**TUGAS AKHIR**

**PERANGKAT LUNAK IDENTIFIKASI**

**KEKABURAN PADA GAMBAR DOKUMEN TEKS**

**MENGGUNAKAN LAPLACIAN OPERATOR**

HALAMAN JUDUL LUAR

****

**Oleh:**

**Robert Antonius 2125250057**

**PROGRAM STUDI INFORMATIKA**

**FAKULTAS ILMU KOMPUTER DAN REKAYASA**

**UNIVERSITAS MULTI DATA PALEMBANG**

**PALEMBANG**

**2025**

**TUGAS AKHIR**

**PERANGKAT LUNAK IDENTIFIKASI**

**KEKABURAN PADA GAMBAR DOKUMEN TEKS**

**MENGGUNAKAN LAPLACIAN OPERATOR**

**Diajukan Sebagai Salah Satu Syarat Untuk Menyelesaikan**

**Program Studi Informatika**

**Jenjang Pendidikan Strata-1**

****

**Oleh:**

**Robert Antonius 2125250057**

**PROGRAM STUDI INFORMATIKA**

**FAKULTAS ILMU KOMPUTER DAN REKAYASA**

**UNIVERSITAS MULTI DATA PALEMBANG**

**PALEMBANG**

**2025**

**TUGAS AKHIR**

**PERANGKAT LUNAK IDENTIFIKASI**

**KEKABURAN PADA GAMBAR DOKUMEN TEKS**

**MENGGUNAKAN *LAPLACIAN OPERATOR***

# HALAMAN PERSETUJUAN

Disusun oleh:

Robert Antonius

(2125250057)

Disetujui oleh:

Novan Wijaya, M.Kom.

Pembimbing Utama

(151117)

Derry Alamsyah, S.Si., M.Kom., M.Pd.

Ketua Program Studi Informatika

(111069)

**PROGRAM STUDI INFORMATIKA**

**FAKULTAS ILMU KOMPUTER DAN REKAYASA**

**UNIVERSITAS MULTI DATA PALEMBANG**

**PALEMBANG**

**2025**

**HALAMAN PERNYATAAN DEWAN PENGUJI**

**DIDAPATKAN SETELAH SIDANG**

PERNYATAAN PERSETUJUAN PUBLIKASI   
TUGAS AKHIR

Sebagai sivitas akademik Universitas Multi Data Palembang, saya yang bertanda tangan di bawah ini :

Nama : Robert Antonius

NIM : 2125250057

Program Studi : Informatika

Jenis Karya : Tugas Akhir

Demi kepentingan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Multi Data Palembang **Hak Bebas Royalti Nonekslusif (*Non-exclusive Royalty-Free Right*)** atas Universitas Multi Data Palembang saya yang berjudul :

**PERANGKAT LUNAK IDENTIFIKASI**

**KEKABURAN PADA GAMBAR DOKUMEN TEKS**

**MENGGUNAKAN *LAPLACIAN OPERATOR***

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non-ekslusif ini Universitas Multi Data Palembang berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), Merawat dan mempublikasikan karya akhir saya tanpa meminta izin dari saya selama tetap mencantumkan saya sebagai penulis.

Demikian pernyataan ini kami buat dengan sebenarnya.

Dibuat di : Palembang

Pada tanggal : Januari 2025

Yang Menyatakan,

Robert Antonius

2125250057

**Fakultas Ilmu Komputer dan Rekayasa  
Universitas Multi Data Palembang**

Program Studi Informatika

Tugas Akhir Sarjana Komputer

Semester Gasal Tahun 2024/2025

**Perangkat Lunak Identifikasi**

**Kekaburan pada Gambar Dokumen Teks**

**Menggunakan *Laplacian Operator***

Robert Antonius 2125250057

Abstrak

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx.

**Kata kunci**: xxxxxxxxxx xxxxxxxxxxxxxx xxxxxxxxxx xxxxxxxxxx xxxxxxxx xxxxxxxxxxx xxxxxxxxx xxxxxxxxx xxxxxxxxxxx xxxxxxxxx xxxxxxx

**Faculty of Computer Science and Engineering  
Universitas Multi Data Palembang**

Diploma/Undergraduate Program in Informatics

Final Project

1st Semester of Academic Year 2024/2025

**Blur Identification Software**

**on Text Document Images**

**Using Laplacian Operator**

Robert Antonius 2125250057

Abstract

*xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx*xxx.

**Keywords**: xxxxxxxxxx xxxxxxxxxxxxxx xxxxxxxxxx xxxxxxxxxx xxxxxxxx xxxxxxxxxxx xxxxxxxxx xxxxxxxxx xxxxxxxxxxx xxxxxxxxx xxxxxxx

KATA PENGANTAR

Puji dan syukur penulis ucapkan kepada Tuhan Yang Maha Esa, sehingga penulis dapat menyelesaikan laporan tugas akhir yang berjudul **“PERANGKAT LUNAK IDENTIFIKASI KEKABURAN PADA GAMBAR DOKUMEN TEKS MENGGUNAKAN *LAPLACIAN OPERATOR*”**.

Laporan skripsi ini penulis susun sebagai salah satu syarat untuk menyelesaikan mata kuliah Tugas Akhir di Universitas Multi Data Palembang. Penulis menyadari bahwa laporan ini masih jauh dari sempurna karena terbatas oleh pengalaman penulis, untuk itu segala saran dan kritik yang membangun akan penulis terima dengan senang hati.

Penulis mengucapkan banyak terima kasih kepada semua pihak yang telah membantu baik secara langsung maupun tidak langsung juga memberikan bantuan, bimbingan, pengarahan, dan ide-ide selama proses penyelesaian laporan tugas akhir ini. Penulis ingin mengucapkan terima kasih sebesar-besarnya kepada:

1. Bapak Dr. Johannes Petrus, S.Kom., M.T.I., selaku Rektor Universitas Multi Data Palembang yang telah memberikan kesempatan untuk pelaksanaan tugas akhir ini.
2. Ibu Dr. Yulistia, S.Kom., M.T.I., selaku Wakil Rektor I Universitas Multi Data Palembang.
3. Ibu Kathryn Sugara, S.E., M.Si., selaku Wakil Rektor II Universitas Multi Data Palembang.
4. Bapak Dedy Hermanto, S.Kom., M.T.I., selaku Wakil Rektor III Universitas Multi Data Palembang.
5. Ibu Charisma Ayu P., M.HRM., selaku Wakil Rektor IV Universitas Multi Data Palembang.
6. Bapak Dr. Wijang Widhiarso, M.Kom., selaku Dekan Fakultas Ilmu Komputer dan Rekayasa Universitas Multi Data Palembang.
7. Ibu Yoannita, M.Kom., selaku Wakil Dekan Fakultas Ilmu Komputer dan Rekayasa Universitas Multi Data Palembang.
8. Bapak Derry Alamsyah, S.Si., M.Kom., M.Pd., selaku Ketua Program Studi Informatika Universitas Multi Data Palembang yang telah memberikan kesempatan dan persetujuan untuk pelaksanaan tugas akhir ini.
9. Bapak Novan Wijaya, M.Kom., selaku pembimbing pelaksanaan tugas akhir yang telah memberikan banyak saran dan masukan terhadap pelaksanaan dan penulisan laporan kerja praktik ini.
10. Ibu Desi dari PT Sukses Wijaya Adikarya, Ibu Phuza dari PT Cahaya Sanubari Sakti, dan Ibu Yuniar dari PT Kesuma Maju Sumatera, selaku perwakilan dari masing-masing perusahaan yang memberikan perspektif perusahaan dalam pelaksanaan tugas akhir.
11. Orang tua, keluarga, dan teman-teman penulis yang telah memberikan dukungan dalam pelaksanaan kerja praktik.
12. Teman-teman dari Unit Kegiatan Mahasiswa Programming dan dari shift malam Program Studi Informatika yang telah memberi masukan dari pengalaman dan pengetahuan mereka sendiri untuk membantu dalam pelaksanaan tugas akhir.
13. Dosen-dosen Universitas Multi Data Palembang yang telah memberikan bimbingan akademik yang menjadi landasan dari pelaksanaan tugas akhir.
14. Segenap staf Universitas Multi Data Palembang yang telah membantu dalam menjalankan operasi harian kampus Universitas Multi Data Palembang.
15. Semua pihak yang tidak dapat disebutkan satu persatu yang telah terlibat baik secara langsung maupun tidak langsung memberi dukungan bagi penulis.

Palembang, Januari 2025

Penulis

DAFTAR ISI

[HALAMAN JUDUL LUAR i](#_Toc185853489)

[HALAMAN PERSETUJUAN ii](#_Toc185853490)

[PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR ii](#_Toc185853491)

[Abstrak ii](#_Toc185853492)

[Abstract ii](#_Toc185853493)

[KATA PENGANTAR ii](#_Toc185853494)

[DAFTAR ISI ii](#_Toc185853495)

[DAFTAR TABEL ii](#_Toc185853496)

[DAFTAR GAMBAR ii](#_Toc185853497)

[PERNYATAAN KEASLIAN TUGAS AKHIR ii](#_Toc185853498)

[SURAT PERNYATAAN BEBAS PLAGIAT ii](#_Toc185853499)

[BAB 1 PENDAHULUAN 2](#_Toc185853500)

[1.1 Latar Belakang 2](#_Toc185853501)

[1.2 Rumusan Masalah 2](#_Toc185853502)

[1.3 Analisis Terhadap Batasan (*Constraint*) 2](#_Toc185853503)

[1.3.1 Analisis dari Aspek Ekonomis 2](#_Toc185853504)

[1.3.2 Analisis dari Aspek Manufakturabilitas 2](#_Toc185853505)

[1.3.3 Analisis dari Aspek Sustainabilitas 2](#_Toc185853506)

[1.4 Analisis Terhadap Karakteristik Solusi 2](#_Toc185853507)

[1.5 Pemilihan Solusi 2](#_Toc185853508)

[1.6 Skenario Pemanfaatan Produk oleh Pengguna 2](#_Toc185853509)

[1.7 Tujuan 2](#_Toc185853510)

[BAB 2 ALTERNATIF DAN PEMILIHAN SOLUSI 2](#_Toc185853511)

[2.1 Alternatif Solusi 2](#_Toc185853512)

[2.1.1 *HaarWavelet Transform* (HWT) 2](#_Toc185853513)

[2.1.2 *Fast Fourier Transform* (FFT) 2](#_Toc185853514)

[*2.1.3* *Laplacian* *Operator* 2](#_Toc185853515)

[2.2 Analisis Pemilihan Solusi 2](#_Toc185853516)

[BAB 3 METODOLOGI 2](#_Toc185853517)

[3.1 Identifikasi Masalah 2](#_Toc185853518)

[3.2 Analisis Solusi 2](#_Toc185853519)

[3.3 Pengumpulan Data 2](#_Toc185853520)

[3.4 Pengembangan Model 2](#_Toc185853521)

[3.5 Pengembangan Perangkat Lunak 2](#_Toc185853522)

[3.6 Pengujian 2](#_Toc185853523)

[3.7 Hasil 2](#_Toc185853524)

[BAB 4 PERANCANGAN 2](#_Toc185853525)

[4.1 Spesifikasi Solusi 2](#_Toc185853526)

[4.2 Rencana Pengujian 2](#_Toc185853527)

[4.3 Perancangan Sistem 2](#_Toc185853528)

[4.4 Verifikasi Perancangan 2](#_Toc185853529)

[BAB 5 IMPLEMENTASI DAN PENGUJIAN 2](#_Toc185853530)

[5.1 Implementasi Perancangan 2](#_Toc185853531)

[5.2 Pengujian 2](#_Toc185853532)

[BAB 6 PENUTUP 2](#_Toc185853533)

[6.1 Kesimpulan 2](#_Toc185853534)

[6.2 Saran 2](#_Toc185853535)

[DAFTAR PUSTAKA 2](#_Toc185853536)

[LAMPIRAN 2](#_Toc185853537)

[Daftar Riwayat Hidup 2](#_Toc185853538)

[Lembar Konsultasi 2](#_Toc185853539)

[Kode Program 2](#_Toc185853540)

[<judul lampiran> 2](#_Toc185853541)

[<judul lampiran> 2](#_Toc185853542)

[Form Hasil Pemeriksaan Tingkat Plagiarisme 2](#_Toc185853543)

[Laporan Hasil Pengecekan Plagiarisme 2](#_Toc185853544)

[Notulen Tugas Akhir 2](#_Toc185853545)

DAFTAR TABEL

[Tabel 1.1 Analisis Aspek Ekonomis oleh Narasumber **Error! Bookmark not defined.**](#_Toc183209807)

[Tabel 1.2 Estimasi Biaya Pembuatan Perangkat Lunak **Error! Bookmark not defined.**](#_Toc183209808)

[Tabel 1.3 Hasil Analisis Aspek Manufakturabilitas **Error! Bookmark not defined.**](#_Toc183209809)

[Tabel 1.4 Standar Pencahayaan Lux **Error! Bookmark not defined.**](#_Toc183209810)

[Tabel 1.5 Hasil Analisis Aspek Sustainabilitas 2](#_Toc183209811)

[Tabel 1.6 Hasil Analisis Karakteristik Solusi 2](#_Toc183209812)

[Tabel 1.7 Ringkasan dan Nilai Performa Penelitian Terkait 2](#_Toc183209813)

[Tabel 1.8 Nilai Performa Metode Terpilih 2](#_Toc183209814)

[Tabel 2.1 Perbandingan Performa Teknik Identifikasi *Blur* 2](#_Toc183209815)

[Tabel 2.2 Estimasi Pelatihan Solusi 2](#_Toc183209816)

[Tabel 2.3 Analisis Pemilihan Solusi berdasarkan Tiga Aspek 2](#_Toc183209817)

[Tabel 3.1 Variasi Pengambilan Gambar Dokumen SmartDoc-QA 2](#_Toc183209818)

[Tabel 4.1 Rencana Pengujian *Black Box* 2](#_Toc183209819)

[Tabel 4.2 Verifikasi dan Bukti dalam Proses Perancangan 2](#_Toc183209820)

DAFTAR GAMBAR

[Gambar 1.1 Proses Pengiriman Dokumen Oleh Klien **Error! Bookmark not defined.**](#_Toc183209821)

[Gambar 1.2 Proses Pemilihan Solusi **Error! Bookmark not defined.**](#_Toc183209822)

[Gambar 1.3 Siklus Pengumpulan Dokumen Tanpa Perangkat Lunak 2](#_Toc183209823)

[Gambar 1.4 Siklus Pengumpulan Dokumen Dengan Perangkat Lunak 2](#_Toc183209824)

[Gambar 2.1 Tipe-tipe Tepi pada Gambar 2](#_Toc183209825)

[Gambar 2.2 Hasil Transformasi 3-*level* HWT 2](#_Toc183209826)

[Gambar 2.3 Hasil Transformasi FFT 2](#_Toc183209827)

[Gambar 2.4 Hasil *Laplacian Operator* dan Nilai Variansnya 2](#_Toc183209828)

[Gambar 3.1 Contoh Gambar KTP 2](#_Toc183209829)

[Gambar 3.2 Contoh Gambar KK 2](#_Toc183209830)

[Gambar 3.3 Contoh Gambar Dokumen pada *Dataset* SmartDoc-QA 2](#_Toc183209831)

[Gambar 3.4 Pengembangan Model 2](#_Toc183209832)

[Gambar 3.5 Siklus Metode SDLC *Iterative* 2](#_Toc183209833)

[Gambar 3.6 Gambar *Prototype* Perangkat Lunak 2](#_Toc183209834)

[Gambar 4.1 Alur Kerja Model pada Perangkat Lunak 2](#_Toc183209835)

[Gambar 4.2 Tahapan Perancangan Sistem 2](#_Toc183209836)

[Gambar 4.3 *Use Case Diagram* Sistem Identifikasi *Blur* 2](#_Toc183209837)

[Gambar 4.4 *Activity Diagram* Pengiriman Gambar oleh Klien 2](#_Toc183209838)

[Gambar 4.5 *Activity Diagram* Pengaksesan Percakapan oleh Pengguna 2](#_Toc183209839)

[Gambar 4.6 *Activity Diagram Forward* Gambar oleh Pengguna 2](#_Toc183209840)

**Fakultas Ilmu Komputer dan Rekayasa  
Universitas Multi Data Palembang**

PERNYATAAN KEASLIAN TUGAS AKHIR

Nama : Robert Antonius

NPM : 2125250057

Program Studi : Informatika

Judul Tugas Akhir : Perangkat Lunak Identifikasi Kekaburan pada Gambar  
Dokumen Teks Menggunakan *Laplacian Operator*

Dengan ini saya menyatakan bahwa dalam Tugas Akhir ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu Perguruan Tinggi manapun, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila dikemudian hari terbukti ternyata saya memberikan pernyataan yang tidak benar, saya bersedia menerima sanksi berupa pencabutan gelar kesarjanaan saya dan sanksi hukum yang berlaku.

Palembang, Januari 2025

(Robert Antonius)

SURAT PERNYATAAN BEBAS PLAGIAT

Bersama surat ini saya:

Nama : Robert Antonius

Alamat : Jalan Karya Baru, Komplek Palem Blok A22

Instansi : Universitas Multi Data Palembang

Menyatakan bahwa paper saya yang berjudul:

Perangkat Lunak Identifikasi Kekaburan pada Gambar Dokumen Teks Menggunakan *Laplacian Operator*

1. Adalah benar **karya saya sendiri** atau **bukan plagiat** hasil karya orang lain.
2. Belum pernah dipublikasikan di media lain.
3. Apabila dikemudian hari terbukti bahwa karya ilmiah ini bukan karya saya sendiri atau plagiat hasil karya orang lain atau ditemukan pada media publikasi lain, maka saya bersedia menerima sanksi sesuai ketentuan perundang-undangan yang berlaku.

Demikian surat pernyataan ini saya buat untuk dipergunakan sebagaimana mestinya.

Palembang, Januari 2025

(Robert Antonius)

# PENDAHULUAN

Bab pendahuluan mencakup tinjauan umum dari masalah yang akan dibahas terdiri dari beberapa subbab, yaitu latar belakang masalah, rumusan masalah, analisis terhadap batasan, analasis terhadap karakteristik solusi, pemilihan solusi, skenario pemanfaatan, dan tujuan penelitian.

## Latar Belakang

Dengan pemopuleran penggunaan kamera *smartphone* meningkat pula pengambilan gambar di luar keadaan pengambilan yang stabil seperti saat menggunakan *scanner* yang menyebabkan gambar dapat memiliki kecacatan seperti *blur* (Rai dkk., 2017). Kekaburan atau *blur* adalah salah satu bentuk distorsi gambar yang paling umum ditemui dalam pengambilan foto dan merepresentasikan bentuk atau area dari gambar yang tidak dapat dilihat dengan baik karena tidak memiliki tepi yang jelas (Perić dkk., 2022). Pada dokumen teks, degradasi terhadap gambar seperti *blur* menyebabkan tulisan sulit untuk dibaca dan diolah (Souibgui dkk., 2022). Pada kasus dengan tingkat *blur* terlalu tinggi, gambar dokumen teks yang diambil tidak dapat terpakai sama sekali sebab banyak dari informasi pentingnya tertutupi oleh *blur* sehingga perlu dihilang terlebih dahulu *blur* tersebut atau mengambil gambar yang baru dengan kualitas lebih tinggi.

Gambar-gambar yang dapat memiliki *blur* perlu diolah terlebih dahulu untuk memastikan bahwa gambar tersebut tidak memiliki *blur* sehingga dapat digunakan pada proses selanjutnya (Tran dkk., 2020). Pengolahan kualitas gambar oleh manusia tidak dapat diimplementasikan untuk bekerja secara *real time* tanpa memerlukan pertimbangan yang luas dengan beban kerja yang besar (Perić dkk., 2022). Proses pengajuan kredit rumah memerlukan banyak dokumen beragam yang perlu dikumpulkan oleh klien (Sa’diah dkk., 2024). Dengan pengecekan manusia, pengajuan kredit rumah memerlukan waktu pengecekan lebih untuk mengecek terlebih dahulu ada atau tidaknya *blur* pada gambar dokumen dari klien sebelum dapat mengecek isi dari dokumen tersebut.

Sebuah pendekatan yang populer untuk menangani *blur* adalah melakukan proses *deblur* untuk menutupi kekaburan pada gambar namun *deblur* tidak dapat bekerja dengan baik pada gambar dokumen teks dibandingkan pada gambar natural karena dapat memunculkan artefak disekitar tulisan yang ingin dihilangkan *blur* (Asad dkk., 2016). Pendekatan lain adalah melakukan estimasi tingkat *blur* pada sebuah gambar untuk mengidentifikasi dan memfilter gambar yang memiliki *blur* untuk mengambil gambar dengan kualitas tinggi (Perić dkk., 2022). Saat diterima gambar yang memiliki *blur*, pengguna akan diminta untuk mengambil ulang gambar untuk memastikan dokumen pada gambar yang dikirim dapat terbaca (Asad dkk., 2016). Pada pengerjaan *capstone project* ini digunakan metode identifikasi *blur* *Laplacian operator* untuk memfilter dan memberi umpan balik terhadap gambar dokumen teks yang dikirim oleh klien untuk membantu dalam pengumpulan dokumen untuk proses pengajuan kredit rumah.

## Rumusan Masalah

Berdasarkan latar belakang di atas, rumusan masalah yang terbentuk adalah bagaimana implementasi sebuah sistem yang dapat mengidentifikasi ada atau tidak *blur* pada gambar dokumen teks dengan *Laplace operator*.

## Analisis Terhadap Batasan (*Constraint*)

Analisis dilakukan untuk menentukan batasan pengerjaan proyek. Analisis terhadap batasan yang dilakukan yaitu analisis terhadap aspek ekonomis, terhadap aspek manufakturabilitas, dan terhadap aspek sustainabilitas.

### Analisis dari Aspek Ekonomis

Analisis terhadap aspek ekonomis dilakukan melalui wawancara sudut pandang *stakeholder*/*client* dan dengan penjabaran perkiraan biaya pembuatan perangkat lunak. Survei wawancara dilakukan dengan tiga perusahaan *developer* rumah yang menjadi *stakeholder* pada proyek ini, yaitu PT Sukses Adikarya, PT Cahaya Sanubharisakti, dan PT Kesuma Maju Sumatera sebagai sudut pandang *client*. Berikut hasil wawancara instansi pengguna terkait aspek ekonomis dapat dilihat pada Tabel 1.1.

Tabel 1.1 Hasil Analisis Aspek Ekonomis

| **Perusahaan/Instansi** | **Harga** |
| --- | --- |
| PT Sukses Adikarya | Rp1.000.000,00 |
| PT Cahaya Sanubharisakti | Rp1.000.000,00 |
| PT Kesuma Maju Sumatera | Rp2.000.000,00 |

Estimasi biaya pembuatan perangkat lunak dilakukan dengan menggunakan informasi terkait *application programming interface* (API), layanan *hosting*, *tools* yang digunakan pada perangkat lunak dan biaya pembuatan seperti upah jasa pengembang.

Biaya terkait API berasal dari penggunaan WhatsApp Cloud API untuk mengintegrasikan perangkat lunak dengan sebuah akun WhatsApp yang dirincikan di laman *pricing* pada *website* WhatsApp Business Platform (WhatsApp business platform pricing, n.d.). WhatsApp Business Platform melakukan pembiayan berdasarkan jumlah dan tipe percakapan yang dilakukan oleh akun. Terdapat lima tipe percakapan, yaitu validasi, pemasaran, utilitas, servis, dan validasi internasional, dengan setiap percakapan terbuka selama 24 jam dan dapat bertumpuk jika lebih dari satu tipe.

Untuk pembuatan perangkat lunak ini digunakan tipe percakapan servis. Tipe percakapan servis dapat digunakan secara gratis sejak 1 November 2024 dengan batasan tidak dapat memulai percakapan dan hanya dapat membalas setelah dihubungi (Pricing updates on the WhatsApp business platform, n.d.). Tipe percakapan ini digunakan karena perangkat lunak menggunakan akun WhatsApp untuk menerima gambar dokumen lalu memberi umpan balik sehingga alur kerja perangkat lunak tidak terbatasi oleh limitasi.

Biaya terkait *hosting* didapat dari penggunaan Vercel untuk melakukan *hosting* aplikasi Flask yang dirincikan oleh Vercel pada (Limits, n.d.; Pricing on Vercel, n.d.; Vercel hobby plan, n.d.). Biaya terkait layanan *hosting* yang digunakan sebesar US$20,00 (atau Rp 320.000,00 dengan kurs Rp16.000,00/US$) per bulan dengan Vercel *pro plan*, namun dapat diturunkan menjadi Rp0,00 jika memanfaatkan pilihan bebas biaya Vercel *hobby plan* untuk melakukan *deployment*.

Biaya perangkat yang digunakan adalah sekitar Rp8.750.000 dengan spesifikasi CPU *quad core processor* 2GHz, RAM 8GB, GPU NVIDIA MX130 dengan VRAM 6GB, dan penyimpanan bertipe SSD. *Tools* yang digunakan pada pembuatan perangkat lunak antara lain Python dan Visual Studio Code tersedia secara *open source* sehingga tidak menambah biaya sedangkan untuk Google Colab Pro memerlukan biaya sebesar Rp160.000,00 per bulan.

Perkiraan perincian biaya sementara terjumlah Rp640.000 untuk 2 bulan Vercel *pro plan*, Rp320.000 untuk 2 bulan Google Colab Pro, dan Rp8.750.000,00 untuk perangkat pengembang. Perincian biaya pembuatan perangkat lunak dapat diminimalisir dengan memanfaatkan opsi-opsi gratis dengan batasan yang tidak mengganggu alur kerja aplikasi. Perincian biaya pengembangan dijabarkan pada Tabel 1.2.

Tabel 1.2 Estimasi Biaya Pembuatan Perangkat Lunak

| **Keterangan** | | **Biaya** |
| --- | --- | --- |
| Layanan | *Hosting* Vercel Pro (dan pajak) | Rp710.400,00 |
| WhatsApp Cloud API | Rp0,00 |
| Python | Rp0,00 |
| Visual Studio Code | Rp0,00 |
| Google Colab Pro (dan pajak) | Rp355.200,00 |
| Subtotal | Rp960.000,00 |
| Perangkat (CPU quad core processor 2GHz, RAM 8GB, GPU NVIDIA MX130 dengan VRAM 6GB, dan penyimpanan bertipe SSD) | | Rp8.750.000,00 |
| **Total** | | Rp9.710.000,00 |

### Analisis dari Aspek Manufakturabilitas

Analisis terhadap aspek manufakturabilitas dilakukan atas dasar wawancara dengan 3 perusahaan yang menjadi *stakeholder* dalam proyek. Wawancara dilakukan untuk menentukan batasan penyelesaian proyek termasuk batasan waktu dan batasan fitur yang diperlukan. Terdapat 4 fitur yang dianalisis, yaitu fitur identifikasi gambar dokumen yang memiliki *blur* yang memerlukan waktu pembuatan 6 minggu, rentang cahaya bagi aplikasi dengan waktu 2 minggu, fitur ketepatan tinggi dalam mengidentifikasi *blur* dengan waktu 2 minggu, dan fitur integrasi dengan sistem yang telah digunakan untuk penerimaan gambar (WhatsApp) dalam waktu 2 minggu. Berikut hasil wawancara instansi pengguna terkait aspek manufakturabiltas dapat dilihat pada Tabel 1.3.

Tabel 1.3 Hasil Analisis Aspek Manufakturabilitas

| **Aspek** | **PT Sukses Adikarya** | **PT Cahaya Sanubharisakti** | **PT Kesuma Maju Sumatera** |
| --- | --- | --- | --- |
| Identifikasi gambar dokumen *blur*  (6 minggu) | OK | OK | OK |
| Dapat digunakan pada pencahayaan rendah sampai tinggi  (2 minggu) | OK | OK | OK |
| Menghindari menangkap gambar tajam sebagai gambar *blur*  (2 minggu) | OK | OK | OK |
| Diintegrasi dengan sistem yang digunakan (Whatsapp)  (2 minggu) | OK | OK | OK |
| **Total**  **3 bulan** |  |  |  |

Pengambilan gambar ideal dilakukan dengan pencahayaan yang cukup supaya objek yang difoto terlihat dengan baik (Purwanto & Veranita, 2018). Perangkat lunak maka ditujukan untuk dapat melakukan identifikasi dalam keadaan pencahayaan rendah sampai pencahayaan tinggi. Rentang yang dipilih merupakan rentang pengambilan gambar pada *dataset* substitut yang digunakan dan telah dikonfirmasi oleh pihak perusahaan *stakeholder*. Standar pengukuran cahaya yang digunakan adalah standar pencahayaan lux yang memiliki 6 kategori (Pratama & Nurdiana, 2020) yang ditunjukkan pada Tabel 1.4.

Tabel 1.4 Standar Pencahayaan Lux

| **Pencahayaan** | | **Lux (lx)** |
| --- | --- | --- |
| Ekstra Rendah | | < 50 |
| Rendah | | 50 – 150 |
| Sedang | | 150 – 175 |
| Tinggi | Tinggi I | 200 |
| Tinggi II | 300 |
| Tinggi III | 450 |
| Sangat Tinggi | | 450 – 700 |
| Ekstra Tinggi | | > 700 |

Perangkat lunak identifikasi *blur* berfungsi untuk menyaring pengumpulan gambar dokumen yang dikirimkan oleh klien. Menurut Yuniar dari PT Kesuma Maju Sejahtera, prosedur yang dilakukan saat diterima gambar *blur* adalah gambar tersebut dicek terlebih dahulu apakah dapat terbaca, jika iya maka tetap diterima walau terdapat *blur* untuk mempercepat proses.

Perangkat lunak identifikasi *blur* maka memiliki *cost* lebih tinggi saat salah mengklasifikasikan gambar tanpa *blur* sebagai gambar *blur* dibandingkan dengan mengklasifikasikan gambar *blur* sebagai gambar tanpa *blur*. Identifikasi *blur* dilakukan untuk memudahkan pengumpulan dokumen untuk proses *BI checking*, sehingga identifikasi *blur* tidak perlu menangkap semua gambar *blur* selagi gambar dokumen masih dapat dibaca oleh staf. Identifikasi *blur* dilakukan terhadap gambar yang diambil dan dikirim oleh klien, sehingga perangkat lunak perlu untuk tidak salah mengklasifikasikan gambar tanpa *blur* sebagai gambar *blur* karena dapat membingungkan klien. Perangkat lunak maka ditujukan untuk memprioritaskan nilai *precision* setelah kecepatan dengan nilai *Fbeta-score* dengan *beta* berupa 0.5 untuk acuan pertimbangan *trade-off* *precision* dan *recall*.

### Analisis dari Aspek Sustainabilitas

Dalam melakukan analisis terhadap aspek sustainibilitas, dilakukan dengan wawancara dengan 3 perusahaan mengenai kinerja perangkat lunak dalam beroperasi. Terdapat satu aspek dalam kinerja perangkat lunak dalam beroperasi, yaitu sistem dapat melakukan identifikasi *blur* pada gambar dokumen dalam waktu singkat (3 detik). Berikut hasil wawancara instansi pengguna terkait aspek sustainabilitas dapat dilihat pada Tabel 1.5.

Tabel 1.5 Hasil Analisis Aspek Sustainabilitas

| **Aspek** | **PT Sukses Adikarya** | **PT Cahaya Sanubharisakti** | **PT Kesuma Maju Sumatera** |
| --- | --- | --- | --- |
| Identifikasi *blur* pada gambar dokumen dapat waktu singkat (3 detik) | OK | OK | OK |

## Analisis Terhadap Karakteristik Solusi

Hasil analisis karakteristik solusi diperoleh dari mengidentifikasi dan menanggapi masalah yang diperoleh melalui wawancara dengan perusahaan terkait. Hasil analisis karakteristik solusi tersebut tercantum pada Tabel 1.6.

Tabel 1.6 Hasil Analisis Karakteristik Solusi

| **No.** | **Masalah** | **Fungsi** |
| --- | --- | --- |
| 1. | Staf tidak dapat selalu mengecek ada tidaknya *blur* pada gambar dokumen seperti saat sangat sibuk atau saat pulang dari kantor | Perangkat lunak mampu mengidentifikasi gambar dengan dan tanpa *blur* dan memberi umpan balik |
| 2. | Semakin lama klien mendapat umpan balik semakin besar kemungkinan klien tidak melihatnya | Identifikasi memprioritaskan kecepatan dengan target memberi deteksi dalam waktu 3 detik |
| 3. | Tidak semua klien fasih menggunakan teknologi | Integrasi WhatsApp yang telah digunakan perusahaan untuk komunikasi dengan klien |
| 4. | Terdapat aplikasi CamScanner yang umum digunakan untuk mengambil gambar | Perangkat lunak juga menerima gambar dokumen dalam bentuk PDF |

Dalam melakukan analisis terhadap karakteristik solusi, dilakukan dengan melihat masalah yang dialami oleh perusahaan berdasarkan hasil wawancara. Masalah tersebut berasal dari jangka waktu antara saat klien mengirim gambar sampai saat seorang staf mengecek dan memberi umpan balik terhadap apakah gambar tersebut jelas yang jika terlalu lama dapat menyebabkan klien berpindah ke aktivitas lain yang menghentikan mereka dari mengecek pesan dan mengirim ulang gambar. Jangka waktu ini akan selalu ditemui selama melibatkan manusia karena dapat saja klien mengirim gambar saat tidak ada staf yang dapat mengeceknya, seperti saat jam *meeting* atau jam pulang kantor. Solusi yang diajukan melibatkan sistem *image processing* untuk secara cepat mengecek apakah ada kekaburan pada gambar tanpa supervisi manusia.

Perangkat lunak *document quality image assessment* (DIQA) dapat digunakan untuk memitigasi masalah tersebut dengan mencegah pengguna mengambil gambar yang kabur atau dengan memberi umpan balik secara instan jika terdapat *blur* pada gambar yang diambil (Rai dkk., 2017). Solusi yang diajukan dalam tugas akhir ini adalah sebuah sistem yang dapat mendeteksi keberadaan *blur* pada gambar dokumen dan memberi pesan umpan balik kepada pengirim foto.

Menurut Desi Arrani, selaku Staf Administrasi PT Sukses Wijaya Adikarya, gambar dokumen *blur* sering diterima dalam proses pengajuan kredit dari klien, umumnya karena klien tidak terbiasa dalam mengunduh dan menggunakan aplikasi seperti CamScanner. Menurut Puzha Manzha K., selaku Staf Pemberkasan PT Cahaya Sanubari Sakti, klien mengirim gambar *blur* tersebut umumnya karena tergesa-gesa dalam mengirim gambar atau karena klien tidak terbiasa menggunakan perangkat mereka untuk mengambil gambar dokumen dan tidak menggunakan aplikasi pemindaian dokumen seperti CamScanner.

Menurut Yuniar Saputri, selaku Staf HRD PT Kesuma Maju Sejahtera, pengiriman gambar dokumen *blur* ini dapat menyebabkan melambannya proses pengajuan kredit sebab klien yang mengirim tidak mengetahui bahwa gambar yang mereka kirim kurang jelas sehingga perlu menunggu seorang karyawan mengecek gambarnya, lalu menunggu klien untuk mengirim gambar baru yang diambil ulang.

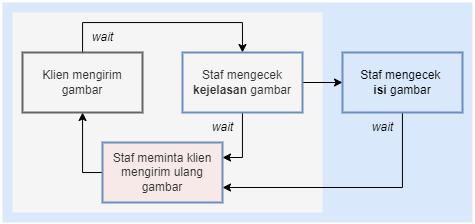
Berdasarkan hasil wawancara dengan ibu Desi, penerimaan gambar dokumen *blur* dapat menyebabkan *delay* dalam proses pengajuan kredit sebab perlu menunggu klien membaca pesan permintaan pengiriman ulang gambar lalu mengambil gambar dokumen yang baru.

Selanjutnya berdasarkan hasil wawancara dengan ibu Puzha, pengiriman gambar dokumen *blur* menyebabkan *delay* karena klien belum tentu langsung membaca pesan untuk mengirim ulang gambar sehingga perlu menunggu saat klien dapat merespons balik sebelum mendapat gambar baru.

Kemudian berdasarkan hasil wawancara dengan ibu Yuniar, penyebab dokumen *blur* dapat menjadi hambatan adalah karena adanya jangka waktu sebelum klien membaca pesan bahwa gambar yang dikirim *blur* sehingga klien telah berpindah ke kesibukan lain dan tidak membaca pesan yang dikirim atau tidak dapat mengirim gambar yang baru.

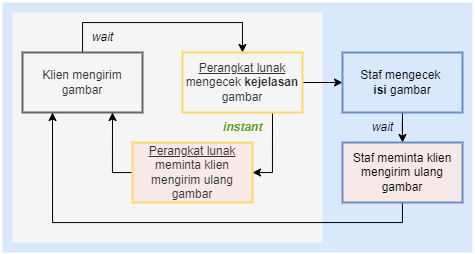
Berdasarkan hasil wawancara tersebut, permasalahan dari gambar dokumen *blur* tersebut ditemukan terjadi karena adanya jangka waktu sebelum klien menerima pesan bahwa gambar yang mereka kirim *blur*, sehingga klien berpindah ke aktivitas lain yang mencegah klien dari mengambil gambar dokumen baru. Dengan adanya sistem yang dapat memberi umpan balik lebih cepat kepada klien terkait apakah gambar yang mereka kirim terdapat *blur* atau tidak, dapat mengurangi penghambatan yang terjadi dalam proses pengajuan kredit.

Permasalahan yang diangkat menghambat alur pengajuan kredit rumah karena menyebabkan siklus saling tunggu antara klien dan staf perusahaan. Staf perusahaan perlu menunggu klien mengunggah gambar dokumen-dokumen guna pengajuan kredit, dan klien perlu menunggu staf perusahaan mengecek apakah gambar dokumen tersebut dapat dibaca dan apakah klien memenuhi persyaratan pengajuan kredit rumah. Fenomena ini menyebabkan munculnya 2 siklus saling tunggu bersarang seperti pada Gambar 1.1 dengan siklus pertama staf memerlukan gambar yang jelas lalu pada siklus kedua staf memerlukan gambar yang tepat dan jika gambar tersebut jelas namun belum tepat siklus kembali ke menunggu gambar jelas.



Gambar 1.1 Siklus Pengumpulan Dokumen Tanpa Perangkat Lunak

Solusi yang diajukan yang menggunakan perangkat lunak untuk melakukan identifikasi *blur* secara otomatis dan cepat bertujuan untuk mengurangi hambatan dengan mempercepat siklus pertama pada pengajuan kredit. Perangkat lunak dapat mempercepat proses pada siklus pertama dengan melakukan penilaian kualitas gambar terlebih dahulu sehingga dapat diperbaiki oleh klien jika sulit dibaca karena *blur* tanpa memerlukan pengecekan manual dari staf. Gambar 1.2 menggambarkan siklus baru dengan penerapan perangkat lunak untuk mengurangi waktu tunggu dari proses pengecekan manual kejelasan gambar.



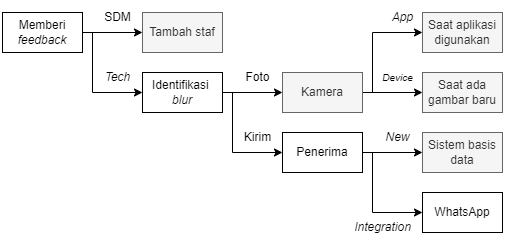
Gambar 1.2 Siklus Pengumpulan Dokumen Dengan Perangkat Lunak

## Pemilihan Solusi

Solusi untuk mencegah *delay* karena diterimanya gambar dokumen *blur* dapat dilakukan dengan berbagai cara, seperti dengan menambah staf untuk mengecek gambar atau dengan menggunakan teknologi untuk membantu dalam pengecekan gambar. Untuk solusi penambahan staf, staf yang telah ada dapat disusun dalam bentuk *shift* untuk membantu mengecek dokumen yang dikirim klien. Masalah utama dengan solusi ini adalah solusi tidak mengubah keadaan seperti semua staf perusahaan telah pulang dari kantor dan tidak dapat dihubungi sehingga tidak dapat mengecek dokumen yang dikirim oleh klien.

Untuk solusi pemanfaatan teknologi ada beberapa metode seperti menggunakan aplikasi kamera yang dapat mencegah diambilnya gambar *blur* atau menyisipkan sistem identifikasi *blur* pada WhatsApp. Untuk penerapan aplikasi kamera dapat menyebabkan kesulitan bagi klien yang perlu mengunduh aplikasi kamera untuk mengirim dokumen, sulit dilakukan oleh klien yang kurang fasih teknologi, dan dapat menyebabkan kekhawatiran terhadap masalah keamanan klien.

Untuk penyisipan sistem identifikasi pada WhatsApp dapat diintegrasi lebih mulus karena tidak memerlukan perubahan dari sisi perusahaan maupun sisi klien dan proses pengiriman dokumen hanya pindah ke kontak berbeda. Solusi yang dipilih diilustrasikan pada Gambar 1.3 adalah pembuatan sistem pemberian umpan balik kepada klien dengan teknologi identifikasi *blur* yang melakukan identifikasi saat gambar dikirim ke penerima dan sistem diintegrasi dengan WhatsApp.



Gambar 1.3 Proses Pemilihan Solusi

Untuk menerapkan solusi tersebut, diperlukan metode untuk membuat perangkat lunak yang dapat melakukan identifikasi ada atau tidaknya kekaburan pada gambar dokumen. Metode yang dilakukan dapat berupa metode untuk mengidentifikasi gambar yang terdapat *blur*.

Metode yang dipilih juga perlu dapat melakukan identifikasi tanpa referensi gambar yang tidak memiliki *blur*, karena dalam skenario penggunaan gambar yang dikirim oleh klien (pengguna) belum tentu terdapat gambar jernih. Keperluan untuk menggunakan metode *non-reference image quality assessment* (NR-IQA) berarti tidak dapat digunakan metode seperti *optical character recognition* (OCR) untuk yang umum digunakan untuk menilai apakah sebuah teks dapat dibaca atau tidak (Rodin dkk., 2021).

Identifikasi *blur* tanpa gambar referensi awal dapat dilakukan dengan berbagai metode-metode dari bidang *computer vision* berbeda, seperti berbasis *convolutional neural network* (CNN), melalui ekstraksi fitur-fitur dari gambar, atau dengan metode hibrida dengan fitur ekstraksi dan CNN (Liu dkk., 2021). Metode identifikasi *blur* berdasarkan CNN dapat dilakukan dengan melatih model untuk mendeteksi area gambar yang memiliki *blur* seperti dengan model *single shot detector* (SSD) *You Only Look Once* *version* 4 (YOLOv4) yang dilakukan oleh (Zhuang dkk., 2021) atau dengan model *faster region-convolutional neural network* (*Faster* R-CNN) AlexNet yang dilakukan oleh (Harron dkk., 2022).

Metode identifikasi *blur* berdasarkan ekstrasi fitur dapat dilakukan dengan metode analisis frekuensi seperti dengan *Fast Fourier transform* yang dilakukan oleh (Zhao dkk., 2019), metode deteksi tepi seperti dengan *Laplacian operator* seperti yang dilakukan oleh (Perić dkk., 2022), atau dengan metode identifikasi tipe-tipe tepi seperti dengan *HaarWavelet transform* yang dilakukan oleh (Tran dkk., 2020). Metode identifikasi *blur* dengan hibrida CNN dan ekstraksi fitur dapat dilakukan dengan menggunakan CNN untuk melakukan deteksi objek lalu melakukan ekstraksi fitur pada area-area gambar untuk menemukan apakah sebuah objek memiliki *blur* seperti pada metode *visual perception nature image quality evaluation* (VP-NIQE) oleh (Wu dkk., 2021).

Pendekatan CNN memiliki keunggulan untuk dapat melakukan klasifikasi pada gambar yang beragam dengan nilai kinerja klasifikasi tinggi, namun memiliki kelemahan pada kecepatan pelatihan penggunaan model yang cenderung lebih lambat karena banyaknya parameter model (Ahmad dkk., 2024). Pendekatan ekstraksi fitur memiliki keunggulan pada kecepatan klasifikasi yang lebih tinggi karena menilai dari lebih sedikit parameter, namun memiliki kelemahan pada pengaplikasiannya yang perlu disesuaikan apakah metode cocok dengan kasus penggunaan dengan pemilihan fitur yang diekstraksi yang tepat (He & Tu, 2024). Pendekatan hibrida dapat mengambil keunggulan dari metode-metode yang digabungkan, namun memiliki kompleksitas komputasi yang serupa dengan metode individualnya (Wu dkk., 2021).

Terdapat beberapa penelitian yang menggunakan metode-metode tersebut seperti yang dilakukan oleh (Wu dkk., 2021). *Visual Perception* NIQE (VP-NIQE) yang digunakan merupakan metode yang dikembangkan atas dasar NIQE dengan mengintegrasikan *object detection* untuk mengidentifikasi dan menilai kualitas objek pada gambar ke dalam proses penilaian kualitas gambar NIQE. VP-NIQE memiliki performa akurasi terbaik pada *database* LIVE, TID2013-sub, IVC-sub, CCID2014, namun pada *database* CSIQ-sub metode dipIQ memiliki performa terbaik dan pada *database* MLIVE metode IL-NIQE memiliki performa terbaik. Rata-rata waktu identifikasi metode VP-NIQE pada perangkat Dell dengan 16 GB RAM dan prosesor 3,2 GHz Intel Core 7 adalah 1.6420 detik per gambar 1280×720 *pixel*.

Penelitian dengan metode *HaarWavelet transform* dilakukan oleh (Tran dkk., 2020). Penelitian dilakukan dengan *dataset* publik umum pada *computer vision* seperti INRIA *Aerial Image Dataset*, Caltech-256 *Object Category*, INRIA *Holidays*, dan LabelMe. Dengan CPU NVIDIA Tegra K1 4 × 2.3 GHz Cortex-A15 dan GPU NVIDIA Kepler “GK20a” (192 cores at 852MHz) dicapai kecepatan identifikasi 0,056 detik per gambar 1024×1024 *pixel*. Pada *dataset* LabelMe dicapai tingkat ketepatan *accuracy* 95,9%, *precision* 97,2%, dan *recall* 89%.

Penelitian terhadap metode *Fast Fourier transform* (FFT) dilakukan oleh (Perić dkk., 2022). Penelitian yang dilakukan bertujuan untuk mengklasifikasi gambar yang akan digunakan sebagai data *input* untuk melatih *decision support system* (DSS) untuk deteksi halangan di rel kereta. *Dataset* yang digunakan berasal dari *dataset* SMART2 ditambah dengan gambar lapangan yang diambil untuk penelitian. FFT mencapai tingkat ketepatan *accuracy* 84,3%, *precision* 67,8%, *recall* 86,9%, dan *F1-score* 76,2% dengan waktu identifikasi 0,1399 detik pada gambar 1024×768 *pixel*.

Selanjutnya terdapat metode penilaian kualitas gambar dengan *Laplacian operator* yang dilakukan oleh (Perić dkk., 2022). Penelitian dilakukan untuk menemukan metode yang dapat digunakan untuk meningkatkan kualitas *dataset* untuk melatih model DSS untuk menanggapi halangan di atas rel kereta. *Dataset* SMART2 dengan gambar-gambar yang diambil di lapangan digunakan sebagai objek penelitian. Pada penelitian metode *Laplacian operator* memerlukan waktu identifikasi rata-rata 0,0931 detik per gambar dengan ukuran 1024×768 *pixel* dengan *accuracy* 83,5%, *precision* 68,5%, *recall* 79,4%, dan *F1-score* 73,5%.

Untuk metode yang memanfaatkan *object detection* terdapat metode dengan *You Only Look Once* versi 4 (YOLOv4) seperti yang dilakukan oleh (Zhuang dkk., 2021). Penelitian tersebut dilakukan untuk memisahkan gambar dengan *motion blur* dengan *dataset* COCO untuk meningkatkan efisiensi model deteksi *gesture*. Penelitian dilakukan dengan CPU Intel(R) Core (TM) i7-5500U CPU @ 2,40GHz pada metode YOLOv4, YOLOv4-*gesture*, dan YOLOv4-*motion*-*blur*-*gesture*. Performa terbaik dicapai dengan model YOLOv4-*motion*-*blur*-*gesture* dengan akurasi identifikasi 97,79%, *mean average precision* (mAP) 96,76%, dan waktu identifikasi 2,68 detik per gambar dengan ukuran 416×416 *pixel*.

Terdapat pula pada penelitian oleh (Harron dkk., 2022) menggunakan beberapa model *convolutional neural network* (CNN) ­*pre-trained*. Penelitian dilakukan untuk kategorisasi gambar pemindaian kanker payudara dengan menggunakan model ResNet-18, ResNet-50, AlexNet, VGG16, dan InceptionV3 untuk ekstraksi fitur dengan *dataset* *digital breast tomosythesis* (DBT) yang diunduh dari Duke University Health System. Performa terbaik didapat oleh model AlexNet dengan waktu identifikasi 0,2 detik per gambar 227×227 *pixel* dan akurasi 95,80% pada prosesor AMD Ryzen 5 3500U CPU @ 2,10GHz dengan RAM 8,0 GB dan ukuran model 227MB.

Identifikasi ada atau tidaknya *blur* pada gambar dokumen pada *capstone project* ini tidak memprioritaskan sensitivitas atau *recall* identifikasi berdasarkan wawancara dengan tiga perusahaan *stakeholder* karena gambar dokumen yang akan diterima tetap akan dilihat oleh karyawan untuk proses *BI checking*. Sistem identifikasi *blur* akan digunakan untuk menyaring gambar-gambar yang memiliki *blur* tinggi sehingga sulit dibaca oleh mata manusia, sedangkan adanya *blur* yang rendah namun masih dapat dibaca masih dapat diterima untuk diproses.

Parameter performa yang diprioritaskan dalam proyek ini adalah kecepatan yang dapat dicapai oleh perangkat lunak dalam melakukan identifikasi. Terkait sistem memiliki tujuan untuk memberi klien umpan balik secepatnya supaya klien belum berpaling dari layar ponsel mereka, metode yang dipilih harus dapat melakukan identifikasi dengan cepat seperti dengan waktu di bawah 3 detik. Untuk menyetarakan ukuran kecepatan metode, kecepatan diubah menjadi bentuk *pixel* / *milisecond* dengan persamaan (1). Tabel 1.7 menunjukkan ringkasan penelitian terdahulu terhadap kecepatan metode yang dipertimbangkan.

|  |  |
| --- | --- |
|  | (1) |

Keterangan:

N : jumlah gambar dalam satu rangkap identifikasi.

A : luas *pixel* per gambar.

T : total waktu identifikasi dalam *milisecond*.

Tabel 1.7 Ringkasan dan Nilai Performa Penelitian Terkait

| **No.** | **Metode** | **Kecepatan**  **(px/ms)** |
| --- | --- | --- |
| 1. | *Visual Perception Nature Image Quality Evaluator* (VP-NIQE) dengan *thresholding* | 561,267 |
| **2.** | ***HaarWavelet transform* dengan *thresholding*** | **18724,571** |
| **3.** | ***Fast Fourier transform* dengan *thresholding*** | **5621,387** |
| **4.** | ***Laplacian operator* dengan *thresholding*** | **8447,175** |
| 5. | *DarkNet* dengan YOLOv4-*motion*-*blur*-*gesture* | 64,573 |
| 6. | AlexNet dengan CNN | 257,645 |

Berdasarkan Tabel 1.7 metode dengan kecepatan identifikasi tercepat adalah *HaarWavelet transform*, *Laplacian operator*, dan *Fast Fourier transform* dengan kecepatan identifikasi FFT dengan 5621 *pixel* per *milisecond* lebih dari 10 kali lebih cepat dari kecepatan identifikasi metode tercepat ke-4, VP-NIQE dengan 561 *pixel* per *milisecond*.

Dipertimbangkan pula mana dari 3 metode yang terpilih yang akan diangkat sebagai topik penelitian. Pada penelitian yang dilakukan oleh (Pagaduan dkk., 2020) yang dipaparkan pada Tabel 1.8 terhadap beberapa metode identifikasi *blur* termasuk *Laplacian operator*, HWT, dan FFT, *Laplacian operator* memiliki nilai performa lebih rendah namun kecepatan lebih tinggi dibanding metode HWT dan FFT. Kecepatan identifikasi dapat melebur ke beberapa aspek diluar aspek sustainabilitas terkait kinerja kecepatan identifikasi, seperti biaya komputasi yang diperlukan untuk menggunakan metode terkait aspek ekonomis dan waktu pelatihan atau pemilihan *threshold* yang berpengaruh dengan aspek manufakturabilitas.

Tabel 1.8 Nilai Performa Metode Terpilih

| **Metode** | **Waktu Identifikasi (detik)** | ***F1-score* (%)** |
| --- | --- | --- |
| *Fast Fourier transform* | 6,2001 | 0,93048 |
| *Laplacian operator* | 1,1482 | 0,87111 |
| *HaarWavele transform* | 6,0370 | 0,96938 |

Berdasarkan paparan sebelumnya, untuk meminimalkan kemungkinan klien berpindah aktivitas sebelum membaca umpan balik *blur* pada gambar dokumen yang dikirim, metode *Laplacian operator* dipilih sebagai metode solusi atas dasar waktu komputasi yang lebih rendah.

## Skenario Pemanfaatan Produk oleh Pengguna

Perangkat lunak yang dibuat merupakan sistem yang dapat mengidentifikasi apakah terdapat *blur* pada gambar dokumen baik dalam format gambar (PNG, JPEG) atau dokumen (PDF). Aplikasi identifikasi diintegrasikan dengan aplikasi WhatsApp. Aplikasi digunakan oleh klien lewat WhatsApp saat klien mengirim dokumen yang diperlukan untuk proses pengajuan kredit lalu aplikasi memberi pesan kepada klien apakah dokumen yang dikirim telah diterima atau perlu difoto ulang karena terdapat *blur*. Aplikasi akan memberi hasil identifikasi maksimal dalam waktu 3 detik.

## Tujuan

Dalam pembuatan *capstone project* ini terdapat dua unsur yang menjadi tujuan proyek, antara lain:

1. Membuat perangkat lunak yang mampu mengidentifikasi ada atau tidaknya *blur* pada gambar dokumen lalu memberi umpan balik kepada pengirim jika gambar dokumen terdapat *blur*.
2. Menerapkan metode *Laplacian operator* dalam mengidentifikasi gambar dokumen yang memiliki *blur*.

# ALTERNATIF DAN PEMILIHAN SOLUSI

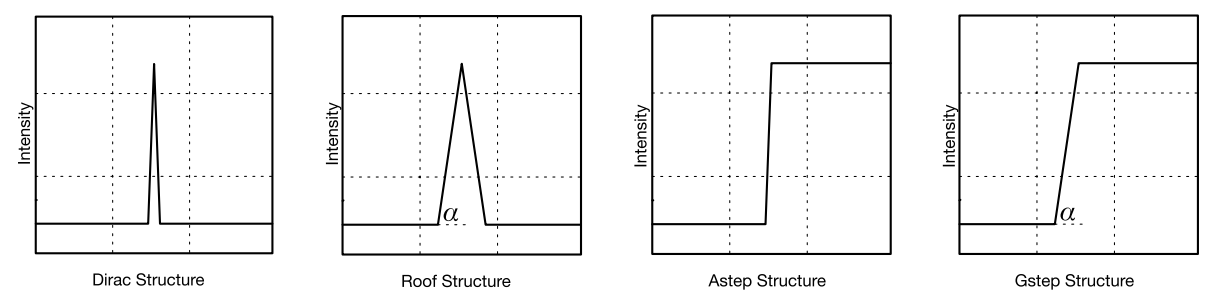
Bab pemilihan solusi berisi alternatif metode yang dipilih sebagai solusi permasalahan dalam penelitian, kajian literatur terkait metode yang diteliti, dan proses pertimbangan dalam pemilihan solusi yang diangkat sebagai metode topik penelitian.

## Alternatif Solusi

Alternatif solusi yang dipilih dijabarkan konsep dan karakteristik metodenya untuk lebih dalam menganalisis metode yang dipertimbangkan. Tiga solusi (metode) yang diajukan adalah *HaarWavelet transform*, *Fast Fourier transform*, dan *Laplacian operator*.

### *HaarWavelet Transform* (HWT)

*HaarWavelet transform* adalah metode ekstraksi fitur yang digunakan dalam inferensi ada atau tidaknya *blur* pada gambar berdasarkan jumlah dan tipe tepi-tepi yang diperoleh dari hasil konvolusi gambar (Sree dkk., 2018).



Gambar 2.1 Tipe-tipe Tepi pada Gambar

Sumber: (Tran dkk., 2020)

Proses inferensi dilakukan melalui empat jenis tepi yang umumnya terdapat pada gambar, yaitu *dirac-structure edge*, *roof-structure edge*, *A-step-structure edge*, dan *G-step-structure edge* seperti pada Gambar 2.1 (Andhavarapu, 2015). Pada gambar yang tidak memiliki *blur* terdapat gabungan dari keempat jenis tepi tersebut dengan intensitas tinggi sementara pada gambar yang memiliki *blur* terdapat lebih banyak tepi *G-step-structure* dan *roof-structure* dengan intensitas rendah dan nilai alfa tinggi (Andhavarapu, 2015).



Gambar 2.2 Hasil Transformasi 3-*level* HWT

Sumber: (G. S. Tran dkk., 2020)

Hasil transformasi yang diperoleh dari gambar seperti pada Gambar 2.2 berupa tepi horizontal, tepi vertikal, dan tepi diagonal gambar disusun di sekitar gambar diulang dengan resolusi gambar setengah dari resolusi iterasi sebelumnya hingga mencapai tingkat HWT ketiga (G. S. Tran dkk., 2020).

Hasil HWT pada gambar lalu digunakan untuk menentukan tipe tepi-tepi pada gambar dengan mengevaluasi tepi yang terdapat dalam area 2×2, 4×4, dan 8×8. Rasio tepi *A-step-structure* terhadap tepi *dirac-structure* digunakan untuk dibandingkan dengan sebuah *threshold* untuk menilai apakah gambar termasuk *blur* atau tidak dan ratio tepi *G-step-structure­* terhadap tepi *roof-structure* digunakan sebagai nilai koefisien keyakinan *blur* pada gambar (Sree dkk., 2018).

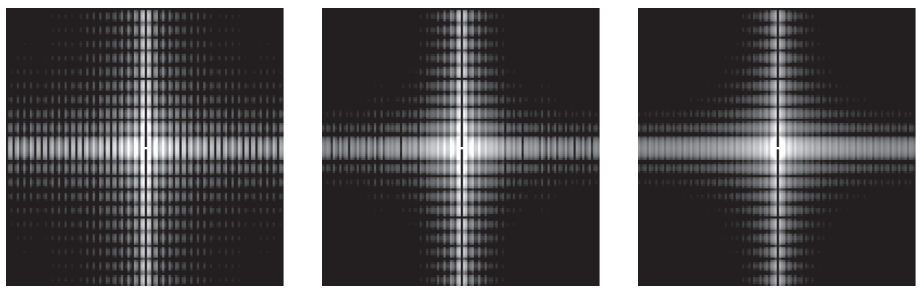
Penelitian yang dilakukan oleh (Sree dkk., 2018) menggunakan identifikasi *blur* dengan *HaarWavelet tranform* pada gambar wajah manusia. HWT digunakan untuk melakukan identifikasi tipe *blur* melalui koefisien keyakinan yang digunakan untuk mengklasifikasikan tipe *blur* lalu melakukan proses pembenaran *blur* berbasis *iterative graph* yang sesuai dengan hasil identifikasi tipe *blur*. Penelitian mencapai nilai akurasi 98,21% pada 1500 gambar tanpa *blur*, 98,33% pada 500 gambar dengan *Gaussian blur*, 99,3% pada 500 gambar dengan *motion blur*, dan 99% pada 500 gambar dengan *out-of-focus blur*.

Penelitian yang dilakukan oleh (G. S. Tran dkk., 2020) menggunakan paralelisasi dan pemanfaatan GPU untuk meningkatkan kecepatan identifikasi *blur* dengan *HaarWavelet transform* pada berbagai gambar dari *dataset* INRIA *Aerial Image Dataset*, Caltech-256 *Object Category*, INRIA Holidays, dan LabelMe. Menggunakan konsep komputasi paralel dengan CPU NVIDIA Tegra K1 4 × 2,3 GHz Cortex-A15 dan GPU NVIDIA Kepler “GK20a” (192 *cores* dengan 852MHz) penelitian mencapai kecepatan identifikasi 0,01290 detik dengan nilai *accuracy* 95,7%, *precision* 95,8%, dan *recall* 89,4%.

### *Fast Fourier Transform* (FFT)

*Fourier transform* adalah metode ekstraksi fitur yang digunakan dalam inferensi ada atau tidaknya *blur* pada gambar melalui frekuensi gelombang-gelombang sinus dan cosinus pada gambar dan *Fast Fourier transform* adalah algoritma *Fourier transform* yang disampel menjadi *Discrete Fourier transform* lalu dioptimasi untuk menurunkan kompleksitasnya hingga menjadi (Zhao dkk., 2019).





Gambar 2.3 Hasil Transformasi FFT

Sumber: (Hinami dkk., 2021)

Hasil transformasi seperti Gambar 2.3 merupakan pemetaan *magnitude* pada peta koordinat polar bersifat simetris dari gelombang sinus dan kosinus yang menyusun gambar dengan jari-jari menyatakan frekuensi gelombang dan sudut menyatakan rotasi gelombang (Hinami dkk., 2021). Identifikasi *blur* menggunakan distribusi frekuensi gelombang pada gambar dengan lebih banyaknya gelombang dengan frekuensi rendah mengindikasikan bahwa gambar memiliki kontras rendah dan memiliki *blur* (Perić dkk., 2022).

Penelitian yang dilakukan oleh (Perić dkk., 2022) menggunakan identifikasi *blur* dengan *Fast Fourier transform* pada gambar rel kereta yang diambil saat kereta berjalan. FFT dilakukan untuk *blur assessment* pada gambar yang diambil untuk meningkatkan kualitas *dataset* yang dibuat. Penelitian terhadap metode FFT mencapai nilai performa *accuracy* 84,3%, *precision* 67,8%, *recall* 86,9%, *F-score* 76,2%, dan waktu komputasi 305 detik pada 872 gambar berukuran 1024×768 *pixel* saat pengujian.

Penelitian yang dilakukan oleh (Pagaduan dkk., 2020) membandingkan performa beberapa metode identifikasi *blur* seperti FFT, HWT, *Laplacian operator*, *Modified Laplacian operator*, dan Tenengrad pada *dataset* gambar area perkantoran dengan *motion blur*, *out-of-focus blur*, dan *synthetic blur*. Penelitian yang dilakukan terhadap metode *Fast Fourier transform* mencapai nilai performa *accuracy* 93,5%, *precision* 100%, *recall* 87%, *F-Measure score* 93,048%, dan waktu identifikasi 6,2001 pada 200 gambar berukuran 640×480 *pixel* dengan CPU Intel Core i7-8750H @ 2,20GHz dan RAM sebesar 8,0 GB.

### *Laplacian* *Operator*

*Laplacian operator* adalah metode ekstraksi fitur yang digunakan dalam inferensi ada atau tidaknya *blur* pada gambar melalui hasil deteksi tepi pada gambar (Harron dkk., 2023). Gambar tepi yang diekstraksi digunakan untuk menghitung nilai-nilai seperti energi dan variasi dari *Laplacian* yang dibandingkan dengan sebuah *threshold* untuk menentukan klasifikasi *blur* gambar (Ali & Mahmood, 2018). Deteksi tepi *Laplacian operator* dilakukan menggunakan kernel yang telah dikomputasi seperti pada persamaan (2) (Bansal dkk., 2016).

|  |  |
| --- | --- |
|  | (2) |

*Laplacian operator* dapat menjadi fitur karena tepi yang didapat menunjukkan perubahan nilai warna (kontras) pada gambar dengan gambar yang memiliki *blur* akan memiliki lebih sedikit kontras sehingga lebih sedikit varians nilai tepi dan sebaliknya (Bansal dkk., 2016). Nilai varians yang dihitung dengan persamaan (3) digunakan karena varians meningkat dengan keberadaan titik bukan tepi dan titik tepi sehingga akan mengindikasi *blur* saat terdapat sedikit tepi pada gambar seperti pada gambar *blur* dan saat terdapat terlalu banyak tepi pada gambar seperti pada gambar dengan *noise* (Bansal dkk., 2016).

|  |  |
| --- | --- |
|  | (3) |

Keterangan:

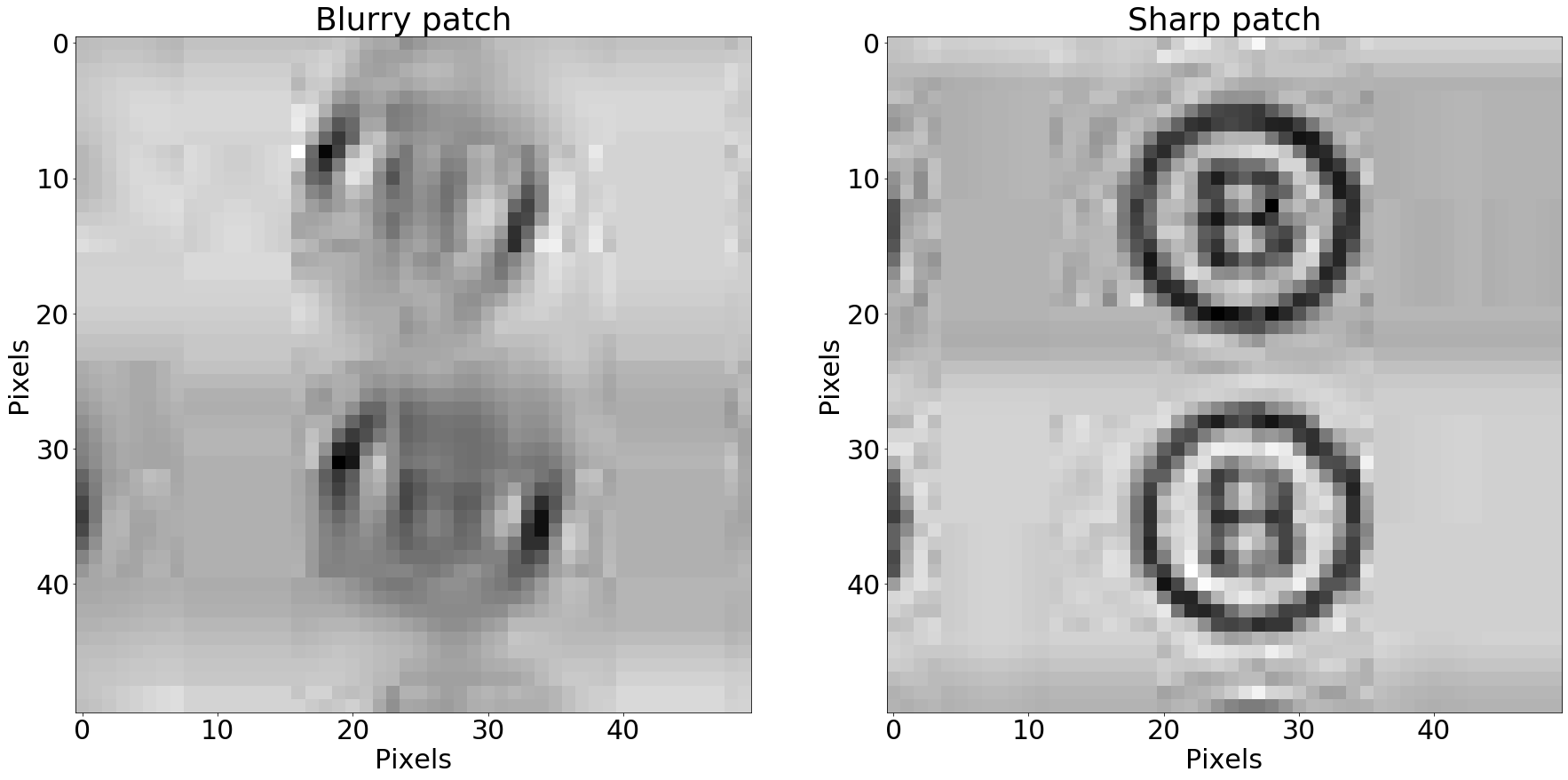
: Nilai varians populasi.

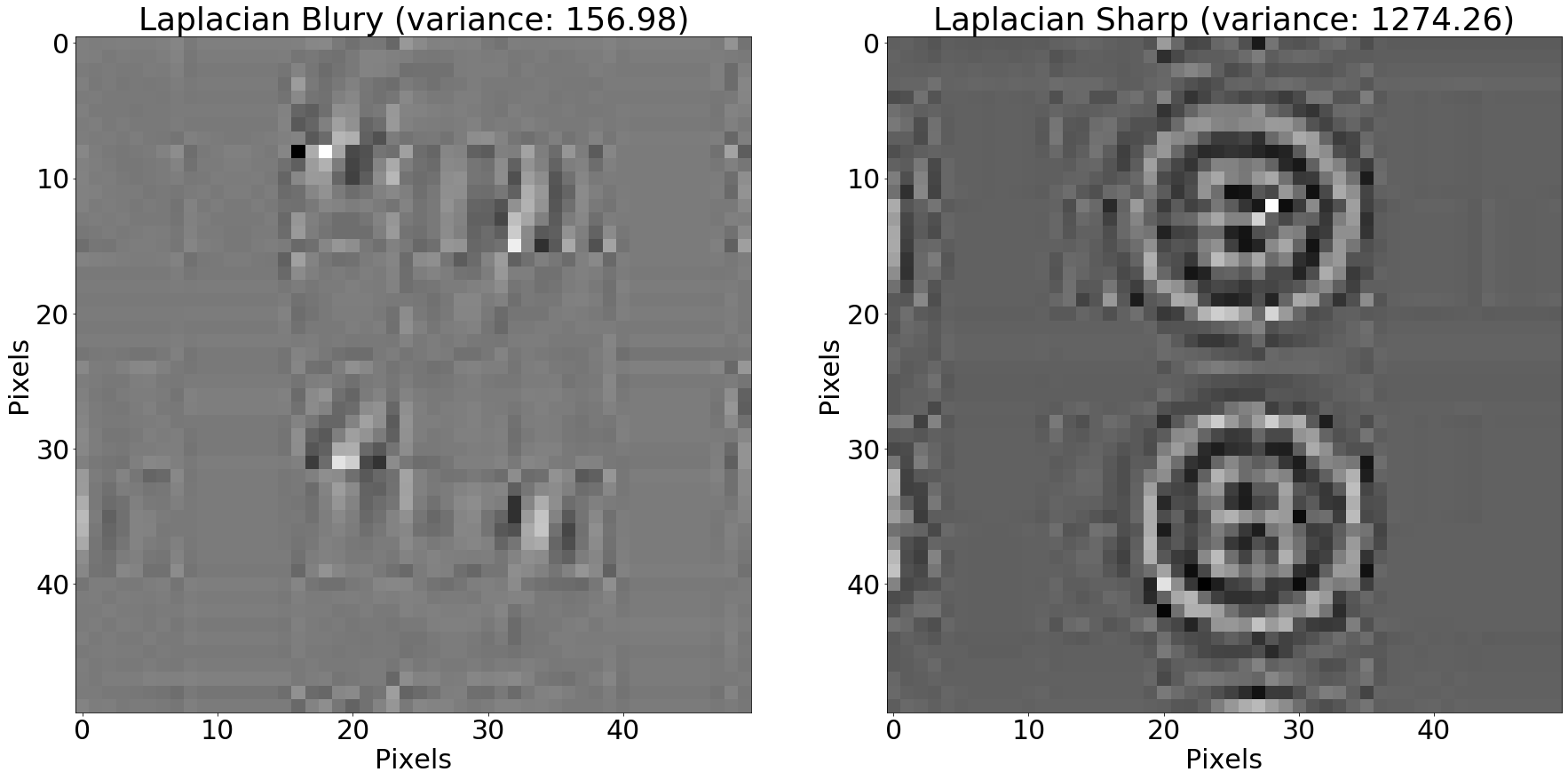
X : Nilai ke-i.

μ : Nilai rata-rata populasi.

N : Jumlah populasi.

Contoh hasil penerapan *laplacian operator* dan perhitungan variansnya dapat dilihat pada Gambar 2.4.





Gambar 2.4 Hasil *Laplacian Operator* dan Nilai Variansnya

Sumber: (*How to identify blurry images*, t.t.)

Penelitian yang dilakukan oleh (Perić dkk., 2022) menggunakan identifikasi *blur* dengan *Laplacian operator* pada gambar rel kereta yang diambil saat kereta berjalan. *Laplacian operator* dilakukan pada *dataset* untuk *quality control* untuk meningkatkan model *decision support system* yang akan dibangun selanjutnya. Penelitian terhadap metode *Laplacian operator* mencapai nilai performa *accuracy* 83,5%, *precision* 68,5%, *recall* 79,4%, *F-score* 73,5%, dan waktu komputasi 203 detik pada 872 gambar berukuran 1024×768 *pixel*.

Penelitian yang dilakukan oleh (Pagaduan dkk., 2020) membandingkan performa beberapa metode identifikasi *blur* seperti FFT, HWT, *Laplacian operator*, *Modified Laplacian operator*, dan Tenengrad pada *dataset* gambar area perkantoran dengan *motion blur*, *out-of-focus blur*, dan *synthetic blur*. Penelitian yang dilakukan terhadap metode *Laplacian operator* mencapai nilai performa *accuracy* 85,5%, *precision* 78,4%, *recall* 98%, *F-Measure score* 87,111%, dan waktu identifikasi 1,1482 pada 200 gambar berukuran 640×480 *pixel* dengan CPU Intel Core i7-8750H @ 2,20GHz dan RAM sebesar 8,0 GB.

## Analisis Pemilihan Solusi

Analisis pemilihan solusi dilakukan terhadap tiga metode yang dipilih untuk mendapat satu metode yang akan digunakan dalam *capstone project*. Analisis dilakukan dengan mempertimbangkan performa dan karakteristik dari masing-masing metode terhadap batasan aspek ekonomis, aspek manufakturabilitas, dan aspek sustainabilitas proyek. Untuk membantu dalam pertimbangan solusi digunakan hasil dari penelitian oleh (Pagaduan dkk., 2020) terhadap kinerja metode-metode dengan CPU Intel Core i7-8750H @ 2,20GHz dan 8,0 GB RAM pada Tabel 2.1 beberapa metode termasuk metode *HaarWavelet transform*, *Fast Fourier transform*, dan *Laplacian operator* yang dipilih.

Tabel 2.1 Perbandingan Performa Teknik Identifikasi *Blur*

Sumber: (Pagaduan dkk., 2020)

| **Solusi** | ***Accuracy* (%)** | ***Precision* (%)** | ***Recall* (%)** | ***F1-score* (%)** | **Waktu / gambar (*milisecond*)** |
| --- | --- | --- | --- | --- | --- |
| *HaarWavelet Transform* (HWT) | **97,0** | 99,0 | 95,0 | **96,938** | 301,850 |
| *Fast Fourier Transform* (FFT) | 93,5 | **100,0** | 87,0 | 93,048 | 310,005 |
| *Laplacian Operator* | 85,5 | 78,4 | **98,0** | 87,111 | **57,410** |

Aspek ekonomis dipertimbangkan dari waktu pelatihan model. Pelatihan model diestimasi melakukan 10 kali identifikasi dengan metode pada 4260 gambar pada *dataset Smartphone Document* *Quality Assurance* (SmartDoc-QA) seperti pada persamaan (4). Waktu pelatihan model kemudian dikalikan dengan biaya penggunaan GPU A100 Google Colab berdasarkan (McCormick, 2024) yang pada 22 April 2024 sebesar US$1.18 per jam (atau Rp18.880,00 per jam dengan kurs Rp16.000,00/US$) dengan persamaan (5).

|  |  |
| --- | --- |
|  | (4) |
|  | (5) |

Keterangan:

T : Waktu pelatihan model.

I : Jumlah iterasi (10).

D : Jumlah gambar untuk identifikasi (4260 gambar).

S : Kecepatan identifikasi dalam detik per gambar.

Price : Total estimasi biaya pelatihan model dalam Rupiah.

C : Biaya penggunaan GPU A100 Google Colab (Rp18.880,00 per jam).

Penjabaran biaya pelatihan model untuk metode *HaarWavelet transform*, *Fast Fourier transform*, dan *Laplacian operator* tertera pada Tabel 2.2. Ditemukan bahwa metode *Laplacian operator* memiliki estimasi biaya pelatihan terendah pada Rp12.819,00 per 10 iterasi.

Tabel 2.2 Estimasi Pelatihan Solusi

| **Solusi** | **Lama Pelatihan (jam)** | **Biaya Total** |
| --- | --- | --- |
| *HaarWavelet Transform* (HWT) | 3,572 | Rp67.439,00 |
| *Fast Fourier Transform* (FFT) | 3,668 | Rp69.251,00 |
| *Laplacian Operator* | 0,679 | Rp12.819,00 |

Aspek manufakturabilitas dipertimbangkan dari nilai kinerja model untuk mengukur apakah metode dapat digunakan untuk identifikasi dan waktu pelatihan model untuk mempertimbangkan waktu yang diperlukan untuk membuat perangkat lunak. Dari segi kinerja masing-masing metode memiliki ukuran kinerja unggul diatas metode lain, dengan HWT memiliki nilai *accuracy* dan *F-score* tertinggi, FFT memiliki nilai *precision* tertinggi, dan *Laplacian operator* memiliki nilai *recall* tertinggi. Dari segi waktu pelatihan diukur dengan persamaan (4) yang hasilnya dijabarkan pada Tabel 2.2 dengan metode *Laplacian operator* memiliki waktu pelatihan tercepat dengan waktu 0,679 jam untuk 10 iterasi pelatihan.

Dengan mempertimbangkan aspek sustainabilitas, metode *Laplacian operator* memiliki kecepatan identifikasi yang lebih cepat dibanding dengan metode HWT dan FFT, dengan waktu identifikasi 1,1482 detik untuk 200 gambar dibanding dengan 6,0370 detik untuk HWT dan 6,2001 detik untuk FFT. Ketiga solusi mampu memenuhi kebutuhan aspek sustainabilitas namun *Laplacian operator* dapat memenuhinya dengan lebih baik dengan waktu deteksi 1,1482 detik per gambar.

Melalui pertimbangan tersebut dipilih *Laplacian operator* sebagai metode yang digunakan. *Laplacian operator* memiliki kecepatan identifikasi tertinggi pada penelitian sehingga *Laplacian operator* dapat lebih baik memberi respons sebelum perangkat ditutup oleh pengirim gambar dan *Laplacian operator* memerlukan waktu pelatihan terendah sehingga memerlukan biaya penggunaan GPU yang lebih rendah dan waktu pembuatan model yang lebih rendah. Karakteristik ini mendukung *Laplacian operator* untuk lebih baik memenuhi fungsi sistem yang dibangun sebagai metode *filter* untuk mempercepat proses pengajuan kredit dibanding metode HWT dan FFT. Hasil analisis solusi dapat dilihat pada Tabel 2.3.

Tabel 2.3 Analisis Pemilihan Solusi berdasarkan Tiga Aspek

| **Solusi** | **Karakteristik Pendekatan Berdasarkan Tiga Aspek Analisis** | **Solusi Terpilih** |
| --- | --- | --- |
| Pendekatan-1  *HaarWavelet Transform* (HWT) | **Aspek Ekonomis** Metode HWT memerlukan biaya pelatihan Rp67.439,00 dengan GPU A100 Google Colab.  **Aspek Manufakturabilitas** Metode HWT memerlukan lama pelatihan 3,572 jam pada CPU Intel Core i7-8750H @ 2,20GHz dan 8,0 GB RAM.  **Aspek Sustainabilitas** Metode HWT memiliki waktu deteksi 6,0370 detik per gambar pada CPU Intel Core i7-8750H @ 2,20GHz dan 8,0 GB RAM. | - |
| Pendekatan-2  *Fast Fourier Transform* (FFT) | **Aspek Ekonomis** Metode FFT memerlukan biaya pelatihan Rp69.251,00 dengan GPU A100 Google Colab.  **Aspek Manufakturabilitas** Metode FFT memerlukan lama pelatihan 3,668 jam pada CPU Intel Core i7-8750H @ 2,20GHz dan 8,0 GB RAM.  **Aspek Sustainabilitas** Metode FFT memiliki waktu deteksi 6,2001 detik per gambar pada CPU Intel Core i7-8750H @ 2,20GHz dan 8,0 GB RAM. | - |
| Pendekatan-3  *Laplacian Operator* | **Aspek Ekonomis** Metode *Laplacian operator* memerlukan biaya pelatihan **Rp12.819,00** dengan GPU A100 Google Colab.  **Aspek Manufakturabilitas** Metode *Laplacian operator* memerlukan lama pelatihan **0,679** jam pada CPU Intel Core i7-8750H @ 2,20GHz dan 8,0 GB RAM.  **Aspek Sustainabilitas** Metode *Laplacian operator* memiliki waktu deteksi **1,1482** detik per gambar pada CPU Intel Core i7-8750H @ 2,20GHz dan 8,0 GB RAM. | ✓ |

# METODOLOGI

Bab metodologi merincikan proses yang dilakukan dalam pembangunan perangkat lunak, seperti proses identifikasi permasalahan yang diangkat, proses analisis solusi beserta alternatifnya, proses yang membentuk rancangan sistem yang akan dibangun, proses pengumpulan *dataset* yang digunakan untuk membangun perangkat lunak, proses implementasi perangkat lunak dengan komponen identifikasi *blur* dan aplikasinya, proses pengujian yang dilakukan untuk pengidentifikasi dan aplikasi, dan hasil yang diharapkan pada akhir proses-proses tersebut.

## Identifikasi Masalah

Identifikasi masalah terkait identifikasi *blur* pada gambar dokumen teks dilakukan dengan membaca jurnal penelitian terkait seperti (Kieu dkk., 2015) dan melalui wawancara dengan 3 perusahaan yang bergerak dibidang *development* dan penjualan rumah yang terdiri dari PT Sukses Wijaya Adikarya, PT Cahaya Sanubari Sakti, dan PT Kesuma Maju Sumatera.

## Analisis Solusi

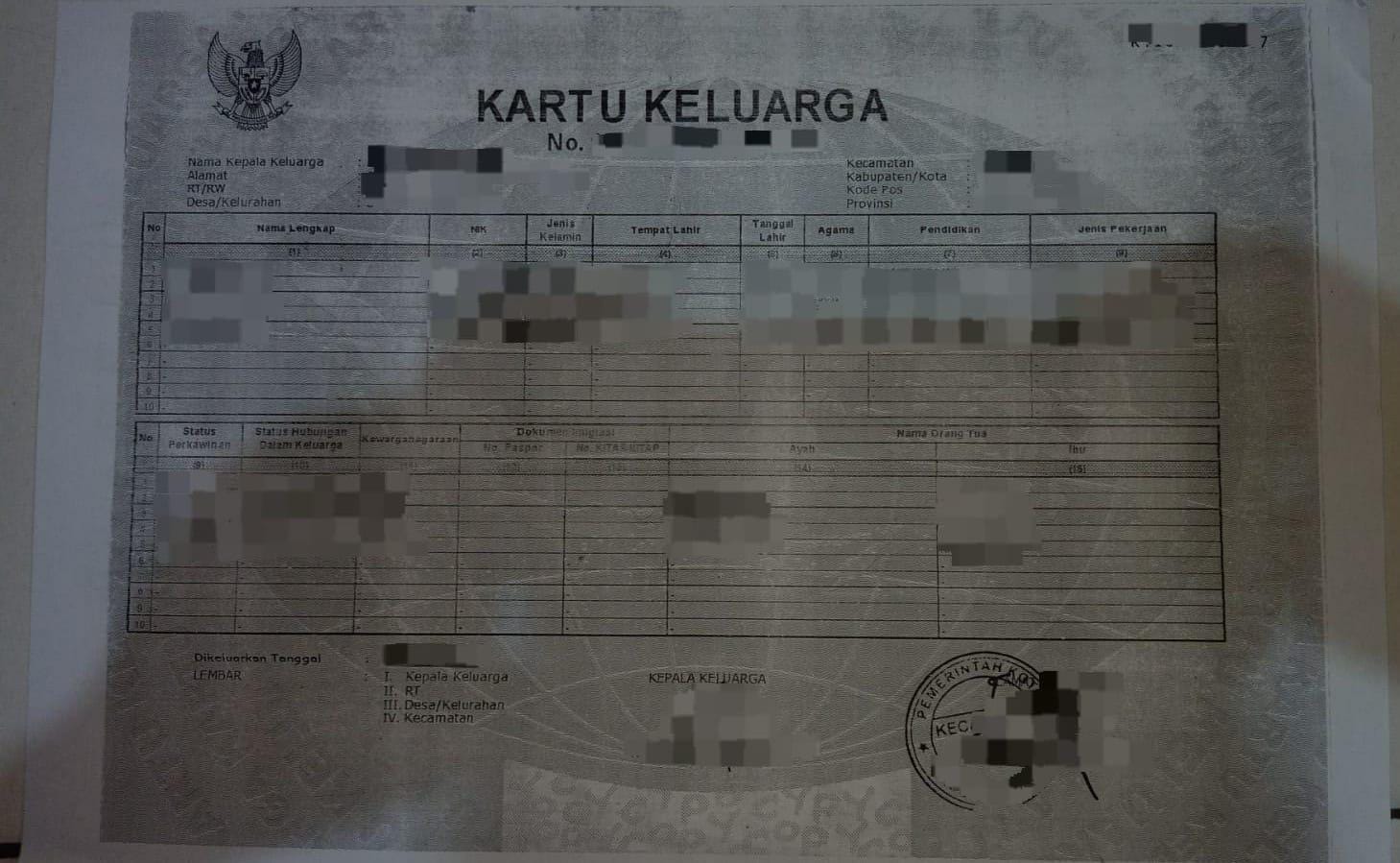
Analisis solusi dilakukan untuk mencari solusi berdasarkan masalah yang didapat dari identifikasi masalah. Solusi didapat dari mencari dan menganalisis penelitian terkait permasalahan. Tiga solusi dipilih dari solusi-solusi yang ditemukan berdasarkan beberapa faktor seperti karakteristik metode, performa solusi, dan aspek lain yang terkait dengan solusi. Solusi yang dipilih dianalisis lebih lanjut dari 3 aspek solusi, yaitu aspek ekonomis, aspek manufakturabilitas, dan aspek sustainabilitas. Solusi yang memenuhi ketiga aspek tersebut dengan paling baik kemudian dipilih sebagai metode yang digunakan.

## Pengumpulan Data

Pengumpulan data pada *capstone project* dilakukan dengan mencari *dataset* yang dapat menjadi pengganti data dokumen asli. Data asli tidak dapat digunakan sebab data memuat informasi sensitif dari klien perusahaan, seperti Kartu Tanda Kependudukan (KTP) dan Kartu Keluarga (KK). Contoh KTP dan KK yang menjadi gambar input untuk identifikasi *blur* dilampirkan pada Gambar 3.1 dan Gambar 3.2.



Gambar 3.1 Contoh Gambar KTP



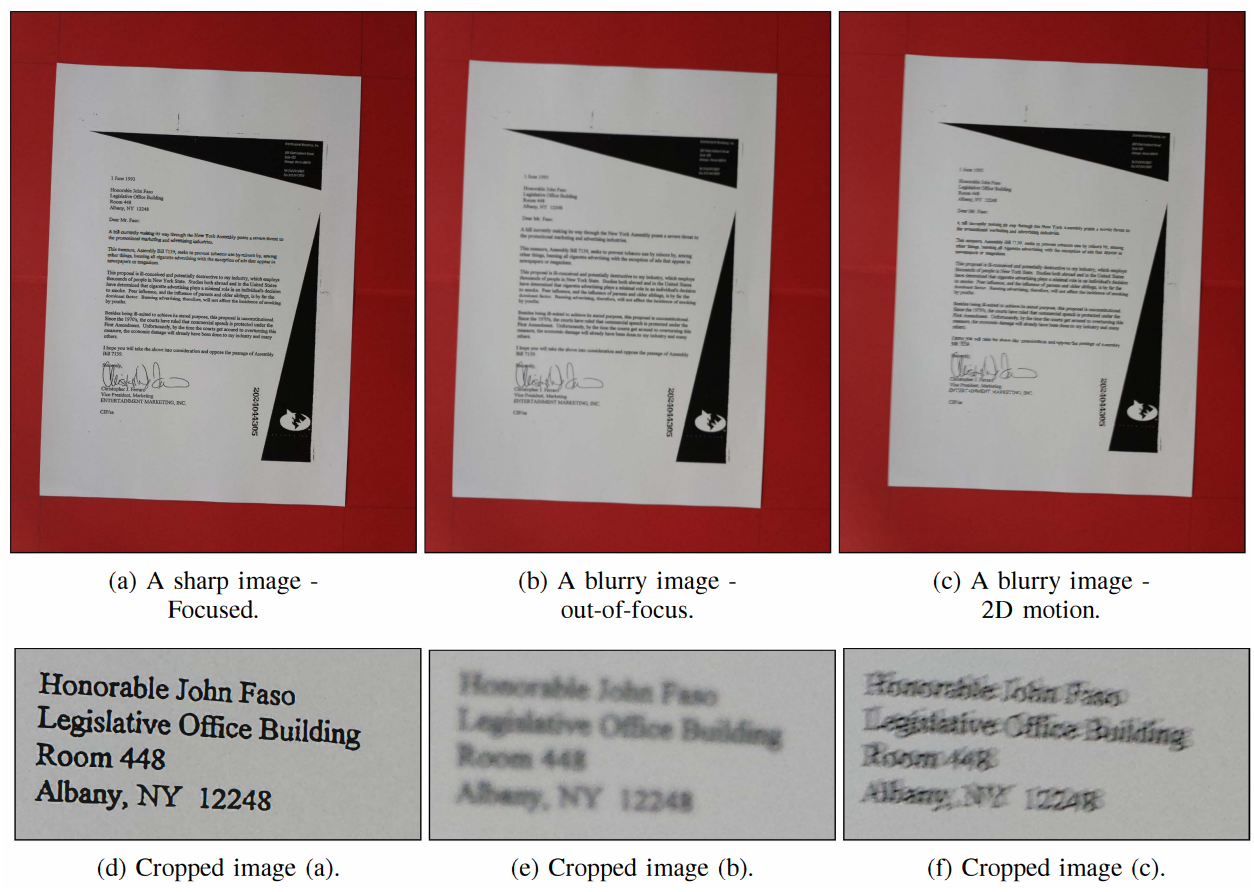
Gambar 3.2 Contoh Gambar KK

Pembuatan *dataset* pengganti dilakukan oleh (Nayef dkk., 2015) yang menyusun *dataset* SmartDoc-QA. SmartDoc-QA berisi 4260 gambar dokumen yang memiliki *blur* dan tidak memiliki *blur* yang diambil menggunakan perangkat *smartphone* difoto dalam variasi pengambilan gambar berbeda seperti sudut pengambilan foto, kondisi pencahayaan, dan tipe *blur* yang ada. SmartDoc-QA dibuat untuk merepresentasikan distorsi pengambilan gambar yang muncul karena keadaan pengambilan, seperti pencahayaan, *out-of-focus blur*, *motion blur*, dan posisi kamera. Variasi pengambilan foto pada SmartDoc-QA dipaparkan pada Tabel 3.1.

Tabel 3.1 Variasi Pengambilan Gambar Dokumen SmartDoc-QA

| **Parameter** | **Variasi** | **Keterangan** |
| --- | --- | --- |
| Dokumen | Dokumen kontemporer | Dicetak dari wiki-books dan resep masakan |
| Dokumen administratif lama | Dicetak dari *dataset* Tobacco, memiliki *salt and pepper noise*, beberapa dokumen memiliki area teks sempit dan tulisan tangan |
| Struk pembelian | Dari toko-toko berbeda, dalam bahasa Perancis |
| *Smartphone* | Samsung Galaxy S4 |  |
| Nokia Lumia 920 |  |
| Pencahayaan | Siang hari | Tanpa cahaya buatan |
| Siang hari + lampu neon langit-langit |  |
| Malam hari + lampu meja |  |
| Lampu meja + area kena bayangan | Lampu meja satu-satunya sumber cahaya |
| Lampu meja + bayangan *grid* | Lampu meja satu-satunya sumber cahaya |
| *Out-of-focus blur* | Fokus 22cm |  |
| Fokus 21cm |  |
| Fokus 20cm |  |
| Fokus 19cm |  |
| Fokus 18cm |  |
| *Motion blur* | Horizontal |  |
| Diagonal |  |
| Posisi | Tanpa rotasi | 0° |
| Rotasi horizontal / sumbu Y | -10°, -5°, 5° |
| Rotasi vertikal / sumbu X | -5°, 5°, 10° |

*Dataset* SmartDoc-QA dapat digunakan karena identifikasi *blur* tidak dilakukan pada isi konten dari dokumen, melainkan dari tampilan gambar berkaitan dengan ada tidaknya distorsi *blur*. SmartDoc-QA digunakan untuk menggantikan gambar dokumen dari klien untuk menjaga privasi klien serta mencegah terjadinya kebocoran data. Penggunaan SmartDoc-QA telah digunakan dalam penelitian serupa, seperti (Asad dkk., 2016; Chazalon dkk., 2017; Courtney, 2020, 2021; Kleber dkk., 2017; Li dkk., 2021; Nayef dkk., 2015; Rodin dkk., 2021). Gambar 3.3 berisi contoh gambar dokumen dengan *blur* dan tanpa *blur* pada *dataset* SmartDoc-QA.

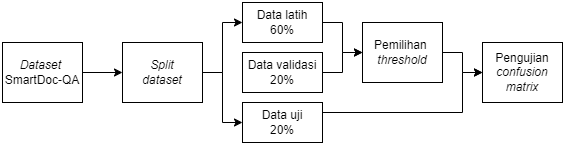


Gambar 3.3 Contoh Gambar Dokumen pada *Dataset* SmartDoc-QA

Sumber: (Nayef dkk., 2015)

## Pengembangan Model

Pengembangan model dimulai dengan mempersiapkan *dataset* SmartDoc-QA yang akan digunakan sebagai pengganti untuk data lapangan nyata karena data tersebut berupa dokumen sensitif. Selanjutnya dilakukan pemisahan data yang akan dilakukan untuk pelatihan dan data yang akan dilakukan untuk pengujian dengan rasio 60% data latih, 20% data validasi, dan 20% data uji. *Threshold* digunakan sebagai *classifier* gambar *blur* dan tanpa *blur* didapat dari menjabarkan hasil nilai *variance Laplacian operator* gambar-gambar dokumen dan memilih nilai yang dapat memisahkan gambar dengan *blur* dan gambar tanpa *blur*. Pengujian dilakukan terhadap data uji untuk menentukan kinerja perangkat lunak. Proses pembuatan model klasifikasi dilakukan dengan proses *supervised learning* dan diilustrasikan pada Gambar 3.4.



Gambar 3.4 Pengembangan Model

Metode yang digunakan untuk melakukan klasifikasi dari hasil *image processing* adalah dengan *thresholding*. *Thresholding* adalah metode klasifikasi yang dilakukan dengan membandingkan hasil ekstrasi fitur gambar terhadap sebuah nilai yang telah ditentukan (de Luna dkk., 2019). *Thresholding* digunakan pada penelitian serupa seperti oleh (Bansal dkk., 2016) karena identifikasi *blur* dengan *Laplacian operator* dilakukan melalui satu nilai hasil *image processing* yaitu *variance* dari gambar sehingga pemanfaatannya dilakukan untuk memperoleh kecepatan tambahan tanpa secara drastis menurunkan performa identifikasi. Pemilihan nilai *threshold* untuk mengklasifikasikan gambar tanpa dan dengan *blur* dilakukan dengan menganalisis hasil konvolusi dengan *Laplacian operator* terhadap gambar-gambar dokumen lalu memilih nilai yang dapat membedakan gambar yang tidak memiliki *blur* dan gambar yang memiliki *blur*. Pustaka seperti OpenCV dan SciPy digunakan untuk mengkomputasi, memvisualisasi, dan menganalisis nilai *variance laplacian operator* dan nilai *threshold*.

Ukuran gambar yang digunakan adalah 1600×1200 piksel untuk gambar *portrait* dan 1200×1600 piksel untuk gambar *landscape*. Resolusi tujuan saat mengubah PDF menjadi gambar adalah 96 *dots per inch* (DPI). Ukuran 1600×1200 dan resolusi 96 DPI digunakan mengikuti kualitas penggunggahan media WhatsApp.

## Pengembangan Perangkat Lunak

Pengembangan perangkat lunak dilakukan dengan metode *software development lifecyle* (SDLC) *iterative* yang dapat dilihat pada Gambar 3.5. Menurut (Arista & Novita, 2024) tahapan pada sebuah siklus metode *iterative* adalah:

1. Perencanaan

Pertama dilakukan identifikasi masalah dan penentuan ruang lingkup sistem melalui observasi dan wawancara.

1. Analisis

Pada tahap ini dipersiapkan data yang telah dikumpulkan dan dilakukan analisis terhadap permasalahan.

1. Perancangan

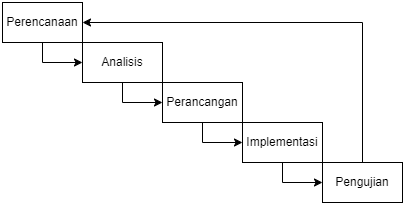
Dilakukan perancangan sistem secara keseluruhan untuk menentukan bagaimana gambaran sistem yang akan dibangun, bagaimana sistem tersebut akan berjalan, dan fungsionalitas lain yang dibutuhkan pada sistem.

1. Implementasi

Lalu dilakukan pembuatan aplikasi dengan bahasa pemrograman yang dipilih sesuai kebutuhan yang telah ditentukan.

1. Pengujian

Dilakukan pengujian sistem yang telah dibangun untuk menemukan kebutuhan yang belum terpenuhi atau bagian sistem yang dapat ditingkatkan.

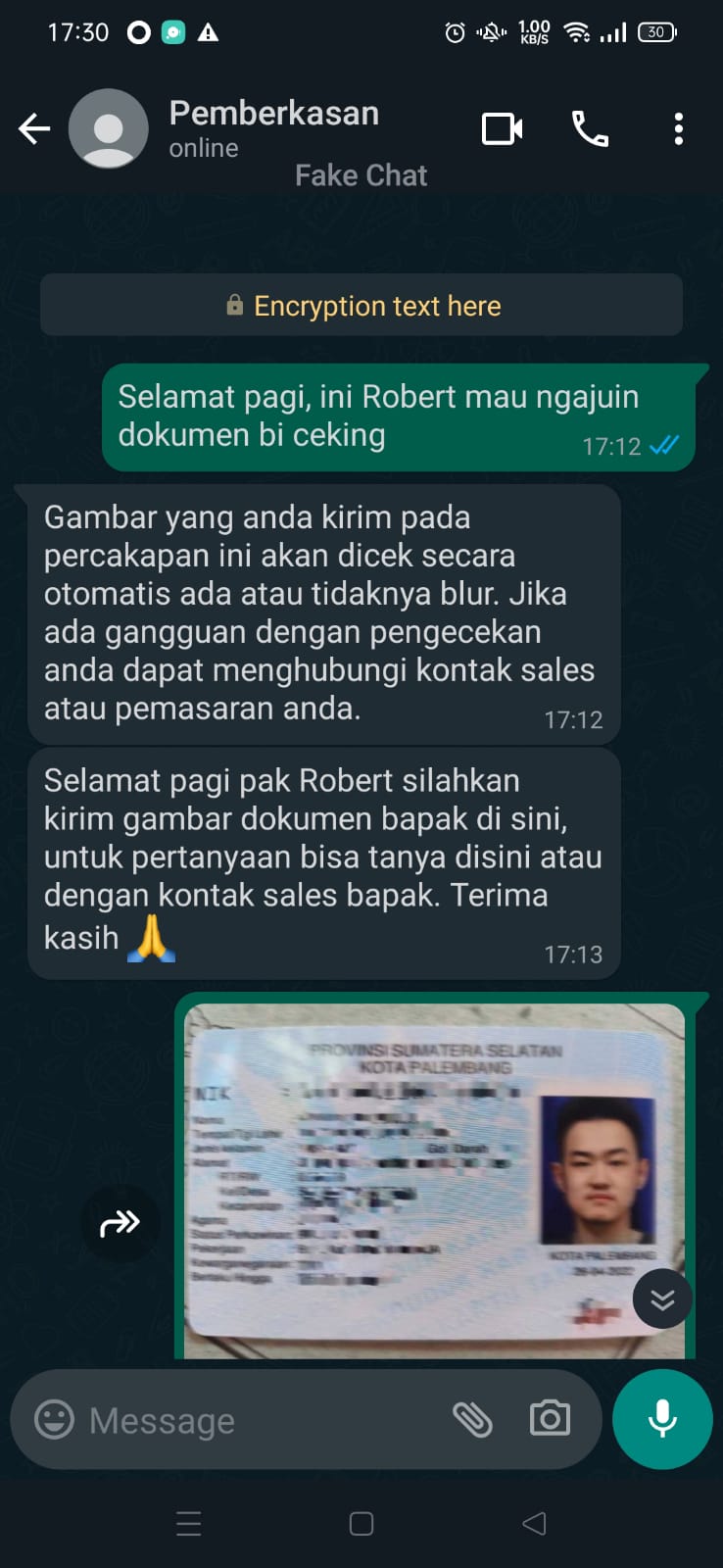
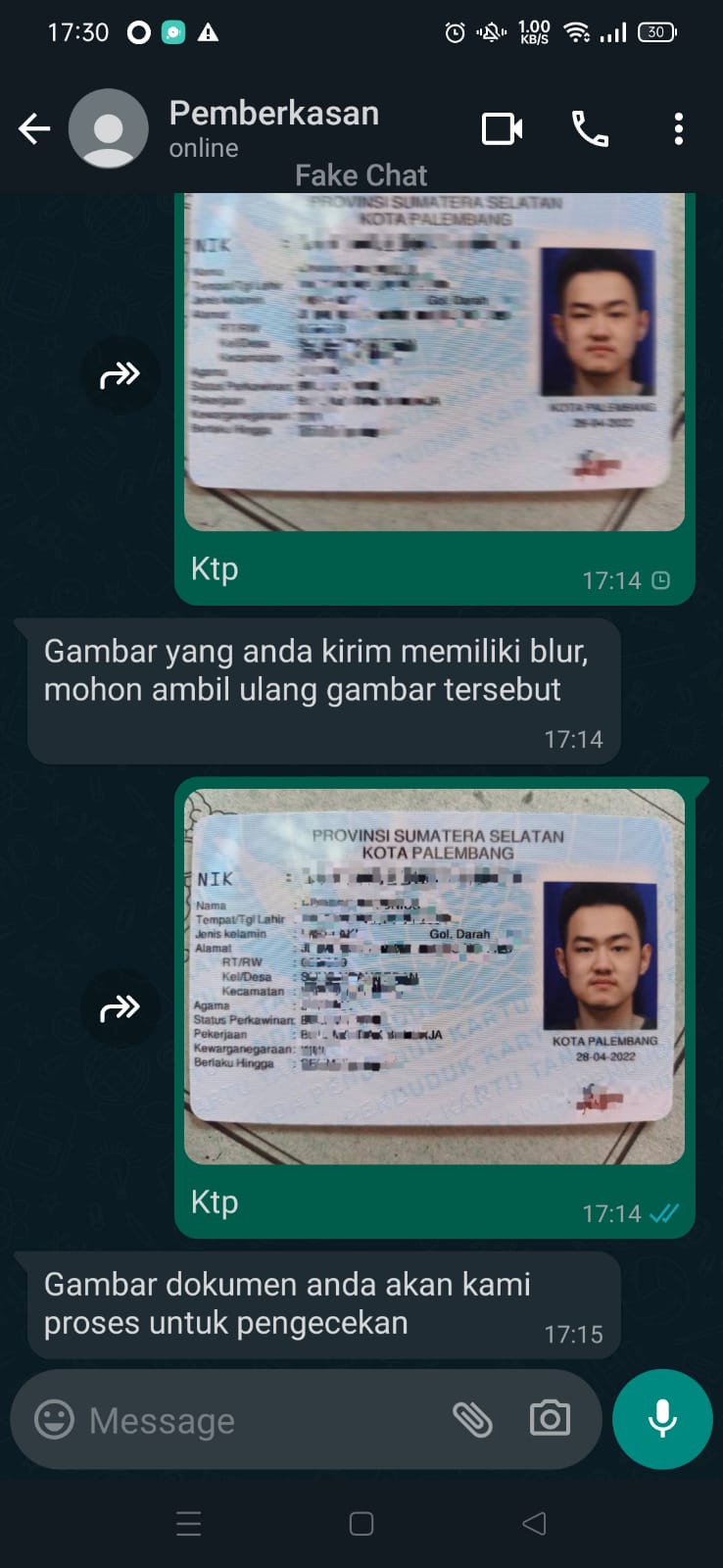


Gambar 3.5 Siklus Metode SDLC *Iterative*

Sumber: (Aliim dkk., 2023)

Metode *iterative* digunakan untuk memanfaatkan karakteristik siklis untuk menghasilkan sebuah *minimum viable product* (MVP) sebelum melakukan pengujian dan karakteristik evaluatif untuk menerapkan penyesuaian berdasarkan hasil pengujian untuk mencapai hasil yang diharapkan.

Sistem yang dibangun menggunakan WhatsApp sebagai *interface* dengan pengguna. Identifikasi *blur* dilakukan pada sebuah percakapan WhatsApp saat klien mengirim gambar dokumen. Klien diberi imbauan bahwa identifikasi *blur* dapat memiliki kesalahan karena keterbatasan pembuatan model dan dengan menggunakan percakapan WhatsApp klien dapat sekaligus menghubungi staf jika ada kesalahan dengan identifikasi *blur*. Gambar 3.6 merupakan gambar rancangan *prototype* perangkat lunak yang dibangun.

Gambar 3.6 Gambar *Prototype* Perangkat Lunak

## Pengujian

Pengujian dilakukan terhadap kedua komponen sistem dengan pengujian *black box* untuk komponen *service* WhatsApp dan pengujian performa model untuk komponen identifikasi *blur*. Pengujian *black box* adalah pengujian yang dilakukan untuk memastikan apakah semua fungsi perangkat lunak telah berjalan semestinya sesuai dengan kebutuhan fungsional yang didefinisikan (Fahrezi dkk., 2022). Pengujian dilakukan untuk memastikan bahwa *service* WhatsApp telah dapat menanggapi semua skenario yang dapat terjadi dalam interaksi dengan pengguna.

## Hasil

Hasil yang didapat adalah perangkat lunak yang dapat menerima gambar dokumen melalui akun WhatsApp lalu memberi umpan balik terhadap ada atau tidaknya *blur* pada gambar. Aplikasi dapat memberi umpan balik melalui WhatsApp pada percakapan yang sama dalam waktu kurang dari 3 detik dan dapat bekerja dengan biaya kurang dari 1 juta rupiah dan waktu selama 2 bulan.

# PERANCANGAN

Bab perancangan berisi rancangan untuk perangkat lunak yang dibangun, seperti spesifikasi dari perangkat lunak, rancangan untuk pengujian kinerja perangkat lunak, rancangan kerja perangkat lunak, verifikasi proses perancangan, dan jadwal pelaksanaan implementasi dan pengujian perangkat lunak.

## Spesifikasi Solusi

Spesifikasi solusi dari permasalahan berisi spesifikasi fungsi dan karakteristik yang memenuhi sifat abstrak, *verifiable*, *traceable*, dan tidak ambigu. Berikut spesifikasi solusi yang dibangun:

1. Perangkat lunak mampu mengidentifikasi gambar dokumen yang memiliki *blur* dari gambar dokumen yang tidak memiliki *blur*.
2. Perangkat lunak mampu mengidentifikasi gambar dokumen dalam *file* format PNG, JPEG, dan PDF.
3. Perangkat lunak mampu dijalankan melalui akun WhatsApp.
4. Perangkat lunak mampu melakukan identifikasi dalam waktu di bawah 3 detik.

## Rencana Pengujian

Rencana pengujian yang diajukan adalah pengujian kinerja dengan *confusion matrix* untuk model dan pengujian *black box* dan *stress test* untuk sisi perangkat lunak. Pengujian ketepatan model dilakukan dengan menggunakan *confusion matrix* untuk menghitung *accuracy*, *precision*, *recall*, dan *F1-score* model. Pengujian *stress test* sistem dilakukan dengan mengukur kemampuan perangkat lunak dan selisih waktu penerimaan pesan *input* dengan pengiriman pesan *output*. Pengujian *black box* dilakukan untuk melihat hasil *output* dari perangkat lunak dengan *input* yang ditentukan sesuai pada Tabel 4.1.

Tabel 4.1 Rencana Pengujian *Black Box*

| **No.** | **Pengujian** |
| --- | --- |
| 1. | Identifikasi gambar dokumen bertipe *file* PNG |
| 2. | Identifikasi gambar dokumen bertipe *file* JPEG |
| 3. | Identifikasi gambar dokumen bertipe file PDF berisi satu halaman |
| 4. | Identifikasi gambar dokumen bertipe file PDF berisi lebih dari satu halaman |
| 5. | Identifikasi gambar dokumen dengan kategori tidak *blur* |
| 6. | Identifikasi gambar dokumen dengan kategori *blur* |
| 7. | Identifikasi gambar dokumen terpotong |
| 8. | Identifikasi gambar dokumen terkena pantulan cahaya berlebihan |
| 9. | Identifikasi gambar dokumen tertutupi oleh objek lain |
| 10. | Identifikasi gambar dokumen terlipat |
| 11. | Identifikasi gambar dokumen dengan tulisan tangan |
| 12. | Identifikasi gambar dokumen dalam pencahayaan sangat rendah (dibawah 50 lux) |
| 13. | Identifikasi gambar dokumen dalam pencahayaan sangat tinggi (diatas 400 lux) |
| 14. | Menerima pesan teks |
| 15. | Menerima pesan selain teks, file JPEG, PNG, dan PDF |

Pengujian performa identifikasi dilakukan dengan pengujian *confusion matrix* dan dengan pengujian kecepatan sistem. Pengujian *confusion matrix* dilakukan untuk mendapat metrik seperti *accuracy*, *precision*, *recall*, dan *Fβ-score* melalui persamaan (6), (7), (8), (9), dan (10) (Pagaduan dkk., 2020). *Fβ-score* yang digunakan adalah β = 0.5 untuk memberi bobot 2 kali lipat lebih pada nilai *precision* daripada nilai *recall.* Pengujian *stress test* kecepatan sistem dilakukan dengan mengirim banyak *request* secara sekaligus lalu mengukur lama *request* melalui selisih waktu penerimaan respons sistem dengan waktu pengiriman gambar oleh klien pada *header* pesan seperti pada persamaan (11).

|  |  |
| --- | --- |
|  | (6) |
|  | (7) |
|  | (8) |
|  | (9) |
|  | (10) |
|  | (11) |

Keterangan:

TP : *true positive*, gambar *blur* yang diprediksi sebagai gambar *blur*.

FN : *false negative*, gambar *blur* yang diidentifikasi sebagai gambar tajam.

FP : *false positive*, gambar tajam yang diprediksi sebagai gambar *blur*.

TN : *true negative*, gambar tajam yang diidentifikasi sebagai gambar tajam.

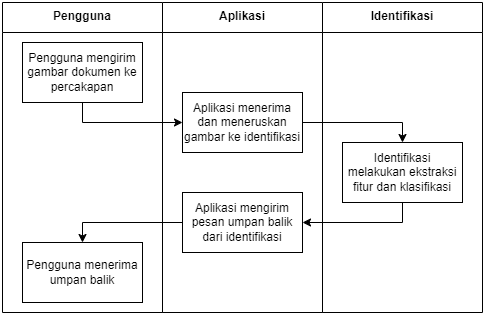
T*total* : durasi waktu eksekusi identifikasi *blur*.

T*sent* : waktu pengiriman gambar.

T*received* : waktu penerimaan gambar.

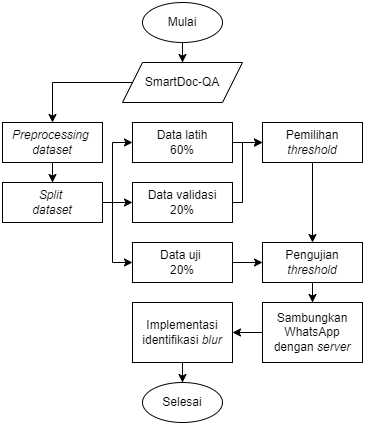
## Perancangan Sistem

Perancangan sistem dilakukan untuk mencapai tujuan *capstone project*, yaitu membuat perangkat lunak yang dapat mengidentifikasi dan memberi umpan balik dari ada atau tidaknya *blur* pada gambar dokumen. Gambar 4.1 menunjukkan tahapan proses identifikasi *blur* oleh perangkat lunak.



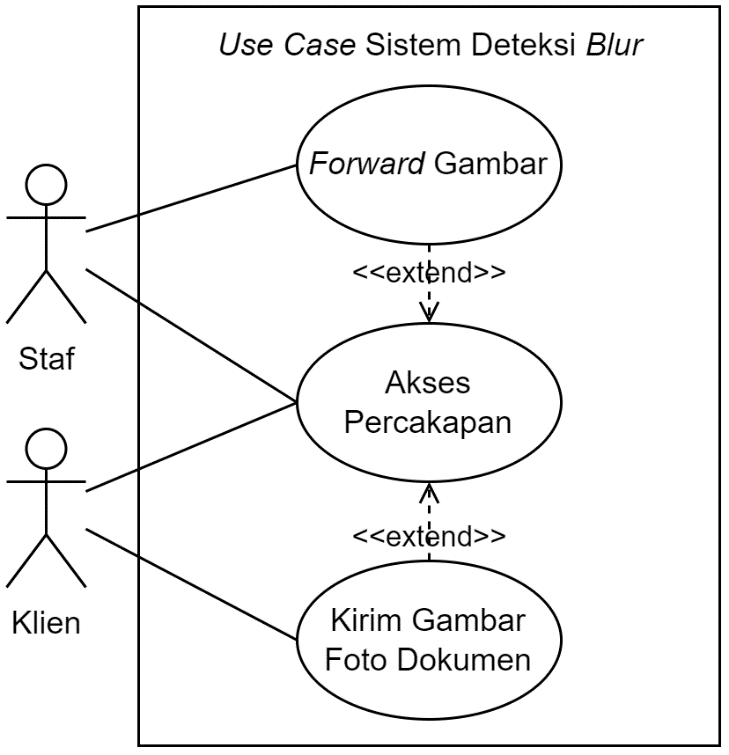
Gambar 4.1 Alur Kerja Perangkat Lunak

Implementasi komponen identifikasi *blur* dilakukan dengan Python dengan pustaka OpenCV, Flask, dan Pickles. Model konstan untuk *thresholding* yang dihasilkan disimpan menjadi suatu *file* sehingga dapat digunakan saat *server* dipanggil. Implementasi aplikasi *service* WhatsApp akan dilakukan *hosting* pada layanan Flask dan disambungkan kepada akun WhatsApp melalui Cloud API yang disediakan oleh WhatsApp Business Platform. Aplikasi *service* akan memanggil komponen identifikasi *blur* saat diterima gambar masuk lalu memberi umpan balik berdasarkan hasil identifikasi model. Implementasi model dan perangkat lunak tertera pada Gambar 4.2.



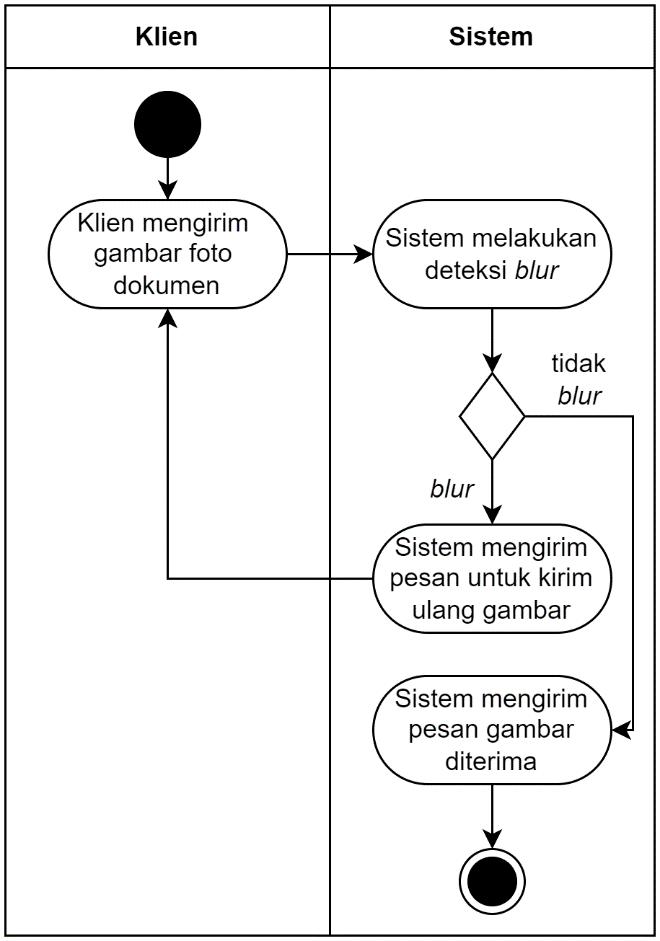
Gambar 4.2 Tahapan Perancangan Sistem

“Klien” yang berupa klien-klien perusahaan yang ingin mengirim gambar dokumen guna pengajuan kredit dan “Staf” yang berupa karyawan yang bertugas memantau, menerima, dan mengelola berkas-berkas untuk pengajuan kredit. Kasus penggunaan sistem oleh “Klien” dan “Staf” diilustrasikan dengan *use case diagram* pada Gambar 4.3.



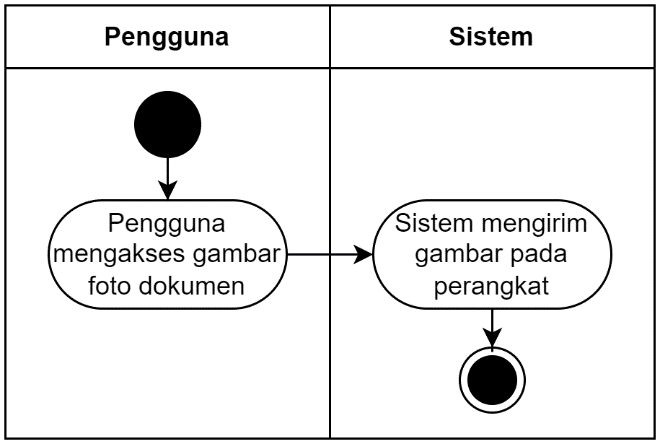
Gambar 4.3 *Use Case Diagram* Sistem Identifikasi *Blur*

Klien pertama membuka dialog pemilihan gambar WhatsApp dan mengirim gambar dokumen yang akan diproses. *File* yang dikirim kemudian akan diterima oleh *server* Flask dengan menggunakan *webhook* dari WhatsApp Cloud API. *Server* Flask kemudian melakukan ekstrasi fitur dengan *Laplacian operator* lalu melakukan klasifikasi. Hasil identifikasi dari model kemudian akan digunakan untuk menentukan pesan apa yang akan menjadi umpan balik untuk pengguna. *Activity diagram* proses interaksi klien dengan perangkat lunak dapat dilihat pada Gambar 4.4.



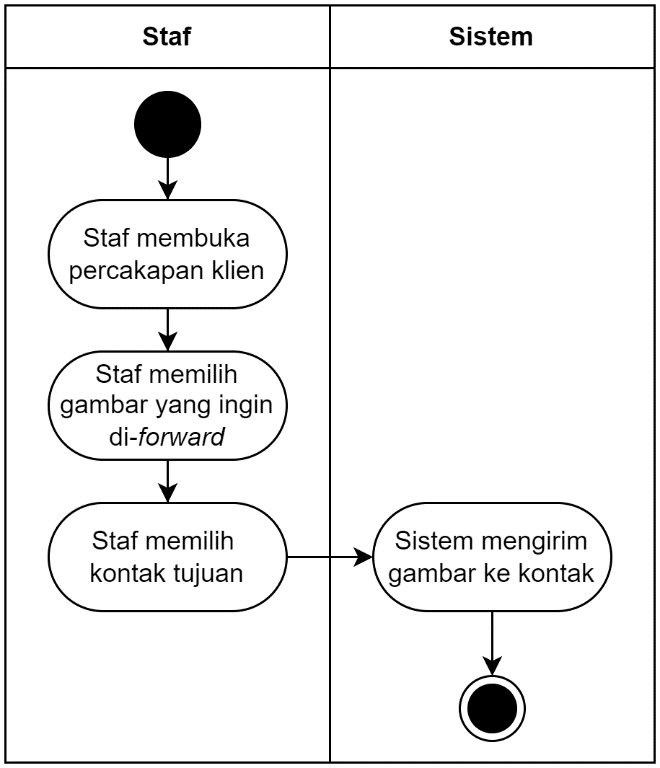
Gambar 4.4 *Activity Diagram* Pengiriman Gambar oleh Klien

Staf dan klien dapat mengakses percakapan dan gambar dokumen yang dikirim melalui aplikasi WhatsApp. *Activity diagram* proses interaksi pengguna dengan perangkat lunak dapat dilihat pada Gambar 4.5.



Gambar 4.5 *Activity Diagram* Pengaksesan Percakapan oleh Pengguna

Staf dapat memajukan gambar dokumen yang dikirim klien menggunakan fungsionalitas *forward* dari WhatsApp. *Activity diagram* proses interaksi pengguna dengan perangkat lunak dapat dilihat pada Gambar 4.6.



Gambar 4.6 *Activity Diagram Forward* Gambar oleh Pengguna

## Verifikasi Perancangan

Verifikasi perancangan dilakukan dengan membuat daftar semua atribut metode yang dilakukan, menjelaskan setiap atribut, dan memberi tanda centang terhadap daftar atribut seperti pada Tabel 4.2.

Tabel 4.2 Verifikasi dan Bukti dalam Proses Perancangan

| **No.** | **Item** | **Verifikasi** | **Keterangan** |
| --- | --- | --- | --- |
| 1. | SmartDoc-QA | ✓ | *Dataset* yang disediakan oleh (Nayef dkk., 2015) |
| 2. | *Laplacian filter* 3×3 | ✓ | *Kernel* konvolusi yang digunakan untuk mengolah gambar berdasarkan *Laplacian operator* |
| 3. | *Split dataset* | ✓ | Pembagian *dataset* dengan rasio 60% data latih, 20% data validasi, dan data uji 20% |
| 4 | *Oversampling* | ✓ | Memperbaiki *class imbalance* supaya model tidak *bias* ke salah satu kelas dengan jumlah sampel lebih |
| 5. | *Thresholding* | ✓ | Metode klasifikasi yang digunakan pada penelitian yang mengklasifikasikan gambar berdasarkan nilai varians hasil ekstraksi fitur *Laplacian operator* |
| 6. | *Black box* | ✓ | Metode pengujian perangkat lunak |
| 7. | *Confusion matrix* | ✓ | Metode pengukur kinerja model |
| 8. | *Timer* | ✓ | Metode pengukur kinerja sistem |
| 9. | WhatsApp Cloud API | ✓ | API yang disediakan WhatsApp yang digunakan untuk menghubungkan *server* dengan akun WhatsApp |

# IMPLEMENTASI DAN PENGUJIAN

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud dolor Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

## Implementasi Perancangan

Implementasi perangkat lunak dibagi menjadi 2 tahap yaitu pembuatan klasifikasi apakah gambar memiliki *blur* atau tidak berdasarkan hasil ekstraksi fitur *Laplacian operator* dan pembuatan server *backend* Flask yang terintegrasi dengan WhatsApp Cloud API untuk membaca dan mengirim pesan pada WhatsApp.

### Pemilihan *Threshold* *Laplacian operator*

Identifikasi *blur* pada gambar dokumen teks dilakukan dengan mengekstraksi fitur nilai varians dari hasil pemberlakuan *Laplacian operator* pada gambar dokumen teks lalu melakukan klasifikasi biner berdasarkan fitur tersebut. Klasifikasi dilakukan dengan memilih *threshold* yang dapat memisahkan gambar *blur* dan tidak *blur* berdasarkan nilai varians *Laplacian operator* dari gambar. *Threshold* dipilih melalui analisis yang mempertimbangkan konteks dari klasifikasi, yaitu *cost* dari *false positive* (gambar tidak *blur* diklasfikasikan sebagai gambar *blur*) lebih tinggi dari *cost* dari *false negative* (gambar *blur* diklasifikasikan sebagai gambar tidak *blur*).

#### Ekstraksi Fitur dari *Dataset*

Untuk melakukan pemilihan *threshold* yang tepat untuk mengklasifikasikan gambar diperlukan nilai varians *Laplacian operator* dari gambar-gambar di dalam *dataset* SmartDoc-QA. Fitur ini diperoleh dengan menjalankan ekstraksi fitur klasifikasi pada *dataset* dan menyimpannya dalam sebuah *file* *Comma Separated Values* (CSV). Kode program yang digunakan tercantum pada Gambar 5.1.

from imutils import paths

import argparse

import csv

import cv2

from tqdm import tqdm

def variance\_of\_laplacian(image):

    return cv2.Laplacian(image, cv2.CV\_64F).var()

ap = argparse.ArgumentParser()

ap.add\_argument("-i", "--images", required=True,

help="path to input directory of images")

args = vars(ap.parse\_args())

with open('lap\_var\_values.csv', 'w', newline='') as csvfile:

    writer = csv.writer(csvfile)

    writer.writerow(['lap\_var','actual','phone', 'width'])

    # loop over the input images

    with tqdm(total=17040) as pbar:

        for image\_path in paths.list\_images(args["images"]):

            image\_name = image\_path.rsplit('\\', 1)[-1]

            if ("Mb" in image\_name or "Ob" in image\_name):

                actu = "Blurry"

            else:

                actu = "Not"

            if ("WP" in image\_name):

                phone = "Nokia"

            else:

                phone = "Samsung"

            image = cv2.imread(image\_path)

            original\_height, original\_width, \_ = image.shape

            aspect\_ratio = original\_height / original\_width

            gray = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)

            pbar.update(1)

            for image\_width in [original\_width, 400, 1600]:

                resized = cv2.resize(gray,

                    (image\_width, int(image\_width \* aspect\_ratio)))

                fm = variance\_of\_laplacian(resized)

                writer.writerow([fm, actu, phone, image\_width])

                pbar.update(1)

print("Finished exporting Laplacian variance values from dataset.")

Gambar 5.1 Kode Program Ekstraksi Fitur *Dataset*

Hasil ekstraksi fitur terhadap *dataset* SmartDoc-QA memuat 4 kolom, yaitu nilai varians *Laplacian operator* dari gambar, apakah gambar tersebut dilabelkan sebagai *blur* atau tidak, model *smartphone* yang digunakan untuk mengambil gambar, dan ukuran gambar. Hasil ekstraksi fitur ini akan menjadi *dataset* untuk menganalisis dan memilih nilai *threshold* untuk klasifikasi gambar dokumen teks dengan dan tanpa *blur*. Hasil ekstraksi fitur tertera pada Tabel 5.1.

Tabel 5.1 Hasil Ekstraksi Fitur *Dataset*

| index | lap\_var | actual | phone | width |
| --- | --- | --- | --- | --- |
| 0 | 392.94 | Not | Samsung | 4128 |
| 1 | 9686.41 | Not | Samsung | 400 |
| 2 | 4152.39 | Not | Samsung | 1600 |
| 3 | 98.09 | Blurry | Samsung | 4128 |
| 4 | 2502.01 | Blurry | Samsung | 400 |
| ... | ... | ... | ... | ... |
| 12775 | 3685.81 | Not | Nokia | 400 |
| 12776 | 1082.51 | Not | Nokia | 1600 |
| 12777 | 410.02 | Not | Nokia | 3264 |
| 12778 | 4887.17 | Not | Nokia | 400 |
| 12779 | 1380.19 | Not | Nokia | 1600 |

#### Persiapan *Runtime* Google Colaboratory

*Runtime* Google Colaboratory dipersiapkan dengan mengimpor modul-modul, menyetel modul yang digunakan, dan menginisialisasi fungsi pendukung. Kode untuk proses persiapan dilampirkan pada Gambar 5.2, Gambar 5.3, dan Gambar 5.4.

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

import seaborn as sns

import cv2

import ipywidgets as widgets

from ipywidgets import interact, interactive, fixed, interact\_manual

from IPython import get\_ipython

from IPython.display import display, HTML

from matplotlib\_inline.backend\_inline import set\_matplotlib\_formats

!pip install myst-nb

import myst\_nb

!pip install kaleido

import kaleido

import plotly

import plotly.graph\_objects as go

import plotly.express as px

from plotly.subplots import make\_subplots

import plotly.io as pio

import math, time, random, itertools

Gambar 5.2 *Import* Modul

np.seterr(divide='ignore', invalid='ignore')

# set up plotly defaults

pio.templates["book"] = go.layout.Template(

    layout=dict(

        margin=dict(l=10, r=10, t=10, b=10),

        autosize=True,

        width=350,

        height=250,

        xaxis=dict(showgrid=True),

        yaxis=dict(showgrid=True),

        title=dict(x=0.5, xanchor="center"),

    )

)

pio.templates.default = "simple\_white+book"

# set up matplotlib defaults

sns.set\_style("whitegrid")

plt.rcParams["figure.figsize"] = (4, 3)

# display options for numpy and pandas

np.set\_printoptions(threshold=20, precision=2, suppress=True)

pd.set\_option("display.max\_rows", 10)

pd.set\_option("display.max\_columns", 30)

pd.set\_option("display.precision", 2)

# stops scientific notation for pandas

# pd.set\_option('display.float\_format', '{:.2f}'.format)

# When two traces share bingroup, plotly thinks they're the same plot

\_clear = ["bingroup"]

Gambar 5.3 Setel Modul

def \_clear\_prop(trace, prop):

    if hasattr(trace, prop):

        trace.update({prop: None})

def \_clear\_props(traces):

    for trace in traces:

        for prop in \_clear:

            \_clear\_prop(trace, prop)

def plots\_in\_row(figures, width=700, height=250, \*\*kwargs):

    """Combine plotly figures side by side"""

    fig = make\_subplots(cols=len(figures), \*\*kwargs)

    fig.update\_layout(width=width, height=height)

    traces = [next(fig.select\_traces()) for fig in figures]

    \_clear\_props(traces)

    for i, trace in enumerate(traces):

        fig.add\_trace(trace, row=1, col=i + 1)

    return fig

def left\_right(left, right, width=700, height=250, \*\*kwargs):

    """Two plotly figures side by side"""

    return plots\_in\_row([left, right],

        width=width, height=height, \*\*kwargs)

Gambar 5.4 Inisialisasi Fungsi Utilitas

#### *Scope*

Bagian *scope* dimulai dengan menelah bentuk data yang dimiliki. Kode dan hasil memuat data hasil ekstraksi fitur dari *dataset* SmartDoc-QA tertera pada Gambar 5.5 dan Tabel 5.2.

ORIGINAL = 0

SMALL = 1

MEDIUM = 2

SAMSUNG = 0

NOKIA = 1

def get\_data(phone=SAMSUNG, width=ORIGINAL):

data = pd.read\_csv("https://raw.githubusercontent.com/roberika/  
 whatsapp-blur-detect-utils/refs/heads/main/lap\_var\_values.csv")

  match width:

    case 0: data = data[(data.width != 400) & (data.width != 1600)];

    case 1: data = data[(data.width == 400)];

    case 2: data = data[(data.width == 1600)];

    case \_: data = data;

  match phone:

    case 0: data = data[(data.phone == "Samsung")];

    case 1: data = data[(data.phone == "Nokia")];

    case \_: data = data;

  return data;

blur = get\_data(phone="all", width=ORIGINAL)

blur

Gambar 5.5 Kode Program Memuat *Dataset*

Tabel 5.2 Hasil Memuat Dataset

| index | lap\_var | actual | phone | width |
| --- | --- | --- | --- | --- |
| 0 | 392.94 | Not | Samsung | 4128 |
| 3 | 98.09 | Blurry | Samsung | 4128 |
| 6 | 30.85 | Blurry | Samsung | 4128 |
| 9 | 32.05 | Blurry | Samsung | 4128 |
| 12 | 414.79 | Not | Samsung | 4128 |
| ... | ... | ... | ... | ... |
| 12765 | 323.92 | Blurry | Nokia | 3264 |
| 12768 | 313.46 | Blurry | Nokia | 3264 |
| 12771 | 246.17 | Not | Nokia | 3264 |
| 12774 | 366.28 | Not | Nokia | 3264 |
| 12777 | 410.02 | Not | Nokia | 3264 |

Nilai yang dimuat adalah semua baris pada *dataset* yang diambil dengan *smartphone* Samsung ataupun Nokia dan semua baris pada *dataset* tanpa mengubah ukuran gambar. Nilai ini dimuat karena *threshold* akan dipilih atas hasil analisis terhadap gambar yang diambil oleh kedua *smartphone* untuk menambah variasi supaya *threshold* tidak hanya baik dipakai untuk gambar yang diambil dari satu model saja dan mengubah ukuran gambar secara exponensial mengubah nilai varians *Laplacian operator* gambar.

f1 = px.histogram(blur.query('actual == "Blurry"'), x="lap\_var",

                  histnorm='probability density', nbins=20)

f2 = px.histogram(blur.query('actual == "Not"'), x="lap\_var",

                  histnorm='probability density', nbins=40)

fig = left\_right(f1, f2, height=250, width=600)

fig.update\_xaxes(title\_text='Blurry image variance of laplacian',

                 row=1, col=1, range=[0, 900])

fig.update\_xaxes(title\_text='Not blurry image variance of laplacian',

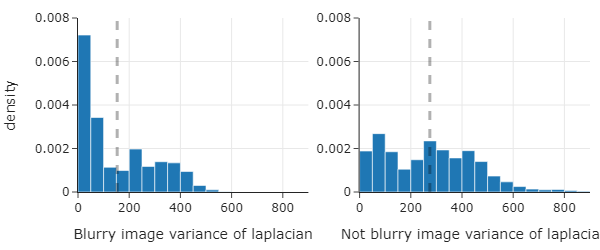
                 row=1, col=2)

fig.update\_yaxes(title\_text='density', row=1, col=1, range=[0, 0.008])

fig.update\_yaxes(row=1, col=2, range=[0, 0.008])

fig.show()

Gambar 5.6 Kode Program Menampilkan Distribusi Data



Gambar 5.7 Hasil Menampilkan Distribusi Data

Dari tampilan distribusi data yang diperoleh pada Gambar 5.6 dan Gambar 5.7 dapat dilihat bahwa distribusi nilai varians *Laplacian operator* cenderung condong ke kanan dalam rentang (0, ∞] dan nilai varians *Laplacian operator* gambar yang memiliki *blur* cenderung lebih rendah daripada gambar yang tidak memiliki *blur*. Observasi pertama mengindikasikan bahwa data lebih baik divisualisasi dengan skala logaritmik dan observasi kedua mengindikasikan bahwa data dapat dipisahkan dengan memiliki nilai *threshold* diantara kedua kelas tersebut. Observasi ketiga adalah ada lebih banyak gambar *blur* daripada gambar tidak *blur*.

def jitter(data, amt=0.2):

    return data + amt \* (np.random.rand(len(data)) - 0.5)

def blur\_as\_01(df):

    return [0 if i == "Not" else 1 for i in df["actual"]]

fig = px.scatter(blur, x=jitter(blur['lap\_var']),

                 y = jitter(blur\_as\_01(blur), 0.5),

                 log\_x=True, width=450, height=250,

                 color="actual")

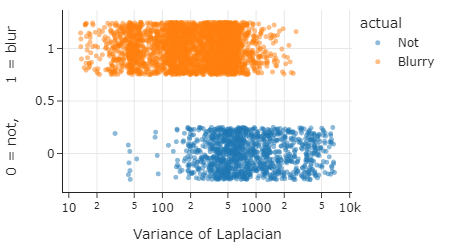
fig.update\_traces(marker=dict(opacity=0.5, size=5))

fig.update\_layout(xaxis\_title="Variance of Laplacian",

                  yaxis\_title="0 = not,       1 = blur")

fig

Gambar 5.8 Kode Program Menampilkan Dua Kelas Data



Gambar 5.9 Kode Program Menampilkan Dua Kelas Data

Ketiga observasi tersebut dikonfirmasi pada Gambar 5.8 dan Gambar 5.9. Data benar lebih baik ditampilkan dalam skala logaritmik, gambar yang memiliki *blur* benar cenderung memiliki nilai *Laplacian operator* yang lebih rendah dari gambar tanpa *blur*, dan gambar yang memiliki *blur* memiliki jumlah lebih besar dari gambar tanpa *blur*.

Untuk memperbaiki *class imbalance* pada data untuk mencegah bias dari jumlah data dalam pemilihan *threshold* dapat diberlakukan teknik *oversampling*, *undersampling*, atau hibrida keduanya. Dalam kasus ini karena *dataset* SmartDoc-QA memuat variasi pengambilan gambar dalam kondisi berbeda-beda dipilih teknik *oversampling* untuk menghindari penghapusan hasil skenario pengambilan foto tersebut dari data latih. Metode *oversampling* yang digunakan adalah menduplikat secara acak gambar dengan kelas tanpa *blur* supaya seimbang rasio data dengan dan tanpa *blur*.

Observasi keempat yang dapat diambil adalah walau kelas dapat dipisahkan dengan sebuah *threshold* terdapat *overlap* dalam nilai fitur kedua kelas. Observasi ini mengindikasikan bahwa akan ada sebuah *trade-off* dalam memilih *threshold* dengan semakin tinggi *precision* mengakibatkan *recall* menurun dan sebaliknya. Untuk mempertimbangkan *trade-off* ini bersama dengan memprioritaskan nilai *precision* akan digunakan nilai *Fβ-score* dengan nilai β pada 0,5 untuk memberi bobot 2 kali lipat pada nilai *precision*.

splits = [10, 20, 50, 100, 200, 500, 1000]

blur["status\_0\_1"] = blur\_as\_01(blur)

blur\_bins = (

    blur["status\_0\_1"]

    .groupby(pd.cut(blur["lap\_var"], splits))

    .agg(["mean", "count"])

    .rename(columns={"mean": "proportion"})

    .assign(lap\_var=lambda df: [i.right for i in df.index])

)

fig = px.scatter(blur\_bins, x='lap\_var', y='proportion',

                 size='count', log\_x=True,

                 labels={'lap\_var':"Variance of Laplacian",

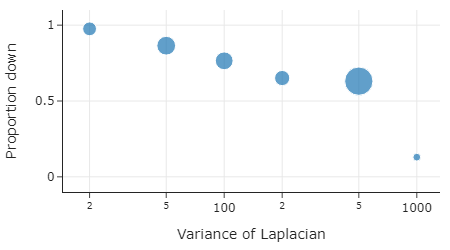
                         'proportion':"Proportion down"},

                 width=450, height=250)

fig.update\_layout(yaxis\_range=[-0.1, 1.1])

fig.show()

Gambar 5.10 Kode Program Menampilkan Proporsi Kelas Data



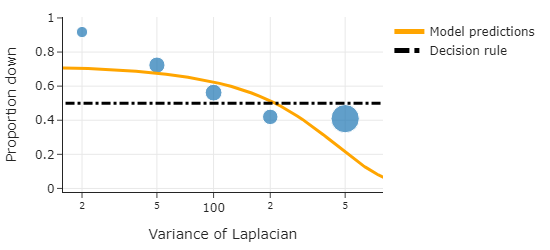
Gambar 5.11 Hasil Menampilkan Proporsi Kelas Data

Observasi keempat dikonfirmasi pada Gambar 5.10 dan Gambar 5.11. Selain pada nilai varians *Laplacian operator* dibawah 100 dan diatas 5000, gambar dengan *blur* dan gambar tanpa *blur* cenderung memiliki nilai varians *Laplacian operator* serupa seperti pada nilai 1000 dimana proporsi kedua kelas mendekati 50:50.

Observasi kelima yang dapat diambil adalah distribusi proporsi kelas menyerupai bentuk curva fungsi sigmoid. Dari observasi tersebut dapat diambil kesimpulan bahwa untuk pemilihan *threshold* dapat dilakukan dengan pertama memodelkan data ke dalam model Regresi Logistik yang juga memliki sebuah *threshold* untuk mengklasifikasikan data. Model Regresi Logistik digunakan untuk memungkinkan menggunakan proses analisis pemilihan *threshold* pada model Regresi Logistik untuk membantu dalam memilih *threshold* pada klasifikasi gambar *blur*.

#### *Modeling*

Sebelum mem-*fit*-kan model, data latih pertama perlu dihitung logaritma dari fitur varians *Laplacian oprator* terlebih dahulu. Model Regresi Logistik tidak dapat menerima nilai yang meningkat secara eksponensial seperti nilai varians karena nilai fitur yang diterima model yang seharusnya bersifat linear. Ditunjukkan pada Gambar 5.12 hasil mem-*fit* model tanpa menggunakan logaritma pada nilai varians *Laplacian operator*.



Gambar 5.12 Hasil *Fit* Model Tanpa Logaritma

Model Regresi Logistik dapat dibuat dengan menggunakan fungsi LogisticRegression dari modul scikit-learn. Data dilakukan proses *undersampling*, lalu dipisahkan menjadi data pelatihan model, data validasi *threshold*, dan data pengujian *threshold* dengan takar 60:20:20. Data pelatihan model kemudian digunakan untuk mem-*fit* model Regresi Logistik.

from sklearn.linear\_model import LogisticRegression

from sklearn.model\_selection import train\_test\_split

from imblearn.over\_sampling import RandomOverSampler

blur = blur.assign(log\_lap\_var=np.log(blur['lap\_var']))

X = blur[['log\_lap\_var']]

y = blur['status\_0\_1']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y,

                                  random\_state=104,

                                  test\_size=0.20,

                                  shuffle=True)

ros = RandomOverSampler(random\_state=103)

X\_res, y\_res = ros.fit\_resample(X\_train, y\_train)

X\_train, X\_val, y\_train, y\_val = train\_test\_split(X\_res, y\_res,

                                  random\_state=105,

                                  test\_size=0.25,

                                  shuffle=True)

lr\_model = LogisticRegression().fit(X\_train, y\_train)

X\_plt = pd.DataFrame(

    {"log\_lap\_var": np.linspace(X\_train["log\_lap\_var"].min(),

                                X\_train["log\_lap\_var"].max(), 50)})

p\_hats = lr\_model.predict\_proba(X\_plt)

X\_orig = np.exp(X\_plt)

fig = px.scatter(

    blur\_bins, x="lap\_var", y="proportion",

    size="count", log\_x=True,

    labels={"lap\_var": "Variance of Laplacian",

            "proportion": "Proportion down"},

    width=550, height=250,)

fig.add\_trace(go.Scatter(

    x=X\_orig["log\_lap\_var"], y=p\_hats[:, 1],

    line=dict(color="orange", width=3),

    name="Model predictions",))

fig.add\_trace(go.Scatter(

    x=X\_orig["log\_lap\_var"],

    y=np.repeat(0.5, len(X\_orig["log\_lap\_var"])),

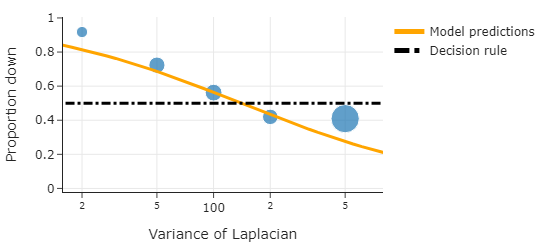
    line=dict(color="black", width=3, dash="dashdot"),

    name="Decision rule",))

fig.update\_layout(xaxis\_range=[1.2, 2.9])

fig

Gambar 5.13 Kode Program *Fit* Model Regresi Logistik



Gambar 5.14 Hasil *Fit* Model Regresi Logistik

Hasil yang di dapat dari Gambar 5.13 dan Gambar 5.14 adalah sebuah model yang merepresentasikan bentuk data yang dapat digunakan untuk melakukan analisis terhadap klasifikasi gambar dokumen teks yang memiliki *blur* dan tidak memiliki *blur* berdasarkan fitur nilai varians *Laplacian operator* gambar. Model sedikit meleset dari bentuk sebenarnya data latih karena data sebenarnya memiliki bias sebab terdapat lebih banyak gambar *blur* sedangkan model dilatih pada data yang sudah diberlakukan *undersampling*.

#### Pemilihan *Threshold*

Model Regresi Logistik merupakan model regresi yang dapat digunakan untuk melakukan klasifikasi dengan memilih *threshold* yang memisahkan hasil regresinya ke dalam kelas-kelas. *Default* nilai *threshold* yang digunakan adalah 0,5 dengan hasil regresi di atas 0,5 akan diklasifikasikan sebagai gambar *blur*.

from sklearn import metrics

def calc\_slope(df):

  return df.precision.diff(periods=1) / (df.recall.diff(periods=1))

def threshold\_predict(model, X, threshold):

    return np.where(model.predict\_proba(X)[:, 1] > threshold, 1.0, 0.0)

def accuracy(threshold, X, y):

    return np.mean(threshold\_predict(lr\_model, X, threshold) == y)

precision, recall, threshold = (

    metrics.precision\_recall\_curve(

        y\_test, lr\_model.predict\_proba(X\_test)[:, 1]))

fbetas = []

for i in threshold:

    y\_pred = pd.DataFrame(

        lr\_model.predict\_proba(X\_test)).map(lambda x: 1 if x > i else 0)

    fbetas.append(metrics.fbeta\_score(

        y\_test.values.reshape(y\_test.values.size,1),

        y\_pred.iloc[:,1].values.reshape(y\_pred.iloc[:,1].values.size,1),

        beta=0.5))

accs = [accuracy(t, X\_test, y\_test) for t in threshold]

tpr\_df = pd.DataFrame({"threshold":threshold,

                       "precision":precision[:-1],

                       "recall": recall[:-1],

                       "fbeta": fbetas,

                       "accuracy": accs})

tpr\_df.round(2)

Gambar 5.15 Kode Program Sampel Kinerja *Threshold*

Tabel 5.3 Hasil Sampel Kinerja *Threshold*

| index | threshold | precision | recall | fbeta | accuracy |
| --- | --- | --- | --- | --- | --- |
| 0 | 0.20 | 0.5 | 1.00 | 0.55 | 0.50 |
| 1 | 0.20 | 0.5 | 1.00 | 0.55 | 0.50 |
| 2 | 0.20 | 0.5 | 1.00 | 0.55 | 0.50 |
| 3 | 0.21 | 0.5 | 1.00 | 0.56 | 0.50 |
| 4 | 0.21 | 0.5 | 1.00 | 0.56 | 0.50 |
| ... | ... | ... | ... | ... | ... |
| 1082 | 0.92 | 1.0 | 0.01 | 0.03 | 0.51 |
| 1083 | 0.93 | 1.0 | 0.01 | 0.02 | 0.50 |
| 1084 | 0.93 | 1.0 | 0.00 | 0.02 | 0.50 |
| 1085 | 0.93 | 1.0 | 0.00 | 0.01 | 0.50 |
| 1086 | 0.93 | 1.0 | 0.00 | 0.00 | 0.50 |

Pemilihan *thresholding* dimulai dengan mengampil sampel *threshold* yang dapat dipilih dan kinerja masing-masing sampel tersebut. Sampel diambil dengan menggunakan fitur-fitur dari data validasi sebagai *threshold*. Kinerja dari sampel tersebut tertera pada Gambar 5.15 dan Tabel 5.3.

Kolom pertama adalah nilai hasil regresi dari fitur data validasi yang akan digunakan sebagai *threshold*. Kolom kedua sampai kelima adalah nilai kinerja *threshold* pada data validasi.

tpr\_df['peak\_indicator'] = calc\_slope(tpr\_df)

jump\_tpr\_df = tpr\_df[(np.isposinf(tpr\_df["peak\_indicator"]))]

fig = make\_subplots(rows=2, cols=1)

fig.add\_trace(go.Scatter(x=tpr\_df["threshold"], y=tpr\_df["precision"],

name="Precision", hovertemplate="<br>".join([

        "Precision: %{y}", ])), row=1, col=1)

fig.add\_trace(go.Scatter(x=tpr\_df["threshold"], y=tpr\_df["recall"],

name="Recall", line=dict(dash='dash'), hovertemplate="<br>".join([

        "Recall: %{y}", ])), row=1, col=1)

fig.add\_trace(go.Scatter(x=tpr\_df["threshold"], y=tpr\_df["fbeta"],

name="F0.5-score",line=dict(dash='dash'), hovertemplate="<br>".join([

        "F0.5-score: %{y}", ])), row=1, col=1)

fig.update\_layout(xaxis\_title="Threshold", yaxis\_title="Proportion")

fig.add\_trace(go.Scatter(x=tpr\_df["recall"], y=tpr\_df["precision"],

    name="Precision-Recall Curve", hoverinfo="skip",

marker={"color": "gray"}), row=2, col=1)

fig.add\_trace(go.Scatter(x=jump\_tpr\_df["recall"], y=jump\_tpr\_df["precision"],

    name="Next Increment", hovertemplate="<br>".join([

        "Recall: %{x}",

        "Precision: %{y}",

    ]), marker={"color": "black"}), row=2, col=1)

fig.add\_scatter(

    x=[0, 0.5],

    y=[1, 0.5],

    mode='lines',

    line\_color='gray',

    name="No-skill Classifier",

    line=dict(dash='dash'),

    row=2, col=1 )

fig['layout']['xaxis'].update(title="Threshold")

fig['layout']['yaxis'].update(title="Proportion", range=[0, 1.01])

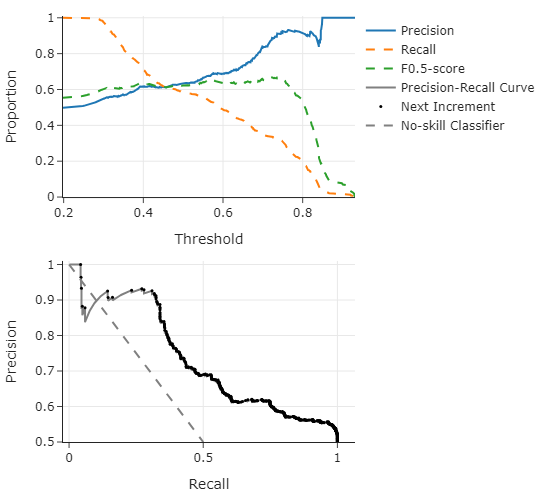
fig['layout']['xaxis2'].update(title="Recall")

fig['layout']['yaxis2'].update(title="Precision", range=[0.5, 1.01])

fig.update\_layout(width=550, height=500)

fig

Gambar 5.16 Kode Program Visualisasi Kinerja *Threshold*



Gambar 5.17 Hasil Visualisasi Kinerja *Threshold*

Dapat dilihat dari Gambar 5.16 dan Gambar 5.17 bahwa terdapat *trade-off* antara nilai *precision* dan nilai *recall*. Semakin tinggi *threshold* yang digunakan semakin tinggi nilai *precision* namun semakin rendah nilai *recall* dan sebaliknya.

Untuk menghindari mendapat *dimishing return* yang terlalu kecil saat mencari nilai *precision* yang tertinggi digunakan metik *Fβ-score* dengan β = 0,5 untuk memberi bobot 2 kali lipat lebih besar pada nilai *precision* namun tetap mempertimbangkan nilai *recall*. Terdapat 2 metrik yang menjadi pertimbangan utama, yaitu *precision* dan *Fβ-score*. Mengikuti curva kinerja model ada dua *threshold* yang dapat dipilih, yaitu *threshold* yang memaksimalkan *Fβ-score* dan *threshold* yang memaksimalkan *precision* dengan kondisi *Fβ-score* tidak kurang dari 0,6 supaya *Fβ-score* tidak menurun drastis.

fbeta\_max = (tpr\_df.sort\_values(by="fbeta",

ascending=False).iloc[0])

fbeta\_max

Gambar 5.18 Kode Program *Threshold Fβ-score* Maksimum

Tabel 5.4 Hasil *Threshold* *Fβ-score* Maksimum

| index | threshold | precision | recall | fbeta | accuracy |
| --- | --- | --- | --- | --- | --- |
| 861 | 0,72 | 0,88 | 0,34 | 0,67 | 0,65 |

Dengan *threshold* yang mencapai *Fβ-score* maksimum dari Gambar 5.18 dan Tabel 5.4 model mencapai nilai *precision* 0,88 yang mengindikasikan sekitar 1 dari 9 gambar tidak *blur* salah diklasifikasikan sebagai gambar *blur* dan nilai *recall* 0,34 yang mengindikasikan sekitar 2 dari 3 gambar *blur* salah diklasifikasikan sebagai gambar tidak *blur*.

p\_fbetalimit = (tpr\_df[tpr\_df["fbeta"] >= 0.60]

.sort\_values(by="precision", ascending=False).iloc[0])

p\_fbetalimit

Gambar 5.19 Kode Program *Threshold* *Precision* Maksimum

Tabel 5.5 Hasil *Threshold Precision* Maksimum

| index | threshold | precision | recall | fbeta | accuracy |
| --- | --- | --- | --- | --- | --- |
| 914 | 0,76 | 0,93 | 0,27 | 0,62 | 0,63 |

Dengan *threshold* yang mencapai *Fβ-score* maksimum dari Gambar 5.19 dan Tabel 5.5 model mencapai nilai *precision* 0,93 yang mengindikasikan sekitar 1 dari 15 gambar tidak *blur* salah diklasifikasikan sebagai gambar *blur* dan nilai *recall* 0,27 yang mengindikasikan sekitar 3 dari 4 gambar *blur* salah diklasifikasikan sebagai gambar tidak *blur*.

*Threshold* kedua memiliki lebih rendah kemungkinan salah klasifikasi gambar tidak *blur* sebagai gambar *blur* namun memiliki nilai *recall* yang lebih rendah pula. Dari kedua *threshold* tersebut dipilih *threshold* pertama karena *threshold* tersebut memiliki nilai *Fβ-score* tertinggi yang mengindikasikan kemungkinan *false positive* dan kemungkinan *false negative* cukup rendah mengikuti batasan dari konteks yang telah ditentukan.

#### Konversi

*Threshold* yang dipilih melakukan klasifikasi berdasarkan hasil regresi model Regresi Logistik bukan atas fitur nilai varians *Laplacian operator* secara langsung sedangkan aplikasi menggunakan nilai varians secara langsung untuk melakukan klasifikasi. Untuk mendapat *threshold* dalam nilai varians dapat digunakan persamaan (12) yang diturunkan dari rumus prediksi model Regresi Logistik.

|  |  |
| --- | --- |
|  | (12) |

Keterangan:

X : nilai varians *Laplacian operator*.

P : nilai probabilistik hasil model Regresi Logistik.

*intercept* : nilai konstan fungsi model.

*coeffiecient* : nilai koefisien fungsi model.

p = fbeta\_max.threshold

[intercept] = lr\_model.intercept\_

[[coef]] = lr\_model.coef\_

actual = np.exp((np.log(p / (1 - p)) - intercept) / coef)

Gambar 5.20 Kode Program Konversi *Threshold* Probabilitas

Probabilitas: 0,72

Aktual      : 39,71

Gambar 5.21 Hasil Konversi *Threshold* Probabilitas

Digunakan kode pada Gambar 5.20 untuk menghitung nilai varians *Laplacian operator* dari *threshold* probabilitas model Regresi Logistik. Hasil yang diperoleh saat nilai probabilitas dikonversi ke nilai varians adalah 39,71 seperti pada Gambar 5.21.

#### Pengujian

import plotly.figure\_factory as ff

from sklearn.metrics import confusion\_matrix

y\_pred = (np.exp(X\_test) < actual).astype(int)

mat = confusion\_matrix(y\_test, y\_pred)

fig = ff.create\_annotated\_heatmap(

    z=mat, x=["False", "True"], y=["False", "True"],

    showscale=False, colorscale=px.colors.sequential.gray\_r,)

fig.update\_layout(font=dict(size=16), width=300, height=250)

fig.add\_annotation(

    x=0, y=0, text="True negative", yshift=30,

    showarrow=False, font=dict(color="white", size=13),)

fig.add\_annotation(

    x=1, y=0, text="False positive", yshift=30,

    showarrow=False, font=dict(color="black", size=13),)

fig.add\_annotation(

    x=0, y=1, text="False negative", yshift=30,

    showarrow=False, font=dict(color="white", size=13),)

fig.add\_annotation(

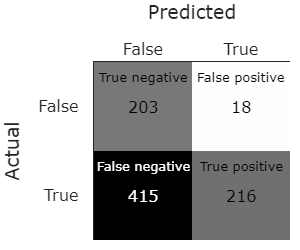
    x=1, y=1, text="True positive", yshift=30,

    showarrow=False, font=dict(color="white", size=13),)

fig.update\_xaxes(title="Predicted")

fig.update\_yaxes(title="Actual", autorange="reversed")

Gambar 5.22 Kode Program *Confusion Matrix*



Gambar 5.23 Hasil *Confusion Matrix*

Berikut hasil *confusion matrix* klasifikasi dengan *threshold* yang dipilih pada data pengujian dilampirkan pada Gambar 5.22 dan Gambar 5.23. Dari 631 gambar *blur*, 216 gambar benar diklasifikasikan sebagai gambar *blur* dan 415 salah diklasifikasikan sebagai gambar tidak *blur*. Dari 221 gambar tidak *blur*, 203 gambar benar diklasifikasi sebagai gambar tidak *blur* dan 18 salah diklasifikasikan sebagai gambar *blur*.

pd.DataFrame({"threshold": [actual],

              "accuracy": [metrics.accuracy\_score(y\_test, y\_pred)],

              "precision": [metrics.precision\_score(y\_test, y\_pred)],

              "recall": [metrics.recall\_score(y\_test, y\_pred)],

              "fbeta": [metrics.fbeta\_score(y\_test, y\_pred, beta=0.5)],})

Gambar 5.24 Kode Program Evaluasi Kinerja *Threshold*

Tabel 5.6 Hasil Evaluasi Kinerja *Threshold*

| threshold | precision | recall | fbeta | accuracy |
| --- | --- | --- | --- | --- |
| 39,71 | 0,49 | 0,92 | 0,34 | 0,69 |

Ditemukan hasil kinerja *threshold* klasifikasi pada Gambar 5.24 dan Tabel 5.6 berupa nilai *precision* 0,92, nilai *recall* 0,60, nilai *Fβ-score*, dan nilai *accuracy* 0,66.

### Server *Backend*

Transaksi data pada aplikasi dilakukan melalui WhatsApp Cloud API yang digunakan untuk meneruskan dan mengirim pesan WhatsApp dan API berbasis Flask yang digunakan untuk menerima gambar, melakukan identifikasi, dan menyusun pesan umpan balik.

WhatsApp Cloud API disediakan oleh WhatsApp Business Platform sebagai cara untuk berinteraksi dengan akun dan percakapan WhatsApp Business. Akun dan percakapan diakses melalui *webhook* yang disediakan oleh WhatsApp. Daftar *endpoint* WhatsApp Cloud API tercantum pada (*WhatsApp business platform: Postman API network*, t.t.).

API Flask digunakan untuk menjalankan fungsi Python seperti identifikasi gambar *blur* saat diterima gambar masuk melalui *webhook* WhatsApp. API Flask di-*host* pada Vercel dan dapat diakses melalui URL <https://whatsapp-blur-detect.vercel.app>. Daftar *endpoint* API Flask tercantum pada Tabel 5.7.

Tabel 5.7 *Entrypoint* API Flask

| **Metode** | ***Endpoint*** | **Fungsi** |
| --- | --- | --- |
| GET | /webhook | Melalukan proses verifikasi dengan WhatsApp Cloud API sebelum membuat koneksi. |
| POST | /webhook | Meneruskan pesan yang diterima dari WhatsApp ke API Flask. |
| POST | /stress-test | Digunakan untuk mensimulasikan kerja API saat menerima pesan berisi gambar. |

WhatsApp Cloud API memerlukan aplikasi yang menggunakan layanan *webhook* untuk memverifikasi koneksi *webhook* sebelum menerima terusan pesan dari WhatsApp. Untuk melakukan verifikasi ini digunakan *endpoint* /webhook pada metode GET. Kode program untuk *endpoint* GET /webhook API Flask dilampirkan pada Gambar 5.25.

@webhook\_blueprint.route("/webhook", methods=["GET"])

def webhook\_get():

    return verify()

# Required webhook verifictaion for WhatsApp

def verify():

    # Parse params from the webhook verification request

    mode = request.args.get("hub.mode")

    token = request.args.get("hub.verify\_token")

    challenge = request.args.get("hub.challenge")

    # Check if a token and mode were sent

    if mode and token:

        # Check the mode and token sent are correct

        if mode == "subscribe" and

token == current\_app.config["VERIFY\_TOKEN"]:

            # Respond with 200 OK and challenge token from the request

            logging.info("WEBHOOK\_VERIFIED")

            return challenge, 200

        else:

            # Responds with '403 Forbidden' if verify tokens do not match

            logging.info("VERIFICATION\_FAILED")

            return jsonify({"status": "error",

"message": "Verification failed"}), 403

    else:

        # Responds with '400 Bad Request' if verify tokens do not match

        logging.info("MISSING\_PARAMETER")

        return jsonify({"status": "error",

"message": "Missing parameters"}), 400

Gambar 5.25 Kode Program *Endpoint* GET /webhook

WhatsApp Cloud API menggunakan *endpoint* yang telah diverifikasi melalui metode POST dari *endpoint* tersebut untuk meneruskan pesan dari WhatsApp ke aplikasi. Kode program untuk *endpoint* POST /webhook API Flask dilampirkan pada Gambar 5.26.

@webhook\_blueprint.route("/webhook", methods=["POST"])

@signature\_required

def webhook\_post():

    return handle\_message()

def handle\_message():

    body = request.get\_json()

    # Check if it's a WhatsApp status update

    if (

        body.get("entry", [{}])[0]

        .get("changes", [{}])[0]

        .get("value", {})

        .get("statuses")

    ):

        logging.info("Received a WhatsApp status update.")

        return jsonify({"status": "ok"}), 200

    try:

        if is\_valid\_whatsapp\_message(body):

            process\_whatsapp\_message(body)

            return jsonify({"status": "ok"}), 200

        else:

            # if the request is not a WhatsApp API event, return an error

            return (

                jsonify({"status": "error",

"message": "Not a WhatsApp API event"}),

                404,

            )

    except json.JSONDecodeError:

        logging.error("Failed to decode JSON")

        return jsonify({"status": "error",

"message": "Invalid JSON provided"}), 400

def process\_whatsapp\_message(body):

    logging.info(body)

    message = body["entry"][0]["changes"][0]["value"]["messages"][0]

    response = generate\_response(message)

    data = get\_text\_message\_input(

current\_app.config["RECIPIENT\_WAID"], response)

    send\_message(data)

Gambar 5.26 Kode Program *Endpoint* POST /webhook

Pesan yang diteruskan melalui POST /webhook kemudian diteruskan ke bagian lain dari server seperti untuk melakukan identifikasi dan penyusunan pesan umpan balik. Kode program identifikasi *blur* dilampirkan pada Gambar 5.27.

def is\_blur(image):

    gray = cvtColor(image, COLOR\_BGR2GRAY)

    fm = Laplacian(gray, CV\_64F).var()

    logging.info(f"Focus Measure: {fm}")

    return True if fm <= 39.71 else False

def identify\_blur(media\_id):

    data, mime\_type = download\_media(media\_id)

    # If it is a PDF file

    if (mime\_type == 'application/pdf'):

        under\_50, blur\_pages = process\_document(data)

        if under\_50:

            if blur\_pages:

                return reply\_document\_blur(blur\_pages)

            else:

                return reply\_document\_clear()

        else:

            if blur\_pages:

                return reply\_document\_blur\_too\_long(blur\_pages)

            else:

                return reply\_document\_clear\_too\_long()

    # If it is an image type file

    elif ('image' in mime\_type):

        if process\_image(data):

            return reply\_image\_blur()

        else:

            return reply\_image\_clear()

    return reply\_unknown()

# Returns on what pages the image is blurred in

def process\_document(data):

    doc = pymupdf.Document(stream=data)

    blur\_pages = []

    pages = doc.page\_count

    for i in range(0, min(pages, 50)):

        page = doc.load\_page(i)

        pixmap = page.get\_pixmap(dpi=96)

        image = np.frombuffer(pixmap.samples\_mv, dtype=np.uint8).reshape((pixmap.height, pixmap.width, -1))

        if is\_blur(image):

            blur\_pages.append(i+1)

    return (True if pages <= 50 else False), blur\_pages

# Returns is the image blurred

def process\_image(data):

    arr = np.asarray(bytearray(data), dtype=np.uint8)

    image = imdecode(arr, -1)

    return is\_blur(image)

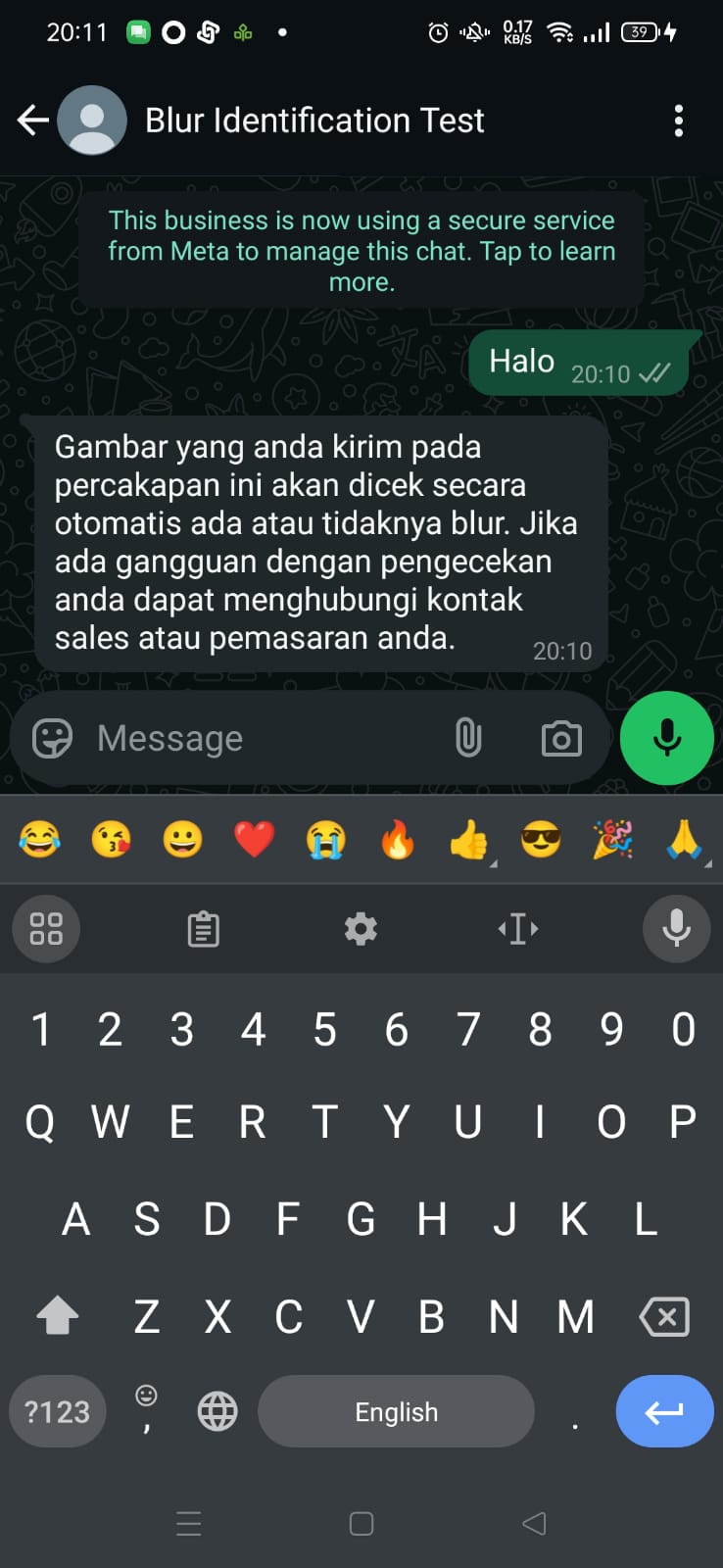
Gambar 5.27 Kode Program Identifikasi *Blur*

### Integrasi WhatsApp

Integrasi dengan WhatsApp memuat beberapa fitur sebagai berikut:

#### Percakapan

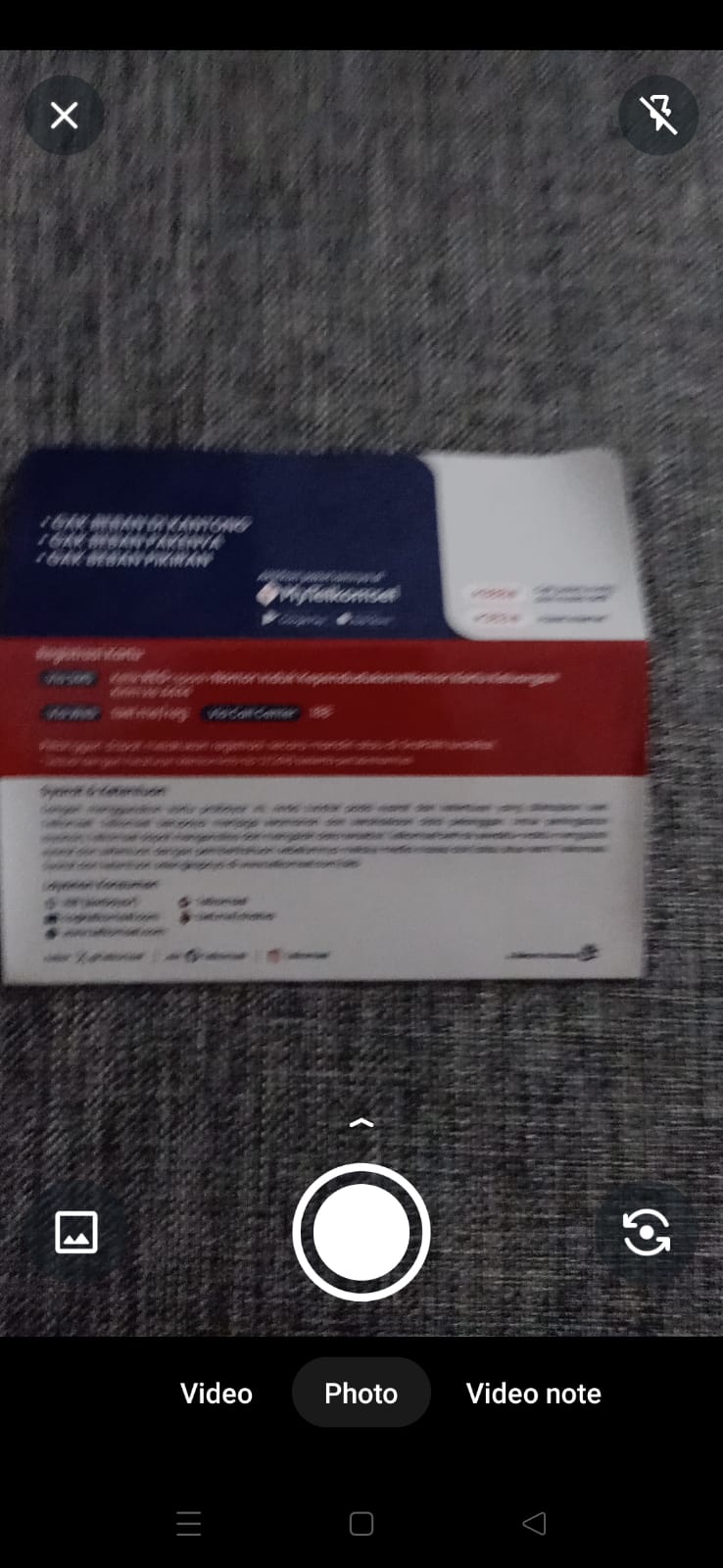
Integrasi dilakukan pada semua percakapan WhatsApp pada akun yang terhubung dengan WhatsApp Cloud API ditandai dengan pesan konfirmasi dari WhatsApp pada *header* percakapan seperti pada Gambar 5.26.



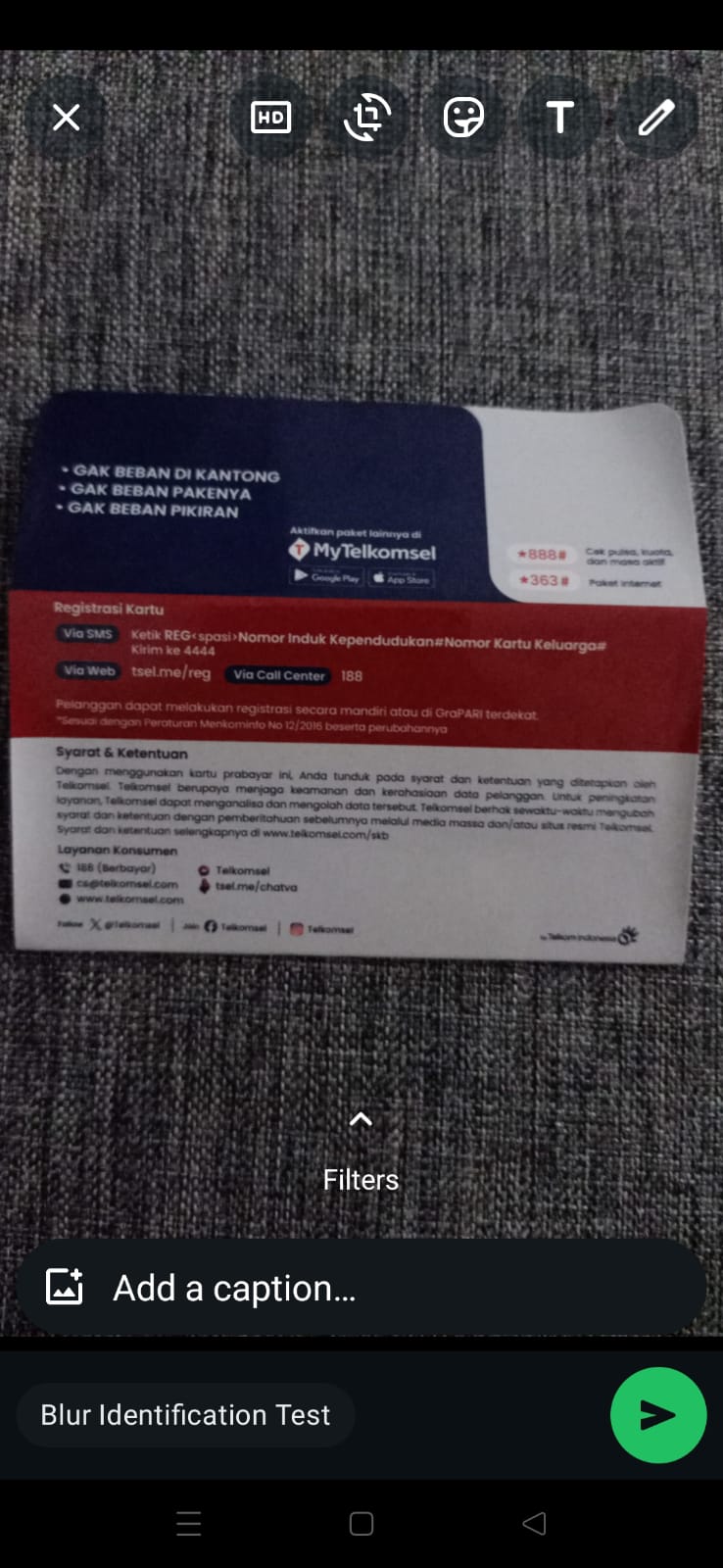
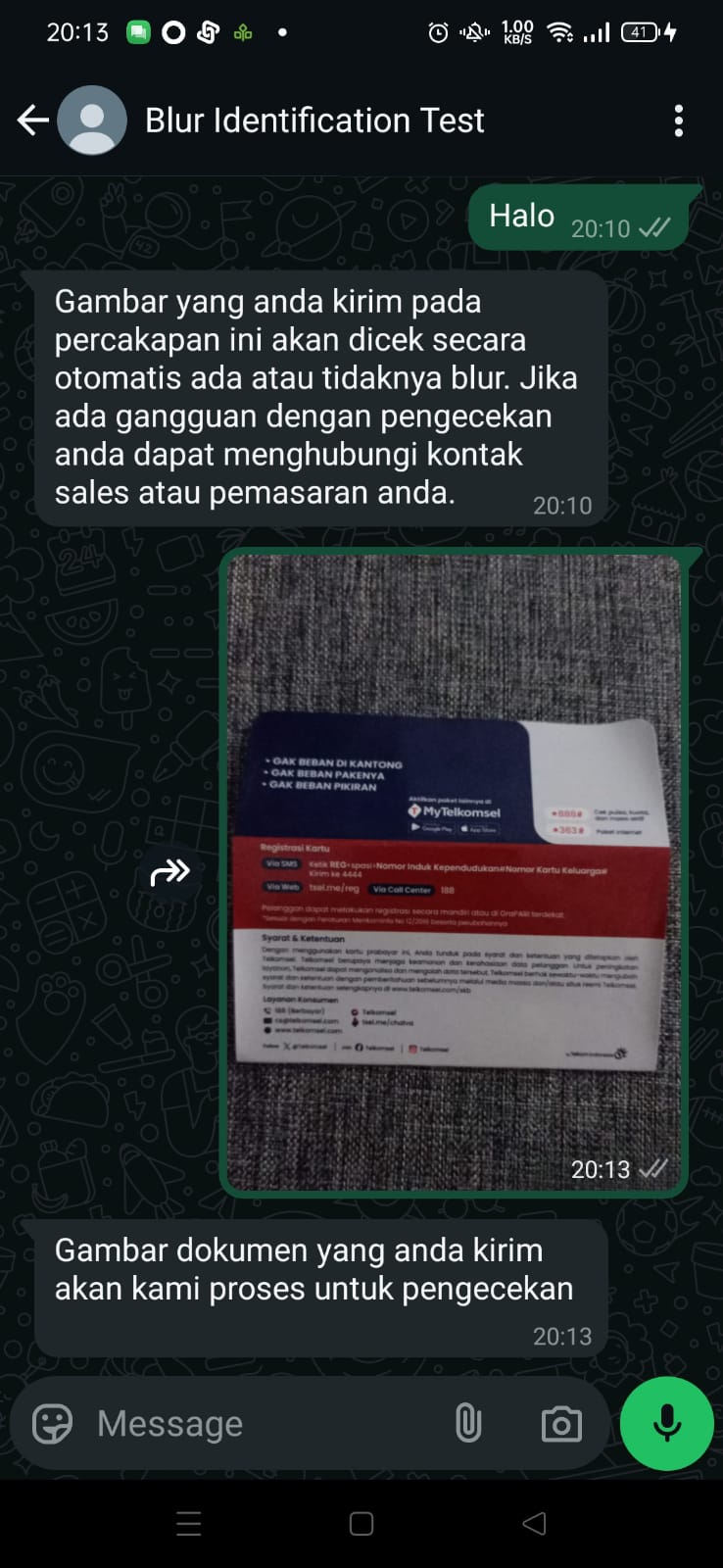
Gambar 5.28 Tampilan Percakapan dengan Akun Terhubung

#### Gambar

Integrasi menerima pesan berisi gambar JPEG atau PNG yang akan dipindai dan diidentifikasi ada atau tidaknya *blur* pada gambar tersebut. Pengiriman gambar dapat dilakukan dengan mengambil gambar baru dengan kamera aplikasi WhatsApp atau dengan memilih gambar yang telah diambil dari penyimpanan perangkat. Proses pengiriman gambar melalui kamera tertera pada Gambar 5.29 dan Gambar 5.30.

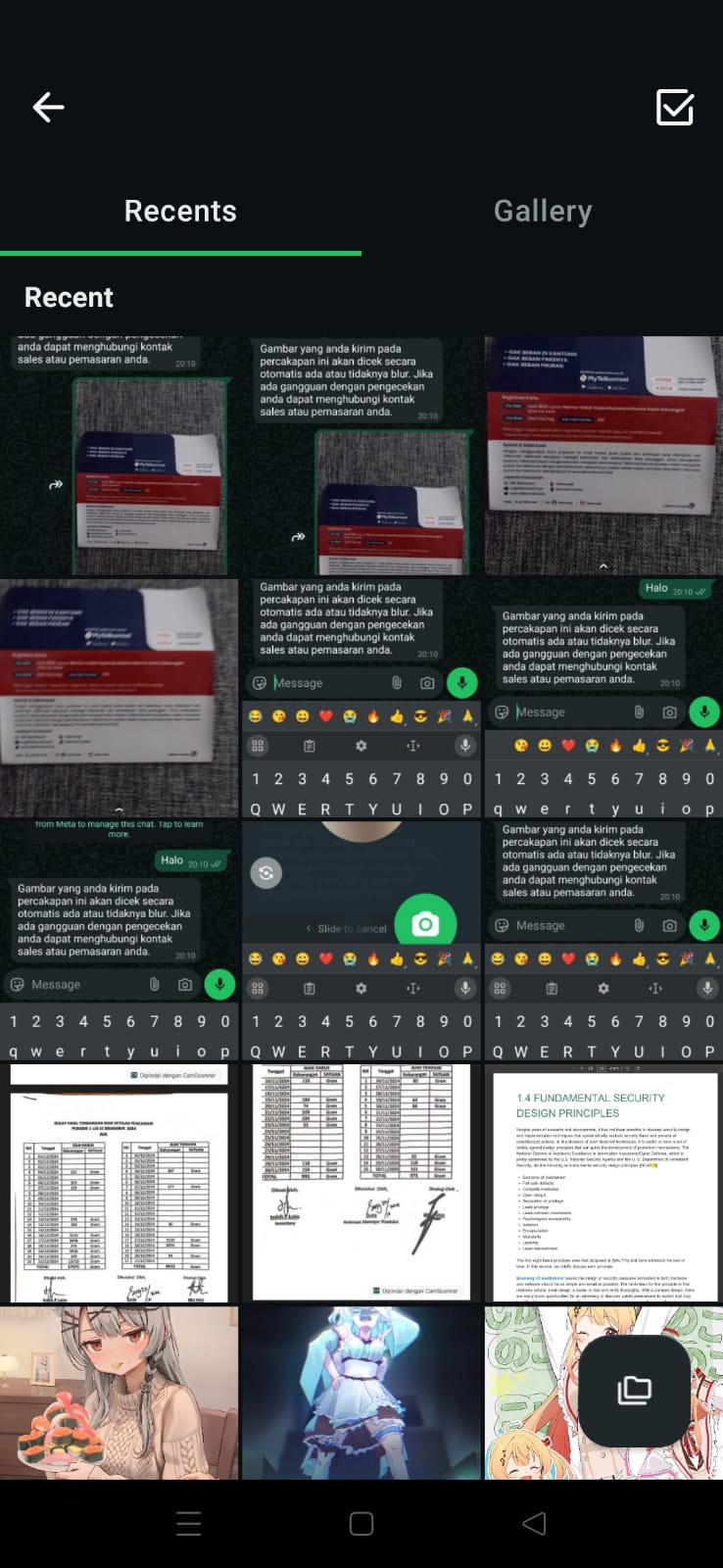
 

Gambar 5.29 Proses Pengiriman Gambar dengan Kamera (Langkah 1 dan 2)

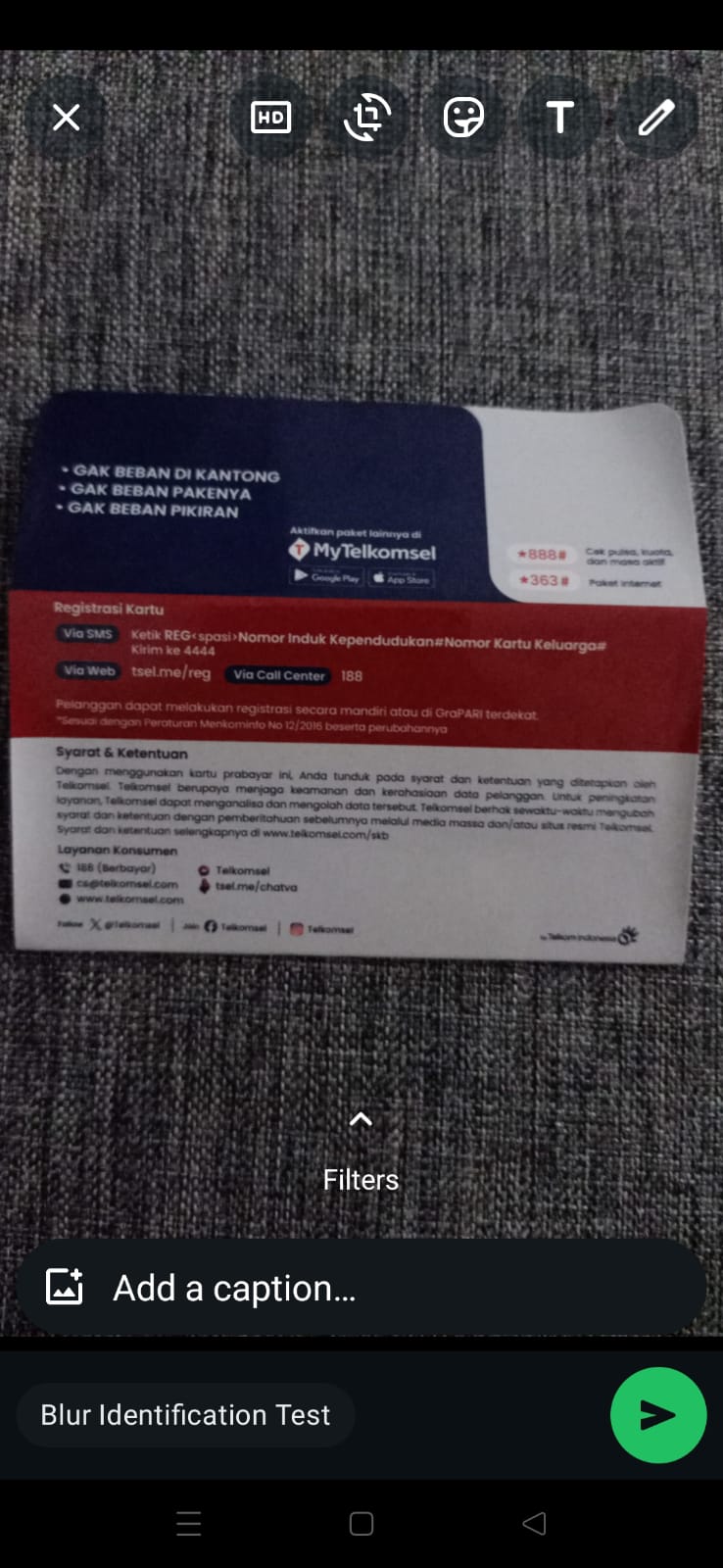
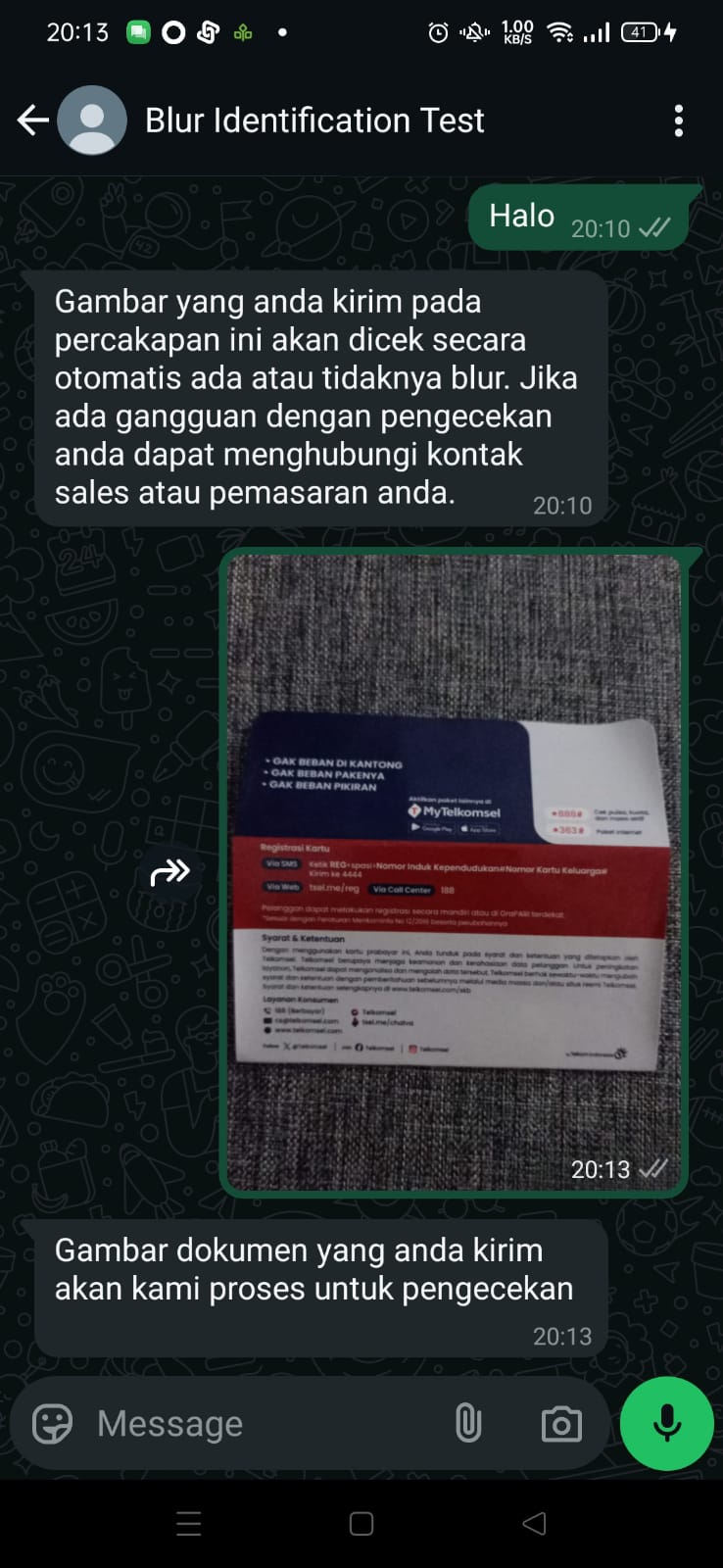
 

Gambar 5.30 Proses Pengiriman Gambar dengan Kamera (Langkah 3 dan 4)

Proses pengiriman gambar melalui penyimpanan tertera pada Gambar 5.31 dan Gambar 5.32.

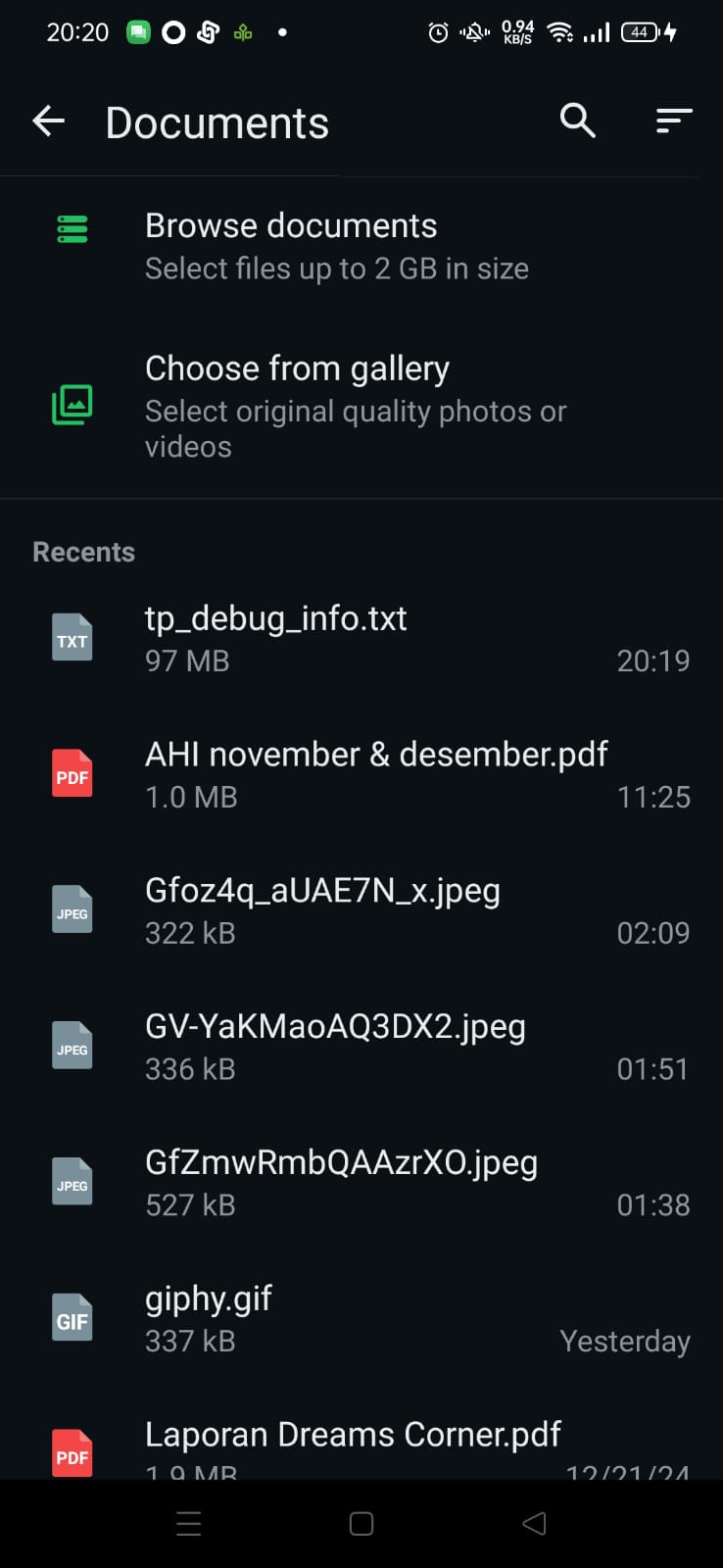
Gambar 5.31 Proses Pengiriman Gambar dari Perangkat (Langkah 1 dan 2)

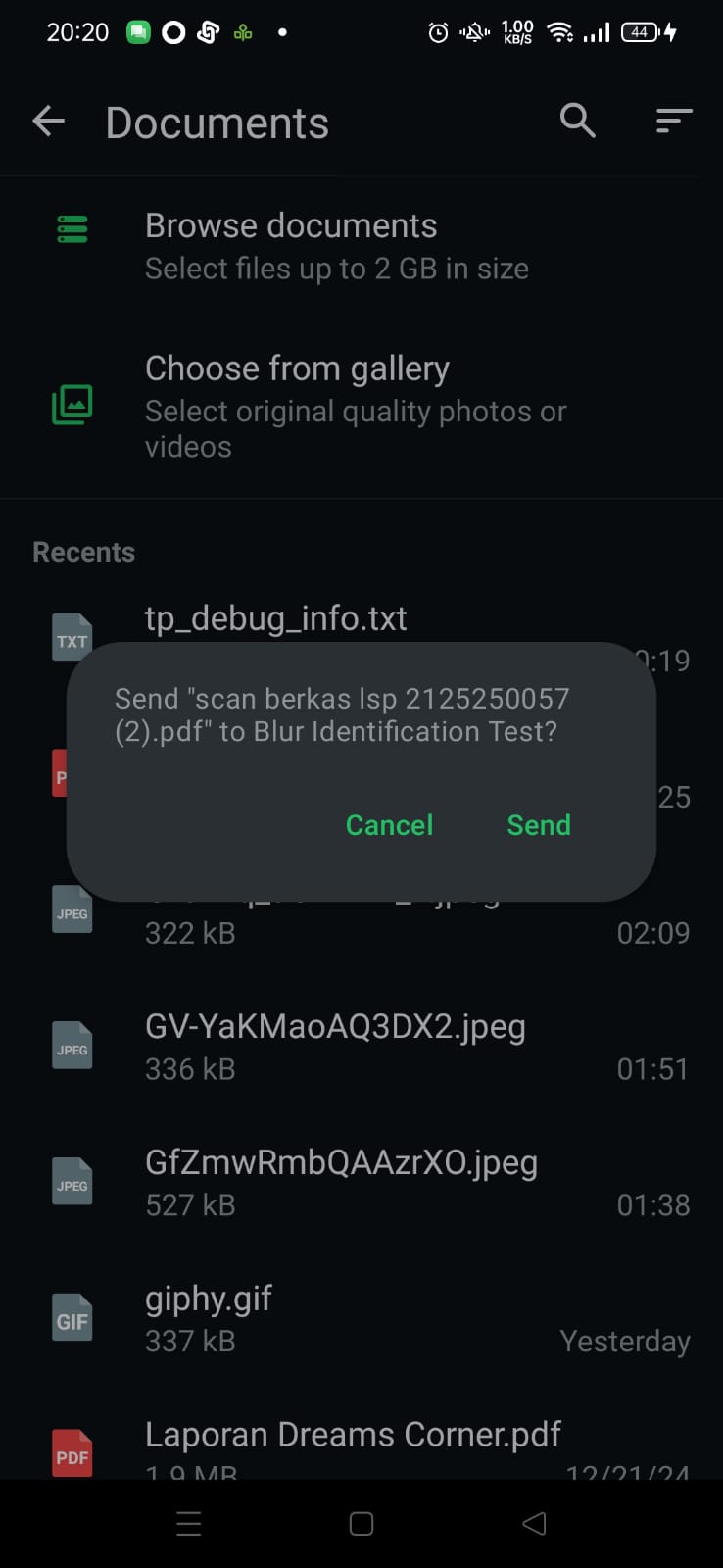
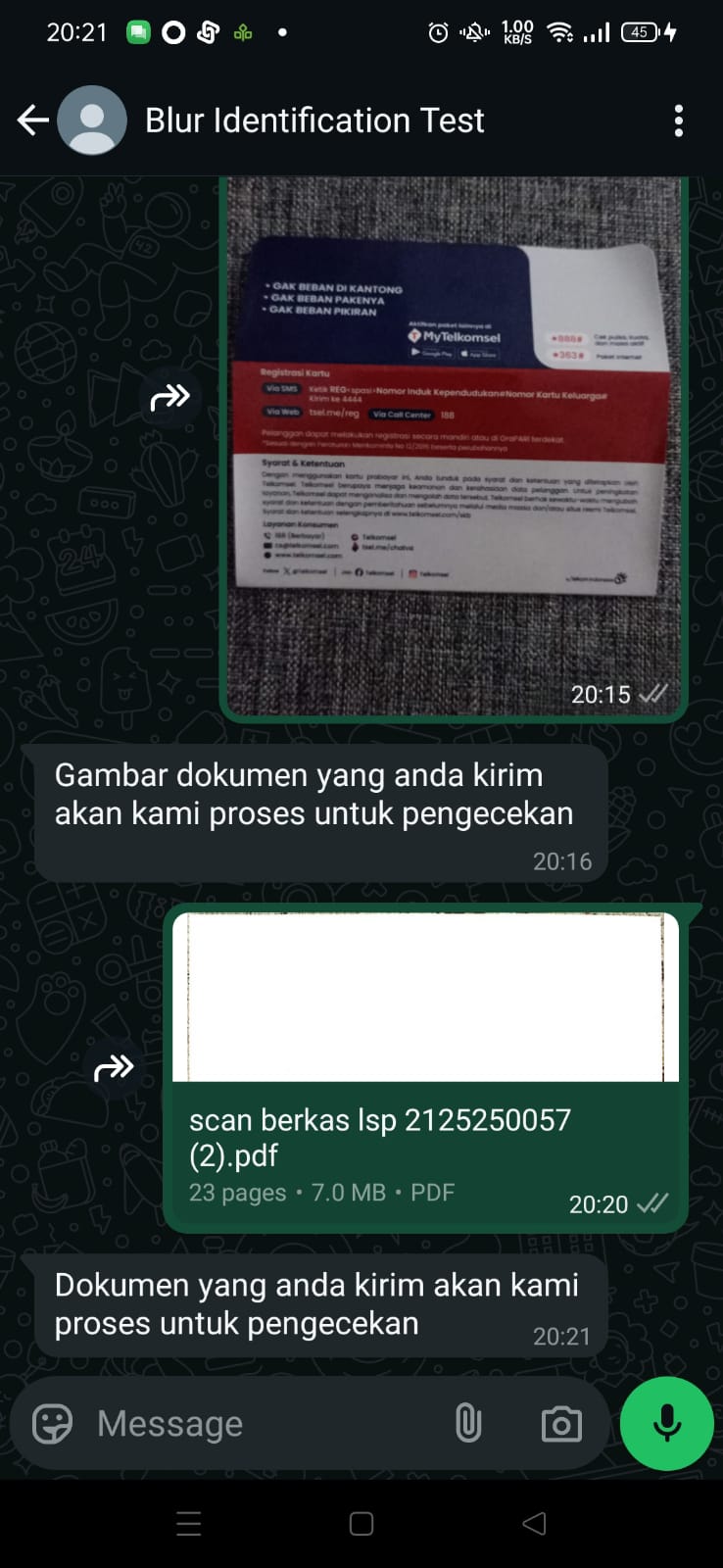
Gambar 5.32 Proses Pengiriman Gambar dari Perangkat (Langkah 3 dan 4)

#### Dokumen PDF

Integrasi menerima pesan berisi dokumen PDF yang akan dipindai dan diidentifikasi jika ada halaman mana yang memiliki *blur* pada dokumen. Proses pengiriman dokumen PDF tertera pada Gambar 5.33 dan Gambar 5.34.

Gambar 5.33 Proses Pengiriman Dokumen (Langkah 1 dan 2)

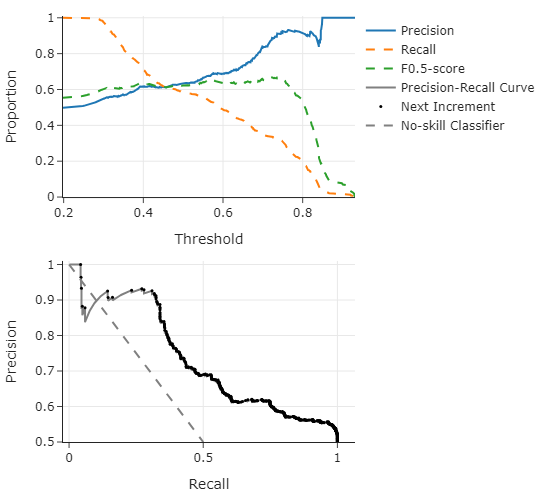
Gambar 5.34 Proses Pengiriman Dokumen (Langkah 3 dan 4)

## Pengujian

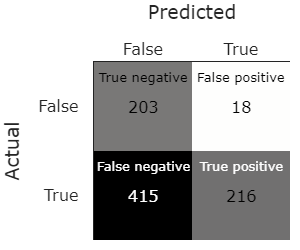
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis:

### Pengujian *Threshold*

Pengujian *threshold* dilakukan dengan menguji kinerja *threshold* terhadap data pengujian untuk mendapatkan *confusion matrix* dan tabel nilai kinerja hasil klasifikasi. Validasi proses pemilihan *threshold* dapat digambarkan melalui grafik kinerja setiap *threshold* dan kurva *precision-recall*. Grafik kinerja dan *precision-recall curve* dilampiran pada Gambar 5.35, *confusion matrix* hasil pengujian dilampirkan pada Gambar 5.36, dan hasil evaluasi *threshold* yang dipilih dilampirkan pada Tabel 5.8.



Gambar 5.35 Grafik Kinerja *Threshold* dan *Precision-recall Curve*



Gambar 5.36 *Confusion Matrix*

Tabel 5.8 Hasil Evaluasi Kinerja *Threshold*

| threshold | precision | recall | fbeta | accuracy |
| --- | --- | --- | --- | --- |
| 39,71 | 0,49 | 0,92 | 0,34 | 0,69 |

### *Stress Test*

*Stress test* dilakukan pada API untuk menguji beban penggunaan yang dapat ditanggung oleh *server* dan tetap memenuhi batasan dari aspek sustainabilitas. Pengujian dilakukan dengan mengirimkan POST *request* kepada *endpoint* */stress-test* dalam jumlah besar secara bersamaan. Kode program yang digunakan untuk melakukan *stress test* dilampirkan pada Gambar 5.37 dan hasil dari *stress test* dilampirkan pada Gambar 5.38.

import asyncio

import aiohttp

import time

from tqdm.asyncio import tqdm

# kirim satu request

async def post(session: aiohttp.ClientSession):

    image = open('image.jpg', 'rb')

    time\_start = time.time()

    async with session.post(

    "https://whatsapp-blur-detect.vercel.app/stress-test",

    data=image) as response:

        await response.text()

        time\_end = time.time()

        return time\_end - time\_start

async def bound\_post(semaphore, session):

    async with semaphore:

        return await post(session)

# kirim n request secara async

async def stress\_test(n):

    print(f"Running {n} times")

    semaphore = asyncio.Semaphore(1000)

    tasks = []

    async with aiohttp.ClientSession() as session:

        for i in range(n):

            task = asyncio.ensure\_future(bound\_post(semaphore, session))

            tasks.append(task)

        times = await tqdm.gather(\*tasks)

        print(f"Tasks        : {len(times)}")

        print(f"Tasks on-time: {sum(

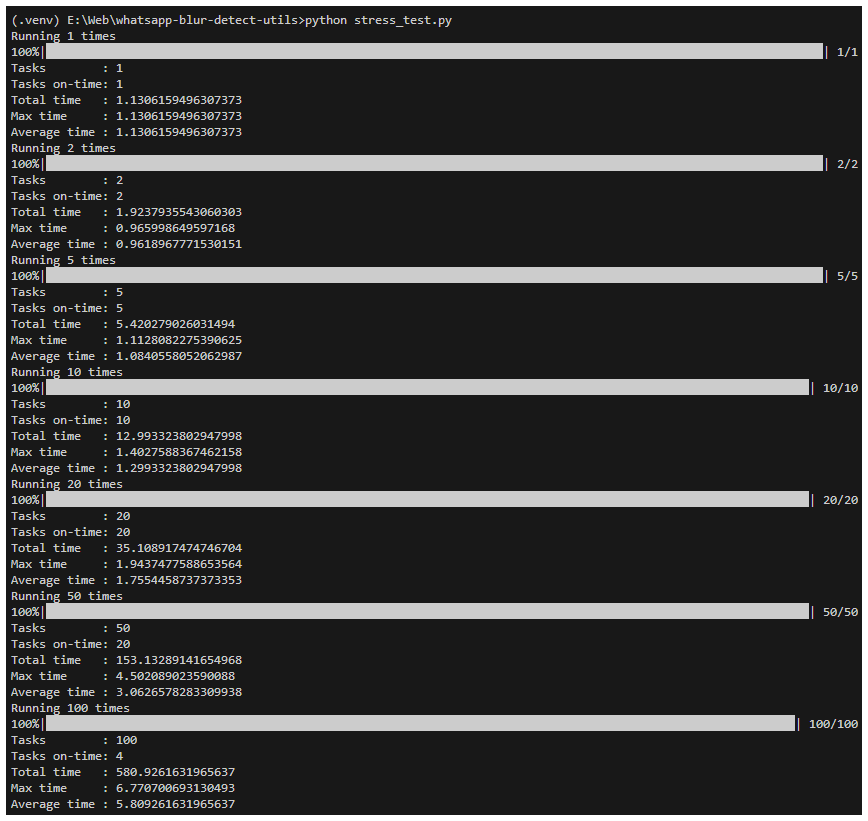
                [1 if t <= 3 else 0 for t in times])}")

        print(f"Total time   : {sum(times)}")

        print(f"Max time     : {max(times)}")

        print(f"Average time : {sum(times) / len(times)}")

Gambar 5.37 Kode Program Pengujian *Stress Test*



Gambar 5.38 Hasil Pengujian *Stress Test*

Hasil pengujian *stress test* menunjukkan bahwa API berhasil berjalan tepat waktu sampai 20 pemanggilan sekaligus. Pada 50 pemanggilan sekaligus walau rata-rata waktu eksekusi masih 3 detik lebih dari setengah panggilan *request* melebihi batasan waktu 3 detik. Pada 100 pemanggilan sekaligus rata-rata waktu eksekusi mencapai 5 detik dan hanya 4 panggilan berhasil dieksekusi kurang dari 3 detik.

### Pengujian Aplikasi

Pengujian yang dilakukan terhadap aplikasi berupa pengujian bersifat *black box*. Pengujian dilakukan secara manual untuk mengetes fitur apakah sudah sesuai rancangan alur aplikasi. Pengujian dilakukan mengikuti rancangan pengujian pada Tabel 4.1. Hasil pengujian dilampirkan pada Tabel 5.9.

Tabel 5.9 Pengujian *Black Box*

| **No.** | **Pengujian** | **Hasil yang Diharapkan** | **Hasil Pengujian** |
| --- | --- | --- | --- |
| 1. | Identifikasi gambar dokumen bertipe *file* PNG |  | Berhasil |
| 2. | Identifikasi gambar dokumen bertipe *file* JPEG |  | Berhasil |
| 3. | Identifikasi gambar dokumen bertipe file PDF berisi satu halaman |  | Berhasil |
| 4. | Identifikasi gambar dokumen bertipe file PDF berisi lebih dari satu halaman |  | Berhasil |
| 5. | Identifikasi gambar dokumen dengan kategori tidak *blur* |  | Berhasil |
| 6. | Identifikasi gambar dokumen dengan kategori *blur* |  | Berhasil |
| 7. | Identifikasi gambar dokumen terpotong |  | Berhasil |
| 8. | Identifikasi gambar dokumen terkena pantulan cahaya berlebihan |  | Berhasil |
| 9. | Identifikasi gambar dokumen tertutupi oleh objek lain |  | Berhasil |
| 10. | Identifikasi gambar dokumen terlipat |  | Berhasil |
| 11. | Identifikasi gambar dokumen dengan tulisan tangan |  | Berhasil |
| 12. | Identifikasi gambar dokumen dalam pencahayaan sangat rendah (dibawah 50 lux) |  | Berhasil |
| 13. | Identifikasi gambar dokumen dalam pencahayaan sangat rendah (dibawah 50 lux) |  | Berhasil |
| 14. | Menerima pesan teks |  | Berhasil |
| 15. | Menerima pesan selain teks, file JPEG, PNG, dan PDF |  | Berhasil |

### Analisis Hasil Pengujian

#### Analisis Pengujian Ekonomis

#### Analisis Pengujian Manufakturabilitas

#### Analisis Pengujian Sustainabilitas

### Pengujian Kepuasan Aplikasi

# PENUTUP

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud dolor Loremp.

DAFTAR PUSTAKA

Ahmad, S., Trahan, G., & Islam, A. (2024). *When do convolutional neural networks stop learning?* https://doi.org/10.48550/arXiv.2403.02473

Ali, U., & Mahmood, M. T. (2018). Analysis of blur measure operators for single image blur segmentation. *Applied Sciences*, *8*(5), 807. https://doi.org/10.3390/app8050807

Aliim, M. S., Supriyanti, R., & Siswantoro, H. (2023). The SDLC analysis for implementation document management system at IPC center of Universitas Jenderal Soedirman. *Engineering, MAthematics and Computer Science (EMACS) Journal*, *5*(2), 47–51. https://doi.org/10.21512/emacsjournal.v5i2.9910

Andhavarapu, S. K. (2015). *Image blur detection with two-dimensional haar wavelet transform*. https://doi.org/10.26076/6aad-79cf

Arista, T. S., & Novita, D. (2024). Sistem pendukung keputusan penentuan teknisi terbaik PT Sapta Karya Manunggal menggunakan metode topsis berbasis website. *3rd MDP Student Conference*, *3*(1), 981–988. https://doi.org/10.35957/mdp-sc.v3i1.7616

Asad, F., Ul-Hasan, A., Shafait, F., & Dengel, A. (2016). High performance OCR for camera-captured blurred documents with LTSM networks. *2016 12th IAPR Workshop on Document Analysis Systems (DAS)*, 7–12. https://doi.org/10.1109/DAS.2016.69

Bansal, R., Raj, G., & Choudhury, T. (2016). Blur image detection using laplacian operator and Open-CV. *2016 International Conference System Modeling & Advancement in Research Trends (SMART)*, 63–67. https://doi.org/10.1109/SYSMART.2016.7894491

Chazalon, J., Gomez-Kramer, P., Burie, J.-C., Coustaty, M., Eskenazi, S., Luqman, M., Nayef, N., Rusinol, M., Sidere, N., & Ogier, J.-M. (2017). SmartDoc 2017 video capture: mobile document acquisition in video mode. *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, 11–16. https://doi.org/10.1109/ICDAR.2017.306

Courtney, J. (2020). CleanPage: Fast and clean document and whiteboard capture. *Journal of Imaging*, *6*(10), 102. https://doi.org/10.3390/jimaging6100102

Courtney, J. (2021). SEDIQA: Sound emitting document image quality assessment in a reading aid for the visually impaired. *Journal of Imaging*, *7*(9), 168. https://doi.org/10.3390/jimaging7090168

de Luna, R. G., Dadios, E. P., Bandala, A. A., & Vicerra, R. R. P. (2019). Size classification of tomato fruit using thresholding, machine learning, and deep learning techniques. *AGRIVITA Journal of Agricultural Science*, *41*(3). https://doi.org/10.17503/agrivita.v41i3.2435

Fahrezi, A., Noer Salam, F., Mahardhika Ibrahim, G., Rahman Syaiful, R., & Saifudin, A. (2022). Pengujian black box testing pada aplikasi inventori barang berbasis web di PT AINO Indonesia. *LOGIC: Jurnal Ilmu Komputer dan Pendidikan*, *1*(1), 1–5. https://www.journal.mediapublikasi.id/index.php/logic/article/view/1262

Harron, N. A., Sulaiman, S. N., Osman, M. K., A. Karim, N. K., & Isa, I. S. (2023). Laplacian-based blur detection algorithm for digital breast tomosynthesis images in improving breast cancer detection. *Journal of Health and Translational Medicine*, *sp2023*(1), 158–164. https://doi.org/10.22452/jummec.sp2023no1.15

Harron, N. A., Sulaiman, S. N., Osman, M. K., Isa, I. S., A. Karim, N. K., & Maruzuki, M. I. F. (2022). Deep learning approach for blur detection of digital breast tomosynthesis images. *Journal of Electrical & Electronic Systems Research*, *21*(OCT2022), 39–44. https://doi.org/10.24191/jeesr.v21i1.006

He, K., & Tu, Y. (2024). Application of computer vision and neural networks in feature extraction and optimization of industrial product design. *Computer-Aided Design and Applications*, 35–49. https://doi.org/10.14733/cadaps.2024.S18.35-49

Hinami, R., Ishiwatari, S., Yasuda, K., & Matsui, Y. (2021). Towards fully automated manga translation. *Proceedings of the AAAI Conference on Artificial Intelligence*, *35*(14), 12998–13008. https://doi.org/10.1609/aaai.v35i14.17537

*How to identify blurry images*. (t.t.). rbaron.net. Diambil 21 November 2024, dari https://rbaron.net/blog/2020/02/16/How-to-identify-blurry-images.html

Kieu, V. C., Cloppet, F., & Vincent, N. (2015). BNRFBE method for blur estimation in document images. Dalam *Advanced Concepts for Intelligent Vision Systems* (hlm. 3–14). Springer International Publishing. https://doi.org/10.1007/978-3-319-25903-1\_1

Kleber, F., Diem, M., Hollaus, F., & Fiel, S. (2017). Mass digitization of archival documents using mobile phones. *Proceedings of the 4th International Workshop on Historical Document Imaging and Processing*, 65–70. https://doi.org/10.1145/3151509.3151526

Li, Z., Yang, C., Shen, Q., & Wen, S. (2021). A document image dataset for quality assessment. *Journal of Physics: Conference Series*, *1828*(1), 012033. https://doi.org/10.1088/1742-6596/1828/1/012033

*Limits*. (t.t.). Vercel. Diambil 6 November 2024, dari https://vercel.com/docs/limits/overview

Liu, Z. J., Ferry, B., & Lacasse, S. (2021). A scalable deep neural network to detect low quality images without a reference. *2020 25th International Conference on Pattern Recognition (ICPR)*, 324–330. https://doi.org/10.1109/ICPR48806.2021.9412549

McCormick, C. (2024, April 23). *Colab GPUs features & pricing*. https://mccormickml.com/2024/04/23/colab-gpus-features-and-pricing/

Nayef, N., Luqman, M. M., Prum, S., Eskenazi, S., Chazalon, J., & Ogier, J.-M. (2015). SmartDoc-QA: A dataset for quality assessment of smartphone captured document images - single and multiple distortions. *Proceedings of the sixth international workshop on Camera Based Document Analysis and Recognition (CBDAR)*. https://doi.org/10.1109/ICDAR.2015.7333960

Pagaduan, R. A., R. Aragon, Ma. C., & Medina, R. P. (2020). iBlurDetect: Image blur detection techniques assessment and evaluation study. *Proceedings of the International Conference on Culture Heritage, Education, Sustainable Tourism, and Innovation Technologies*, 286–291. https://doi.org/10.5220/0010307702860291

Perić, S., Milojković, M., Stan, S.-D., Banić, M., & Antić, D. (2022). Dealing with low quality images in railway obstacle detection system. *Applied Sciences*, *12*(6). https://doi.org/10.3390/app12063041

Pratama, P. A., & Nurdiana, N. (2020). Evaluasi kualitas penerangan ruang kuliah fakultas teknik Universitas PGRI Palembang. *Jurnal Ampere*, *5*(2), 75. https://doi.org/10.31851/ampere.v5i2.5058

*Pricing on Vercel*. (t.t.). Vercel. Diambil 6 November 2024, dari https://vercel.com/docs/pricing

*Pricing updates on the WhatsApp business platform*. (t.t.). Meta. Diambil 6 November 2024, dari https://developers.facebook.com/docs/whatsapp/pricing/updates-to-pricing

Purwanto, Y. S., & Veranita, M. (2018). Pelatihan fotografi dasar bagi pelaku usaha kecil dan menengah (UKM) Kecamatan Lengkong Kota Bandung. *Dharma Bhakti Ekuitas*, *2*(2). https://doi.org/10.52250/p3m.v2i2.74

Rai, P. K., Maheshwari, S., Mehta, I., Sakurikar, P., & Gandhi, V. (2017). Beyond OCRs for document blur estimation. *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, *01*, 1101–1107. https://doi.org/10.1109/ICDAR.2017.182

Rodin, D., Loginov, V., Zagaynov, I., & Orlov, N. (2021). Document image quality assessment via explicit blur and text size estimation. Dalam J. Lladós, D. Lopresti, & S. Uchida (Ed.), *Document Analysis and Recognition – ICDAR 2021* (hlm. 281–292). Springer International Publishing. https://doi.org/10.1007/978-3-030-86337-1\_19

Sa’diah, F. A., Syarif, R., & Azhar, Z. (2024). Analisis atas prosedur pembiayaan kredit pemilikan rumah BTN Platinum. *Jurnal Aplikasi Bisnis Kesatuan*, *4*(3), 361–370. https://doi.org/10.37641/jabkes.v4i3.1927

Souibgui, M. A., Biswas, S., Jemni, S. K., Kessentini, Y., Fornes, A., Llados, J., & Pal, U. (2022). DocEnTr: An end-to-end document image enhancement transformer. *2022 26th International Conference on Pattern Recognition (ICPR)*, 1699–1705. https://doi.org/10.1109/ICPR56361.2022.9956101

Sree, H., Narayana, L., & Rao, S. (2018). Robust face recognition for blurred images with iterative graph based restoration using linear collaborative discriminant regression classification (LCDRC). *ARPN Journal of Engineering and Applied Sciences*, *13*(22), 8780–8787. https://www.arpnjournals.org/jeas/research\_papers/rp\_2018/jeas\_1118\_7389.pdf

Tran, G. S., Nghiem, T. P., & Burie, J.-C. (2020). Fast parallel blur detection on GPU. *Journal of Real-Time Image Processing*, *17*(4), 903–913. https://doi.org/10.1007/s11554-018-0837-1

*Vercel hobby plan*. (t.t.). Vercel. Diambil 6 November 2024, dari https://vercel.com/docs/accounts/plans/hobby

*WhatsApp business platform: Postman API network*. (t.t.). Postman. Diambil 25 Desember 2024, dari https://www.postman.com/meta/whatsapp-business-platform/

*WhatsApp business platform pricing*. (t.t.). Meta. Diambil 6 November 2024, dari https://business.whatsapp.com/products/platform-pricing

Wu, L., Zhang, X., Chen, H., Wang, D., & Deng, J. (2021). VP-NIQE: An opinion-unaware visual perception natural image quality evaluator. *Neurocomputing*, *463*, 17–28. https://doi.org/10.1016/j.neucom.2021.08.048

Zhao, M., Li, D., Shi, Z., Du, S., Li, P., & Hu, J. (2019). Blur feature extraction plus automatic KNN matting: A novel two stage blur region detection method for local motion blurred images. *IEEE Access*, *7*, 181142–181151. https://doi.org/10.1109/ACCESS.2019.2959004

Zhuang, H., Xia, Y., Wang, N., & Dong, L. (2021). High inclusiveness and accuracy motion blur real-time gesture recognition based on YOLOv4 model combined attention mechanism and DeblurGanv2. *Applied Sciences*, *11*(21). https://doi.org/10.3390/app11219982

# LAMPIRAN

*Tuliskan lampiran-lampiran yang ada sesuai dengan tugas akhir yang dipilih*

Misalnya:

Daftar Riwayat Hidup

Lembar Konsultasi dari SIMPONI

Kode Program

Contoh-contoh dataset (sesuai topik tugas akhir)

Lembar-lembar bukti wawancara (jika ada)

Foto bukti wawancara / bukti pengambilan data (jika ada)

Surat pengantar dari kampus (jika ada)

Surat balasan dari perusahaan, dan surat-surat lainnya (jika ada)

Bukti pengambilan data di perusahaan (jika ada)

bukti plagiarism checker

notulen tugas akhir

dan lain lain

## Daftar Riwayat Hidup

****

**Data Pribadi**

Nama Lengkap : Xxxxxxxxx

Tempat, Tanggal Lahir : Xxxxxx, 00 Mei XXXX

Jenis Kelamin : Xxxxx

Alamat : Xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

Nomor Telepon : +62 123456789

Agama : Xxxxxx

Email : Xxxxxx@gmail.com

Riwayat Pendidikan :

1. xxxx – xxxx : TK XXXXX
2. xxxx – xxxx : SD XXXXX
3. xxxx – xxxx : SMP XXXXX
4. xxxx – xxxx : SMA XXXXX
5. xxxx – Sekarang : Universitas Multi Data Palembang

Pengalaman Organisasi :

1. XXXXXXXXXXXX
2. XXXXXXXXXXXX

Prestasi :

1. XXXXXXXXXXXXXXX
2. XXXXXXXXXXXXXXX
3. XXXXXXXXXXXXXXX

## Lembar Konsultasi

|  |  |
| --- | --- |
|  | **LEMBAR KONSULTASI**  **TUGAS AKHIR**  **(CONTOH) (FILE ASLI HARUS DIUNDUH DI SIMPONI)** |
| XX-XXX-XXX-XX.XX |

TAHUN AKADEMIK : 202X/202X Hal x dari x

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Nama / NPM | | | | Xxxxxxxxxxx | | xxxxxxx | | |
| Program Studi | | | | XXXXXXX | | | | |
| Topik | | | | Xxxxxxxxxxxxxxxxx | | | | |
| Judul | | | | Xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx | | | | |
| Pembimbing | | | | **(lembar diunduh pada menu di SIMPONI pada akhir bimbingan saat mendaftar sidang TA)** | | | | |
|  | | | |  | | | | |
| **No** | **Tanggal** | | **Uraian** | | | | **Paraf**  **Pembimbing** | |
| **I** | **II** |
|  |  | |  | | | |  |  |
|  |  | |  | | | |  |  |
| Persetujuan Pembimbing | | | | | JUDUL TUGAS AKHIR :  Xxxxxxxxxxxxxxxxxxx | | | |
| Pembimbing 1 | | Pembimbing 2 | | |

## Kode Program

XXXXXXXXXXXXXXXXXXXX

jika kode program terlalu banyak, seperti aplikasi website, cukup ambil contoh/intinya saja, misal kode program terkait penjualan

XXXXXXXXXX

Kode Program – Rancangan Database

Kode Program Menu XXXXX

Kode Program Laman XXXXXX

Kode Program CSS – Desain Menu XXXX

## <judul lampiran>

## <judul lampiran>

## Form Hasil Pemeriksaan Tingkat Plagiarisme

|  |  |
| --- | --- |
|  | **FORM HASIL PEMERIKSAAN TINGKAT PLAGIARISME**  **TUGAS AKHIR** |

Nama Mahasiswa : ………………………………………………………...

NPM : ……………………………………………………..….

Pembimbing I : …………...…………………………..……… (paraf: )

Pembimbing II : …………...………………………..…...….… (paraf: )

Tahun Akademik : …..….. /…..….. Semester : Gasal / Genap\*

|  |  |  |  |
| --- | --- | --- | --- |
| **BAB** | **Tingkat Plagiarisme (%)** | | **Keterangan** |
| **Sebelum** | **Sesudah** |
| I |  |  |  |
| II |  |  |  |
| III |  |  |  |
| IV |  |  |  |
| V |  |  |  |

\*) Coret yang tidak perlu

Palembang, ……………………………..

Mengetahui,

Ketua Program Studi ………………..

**Keterangan:**

|  |  |
| --- | --- |
| Bab | Tugas Akhir |
| I | 30% |
| II | 50% |
| III | 40% |
| IV | 30% |
| V | 30% |

Kolom Keterangan diisi dengan berita setelah pengecekan Plagiarisme

(contoh: diterima atau ditolak untuk mendaftar Ujian Komprehensif).

Laporan hasil pengecekan Plagiarisme yang **dilampirkan** bersama dengan Form Hasil Pemeriksaan Tingkat Plagiarisme Tugas Akhir, antara lain:

- Laporan pada saat ada indikasi Plagiarisme.

- Laporan pada saat setelah dilakukan perbaikan Plagiarisme.

Tingkat Plagiarisme **maksimal** yang diizinkan oleh Universitas Multi Data Palembang mengikuti ketentuan berikut ini

## Laporan Hasil Pengecekan Plagiarisme

Lampirkan bukti-bukti hasil pengecekan plagiarisme

## Notulen Tugas Akhir