*cropping*) melalui aplikasi *mobile* dan kemudian dikirimkan pada *server* untuk melakukan penilaian otomatis melalui *endpoint RESTful API* yang telah dikembangkan. Aplikasi *mobile* yang telah selesai mengirimkan gambar desain akan menunggu *response* dari *server* berupa hasil penilaian dalam bentuk *JSON*.

Server sebagai penghubung antara *model machine learning* dengan aplikasi *mobile* akan memasukan gambar pada *model machine learning* hingga semua gambar desain telah dinilai. Ketika *model* telah selesai menilai semua gambar desain maka *server* akan mengembalikan permintaan penilaian dalam bentuk *JSON* sebagai *response* dari permintaan *RESTful API* yang telah dilakukan oleh aplikasi *mobile*. Hasil penilaian yang dikembalikan oleh *server* akan ditampilkan bersamaan dengan gambar desain fasad bangunan yang dinilai.

3.4.2.2 Use Case Diagram

Use case diagram merupakan sebuah diagram yang ditujukan untuk menggambarkan interaksi apa saja yang dapat terjadi pada sebuah sistem. Pada Gambar 3. 4 dapat dilihat bahwa pengguna dapat mengambil gambar dan melihat hasil penilaian gambar desain fasad bangunan. Pada *use case diagram* tersebut, pengguna yang dimaksud adalah dosen arsitektur yang hendak melakukan penilaian pada sebuah desain fasad bangunan. Selain itu, pengguna dapat melakukan sebuah aksi dimana setelah pengguna mengambil gambar desain fasad bangunan melalui kamera perangkat maupun melalui gambar desain yang telah tersimpan pada memori perangkat, pengguna dapat melakukan pemotongan (*cropping*) pada gambar yang telah dipilih/diambil tersebut. Gambar-gambar yang telah dipotong tersebut kemudian akan dikirimkan pada *server* sehingga dapat dinilai menggunakan *model machine learning*.

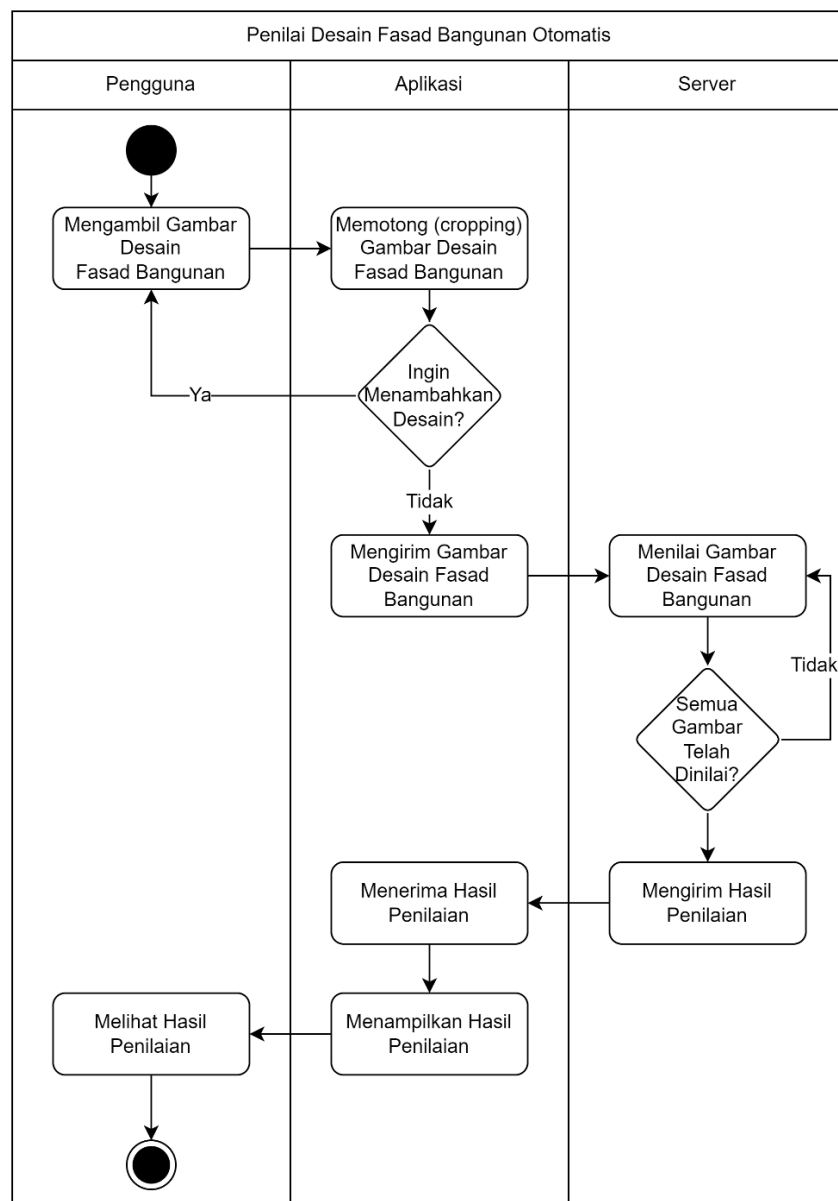
Server selaku salah satu aktor pada sistem dapat melakukan sebuah aksi berupa menilai gambar desain fasad bangunan. Desain yang telah selesai dinilai menggunakan *model machine learning* kemudian akan dikirimkan kembali pada pengguna (aplikasi *mobile*). Demikian, pengguna dapat melihat hasil penilaian ketika *server* telah mengembalikan permintaan penilaian dari pengguna melalui aplikasi (*mobile*).



Gambar 3. 4 Use Case Diagram

3.4.2.3 Activity Diagram

Activity diagram merupakan sebuah diagram yang ditujukan untuk menggambarkan alur aktivitas dari sebuah sistem. Pada Gambar 3. 5 dapat dilihat bahwa proses diawali dari pengguna yang mengambil gambar desain fasad bangunan melalui kamera perangkat ataupun melalui gambar desain yang telah tersimpan pada memori perangkat. Proses pengambilan gambar desain dapat dilakukan oleh pengguna beberapa kali dengan tiap pengambilan aplikasi akan memperlihatkan menu untuk melakukan pemotongan (*cropping*) gambar.

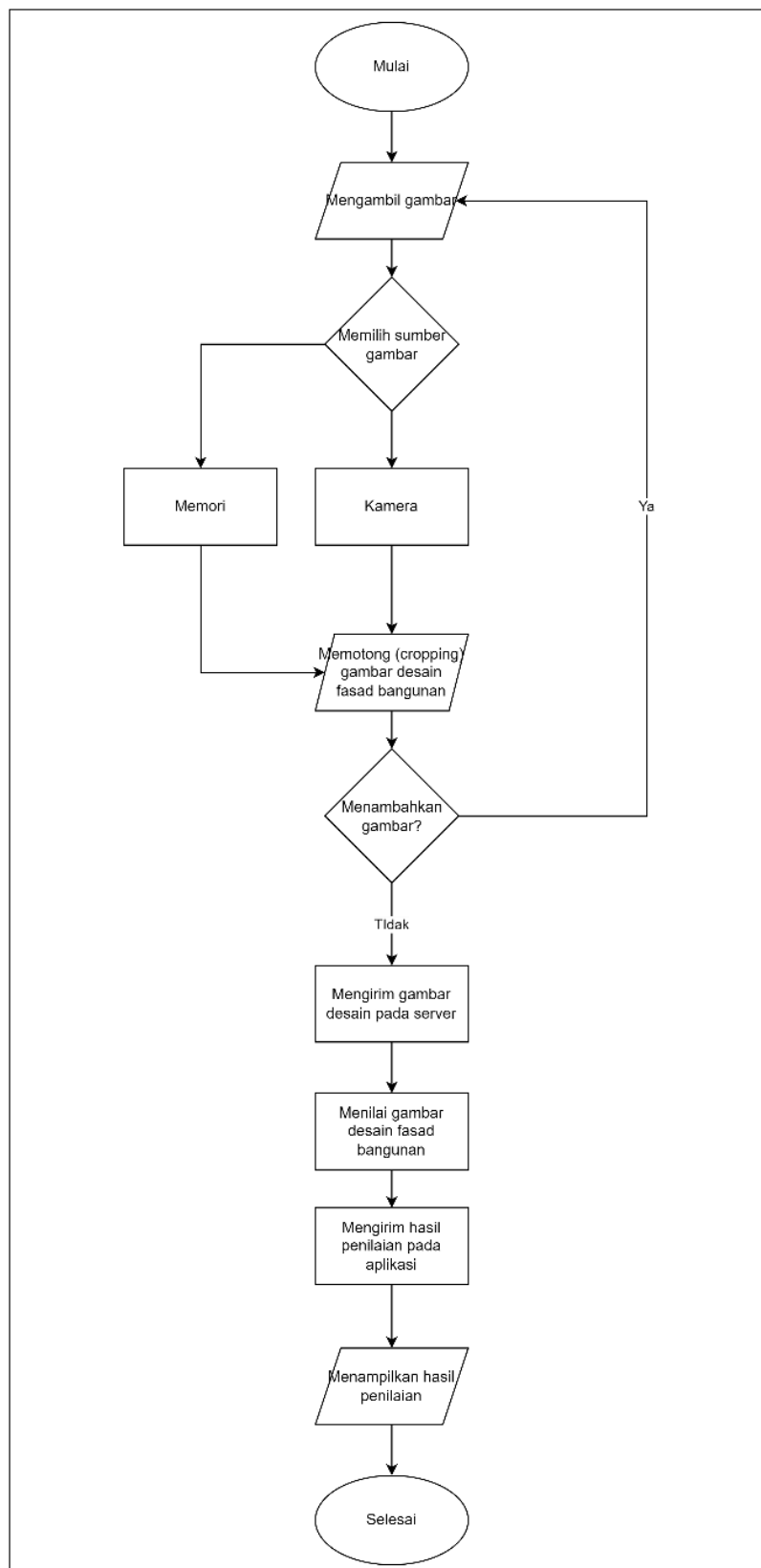


Gambar 3. 5 Activity Diagram

Pengguna yang telah merasa cukup memasukkan gambar yang akan dinilai, aplikasi akan mengirimkan semua gambar desain tersebut pada *server* yang kemudian akan dilanjutkan penilaian oleh *model machine learning*. *Model* tersebut akan melakukan penilaian pada semua gambar yang dikirimkan oleh pengguna melalui aplikasi *mobile*. Setelah selesai melakukan penilaian, *server* akan mengirimkan hasil penilaian kepada aplikasi sehingga pengguna dapat melihat semua hasil penilaian gambar desain fasad bangunan.

3.4.2.4 Rancangan Alur Proses Penggunaan

Pada saat tahap pengembangan telah selesai, penggunaan aplikasi *mobile* akan memiliki alur yang dapat dilihat pada Gambar 3. 6. Pada gambar tersebut, pengguna hanya akan memberikan masukan berupa gambar desain fasad bangunan yang dapat dimasukan dengan cara memilih gambar yang ada pada memori maupun mengambil gambar melalui kamera dan melakukan pemotongan (*cropping*) pada gambar yang dimasukan. Setelah memberikan masukan-masukan tersebut, semua proses akan dilakukan secara otomatis dengan mengirimkan gambar desain pada *server* dan menunggu hasil penilaian ditampilkan pada aplikasi *mobile*. Hasil penilaian tersebut akan ditampilkan bersamaan/disandingkan dengan desain fasad bangunan yang dinilai.



Gambar 3. 6 Alur Proses Penggunaan

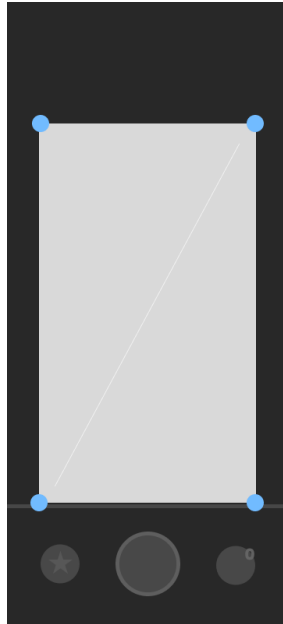
3.4.2.5 Rancangan Antarmuka (Interface)

Sebuah aplikasi *mobile* membutuhkan sebuah antarmuka sehingga pengguna dapat berinteraksi dengan aplikasi lebih cepat dan lebih mudah. Pada tahap perancangan (*planning*), antarmuka aplikasi dirancang yang kemudian nantinya akan digunakan sebagai acuan dalam mengembangkan aplikasi. Adapun empat tampilan utama pada aplikasi yang dapat dilihat pada Gambar 3. 7, Gambar 3. 8, Gambar 3. 9, dan Gambar 3. 10 berikut.



Gambar 3. 7 Layar Utama Aplikasi

Pada saat pengguna membuka aplikasi, pengguna akan mendapatkan antarmuka yang tergambarkan pada Gambar 3. 7. Pada antarmuka ini pengguna dapat melakukan pengambilan gambar desain melalui kamera *smartphone android* maupun mengambil gambar desain yang telah tersimpan pada memori perangkat *android*. Selain itu, jumlah dari gambar desain yang akan dinilai dapat dilihat oleh pengguna pada angka yang ada pada disebelah tombol mengambil gambar.



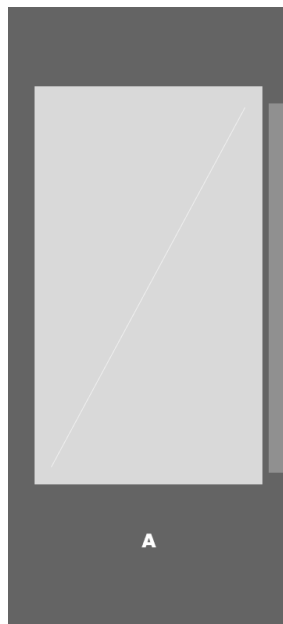
Gambar 3. 8 Layar Pemotongan (*Cropping*) Desain

Pada saat pengguna mengambil sebuah gambar desain menggunakan kamera maupun mengambil gambar desain yang telah ada pada *smartphone*, antarmuka Gambar 3. 8 akan ditampilkan pada pengguna. Pada antarmuka ini, pengguna dapat melakukan pemotongan (*cropping*) gambar desain sehingga gambar yang akan dinilai menjadi lebih terpusat pada objek penilaian desain fasad bangunan. Demikian, pemotongan pada gambar desain dapat mengurangi kemungkinan untuk terjadinya kesalahan penilaian yang dilakukan oleh *machine learning*.

Pada saat pengguna telah memasukan gambar desain fasad bangunan yang akan dinilai maka aplikasi akan melakukan proses penilaian otomatis. Pada saat proses penilaian tersebut telah selesai dilakukan, pengguna akan melihat antarmuka Gambar 3. 9 yang berisikan daftar gambar desain fasad bangunan yang telah dinilai. Ketika pengguna memilih salah satu gambar desain fasad yang ada pada daftar maka pengguna akan melihat tampilan antarmuka Gambar 3. 10. Pada Gambar 3. 10, pengguna dapat melihat nilai dari suatu desain fasad bangunan beserta desain fasad bangunan yang dinilai tersebut.



Gambar 3. 9 Daftar Desain Fasad Bangunan yang Dinilai



Gambar 3. 10 Hasil Penilaian Desain Fasad Bangunan

3.4.3 Pengembangan (Coding)

Tahap pengembangan merupakan tahapan dimana semua rancangan yang telah dirancang diimplementasikan menjadi sebuah aplikasi *mobile* yang siap digunakan. Pada tugas akhir ini, tahap pengembangan akan dibagi menjadi dua tahapan berupa tahap pengembangan *server* dan tahap pengembangan aplikasi *mobile*. Pemecehan

tahapan tersebut dikarenakan aplikasi *mobile* akan menggunakan sebuah *server* yang akan memproses semua permintaan penilaian desain fasad bangunan. Oleh sebab itu, tahap pengembangan akan diawali pada pengembangan *server* yang kemudian akan dilanjutkan pada pengembangan aplikasi *mobile*. Selain itu, jika dibutuhkannya perubahan kode program pada *server* ketika mengembangkan aplikasi *mobile* maka akan dilakukan sebuah *refactoring* sehingga *server* dapat digunakan oleh *mobile*.

3.4.4 Validasi

Tahap validasi merupakan tahapan yang akan dilalui setelah tahapan pengembangan telah selesai. Pada tahapan ini, kode program yang telah dikembangkan akan diuji menggunakan dua metode pengujian yakni *unit testing* dan *acceptance testing*. Pengujian dilakukan agar dapat memastikan apakah kode program yang dikembangkan tidak memiliki *bugs* dan kesalahan serta dapat memenuhi kebutuhan fungsional maupun kebutuhan non-fungsional. Pada penelitian tugas akhir ini, pada saat pengembangan *server* pengujian *acceptance testing* tidak akan dilakukan. Hal ini dikarenakan tidak memungkinkannya pengguna untuk menguji sebuah sistem yang belum memiliki antarmuka.

3.4.5 Perbaikan (Adjustment)

Tahap perbaikan merupakan tahapan yang hanya akan dilalui jika kode program tidak bisa melewati rangkaian pengujian yang telah didefinisikan. Tahapan ini akan terus dilewati jika setelah melewati perbaikan dan kode program tetap tidak bisa melewati proses pengujian. Demikian, kode program yang dikembangkan dapat dipastikan berfungsi dengan baik dan sesuai dengan semua kebutuhan yang telah didefinisikan.

3.4.6 Peningkatan (Incremental)

Tahapan peningkatan merupakan tahapan yang hanya akan dilalui jika fase pengembangan masih belum berada pada fase pengembangan terakhir. Tahapan ini akan mengulang tahapan pengembangan pada tahapan perencanaan dengan fitur/fungsi baru yang akan dikembangkan. Demikian, pengembangan menggunakan XP akan terdiri dari beberapa iterasi yang terdiri dari fitur/fungsi yang telah teruji dengan baik.

3.4.7 Pengerahan (Deployment)

Tahap pengerahan merupakan tahapan terakhir pada SDLC *eXtreme Programming* (XP). Tahapan ini hanya akan dilewati jika tidak terjadi peningkatan lagi pada tahap pengembangan. Demikian, proses pengembangan aplikasi *mobile* dapat dikatakan telah selesai dan telah sesuai dengan semua kebutuhan pengguna serta telah teruji dengan baik.

3.5 Rancangan Pengujian

Pada penelitian tugas akhir ini, terdapat beberapa metode pengujian yang akan digunakan. Metode pengujian yang digunakan ialah *unit test*, *acceptance test*, dan pengujian efektifitas penggunaan aplikasi *mobile* dalam melakukan penilaian. Adapun rincian dari rancangan pengujian yang akan digunakan pada penelitian tugas akhir ini sebagai berikut:

3.5.1 Unit Testing

Pada tahap pengembangan menggunakan metode pengembangan *eXtreme Programming* (XP), terdapat dua macam pengembangan berupa *server* untuk melakukan penilaian dan aplikasi *mobile* sebagai perantara untuk melakukan penilaian otomatis. Demikian, pengujian menggunakan *unit test* terbagi mejadi dua bagian berupa *server* dan *mobile*. Adapun rangkaian pengujian yang telah dirancang sedemikian rupa sebagai berikut:

3.5.1.1 Server

Pengujian *server* yang dikembangkan menggunakan *unit test* ditujukan untuk memastikan fungsi/fitur yang dikembangkan dapat berjalan dengan baik. Demikian, jika terjadi perubahan pada *source code* maka rangkaian pengujian tersebut dapat digunakan sebagai rujukan untuk memastikan fungsi/fitur yang dilakukan perubahan dapat berjalan sebagaimana mestinya. Rangkaian pengujian menggunakan metode *unit test* pada sisi *server* dapat dilihat pada Tabel 3. 5 berikut.

Tabel 3. 5 Server User Testing

| No | Method | Input | Output |
|----|-------------------------|--|---|
| 1. | uploadMw(req,res,next) | Kumpulan gambar desain fasad bangunan. | Gambar desain fasad bangunan disimpan pada <i>server</i> sementara. |
| 2. | predictMw(req,res,next) | Kumpulan gambar desain fasad bangunan. | Fungsi predictDesign terpanggil. |
| 3. | predictDesign(files) | Kumpulan gambar desain fasad bangunan. | Kumpulan nilai gambar desain fasad bangunan. |
| 4. | returnMw(req,res) | Kumpulan nilai gambar desain fasad bangunan. | Kumpulan nilai gambar desain fasad bangunan (<i>JSON response</i>). |

Pada Tabel 3. 5 dapat dilihat bahwa terdapat empat rangkaian pengujian yang akan diujikan pada *server* yang dikembangkan. Pada Tabel 3. 5 terdapat fungsi-fungsi yang akan diujikan dengan masukan dan keluaran yang diharapkan. Menggunakan *unit test*, fungsi tersebut akan diberikan masukan yang ada pada Tabel 3. 5 dan jika fungsi tersebut memberikan keluaran yang sama pada keluaran yang diharapkan maka fungsi tersebut dapat dinyatakan berjalan dengan baik dan sesuai dengan yang diharapkan.

3.5.1.2 Mobile

Pengujian *mobile* yang dikembangkan menggunakan *unit test* ditujukan untuk memastikan fungsi/fitur yang ada pada aplikasi *mobile* dapat berjalan sebagaimana mestinya. Jika terjadi sebuah perubahan pada *source code* maka rangkaian pengujian akan berfungsi untuk memastikan keluaran/hasil dari fungsi yang diujikan berfungsi dengan baik atau tidak. Rangkaian pengujian menggunakan metode *unit test* pada sisi *mobile* dapat dilihat pada Tabel 3. 6 berikut.

Tabel 3. 6 Mobile User Testing

| No | Method | Input | Output |
|----|-----------------|-------------------------------|---|
| 1. | setDesign(file) | Gambar desain fasad bangunan. | Kumpulan gambar desain fasad bangunan dengan salah satu <i>item</i> |

| No | Method | Input | Output |
|----|------------------------------|---|--|
| | | | berupa gambar desain yang dimasukan. |
| 2. | updateDesign(filename, file) | Nama <i>file</i> yang tersimpan pada <i>state</i> dan gambar desain fasad bangunan. | Kumpulan gambar desain fasad bangunan yang telah diperbarui. |
| 3. | deleteDesign(filename) | Nama <i>file</i> gambar desain fasad bangunan. | Kumpulan gambar desain fasad bangunan tanpa adanya <i>item</i> dengan nama yang sama dengan masukan. |
| 4. | predictDesign (files) | Kumpulan gambar desain fasad bangunan. | Kumpulan nilai gambar desain fasad bangunan (<i>JSON response</i>). |

Pada Tabel 3. 6 dapat dilihat bahwa terdapat empat rangkaian pengujian yang akan diujikan pada aplikasi *mobile* yang dikembangkan. Pada rangkaian pengujian tersebut terdapat beberapa fungsi yang akan diujikan dengan memberikan masukan dan luaran dari fungsi tersebut yang akan dibandingkan dengan luaran yang diharapkan. Fungsi-fungsi tersebut akan diujikan menggunakan *unit test* dan jika fungsi tersebut memberikan luaran yang diharapkan maka fungsi tersebut dapat dinyatakan berjalan dengan baik dan sesuai dengan yang diharapkan.

3.5.2 Acceptance Testing

Acceptance testing merupakan pengujian yang dilakukan untuk memastikan aplikasi yang dikembangkan telah memenuhi kebutuhan-kebutuhan fungsional/non-fungsional atau tidak. Pada penelitian tugas akhir ini, *acceptance testing* hanya akan dilakukan setelah aplikasi *mobile* telah selesai dikembangkan. Hal ini dikarenakan pada pengembangan *server*, pengguna tidak memiliki sebuah antarmuka (*interface*) yang memungkinkan untuk melakukan pengujian pada *server* yang dikembangkan. Rangkaian pengujian *acceptance testing* dapat dilihat pada Tabel 3. 7 berikut.

Tabel 3. 7 Acceptance Testing

| No | Acceptance Requirements |
|----|--|
| 1. | Aplikasi <i>mobile</i> dapat menerima masukan berupa gambar desain fasad depan bangunan dengan format JPG. |
| 2. | Aplikasi <i>mobile</i> dapat melakukan pemotongan (<i>cropping</i>) pada masukan gambar desain fasad depan bangunan. |
| 3. | Aplikasi <i>mobile</i> dapat melakukan penilaian otomatis terhadap masukan gambar desain fasad depan bangunan. |

3.5.3 Pengujian Efektivitas

Pengujian efektivitas merupakan sebuah rangkaian pengujian yang ditujukan untuk memastikan apakah aplikasi *mobile* dapat menyelesaikan masalah yang ditemukan atau tidak. Pengujian ini dilakukan dengan cara membandingkan jumlah waktu yang dibutuhkan serta keakuratan aplikasi dalam melakukan penilaian. Demikian, berdasarkan hasil perbandingan tersebut dapat ditarik kesimpulan apakah aplikasi yang dikembangkan dapat menyelesaikan permasalahan yang ditemukan atau tidak.

3.5.3.1 Waktu Penilaian

Pengujian waktu penilaian merupakan pengujian yang dilakukan untuk mengetahui apakah aplikasi dapat mengurangi waktu yang dibutuhkan untuk menilai sebuah desain fasad bangunan atau tidak. Pengujian akan dilakukan menggunakan konsep membandingkan yang dilakukan oleh Ojala et al. [41] dalam penelitiannya. Pengujian akan dilakukan dengan beberapa dosen menilai sebuah desain fasad bangunan dan kemudian waktu yang dibutuhkan oleh dosen akan dibandingkan dengan waktu yang dibutuhkan oleh aplikasi dalam melakukan penilaian. Proses membandingkan waktu akan dilakukan pada semua hasil penilaian yang dilakukan oleh dosen yang ikut berpartisipasi. Adapun tabel perbandingan yang akan digunakan pada pengujian ini pada Tabel 3. 8 berikut.

Tabel 3. 8 Perbandingan Waktu Penilaian

| Desain ke-i | Aplikasi (s) | Dosen 1 (s) | Dosen 2 (s) | Dosen 3 (s) | Dosen 4 (s) | Dosen 5 (s) |
|-------------|----------------|------------------|------------------|------------------|------------------|------------------|
| 1 | A ₁ | D _{1,1} | D _{1,2} | D _{1,3} | D _{1,4} | D _{1,5} |

| Desain ke-i | Aplikasi (s) | Dosen 1 (s) | Dosen 2 (s) | Dosen 3 (s) | Dosen 4 (s) | Dosen 5 (s) |
|-------------|--------------|-------------|-------------|-------------|-------------|-------------|
| 2 | A_2 | $D_{2,1}$ | $D_{2,2}$ | $D_{2,3}$ | $D_{2,4}$ | $D_{2,5}$ |
| ... | ... | ... | ... | ... | ... | ... |
| 10 | A_{10} | $D_{10,1}$ | $D_{10,2}$ | $D_{10,3}$ | $D_{10,4}$ | $D_{10,5}$ |

Keterangan:

A_i : waktu aplikasi dalam menilai desain ke-i (s)

$D_{i,j}$: waktu dosen ke-j dalam menilai desain ke-i (s)

Pada Tabel 3. 8 dapat dilihat bahwa pengujian akan dilakukan sebanyak tiga kali dengan tiga desain fasad bangunan yang berbeda. Pengujian dilakukan dengan lima dosen arsitektur akan menilai tiga desain fasad bangunan berbeda. Pemilihan lima dosen sebagai pembanding waktu dilakukan dengan mengikuti rekomendasi yang disebutkan oleh Yusoff [42], yakni lima hingga delapan orang ahli. Setiap waktu yang dibutuhkan oleh tiap dosen dalam menilai suatu desain fasad akan di tabulasikan kedalam tabel. Layaknya yang dilakukan oleh dosen yang berpartisipasi, pengujian pada aplikasi akan dilakukan dengan cara mencatat waktu yang dibutuhkan oleh aplikasi dalam menilai tiap desain fasad bangunan yang ada. Setelah mengisi Tabel 3. 8, pengujian dilanjutkan dengan menghitung berapa banyak kejadian dimana waktu yang dibutuhkan A_i lebih kecil dibandingkan dengan D_{ij} . Proses pengujian akan diakhiri dengan penarikan kesimpulan apakah aplikasi dapat menilai desain fasad bangunan dengan waktu yang lebih cepat dibandingkan dengan dosen yang berpartisipasi atau tidak.

3.5.3.2 Hasil Penilaian

Pengujian hasil penilaian merupakan pengujian yang dilakukan untuk memastikan apakah aplikasi dapat memberikan hasil penilaian mendekati maupun menyerupa penilaian dosen atau tidak. Pengujian ini dilakukan menggunakan konsep membandingkan yang dilakukan oleh Ojala et al. [41] dalam penelitiannya. Pengujian akan dilakukan dengan beberapa dosen menilai sebuah desain fasad bangunan dan kemudian hasil penilaian dosen tersebut dibandingkan dengan hasil penilaian yang dilakukan oleh aplikasi. Proses membandingkan hasil penilaian akan dilakukan pada

semua hasil penilaian yang dilakukan oleh dosen yang ikut berpartisipasi. Adapun tabel perbandingan yang akan digunakan pada pengujian ini pada Tabel 3. 9 berikut.

Tabel 3. 9 Perbandingan Hasil Penilaian

| Desain ke-i | Aplikasi | Dosen 1 | Dosen 2 | Dosen 3 | Dosen 4 | Dosen 5 |
|--------------------|-----------------|----------------|----------------|----------------|----------------|----------------|
| 1 | A_1 | $D_{1,1}$ | $D_{1,2}$ | $D_{1,3}$ | $D_{1,4}$ | $D_{1,5}$ |
| 2 | A_2 | $D_{2,1}$ | $D_{2,2}$ | $D_{2,3}$ | $D_{2,4}$ | $D_{2,5}$ |
| ... | ... | ... | ... | ... | ... | ... |
| 10 | A_{10} | $D_{10,1}$ | $D_{10,2}$ | $D_{10,3}$ | $D_{10,4}$ | $D_{10,5}$ |

Keterangan:

A_i : hasil penilaian aplikasi dalam menilai desain ke-i

$D_{i,j}$: hasil penilaian dosen ke-j dalam menilai desain ke-i

Pada Tabel 3. 9 dapat dilihat bahwa pengujian akan dilakukan sebanyak tiga kali dengan tiga desain fasad bangunan yang berbeda. Pengujian dilakukan dengan lima dosen arsitektur akan menilai tiga desain fasad bangunan berbeda. Pemilihan lima dosen sebagai pembanding nilai dikarenakan Yusoff [42] dalam penelitiannya menyebutkan jumlah ahli yang direkomendasikan ialah lima hingga delapan ahli. Setiap hasil penilaian yang diberikan oleh dosen akan dikonversikan kedalam nilai mutu jika dosen memberikan nilai dengan rentang 0 s/d 100. Demikian, hasil penilaian yang ada pada Tabel 3. 9 terdiri dari nilai mutu yang siap untuk dibandingkan dengan hasil penilaian yang diberikan oleh aplikasi. Pengujian dilanjutkan dengan menghitung berapa banyak kejadian dimana nilai yang diberikan oleh A_i sama dengan D_{ij} . Proses pengujian akan diakhiri dengan penarikan kesimpulan apakah aplikasi dapat menilai desain fasad bangunan layaknya penilaian yang diberikan oleh dosen atau tidak.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil Penelitian

Dijabarkan dalam bentuk pseudocode dan dijelaskan per bagian.

Bagi yang membuat alat dijelaskan alat yang jadi dalam bentuk apa.

Bagi yang membuat aplikasi dijelaskan aplikasi yang jadi dalam bentuk seperti apa.

4.2 Hasil Pengujian

Beri hasil pengujian dari poin rancangan pengujian Bab 3.

4.3 Analisis Hasil Penelitian

Berisi analisis hasil penelitian, berupa data yang didapatkan dari pengerjaan tugas akhir yang sudah Anda kerjakan

4.3.1 Analisis Hasil Data 1

Pastikan penggunaan tabel juga menggunakan cross-reference seperti Tabel *Error! No text of specified style in document..1*, berikut:

Tabel *Error! No text of specified style in document..1* Tabel sama seperti gambar, penjelasan diberikan caption

| Pengujian | Metode 1 | Metode 2 |
|-----------|----------|----------|
| Kecepatan | 10 ms | 12 ms |
| Memory | 10 mb | 10 mb |

Tabel yang Panjang dan melebihi 1 halaman, untuk header wajib menggunakan repeat header.

4.3.2 Analisis Hasil Data 2

Berisi data lainnya yang sudah didapatkan, dapat berupa:

1. Hasil pengujian
2. Hasil kuesioner
3. Aplikasi yang dikembangkan
4. UI / UX yang dikembangkan

4.4 Pembahasan

Berisi pembahasan terkait hasil yang sudah didapatkan / dipaparkan sebelumnya, berupa penutup yang dapat menjelaskan mengenai kelebihan hasil tugas akhir dan kekurangannya dibandingkan dengan penelitian atau produk lain yang serupa atau mirip. Penulis dapat menggunakan tabel untuk mempermudah perbandingan dan kemudian menjelaskannya.

4.5 Pengujian

Berisi hasil analisis pengujian.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berisi kesimpulan dari hasil dan pembahasan terkait penelitian yang dilakukan, dapat juga berupa temuan yang Anda dapatkan setelah melakukan penelitian atau analisis terhadap tugas akhir Anda. Berhubungan dengan poin pada rumusan masalah dan tujuan.

5.2 Saran

Berisi saran mengenai aspek tugas akhir atau temuan yang dapat dikembangkan dan diperkaya di tugas akhir selanjutnya.

DAFTAR PUSTAKA

- [1] R. N. Putri, “Indonesia dalam Menghadapi Pandemi Covid-19,” *Jurnal Ilmiah Universitas Batanghari Jambi*, vol. 20, no. 2, hlm. 705, Jul 2020, doi: 10.33087/jiubj.v20i2.1010.
- [2] M. N. I. Opu, M. R. Islam, M. A. Kabir, M. S. Hossain, dan M. M. Islam, “Learn2Write: Augmented Reality and Machine Learning-Based Mobile App to Learn Writing,” *Computers*, vol. 11, no. 1, 2022, doi: 10.3390/computers11010004.
- [3] D. Anggraini dan D. H. Rahmi, “Karakteristik fasad bangunan Indis di kawasan jalan Prawitotaman Yogyakarta,” *ARTEKS : Jurnal Teknik Arsitektur*, vol. 4, no. 1, 2019, doi: 10.30822/arteks.v4i1.78.
- [4] M. Sethu *dkk.*, “Using Artificial Intelligence To Mitigate Human Factor Errors In Nuclear Power Plants: A Review,” 2021.
- [5] J. Alves, D. Moreira, P. Alves, L. Rosado, dan M. Vasconcelos, “Automatic focus assessment on dermoscopic images acquired with smartphones,” *Sensors (Switzerland)*, vol. 19, no. 22, 2019, doi: 10.3390/s19224957.
- [6] F. M. Garcia-Moreno, M. Bermudez-Edo, E. Rodríguez-García, J. M. Pérez-Mármol, J. L. Garrido, dan M. J. Rodríguez-Fórtiz, “A machine learning approach for semi-automatic assessment of IADL dependence in older adults with wearable sensors,” *Int J Med Inform*, vol. 157, 2022, doi: 10.1016/j.ijmedinf.2021.104625.
- [7] N. I. Mohammed, A. Jarde, G. Mackenzie, U. D’Alessandro, dan D. Jeffries, “Deploying Machine Learning Models Using Progressive Web Applications: Implementation Using a Neural Network Prediction Model for Pneumonia Related Child Mortality in The Gambia,” *Front Public Health*, vol. 9, 2022, doi: 10.3389/fpubh.2021.772620.
- [8] F. Awanda Alviansyah dan E. Ramadhani, “Implementasi Dynamic Application Security Testing pada Aplikasi Berbasis Android,” 2021.
- [9] L. P. Dewi, A. Noertjahyana, dan T. J. Wahono, “Development of an android document management,” dalam *IOP Conference Series: Materials Science and Engineering*, 2019, vol. 673, no. 1. doi: 10.1088/1757-899X/673/1/012050.

- [10] S. Al-Saqqa, S. Sawalha, dan H. Abdelnabi, "Agile software development: Methodologies and trends," *International Journal of Interactive Mobile Technologies*, vol. 14, no. 11, 2020, doi: 10.3991/ijim.v14i11.13269.
- [11] A. K. M. Z. Islam dan Dr. A. Ferworn, "A Comparison between Agile and Traditional Software Development Methodologies," *Global Journal of Computer Science and Technology*, 2020, doi: 10.34257/gjcstcvol20is2pg7.
- [12] I. B. Prayoga Bhiantara, G. Indrawan, dan K. Y. E. Aryanto, "Pengembangan Sistem Informasi Pariwisata Terintegrasi E-Ticket Mobile dengan Metode Extreme Programming (Studi Kasus Dinas Pariwisata Karangasem)," *Jurnal SISKOM-KB (Sistem Komputer dan Kecerdasan Buatan)*, vol. 5, no. 1, 2021, doi: 10.47970/siskom-kb.v5i1.227.
- [13] A. Sucipto, H. W. David, S. D. Riskiono, dan S. Ahdan, "SISTEM PEMELIHARAAN MENARA BTS (BASE TRANSCEIVER STATION) BERBASIS MOBILE," *Jurnal SAINTEKOM*, vol. 12, no. 1, 2022, doi: 10.33020/saintekom.v12i1.196.
- [14] P. Yang, Z. Liu, J. Xu, Y. Huang, dan Y. Pan, "An Empirical Study on the Ability Relationships between Programming and Testing," *IEEE Access*, vol. 8, 2020, doi: 10.1109/ACCESS.2020.3018718.
- [15] Noerlina, A. Chandra, dan T. N. Mursitama, "Sales Revenue Sharing Model using Dynamics NAV Modification in Health Industries," dalam *IOP Conference Series: Earth and Environmental Science*, Mar 2020, vol. 426, no. 1. doi: 10.1088/1755-1315/426/1/012162.
- [16] A. Anharudin, S. Siswanto, dan R. M. Syakira, "Rancang Bangun Data Storage System berbasis Web Dengan Metode Extreme Programming," *Jurnal Tekno Kompak*, vol. 16, no. 1, 2022, doi: 10.33365/jtk.v16i1.1454.
- [17] R. Fauzan, D. Siahaan, S. Rochimah, dan E. Triandini, "A Different Approach on Automated Use Case Diagram Semantic Assessment," *International Journal of Intelligent Engineering and Systems*, vol. 14, no. 1, 2021, doi: 10.22266/IJIES2021.0228.46.
- [18] R. S. Pressman, *Software Engineering A Practitioner's Approach 7th Ed - Roger S. Pressman*. 2009. doi: 10.1017/CBO9781107415324.004.
- [19] I. Sommerville, *Sommerville Software Engineering*, vol. 291. 2011.

- [20] S. Al-Fedaghi, "Validation: Conceptual versus Activity Diagram Approaches," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 6, 2021, doi: 10.14569/IJACSA.2021.0120632.
- [21] T. C. D. Lucas, T. M. Pollington, E. L. Davis, dan T. D. Hollingsworth, "Responsible modelling: Unit testing for infectious disease epidemiology," *Epidemics*, vol. 33, 2020. doi: 10.1016/j.epidem.2020.100425.
- [22] X. Guo, H. Okamura, dan T. Dohi, "Automated Software Test Data Generation With Generative Adversarial Networks," *IEEE Access*, vol. 10, 2022, doi: 10.1109/ACCESS.2022.3153347.
- [23] S. Tatale dan V. Chandra Prakash, "Enhancing acceptance test driven development model with combinatorial logic," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 10, 2020, doi: 10.14569/IJACSA.2020.0111036.
- [24] J. Bogner dan M. Merkel, "To Type or Not to Type? A Systematic Comparison of the Software Quality of JavaScript and TypeScript Applications on GitHub," dalam *IEEE International Conference on Program Comprehension*, 2022, vol. 2022-March. doi: 10.1145/nnnnnnnn.nnnnnnnn.
- [25] A. Wirfs-Brock dan B. Eich, "JavaScript: The first 20 years," *Proceedings of the ACM on Programming Languages*, vol. 4, no. HOPL, 2020, doi: 10.1145/3386327.
- [26] K. Ramírez Pulido, J. L. Ortega-Arjona, dan L. del Carmen González Huesca, "Gradual Typing Using Union Typing With Records," *Electron Notes Theor Comput Sci*, vol. 354, 2020, doi: 10.1016/j.entcs.2020.10.013.
- [27] K. Jesse, P. T. Devanbu, dan T. Ahmed, "Learning type annotation: Is big data enough?," dalam *ESEC/FSE 2021 - Proceedings of the 29th ACM Joint Meeting European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2021. doi: 10.1145/3468264.3473135.
- [28] G. S. M. Diyasa, G. S. Budiwitjaksono, H. Amarul Ma'rufi, dan I. Ade Widya Sampurno, "Comparative Analysis of Rest and GraphQL Technology on Nodejs-Based Api Development," 2021. doi: 10.11594/nstp.2021.0908.
- [29] M. Bawane, "A Review on Technologies used in MERN stack," *Int J Res Appl Sci Eng Technol*, vol. 10, no. 1, 2022, doi: 10.22214/ijraset.2022.39868.
- [30] Ruqiya, N. Islam, A. Rai, dan N. Khan, "Usability analysis of android and iOS operating systems," *International Journal of Engineering Trends and*

- Technology*, vol. 68, no. 10, 2020, doi: 10.14445/22315381/IJETT-V68I10P218.
- [31] R. Mayrhofer, J. vander Stoep, C. Brubaker, dan N. Kravovich, “The Android Platform Security Model,” *ACM Transactions on Privacy and Security*, vol. 24, no. 3, 2021, doi: 10.1145/3448609.
- [32] I. M. Almomani dan A. al Khayer, “A Comprehensive Analysis of the Android Permissions System,” *IEEE Access*, vol. 8, 2020, doi: 10.1109/ACCESS.2020.3041432.
- [33] M. A. Karim dan A. R. Adriansyah, “Analisis dan Perancangan Aplikasi Mobile untuk Donasi menggunakan Metode Hybrid berbasis React Native,” *Jurnal Informatika Terpadu*, vol. 8, no. 1, 2022, doi: 10.54914/jit.v8i1.394.
- [34] E. GÜLCÜOĞLU, A. B. USTUN, dan N. SEYHAN, “Comparison of Flutter and React Native Platforms,” *Journal of Internet Applications and Management*, 2021, doi: 10.34231/iuyd.888243.
- [35] M. F. Santoso, “TEKNIK SINGLE PAGE APPLICATION (SPA) LAYOUT WEB DENGAN MENGGUNAKAN REACT JS DAN BOOTSTRAP,” *Jurnal Khatulistiwa Informatika*, vol. 9, no. 2, 2021, doi: 10.31294/jki.v9i2.11357.
- [36] M. Mohan, *Advanced Web Development with React, SSR and PWA with Next.js using React with advanced concepts*, 1 ed. Darya Ganj: BPB Publications, 2020.
- [37] A. Biørn-Hansen, C. Rieger, T. M. Grønli, T. A. Majchrzak, dan G. Ghinea, “An empirical investigation of performance overhead in cross-platform mobile development frameworks,” *Empir Softw Eng*, vol. 25, no. 4, 2020, doi: 10.1007/s10664-020-09827-6.
- [38] T. Zohud dan S. Zein, “Cross-Platform Mobile App Development in Industry: A Multiple Case-Study,” *International Journal of Computing*, vol. 20, no. 1, 2021, doi: 10.47839/ijc.20.1.2091.
- [39] C. Janiesch, P. Zschech, dan K. Heinrich, “Machine learning and deep learning,” *Electronic Markets*, vol. 31, 2021, doi: 10.1007/s12525-021-00475-2/Published.
- [40] X. Li, B. Dai, H. Sun, dan W. Li, “Corn classification system based on computer vision,” *Symmetry (Basel)*, vol. 11, no. 4, 2019, doi: 10.3390/sym11040591.
- [41] R. Ojala, J. Vepsäläinen, dan K. Tammi, “Motion detection and classification: ultra-fast road user detection,” *J Big Data*, vol. 9, no. 1, 2022, doi: 10.1186/s40537-022-00581-8.

- [42] M. S. B. Yusoff, “ABC of Content Validation and Content Validity Index Calculation,” *Education in Medicine Journal*, vol. 11, no. 2, hlm. 49–54, Jun 2019, doi: 10.21315/eimj2019.11.2.6.

LAMPIRAN

LAMPIRAN 1 Isi Lampiran

Isian lampiran, dapat berupa:

1. Foto pengujian di lapangan
2. Gambar hasil aplikasi
3. Tampilan UI / UX
4. Hasil pengujian, contoh : kuesioner, wawancara, bukti pengujian
5. Source code