TUGAS AKHIR – KI141502

SISTEM INFORMASI PENGELOLAAN KEUANGAN SEKOLAH (SIPKS) DENGAN *GENERATIVE ADVERSARIAL NETWORKS* UNTUK *FACE RECOGNITION*

ATIKA RIZKI NURAKHMAH
NRP 05111740000015

Dosen Pembimbing
Ir. Siti Rochimah, MT., Ph.D.
Dwi Sunaryono, S.Kom., M.Kom.

DEPARTEMEN INFORMATIKA
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2021

TUGAS AKHIR – KI141502

SISTEM INFORMASI PENGELOLAAN KEUANGAN SEKOLAH (SIPKS) DENGAN *GENERATIVE ADVERSARIAL NETWORKS* UNTUK *FACE RECOGNITION*

ATIKA RIZKI NURAKHMAH
NRP 05111740000015

Dosen Pembimbing
Ir. Siti Rochimah, MT., Ph.D.
Dwi Sunaryono, S.Kom., M.Kom.

DEPARTEMEN INFORMATIKA
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2021

[Halaman ini sengaja dikosongkan]

UNDERGRADUATE THESIS – KI141502

SCHOOL FINANCIAL MANAGEMENT INFORMATION SYSTEM (SIPKS) WITH GENERATIVE ADVERSARIAL NETWORKS FOR FACE RECOGNITION

ATIKA RIZKI NURAKHMAH
NRP 05111740000015

Advisor
Ir. Siti Rochimah, MT., Ph.D.
Dwi Sunaryono, S.Kom., M.Kom.

INFORMATICS DEPARTEMENT
Faculty Of Intelligent Electrical And Informatics Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2021

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

SISTEM INFORMASI PENGELOLAAN KEUANGAN SEKOLAH DENGAN *GENERATIVE ADVERSARIAL NETWORKS* UNTUK *FACE RECOGNITION*

TUGAS AKHIR

Diajukan Guna Memenuhi Salah Satu Syarat Memperoleh Gelar
Sarjana Komputer pada Rumpun Mata Kuliah Rekayasa
Perangkat Lunak Program Studi S-1 Departemen Informatika
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Oleh :

ATIKA RIZKI NURAKHMAH
NRP : 05111740000015

Disetujui oleh Dosen Pembimbing Tugas Akhir :

Dr. Ir. Siti Rochimah, M.T.

NIP: 196810021994032001 (pembimbing 1)

Dwi Sunaryono, S.Kom., M.Kom.

NIP: 197205281997021001 (pembimbing 2)

SURABAYA
JANUARI 2021

SISTEM INFORMASI PENGELOLAAN KEUANGAN SEKOLAH (SIPKS) DENGAN *GENERATIVE ADVERSARIAL NETWORKS* UNTUK *FACE RECOGNITION*

Nama Mahasiswa	: ATIKA RIZKI NURAKHMAH
NRP	: 05111740000015
Jurusan	: Informatika ITS
Dosen Pembimbing I	: Dr. Ir. Siti Rochimah, M.T.
Dosen Pembimbing II	: Dwi Sunaryono, S.Kom., M.Kom.,

Abstrak

Proses pengajuan pencairan dana anggaran pada mayoritas sekolah masih dilakukan secara manual. Proses manual ini membutuhkan waktu yang cukup lama dan memperbesar resiko terjadinya kesalahan, kekeliruan, maupun kecurangan dalam prosesnya, sehingga menimbulkan kerugian bagi sekolah. Oleh karena itu, dibuatlah aplikasi Sistem Informasi Pengelolaan Keuangan Sekolah (SIPKS).

Tugas akhir ini membahas perancangan dan implementasi SIPKS pada platform website dan android. Pada SIPKS diterapkan teknologi face recognition untuk validasi persetujuan pengajuan dengan sample yang diperoleh menggunakan Generative Adversarial Networks, algoritma deep learning yang memungkinkan wajah untuk dikenali meskipun kualitas foto wajah blur. Algoritma ini menghasilkan gambar baru dengan kualitas fokus yang lebih baik, sehingga memperbaiki data set untuk face recognition. Aplikasi SIPKS diharapkan dapat meminimalisir terjadinya kecurangan dan kekeliruan dalam pengelolaan keuangan sekolah.

Kata kunci: SIPKS, Generative Adversarial Networks, Android, Website, Anggaran Dana

SCHOOL FINANCIAL MANAGEMENT INFORMATION SYSTEM (SIPKS) WITH GENERATIVE ADVERSARIAL NETWORKS FOR FACE RECOGNITION

Name	: ATIKA RIZKI NURAKHMAH
NRP	: 05111740000015
Major	: Informatika ITS
Supervisor I	: Dr. Ir. Siti Rochimah, M.T.
Supervisor II	: Dwi Sunaryono, S.Kom., M.Kom.,

Abstract

The process for submitting budget funds to schools is still done manually. This manual process takes a long time, allow mistakes and cheating to occur in the process, which can cause losses for schools itself. Therefore, the School Financial Management Information System (SIPKS) application was made.

This final project discusses the design and implementation of SIPKS on website and android platforms. SIPKS applies face recognition technology to validate submission agreements with samples obtained using Generative Adversarial Networks, a deep learning algorithm that allows faces to be recognized even though the quality of facial photos is blurry. This algorithm produces new images with better focus quality, thus improving the dataset for face recognition. The SIPKS application is expected to minimize damage and mistakes in managing school finances.

Keywords: SIPKS, Generative Adversarial Networks, Android, Website, School Finances

KATA PENGANTAR

Segala puji dan syukur penulis panjatkan atas kehadirat Allah SWT, yang oleh karena anugrahNya penulis dapat menyelesaikan Tugas Akhir yang berjudul "SISTEM INFORMASI PENGELOLAAN KEUANGAN SEKOLAH (SIPKS) DENGAN *GENERATIVE ADVERSARIAL NETWORKS* UNTUK *FACE RECOGNITION*".

Pengerjaan Tugas Akhir ini menjadi suatu pengalaman yang baik bagi penulis. Tugas Akhir ini selesai karena tidak lepas dari bantuan dan dukungan berbagai pihak. Pada kesempatan kali ini penulis ingin mengucapkan terima kasih kepada:

1. Allah SWT, Atas segala rahmat dan berkatNya bagi penulis.
2. Bapak Wahyudi, Mama Arie Sudiastuti, Budhe Surantini, Mas Reza, Mas Mukti, Mbak Ika, Rashif dan Azzahra yang senantiasa menyemangati memberikan penulis do'a, dukungan dan semangat dari rumah untuk penulis.
3. Ibu Siti Rochimah selaku dosen pembimbing 1 yang telah membantu penulis untuk menyelesaikan Tugas Akhir ini.
4. Bapak Dwi Sunaryono selaku dosen pembimbing 2 yang telah membantu penulis untuk menyelesaikan Tugas Akhir ini.
5. Mas John Harison dan Mas Irzal Ahmad Sabilla, yang telah membimbing penulis dan membantu penulis selama pembuatan Tugas Akhir ini.
6. Chaniyah Zulfa dan Lutfiyanti yang telah melengkapi Tugas Akhir ini bersama penulis dan menjadi teman berjuang sekaligus bermain selama mengerjakan Tugas Akhir ini.
7. Roficho, Adelia, Debora dan Ficky yang telah menemani, menghibur, dan menyemangati penulis selama di ITS.
8. Karina Soraya yang telah menjadi teman satu kamar selama penulis berada di Surabaya.
9. Teman-teman IDC, UKTK ITS, IFLS ITS, Victoria DC, dan Valkyrie DC yang telah menjadi penyemangat juga menemani penulis melakukan hobinya selama di ITS dan Surabaya.

10. Teman-teman angkatan 2017 yang telah membantu, menyemangati, dan menemani penulis selama berada di Departemen Informatika ITS.
11. Semua pihak terkait yang tidak bisa saya sebut satu persatu.

Penulis menyadari bahwa Tugas Akhir ini masih memiliki banyak kekurangan, sehingga dengan kerendahan hati, penulis mengharapkan kritik dan saran dari pembaca untuk perbaikan ke depannya.

Surabaya, Desember 2020

Atika Rizki Nurakhmah

DAFTAR ISI

Abstrak ……vi

Abstract ……..vii

KATA PENGANTAR……..viii

DAFTAR ISI……x

DAFTAR GAMBAR ……..xvii

DAFTAR TABEL ……...xxii

DAFTAR KODE SUMBER ……...xxv

1 BAB I PENDAHULUAN……. 1
1.1 Latar Belakang……. 1
1.2 Rumusan Masalah…… 3
1.3 Batasan Masalah…….. 3
1.4 Tujuan…….. 4
1.5 Manfaat…… 4
1.6 Metodologi……4
1.7 Sistematika Penulisan…….. 6
2 BAB II TINJAUAN PUSTAKA…….. 7
2.1 Pengenalan Wajah…… 7
2.2 *Generative Adversarial Networks (GANs)*……... 8
2.3 *DeblurGAN*……..... 11
2.4 PHP……. 11
2.5 Laravel…… 11
2.6 Python……..… 11
2.7 MobileNet……... 12
2.8 Android……... 12
2.9 Java……. 12
3 BAB III ANALISIS DAN PERANCANGAN……... 13
3.1 Analisis……13
3.1.1 Analisis Permasalahan……13

3.1.2 Deskripsi Umum Sistem……..16
3.1.3 Spesifikasi Perangkat Lunak……...17
3.1.3.1 Aktor……17
3.1.3.2 Kebutuhan Fungsional……..18
3.1.3.3 Kebutuhan Non Fungsional……..19
3.1.3.4 Diagram Kasus penggunaan……19
3.1.3.4.1 Kasus Penggunaan Menambahkan Pengajuan Dana Anggaran…….21
3.1.3.4.2 Kasus Penggunaan Melihat Pengajuan Dana Anggaran (Website)……23
3.1.3.4.3 Kasus Penggunaan Mengubah Pengajuan Dana Anggaran…….……25
3.1.3.4.4 Kasus Penggunaan Menghapus
3.1.3.4.5 Pengajuan Dana Anggaran…….27
3.1.3.4.6 Kasus Penggunaan Melihat Detail Pengajuan Dana Anggaran…….28
3.1.3.4.7 Kasus Penggunaan Mengubah Detail Pengajuan Dana Anggaran…….30
3.1.3.4.8 Kasus Penggunaan Menghapus Detail Pengajuan Dana Anggaran…….32
3.1.3.4.9 Kasus Penggunaan Melihat Pengajuan Dana Anggaran (Mobile)……...34
3.1.3.4.10 Kasus Penggunaan Mengunggah Sampel Wajah…….35
3.1.3.4.11 Kasus Penggunaan Melakukan Training Sampel Wajah……37
3.1.3.4.12 Kasus Penggunaan Memberikan Persetujuan Pengajuan dengan Pengenalan Wajah……40
3.1.3.4.13 Kasus Penggunaan Menolak Pengajuan Dana Anggaran……...42
3.2 Perancangan……44
3.2.1 Perancangan Arsitektur Sistem……..44
3.2.2 Perancangan Basis Data……45
3.2.2.1 Penjelasan Tabel……..46
3.2.2.1.1 Tabel User……46

3.2.2.1.2 Tabel Jabatan……..46
3.2.2.1.3 Tabel Sekolah…….46
3.2.2.1.4 Tabel Pengajuan…….46
3.2.2.1.5 Tabel Detail Pengajuan……..46
3.2.2.1.6 Tabel Mapping Detail Pengajuan……...46
3.2.3 Perancangan Proses…….46
3.2.3.1 Perancangan Proses *Generate* Perbaikan Foto.46
3.2.3.2 Perancangan Proses Pengenalan Wajah..……47
3.2.4 Perancangan Antarmuka Pengguna……48
3.2.4.1 Rancangan Antarmuka Halaman *Login* Website……48
3.2.4.2 Rancangan Antarmuka Halaman List Data Pengajuan…….49
3.2.4.3 Rancangan Antarmuka Halaman Tambah Pengajuan…….51
3.2.4.4 Rancangan Antarmuka Halaman Edit Pengajuan…….54
3.2.4.5 Rancangan Antarmuka Halaman Hapus Pengajuan…….56
3.2.4.6 Rancangan Antarmuka Halaman Tambah Detail Pengajuan……..57
3.2.4.7 Rancangan Antarmuka Halaman List Detail Pengajuan…….60
3.2.4.8 Rancangan Antarmuka Halaman Edit Detail Pengajuan…….61
3.2.4.9 Rancangarn Antarmuka Halaman Hapus Detail Pengajuan……..63
3.2.4.10 Rancangan Antarmuka Halaman *Start Page* Android……64
3.2.4.11 Rancangan Antarmuka Halaman *Login* Android……66
3.2.4.12 Rancangan Antarmuka Halaman *Dashboard* Android……67
3.2.4.13 Rancangan Antarmuka Halaman Menu Sidebar…….69

3.2.4.14 Rancangan Antarmuka Halaman Pengajuan Belum Disetujui…….....72
3.2.4.15 Rancangan Antarmuka Halaman Upload Wajah……74
3.2.4.16 Rancangan Antarmuka Halaman *Training* Wajah……....76
3.2.4.17 Rancangan Antarmuka Halaman Pengajuan Sudah Disetujui…….77
3.2.4.18 Rancangan Antarmuka Halaman Pengajuan Ditolak……..78
3.2.4.19 Rancangan Antarmuka Halaman Berhasil Disetujui…….. 80
3.2.5 Perancangan API *Endoint*…….. 82
3.2.5.1 Struktur JSON API *Endpoint*……...82
4 BAB IV IMPLEMENTASI…….83
4.1 Lingkungan Implementasi……...83
4.2 Implementasi Antarmuka Sistem……84
4.2.1 Implementasi Antarmuka Halaman *Login* Website……. 84
4.2.2 Implementasi Antarmuka Halaman Data Pengajuan…….85
4.2.3 Implementasi Antarmuka Halaman Tambah Pengajuan……..85
4.2.4 Implementasi Antarmuka Halaman Tambah Detail Pengajuan……...86
4.2.5 Implementasi Antarmuka Halaman Edit Pengajuan……..87
4.2.6 Implementasi Antarmuka Halaman Hapus Pengajuan……..88
4.2.7 Implementasi Antarmuka Halaman List Detail Pengajuan……..88
4.2.8 Implementasi Antarmuka Halaman Edit Detail Pengajuan……..89
4.2.9 Implementasi Antarmuka Halaman Hapus Detail Pengajuan……...89

4.2.10 Implementasi Antarmuka Halaman *Start Page* Android…….90
4.2.11 Implementasi Antarmuka Halaman *Login* Android……...90
4.2.12 Implementasi Antarmuka Halaman *Dashboard* Android…….91
4.2.13 Implementasi Antarmuka Halaman Menu Sidebar……92
4.2.14 Implementasi Antarmuka Halaman Pengajuan Belum Disetujui……93
4.2.15 Implementasi Antarmuka Halaman Upload Wajah……...94
4.2.16 Implementasi Antarmuka Halaman *Training* Wajah…….95
4.2.17 Rancangan Antarmuka Halaman List Pengajuan Sudah Disetujui……...96
4.2.18 Implementasi Antarmuka Halaman List Pengajuan Ditolak…….97
4.2.19 Implementasi Antarmuka Halaman Berhasil Disetujui……98
4.3 Implementasi Kasus Penggunaan Aplikasi…….99
4.3.1 Implementasi Kasus Penggunaan Menambahkan Pengajuan Dana Anggaran……..100
4.3.2 Implementasi Kasus Penggunaan Melihat Pengajuan Dana Anggaran (Website)……..102
4.3.3 Implementasi Kasus Penggunaan Mengubah Pengajuan Dana Anggaran……..103
4.3.4 Implementasi Kasus Penggunaan Menghapus Pengajuan Dana Anggaran……..104
4.3.5 Implementasi Kasus Penggunaan Melihat Detail Pengajuan Dana Anggaran……...105
4.3.6 Implementasi Kasus Penggunaan Mengubah Detail Pengajuan Dana Anggaran……...106
4.3.7 Implementasi Kasus Penggunaan Menghapus Detail Pengajuan Dana Anggaran……...108

4.3.8 Implementasi Kasus Penggunaan Melihat Pengajuan Dana Anggaran (Mobile)……..109
4.3.9 Implementasi Kasus Penggunaan Mengunggah Sampel Wajah……110
4.3.10 Implementasi Kasus Penggunaan Melakukan Training Sampel Wajah……..112
4.3.11 Implementasi Kasus Penggunaan Memberikan Persetujuan Pengajuan dengan Pengenalan Wajah……..114
4.3.12 Implementasi Kasus Penggunaan Menolak Pengajuan Dana Anggaran……..115
4.4 Implementasi Proses Perbaikan Foto……117
4.5 Implementasi Proses Pengenalan Wajah..……118
4.6 Penggunaan API *Endpoint*...……119
4.6.1 Implementasi API *Endpoint Login*……119
4.6.2 Implementasi API *Endpoint Profile*……120
4.6.3 Implementasi API *Endpoint* Pengajuan……..121
4.6.4 Implementasi API *Endpoint* Detail Pengajuan…122
4.6.5 Implementasi API *Endpoint* Mengirim Foto Wajah……..123
4.6.6 Implementasi API *Endpoint Training* Foto Wajah……125
4.6.7 Implementasi API *Endpoint* Menyetujui Pengajuan……126
5 UJI COBA DAN EVALUASI…….. 127
5.1 Lingkungan Uji Coba……127
5.2 Pengujian Kasus Penggunaan……128
5.2.1 Pengujian Menambahkan Pengajuan Dana Anggaran……128
5.2.2 Pengujian Melihat Pengajuan Dana Anggaran (Website)……131
5.2.3 Pengujian Mengubah Pengajuan Dana Anggaran……132
5.2.4 Pengujian Menghapus Pengajuan Dana Anggaran……135

5.2.5 Pengujian Melihat Pengajuan Dana Anggaran (mobile)……137
5.2.6 Pengujian Mengunggah Sampel Wajah……...140
5.2.7 Pengujian Melakukan *Training* Foto Wajah…142
5.2.8 Pengujian Memberikan Persetujuan dengan Pengenalan Wajah……...144
5.2.9 Pengujian Memberikan Penolakan Pengajuan Dana Anggaran……148
5.3 Pengujian Koneksi Internet……..150
5.4 Pengujian Kasus Pengenalan Wajah……151
6 KESIMPULAN DAN SARAN……155
6.1 Kesimpulan…….....155
6.2 Saran……156
7 DAFTAR PUSTAKA……..157
8 LAMPIRAN…….....159

BIODATA PENULIS……172

DAFTAR GAMBAR

Gambar 2.1. *Generative* model dalam *GANS*……..8
Gambar 2.2. *Discriminator* model dalam *GANS*…….....9
Gambar 2.3. *Output training discriminator*…….....9
Gambar 2.4. Proses *GANS*….……10
Gambar 2.5. Contoh penggunaan algoritma *GANS*……...10
Gambar 2.6. Arsitektur Generator *DeblurGAN*…….....11
Gambar 3.1. Proses Bisnis Umum SIPKS……...…..14
Gambar 3.1.1. Proses Bisnis dengan Pengenalan Wajah ...……...15
Gambar 3.2. Deskripsi Umum Sistem……...17
Gambar 3.3. Diagram Kasus Penggunaan Website SIPKS…… ..20
Gambar 3.4. Diagram Kasus Penggunaan Aplikasi Android……20
Gambar 3.5. Diagram Aktivitas Menambahkan Pengajuan Dana.23
Gambar 3.6 Diagram Aktivitas Melihat Pengajuan Dana…….. .24
Gambar 3.7. Diagram Aktivitas Mengubah Pengajuan Dana…. .26
Gambar 3.8. Diagram Aktivitas Menghapus Pengajuan Dana… 28
Gambar 3.9. Diagram Aktivitas Melihat Detail Pengajuan Dana Anggaran……. 29
Gambar 3.10. Diagram Aktivitas Mengubah Detail Pengajuan Dana Anggaran……31
Gambar 3.11. Diagram Aktivitas Menghapus Detail Pengajuan Dana Anggaran……33
Gambar 3.12. Diagram Aktivitas Melihat Pengajuan Dana Anggaran (Mobile)……..35
Gambar 3.13. Diagram Aktivitas Mengunggah Sampel Wajah……..37
Gambar 3.14. Diagram Aktivitas Melakukan *Training* Sampel Wajah ……...……..39
Gambar 3.15. Diagram Aktivitas Memberikan Persetujuan Pengajuan dengan Pengenalan Wajah…… 42
Gambar 3.16. Diagram Aktivitas Menolak Pengajuan Dana Anggaran……44
Gambar 3.17. Ilustrasi Arsitektur *Platform* Website ……..45
Gambar 3.18. Ilustrasi Arsitektur *Platform Mobile*……....45

Gambar 3.19. Proses *Generate* Perbaikan Foto ……..……...47
Gambar 3.20. Proses Pengenalan Wajah ……..47
Gambar 3.21. Rancangan Halaman Antarmuka *Login* Website….....49
Gambar 3.22. Rancangan Antarmuka Halaman Data Pengajuan.....50
Gambar 3.23. Rancangan Antarmuka Halaman Tambah Pengajuan……..52
Gambar 3.24 Rancangan Antarmuka Halaman Edit Pengajuan….....56
Gambar 3.25 Rancangan Antarmuka Halaman Hapus Pengajuan.....57
Gambar 3.26 Rancangan Antarmuka Halaman Tambah Detail Pengajuan……..58
Gambar 3.27 Rancangan Antarmuka Halaman List Detail Pengajuan…….…61
Gambar 3.28 Rancangan Antarmuka Halaman Edit Detail Pengajuan…….……62
Gambar 3.29. Rancangan Antarmuka Halaman Hapus Detail Pengajuan…….……64
Gambar 3.30. Rancangan Antarmuka Halaman *Start Page* Android……65
Gambar 3.31. Rancangan Antarmuka Halaman *Login* Android……66
Gambar 3.32. Rancangan Antarmuka Halaman Dashboard Android……69
Gambar 3.33 Rancangan Antarmuka Halaman Menu Sidebar……..70
Gambar 3.34. Rancangan Antarmuka Halaman Pengajuan Belum Disetujui……72
Gambar 3.35. Rancangan Antarmuka Halaman Upload Wajah….....75
Gambar 3.36 Rancangan Antarmuka Halaman *Training* Wajah……76

Gambar 3.37. Rancangan Antarmuka Halaman Pengajuan Sudah Disetujui……77
Gambar 3.38. Rancangan Antarmuka Halaman Pengajuan Ditolak……79
Gambar 3.39. Rancangan Antarmuka Halaman Berhasil Disetujui……81
Gambar 4.1 Implementasi Antarmuka *Login* Website……..84
Gambar 4.2 Implementasi Antarmuka *Login* Website Error…….84
Gambar 4.3. Implementasi Antarmuka Halaman Data Pengajuan……...85
Gambar 4.4. Implementasi Antarmuka Halaman Tambah Pengajuan……...86
Gambar 4.5. Implementasi Antarmuka Halaman Tambah ……..86
Gambar 4.6. Implementasi Antarmuka Halaman Tambah Detail Pengajuan……...87
Gambar 4.7. Implementasi Antarmuka Halaman Detail Pengajuan……...87
Gambar 4.8. Implementasi Antarmuka Halaman Hapus Pengajuan……...88
Gambar 4.9. Implementasi Antarmuka Halaman List Detail Pengajuan……...88
Gambar 4.10. Implementasi Antarmuka Halaman Edit Detail Pengajuan…….…..88
Gambar 4.11. Implementasi Antarmuka Halaman Hapus Detail Pengajuan…….……..89
Gambar 4.12. Implementasi Antarmuka *Start Page* Android……90
Gambar 4.13. Implementasi Antarmuka *Login* Android……..91
Gambar 4.14. Implementasi Antarmuka *Dashboard* Android…...92
Gambar 4.15. Implementasi Antarmuka Menu *Sidebar* Android..93
Gambar 4.16. Implementasi Antarmuka *Start Page* Android …...94
Gambar 4.17. Implementasi Antarmuka Halaman Upload Wajah……95
Gambar 4.18. Implementasi Antarmuka Halaman *Training* Wajah……96

Gambar 4.19. Implementasi Antarmuka Halaman List Pengajuan Sudah Disetujui…….…….97
Gambar 4.20. Implementasi Antarmuka Halaman List Pengajuan Ditolak…….……..98
Gambar 4.21. Implementasi Antarmuka Halaman Berhasil Disetujui……99
Gambar 4.22. Contoh Hasil *Preprocessing*……..117
Gambar 4.23. Hasil Proses Perbaikan Foto……..118
Gambar 4.24 Hasil Pengenalan Wajah ……119
Gambar 5.1 Proses Pengisian Form Pengajuan……130
Gambar 5.2 Proses Pengisian Form Detail Pengajuan…….130
Gambar 5.3 Hasil Pengujian PWU-001 Pengajuan yang Ditambahkan Berhasil Di Tampilkan……...130
Gambar 5.4 Hasil Pengaksesan Halaman Dashboard……..132
Gambar 5.5. Hasil Pengujian PWU-002 List Tabel Pengajuan Berhasil Di Tampilkan…….....132
Gambar 5.6. Proses Pemilihan Pengajuan yang Ingin di Edit…..134
Gambar 5.7. Proses Pengeditan pada Form Edit Pengajuan……134
Gambar 5.8 Hasil Pengujian PWU-003 Pengajuan yang Telah Di Edit Berhasil Di Tampilkan…….135
Gambar 5.9. Proses Penghapusan Pengajuan yang dipilih……..136
Gambar 5.10 Hasil Pengujian PWU-004 Pengajuan yang Telah Dihapus Berhasil Terhapus dari List Pengajuan……..137
Gambar 5.11 Proses *login* ke aplikasi android…….138
Gambar 5.12 Proses mengakses halaman *dashboard* android….138
Gambar 5.13 Proses mengakses menu sidebar untuk memilih status pengajuan…….…139
Gambar 5.14 Hasil pengujian PAU-001 untuk status pengajuan disetujui berhasil ditampilkan……..139
Gambar 5.15 Hasil pengujian PAU-001 untuk status pengajuan ditolak berhasil ditampilkan…….140
Gambar 5.16 Hasil Pengujian PAU-002 Upload Wajah sebanyak 1 kali……....142
Gambar 5.17 Hasil Pengujian PAU-002 Upload Wajah sebanyak 5 kali……..…..142

Gambar 5.18 Proses mengakses halaman *training* wajah……144
Gambar 5.19 Hasil Pengujian PAU-003 Sampel Wajah berhasil di*training*…….…….144
Gambar 5.20 Proses mengakses halaman detail pengajuan yang belum disetujui……..……...146
Gambar 5.21 Proses mengakses halaman ambil foto wajah……146
Gambar 5.22 Hasil Pengujian PAU-004 Pengajuan Berhasil Disetujui Dengan Pengenalan Wajah……...147
Gambar 5.23 Hasil Pengujian PAU-004 Daftar Pengajuan Sudah Disetujui ……..147
Gambar 5.24 Proses mengakses halaman detail pengajuan yang belum disetujui…….149
Gambar 5.25 Hasil Pengujian PAU-005 Persetujuan berhasil ditolak……...149
Gambar 5.26 Hasil Pengujian PAU-005 Persetujuan yang ditolak berhasil masuk ke daftar pengajuan ditolak…….150

DAFTAR TABEL

Tabel 3.1 Spesifikasi Karakter Aktor pada Sistem……17
Tabel 3.2. Kebutuhan Fungsional Sistem …….....18
Table 3.3 Spesifikasi Kasus Penggunaan Menambahkan Pengajuan Dana Anggaran…….....21
Table 3.4 Spesifikasi Kasus Penggunaan Melihat Pengajuan Dana Anggaran……...24
Table 3.5. Spesifikasi Kasus Penggunaan Mengubah Pengajuan Dana Anggaran……...25
Table 3.6. Spesifikasi Kasus Penggunaan Menghapus Pengajuan Dana Anggaran……...27
Table 3.7 Spesifikasi Kasus Penggunaan Melihat Detail Pengajuan Dana Anggaran…….29
Table 3.8. Spesifikasi Kasus Penggunaan Mengubah Detail Pengajuan Dana Anggaran…….30
Table 3.9 Spesifikasi Kasus Penggunaan Menghapus Detail Pengajuan Dana Anggaran…….32
Table 3.10. Spesifikasi Kasus Penggunaan Melihat Pengajuan Dana Anggaran (Mobile)……...34
Table 3.11 Spesifikasi Kasus Penggunaan Mengunggah Sampel Wajah…….36
Table 3.12 Spesifikasi Kasus Penggunaan Melakukan Training Sampel Wajah……38
Table 3.13 Spesifikasi Kasus Penggunaan Memberikan Persetujuan Pengajuan dengan Pengenalan Wajah……..…….40
Table 3.14 Spesifikasi Kasus Penggunaan Menolak Pengajuan Dana Anggaran……...……43
Tabel 3.15 Atribut Antarmuka Halaman *Login* Website ..……49
Tabel 3.16 Atribut Antarmuka Halaman List Data Pengajuan…..50
Tabel 3.17 Atribut Antarmuka Halaman Tambah Pengajuan……52
Tabel 3.18 Atribut Antarmuka Halaman Edit Pengajuan……..54
Tabel 3.19 Atribut Antarmuka Halaman Hapus Pengajuan……...56
Tabel 3.20 Atribut Antarmuka Halaman Tambah Detail Pengajuan……...58
Tabel 3.21 Atribut Antarmuka Halaman List Detail Pengajuan…60

Tabel 3.22 Atribut Antarmuka Halaman Edit Detail Pengajuan....62
Tabel 3.23 Atribut Antarmuka Halaman Hapus Detail Pengajuan……...63
Tabel 3.24 Atribut Antarmuka Halaman *Start Page* Android…...65
Tabel 3.25 Atribut Antarmuka Halaman *Login* Android……...66
Tabel 3.26 Atribut Antarmuka Halaman Dashboard Android…...67
Tabel 3.27 Atribut Antarmuka Halaman Menu Sidebar……70
Tabel 3.28 Atribut Antarmuka Halaman Pengajuan Belum Disetujui……73
Tabel 3.29 Atribut Antarmuka Halaman Upload Wajah……...74
Tabel 3.30 Atribut Antarmuka Halaman *Training* Wajah……76
Tabel 3.31 Atribut Antarmuka Halaman Pengajuan Sudah Disetujui……77
Tabel 3.32 Atribut Antarmuka Halaman Pengajuan Ditolak…….79
Tabel 3.33 Atribut Antarmuka Halaman Berhasil Disetujui……..81
Tabel 3.34 API yang Akan Digunakan……..82
Tabel 4.1 Lingkungan Implementasi Perangkat Lunak…….83
Table 4.2. Penjelasan Fungsi Menambahkan Pengajuan Dana Anggaran……..101
Table 4.3. Penjelasan Fungsi Menambahkan Detail Pengajuan Dana Anggaran……..101
Table 4.4. Penjelasan Fungsi Melihat Pengajuan Dana Anggaran (Website)…….....102
Table 4.5. Penjelasan Fungsi Mengubah Pengajuan Dana Anggaran…….103
Table 4.6. Penjelasan Fungsi Menghapus Pengajuan Dana Anggaran…….105
Table 4.7. Penjelasan Fungsi Melihat Detail Pengajuan Dana Anggaran…….105
Table 4.8. Penjelasan Fungsi Mengubah Pengajuan Dana Anggaran…….107
Table 4.9. Penjelasan Fungsi Menghapus Detail Pengajuan Dana Anggaran……..108
Table 4.10. Penjelasan Fungsi Melihat Pengajuan Dana

Anggaran (Mobile)……...110
Table 4.11. Penjelasan Fungsi Mengunggah Sampel Wajah…...111
Table 4.12. Penjelasan Fungsi Melakukan *Training* Sampel Wajah……...112
Table 4.13. Penjelasan Fungsi Memberikan Persetujuan Pengajuan dengan Pengenalan Wajah……..……114
Table 4.14. Penjelasan Fungsi Menolak Pengajuan Dana Anggaran……..116
Tabel 5.1 Lingkungan Pengujian Aplikasi Web …….127
Tabel 5.2 Lingkungan Pengujian Aplikasi Android……127
Tabel 5.3 Lingkungan Server Aplikasi ……...…...128
Tabel 5.4 Kasus Pengujian Menambahkan Pengajuan Dana Anggaran……..128
Tabel 5.5 Kasus Pengujian Melihat Pengajuan Dana Anggaran (Website)……..131
Tabel 5.6 Kasus Pengujian Mengubah Pengajuan Dana Anggaran……..132
Tabel 5.7 Kasus Pengujian Menghapus Pengajuan Dana Anggaran……..135
Tabel 5.8 Kasus Pengujian Melihat Pengajuan Dana Anggaran..137
Tabel 5.9 Kasus Pengujian Mengunggah Sampel Wajah……140
Tabel 5.10 Kasus Pengujian Melakukan Training Foto Wajah...142
Tabel 5.11 Kasus Pengujian Memberikan Persetujuan dengan Pengenalan Wajah……144
Tabel 5.12 Kasus Pengujian Memberikan Penolakan Pengajuan Dana Anggaran……..……..148
Tabel 5.13 Hasil Uji Kasus Pengenalan Wajah……...151

DAFTAR KODE SUMBER

Kode sumber 4.1 Fungsi Menambahkan Pengajuan Dana Anggaran…….100
Kode sumber 4.2 Fungsi Menambahkan Detail Pengajuan Dana Anggaran ……..….102
Kode sumber 4.3 Fungsi Melihat Pengajuan Dana Anggaran (Website)…….…103
Kode sumber 4.4 Fungsi Mengubah Pengajuan Dana Anggaran…….104
Kode sumber 4.5 Fungsi Menghapus Pengajuan Dana……...104
Kode sumber 4.6 Fungsi Melihat Detail Pengajuan Dana……...106
Kode sumber 4.7 Fungsi Mengubah Detail Pengajuan Dana 107
Kode sumber 4.8 Fungsi Menghapus Detail Pengajuan Dana …108
Kode sumber 4.9 Fungsi Melihat Pengajuan Dana Anggaran (Mobile)……...109
Kode sumber 4.10 Fungsi Mengunggah Sampel Wajah…….111
Kode sumber 4.11 Fungsi Melakukan *Training* Sampel Wajah.113
Kode sumber 4.12 Fungsi Memberikan Persetujuan Pengajuan dengan Pengenalan Wajah……..115
Kode sumber 4.13 Fungsi Menolak Pengajuan Dana Anggaran…….116
Kode sumber 4.14 Implementasi API *Endpoint Login* dengan metode POST……..120
Kode sumber 4.15 Implementasi API *Endpoint Profile* dengan metode GET……120
Kode sumber 4.16 Implementasi API *Endpoint* Pengajuan dengan metode GET……122
Kode sumber 4.17 Implementasi API *Endpoint* Detail Pengajuan dengan metode GET……...123
Kode sumber 4.18 Implementasi API *Endpoint* Mengirim Foto Wajah dengan metode POST……..…….124
Kode sumber 4.19 Implementasi API *Endpoint Training* Foto Wajah dengan metode GET……125
Kode sumber 4.20 Implementasi API *Endpoint* Menyetujui Pengajuan dengan metode POST…...……..126

BAB I

PENDAHULUAN

Bab pendahuluan membahas garis besar penyusunan Tugas Akhir yang meliputi latar belakang, tujuan pembuatan, rumusan dan batasan permasalahan, metodologi penyusunan Tugas Akhir, dan sistematika penulisan.

1.1 Latar Belakang

Menurut Data Pokok Pendidikan Dasar dan Menengah Kementerian Pendidikan dan Kebudayaan (Dapo Kemendikbud) tahun 2019, jumlah sekolah di Kota Surabaya mencapai 1.290 sekolah. Jumlah tersebut terbagi menjadi 398 sekolah negeri dan 892 sekolah swasta dari jenjang dasar sampai menengah (SD, SMP, SMA, SMK)[1]. Keberlangsungan kegiatan di sekolah tidak luput dari pengelolaan keuangan di sekolah tersebut. Sekolah swasta mendapatkan dana dari peserta didik sendiri, biasanya berbentuk iuran bulanan atau SPP, uang gedung, dan uang pangkal. Sedangkan sekolah negeri mendapatkan dana dari pemerintah. Dana pendidikan tersebut disalurkan ke setiap sekolah negeri. Setiap sekolah wajib memberikan anggaran dana beserta laporan penggunaan dana anggaran sebagai bentuk pertanggungjawaban kepada pemerintah. Pengelolaan keuangan sekolah menjadi hal yang sangat penting sehingga membutuhkan perhatian dalam pelaksanaannya. Namun, banyak sekolah negeri yang melakukan proses pengelolaan secara manual, seperti pencatatan di buku, perhitungan manual, dan administrasi manual. Proses manual membutuhkan waktu yang cukup lama dan resiko kesalahan pengelolaan keuangan juga cukup besar. Terlebih lagi jika berkaitan dengan persetujuan tentang dana anggaran. Persetujuan harus benar-benar dilakukan oleh pejabat sekolah yang berwenang, dalam hal ini Bendahara dan Kepala Sekolah. Sebagai legalitas pengajuan, Bendahara sekolah dan Kepala sekolah harus menandatangani dokumen pengajuan jika ingin memberikan persetujuan. Untuk memastikan keaslian tanda tangan cukup sulit

karena tanda tangan rawan untuk dipalsukan. Oleh karena itu, proses pengelolaan keuangan sekolah yang masih manual merupakan masalah yang dialami banyak sekolah terutama berkaitan dengan autentifikasi pemberi persetujuan pencairan anggaran dana.

Berdasarkan permasalahan tersebut, maka dibutuhkan sebuah sistem yang dapat memudahkan pihak sekolah dalam hal pengelolaan dana anggaran sekolah. Sistem ini harus memiliki jaminan terhadap autentifikasi atau keaslian pemberi persetujuan pengajuan. Solusi ini dapat diwujudkan dengan memanfaatkan teknologi yaitu teknologi pengenalan wajah (*face recognition*) melalui ponsel cerdas dalam proses pengajuan pencairan dana anggaran sekolah, sehingga persetujuan yang diberikan oleh bendahara dan kepala sekolah terjamin keasliannya. Dalam praktiknya, dataset yang digunakan pada proses training pengenalan wajah tidak selalu memiliki kualitas yang baik. Terkadang kualitas foto yang diperoleh kurang bagus dikarenakan spesifikasi ponsel yang digunakan pengguna berbeda-beda. Selain itu, kondisi saat pengambilan gambar yang berbeda pula antara satu pengguna dengan pengguna lainnya. Contohnya saat mobilitas pengguna yang padat, sehingga pengguna terburu-buru dalam mengambil foto dan akhirnya foto yang dihasilkan blur. Foto blur seringkali tidak berhasil dikenali pada proses deteksi objek hingga *face recognition*[14]. Pengguna harus melakukan pengambilan gambar ulang hingga hasilnya cocok, hal ini kurang efektif. Oleh karena itu, dibutuhkan teknologi yang dapat memperbaiki dataset untuk proses pengenalan wajah agar wajah dapat dikenali walaupun pada kualitas lingkungan yang kurang baik. Penggunaan teknologi ini erat kaitannya dengan algoritma deep learning, salah satunya adalah algoritma *Generative Adversarial Networks* (*GANS*). *GANS* sudah banyak dimanfaatkan untuk proses pengenalan wajah. Algoritma ini memungkinkan gambar wajah untuk dikenali meskipun terdapat kekurangan dalam kualitas dataset. Algoritma ini juga memungkinkan adanya wajah baru yang dihasilkan dari data set yang ada user tidak perlu

menginputkan terlalu banyak data wajah untuk *training*. Teknologi pengenalan wajah memanfaatkan ponsel cerdas dalam proses pengambilan gambar wajah. Selain menggunakan ponsel cerdas, sistem informasi pengelolaan keuangan sekolah ini juga dibangun pada platform website. Tujuannya agar input data dapat dilakukan dengan mudah. Maka dari itu, tugas akhir ini akan mengimplementasikan teknologi pengenalan wajah dengan menggunakan *GANS* pada Sistem Informasi Pengelolaan Keuangan Sekolah yang akan dibangun dalam aplikasi *mobile* dan aplikasi web.

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam Tugas Akhir ini dapat dipaparkan sebagai berikut

1. Bagaimanakah proses pengajuan anggaran dana melalui aplikasi SIPKS?
2. Bagaimana cara melakukan persetujuan pengajuan anggaran dana melalui aplikasi SIPKS?
3. Bagaimana cara memperbaiki foto wajah yang blur dan menghasilkan foto baru menggunakan *Generative Adversarial Networks*?
4. Bagaimana cara melakukan pengenalan wajah pada aplikasi SIPKS?

1.3 Batasan Masalah

Permasalahan yang dibahas dalam Tugas Akhir memiliki beberapa batasan antara lain:

1. Teknologi yang dalam pengembangan aplikasi ini adalah bahasa pemrograman Android (Java), bahasa pemrograman web (PHP, HTML, CSS, JavaScript) , kerangka kerja Laravel, dan basis data MySQL.

2. Sample wajah yang di*generate* menggunakan GAN adalah perbaikan dari sample wajah blur dari dataset.

3. Pengenalan wajah dengan MobileNET menggunakan data hasil *Generative Adversarial Networks.*
4. Aplikasi website hanya dapat diakses oleh Pegawai Tata Usaha sekolah untuk menambahkan pengajuan, sedangkan aplikasi *mobile* hanya dapat diakses Kepala Sekolah dan Bendahara Sekolah untuk melakukan persetujuan.
5. Sistem Informasi Pengelolaan Keuangan Sekolah (SIPKS) hanya terbatas pada Sekolah Negeri di Surabaya.

1.4 Tujuan

Tugas Akhir ini mempunyai tujuan yaitu membuat aplikasi berbasis android dan web yang dapat digunakan untuk melakukan persetujuan pengajuan dana anggaran dengan *face recognition*.

1.5 Manfaat

Tugas akhir ini diharapkan dapat memberikan manfaat sebagai berikut :

1. Bagi sekolah, tugas akhir ini diharapkan dapat mempermudah proses pencairan dana anggaran sekolah dan menjaga sekuritas pencairan dana anggaran sekolah.
2. Bagi pemerintah, tugas akhir ini diharapkan dapat memberikan alternatif cara untuk pengembangan cara persetujuan dengan digital, terutama *face recognition.*

1.6 Metodologi

Tahap yang dilakukan untuk menyelesaikan Tugas Akhir ini adalah sebagai berikut:

1. Penyusunan Proposal Tugas Akhir

Penyusunan proposal tugas akhir merupakan tahap pertama pengerjaan tugas akhir. Tugas akhir yang diajukan adalah Sistem Informasi Pengelolaan Keuangan Sekolah (SIPKS) dengan *Generative Adversarial Networks* Untuk *Face Recognition*. Sistem Informasi ini akan diimplementasikan ke

dalam dua platform, yaitu website dan android.

2. **Studi Literatur**

   Studi literatur dilakukan untuk memperoleh literatur yang relevan sebagai referensi pengerjaan tugas akhir. Studi literatur yang dibutuhkan antara lain yaitu pemrograman website dengan framework Laravel, implementasi *Generative Adversarial Networks*, pemrograman Android dengan bahasa pemrograman Java, serta materi pendukung lainnya.

3. **Analisis dan Desain Perangkat Lunak**

   Pada tahap ini dilakukan analisis kebutuhan dan perancangan sistem yang dapat menghasilkan solusi untuk permasalahan yang diajukan. Solusi tersebut kemudian akan dijadikan sebagai acuan dalam desain perangkat lunak. Desain perangkat lunak meliputi deskripsi sistem yang akan dibuat, antara lain perancangan arsitektur perangkat lunak, desain antarmuka, desain basis data, dan diagram-diagram pendukung lainnya.

4. **Implementasi**

   Implementasi merupakan tahap membangun rancangan program yang telah dibuat. Pada tahapan ini merealisasikan rancangan yang terdapat pada tahapan sebelumnya, sehingga menjadi sebuah program yang sesuai dengan apa yang telah direncanakan.

5. **Uji Coba dan Evaluasi**

   Pada tahapan ini dilakukan uji coba pada perangkat lunak dengan pengujian berdasarkan fungsionalitas dan kasus penggunaan sistem.

6. **Penyusunan Buku Tugas Akhir**

   Pada tahapan ini disusun buku yang membuat dokumentasi

mengenai pembuatan serta hasil dari implementasi perangkat lunak yang telah dibuat.

1.7 Sistematika Penulisan

Buku Tugas Akhir ini terdiri atas beberapa bab yang tersusun secara sistematis, yaitu sebagai berikut.

1. **Bab I. Pendahuluan**
   Bab pendahuluan berisi penjelasan mengenai latar belakang masalah, rumusan masalah, batasan masalah, tujuan, manfaat dan sistematika penulisan Tugas Akhir.
2. **Bab II. Tinjauan Pustaka**
   Bab tinjauan pustaka berisi penjelasan mengenai dasar teori yang mendukung pengerjaan Tugas Akhir.
3. **Bab III. Analisis dan Perancangan**
   Bab ini berisi tentang desain sistem, rancangan basis data, diagram kasus penggunaan, diagram aktivitas dan rancangan antarmuka pengguna.
4. **Bab IV. Implementasi**
   Bab ini membahas implementasi dari desain yang telah dibuat pada bab sebelumnya. Penjelasan berupa tampilan antarmuka yang telah dibuat dan dapat berfungsi untuk mengakomodir kebutuhan fungsional yang ada.
5. **Bab V. Uji Coba dan Evaluasi**
   Bab ini menjelaskan kemampuan perangkat lunak dengan melakukan pengujian kebenaran dan pengujian kinerja dari sistem yang telah dibuat.
6. **Bab VI. Kesimpulan dan Saran**
   Bab ini merupakan bab terakhir yang menyampaikan kesimpulan dari hasil uji coba yang dilakukan dan saran untuk pengembangan perangkat lunak ke depannya.

BAB II

TINJAUAN PUSTAKA

Bab tinjauan pustaka berisi mengenai penjelasan teori yang berkaitan dengan implementasi perangkat lunak. Penjelasan tersebut bertujuan untuk memberikan gambaran mengenai sistem yang akan dibangun dan berguna sebagai pendukung dalam pengembangan perangkat lunak.

2.1 Pengenalan Wajah

Teknologi pengenalan wajah atau yang biasa disebut *face recognition* merupakan teknologi yang dapat mengidentifikasi seseorang dari data gambar digital maupun video. Teknologi ini bekerja dengan cara membandingkan beberapa fitur wajah dengan dataset yang sudah diambil dan dilakukan *training* [2]. Pengenalan wajah biasanya dilakukan untuk Tujuan keamanan, keselamatan, dan kepemilikan. Adapun cara kerja pengenalan wajah adalah sebagai berikut :

1. Gambar wajah diambil melalui kamera, foto, atau video.
2. Software atau algoritma pengenalan wajah membaca geometri dari gambar wajah yang diambil. Salah satu faktor kunci dari wajah yaitu jarak antara kedua mata serta jarak dari kening ke dagu. Software mengidentifikasi ciri khas wajah (landmark). Hasil dari software ini adalah fitur dari gambar wajah.
3. Fitur wajah yang didapat kemudian dicocokan dengan algoritma pengenalan wajah dan basis data yang dimiliki [3].

Face recognition memiliki keterbatasan, salah satunya yaitu halangan. Halangan yang dimaksud merupakan benda yang ada di gambar wajah selain wajah itu sendiri, contohnya aksesoris wajah. Selain itu pengenalan wajah juga dibatasi oleh pose yang dapat membuat fitur yang dikenali memiliki nilai yang berbeda, sehingga sulit dikenali. Umumnya pengenalan wajah akan bekerja maksimal dalam pose wajah yang menghadap lurus ke depan. Pencahayaan, basis data yang dimiliki, serta *data processing* menjadi

keterbatasan *face recognition* lainnya [4].

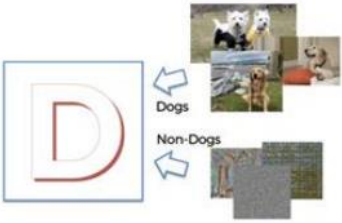
2.2 Generative Adversarial Network (GANs)

Generative Adversarial Networks(GANs) adalah deep learning berdasarkan generative model. GANs adalah arsitektur model untuk mentraining generative model. GANs pertama kali muncul tahun 2014 dalam paper Ian Goodfellow berjudul "*Generative Adversarial Networks*". Awalnya model yang dihasilkan sulit untuk ditraining dan tidak stabil. Tahun 2015 Alec Radford mempublikasikan pendekatan *Deep Convolutional Generative Adversarial Networks* (DCGAN) yang dinilai lebih stabil. Algoritma GANs memiliki dua sub-model yaitu *generator* untuk menghasilkan contoh baru dan *discriminator* untuk mengklasifikasikan apakah contoh yang digenerate *generator* nyata (dari domain) atau palsu (hasil generator)[5]. Generator model mengambil input sebagai random *noise* signal kemudian menghasilkan sebuah gambar. Gambar 2.1 merupakan ilustrasi dari generator model.

Gambar 2.1. Generator model dalam GANS

Sementara itu, *discriminator* model merupakan lawan dari generator. *Discriminator* dapat melakukan learning terhadap sebuah objek. Contohnya data gambar anjing. *Discriminator* dapat menemukan perbedaan mana gambar anjing yang asli, mana gambar yang bukan anjing. Ilustrasi *discriminator* model terdapat pada Gambar 2.2.

Dengan contoh ini, ketika *discriminator* ditraining, gambar yang bukan anjing akan mengembalikan nilai 0, sementara yang anjing akan mengembalikan nilai 1. Ilustrasi dapat dilihat pada Gambar 2.3.



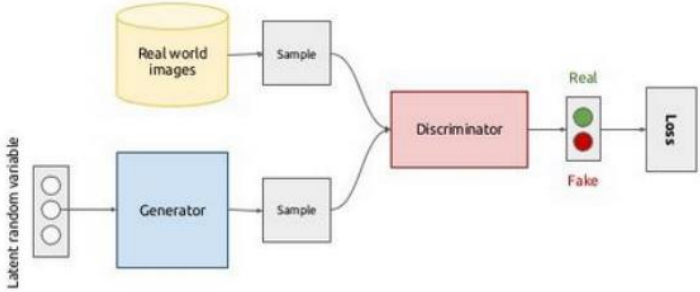
Gambar 2.2. Discriminator model dalan GANS

Gambar 2.3. *Output training discriminator*

Proses algoritma GAN:

1. Input random noise signal ke dalam generator. Generator akan membuat beberapa gambar yang akan digunakan untuk mentraining discriminator. Pada discriminator disediakan beberapa fitur atau gambar. Discriminator akan memberikan kemungkinan outputnya. Kemungkinan ini bisa jadi tinggi karena training yang dilakukan discriminator masih awal. Nilai –nilai yang dihasilkan kemudian diidentifikasi. Kemudian error akan dihitung. Kemudian dikembalikan lagi ke discriminator untuk melakukan update bobot.
2. Selanjutnya, generator ditraining. Gambar yang dihasilkan generator tadi dimasukkan ke dalam discriminator. Kali ini tidak disertai dengan gambar fitur. Generator akan melakukan learning dengan mengakali discriminator untuk menghasilkan false positives. Discriminator akan memberikan kemungkinan output. Nilai –nilai yang dihasilkan kemudian diidentifikasi. Kemudian error akan dihitung. Kemudian dikembalikan lagi ke discriminator untuk melakukan update bobot.
3. Langkah ketiga sama dengan kedua langkah sebelumnya, namun generator dan discriminator lebih dilatih (training) lagi.

Melalui backpropagation, generator memahami kesalahannya dan mulai membuatnya lebih seperti fitur asli.



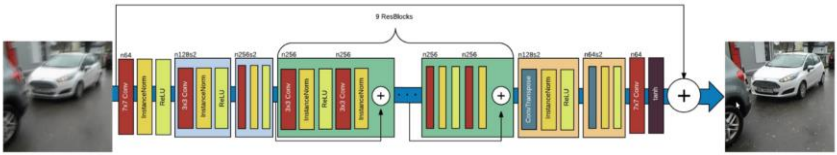
Gambar 2.4. Proses GANS

Gambar 2.4 merupakan skema proses GANs. GANs dapat digunakan dalam generating image, modifikasi gambar, resolusi super, assisting artists, foto yang terlihat realistis, speech generation, face ageing . Contoh penggunaan algoritma GANs dapat dilihat pada Gambar 6. [6].

Gambar 2.5. Contoh penggunaan algoritma GAN

2.3 DeblurGAN

DeblurGAN adalah salah satu pengembangan dari generative adversarial networks yang digunakan untuk image deblurring. Input generator pada DeblurGAN adalah foto blur. Generator bertujuan menghasilkan sharp images. Network yang digunakan adalah ResNet Blocks yang menelusuri evolusi dari gambar blur original. Discriminator bertujuan untuk menentukan apakah gambar input merupakan gambar nyata atau palsu. Oleh karena itu, discriminator bersifat konvolusional dan menghasilkan output tunggal[7].



Gambar 2.6. Arsitektur Generator DeblurGAN

2.4 PHP

Hypertext Preprocessor (PHP) adalah bahasa skrip server-side yang dapat disisipkan ke dalam HTML dan dirancang khusus untuk web. PHP juga bisa menyelesaikan tugas-tugas pemrograman dalam bentuk antarmuka baris perintah (CLI). Kode PHP ditafsirkan di server web dan menghasilkan HTML[8].

2.5 Laravel

Laravel merupakan salah satu *framework* dengan konsep MVC (Model, Controller, View) dalam bahasa pemrograman PHP. Laravel menyediakan sintaks yang ekspresif, jelas, dan menghemat waktu[9].

2.6 Python

Python adalah salah satu bahasa pemograman tingkat tinggi yang popular. Python bersifat interpreter, interactive, dan *object oriented.* Python sering dimanfaatkan dalam pengembangan web, perangkat lunak, penelitian, dan system scripting. Python dapat

digunakan untuk menangani data besar dan melakukan operasi matematika yang kompleks. Python dapat beroperasi di berbagai platform seperti Mac, Linux, dan Windows. Python termasuk bahasa pemograman yang mudah dipelajari karena sintaks yang jelas, dapat dikombinasikan dengan penggunaan modu lmodul siap pakai, dan struktur data tingkat tinggi yang efisien[10].

2.7 MobileNet

MobileNet, merupakan salah satu arsitektur *convolutional neural network* (CNN) yang dapat digunakan untuk mengatasi kebutuhan akan computing resource berlebih. MobileNet digunakan untuk ponsel. Perbedaan antara arsitektur MobileNet dan arsitektur CNN pada umumnya adalah penggunaan lapisan atau layer konvolusi dengan ketebalan filter yang sesuai dengan ketebalan dari input image. MobileNet membagi konvolusi menjadi *depthwise convolution* dan *pointwise convolution*[11].

2.8 Android

Android adalah sistem operasi yang dirancang untuk perangkat bergerak dengan layar sentuh seperti smartphone atau tablet dengan didukung kernel yang berbasis Linux. Android merupakan platform tak terbatas dimana aplikasi berbasis Android sangat banyak diakses secara gratis[12].

2.9 Java

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada C dan C++ namun dengan sintaksis model objek yang lebih sederhana serta dukungan rutin-rutin aras bawah yang minimal. Karena fungsionalitasnya yang memungkinkan, aplikasi java mampu berjalan di beberapa platform sistem operasi yang berbeda. Saat ini java merupakan bahasa pemrograman yang paling populer digunakan, dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi [13].

BAB III

ANALISIS DAN PERANCANGAN

Pada Bab ini akan dijelaskan mengenai analisis kebutuhan dan perancangan perangkat lunak untuk mencapai tujuan dari Tugas Akhir.

3.1 Analisis

Tahap analisis dibagi menjadi beberapa bagian, antara lain analisis permasalahan, deskripsi umum sistem, dan spesifikasi kebutuhan perangkat lunak.

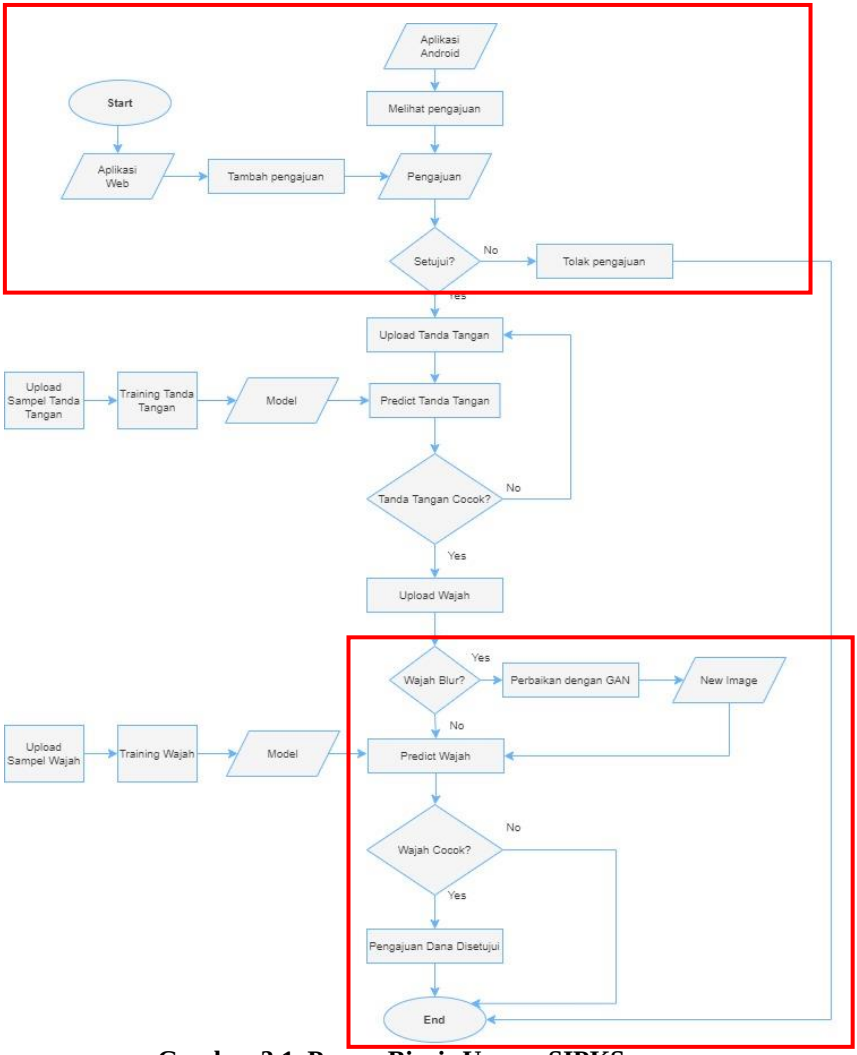
3.1.1 Analisis Permasalahan

Permasalahan utama yang diangkat dalam Tugas Akhir ini adalah bagaimana membangun aplikasi Sistem Informasi Pengelolaan Keuangan Sekolah (SIPKS) berbasis Android dan website dengan menggunakan *face recognition* pada validasi persetujuan pencairan dana anggaran sekolah.

Pengajuan dana anggaran sekolah masih dilakukan secara manual di kebanyakan sekolah negeri di Kota Surabaya. Terdapat resiko *human error* yang lumayan tinggi dan waktu yang cukup lama pada proses manual. Selain itu, akuntabilitas pemberian persetujuan terhadap pengajuan kurang terjamin keasliannya, bisa saja terjadi pemalsuan dalam pemberian persetujuan. Hal ini menjadi krusial karena berkaitan dengan dana anggaran sekolah.

Untuk mengatasi permasalahan tersebut, penulis akan mengimplementasikan pembuatan aplikasi SIPKS (Sistem Informasi Pengelolaan Keuangan Sekolah) untuk mempermudah proses pencairan dana anggaran sekolah melalui website dan aplikasi android dengan memanfaatkan *face recognition.* Adapun flowchart proses bisnis umum Sistem Informasi Pengelolaan Keuangan Sekolah (SIPKS) dapat dilihat pada Gambar 3.1.

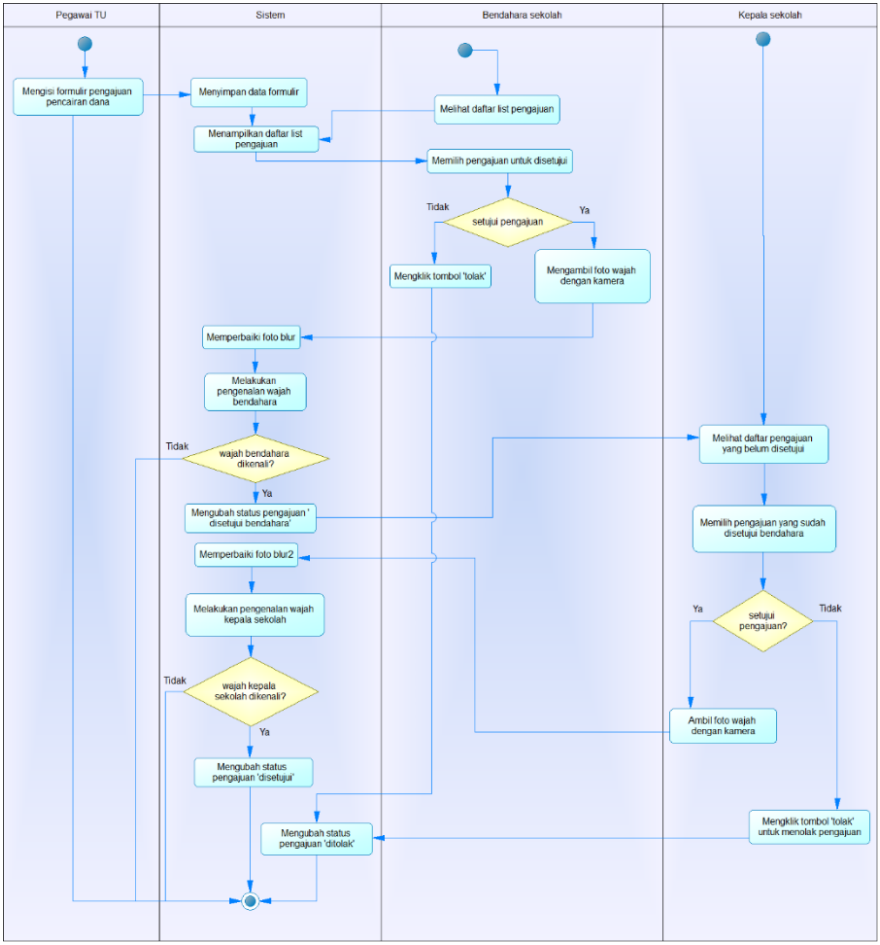
Dari proses bisnis besar pada Gambar 3.1 terdapat dua alur pengenalan wajah sesuai dengan input foto wajah, yaitu alur normal dan alur saat input foto wajah blur.



Gambar 3.1. Proses Bisnis Umum SIPKS

Alur normal yaitu alur yang berjalan saat foto wajah memiliki kualitas yang baik karena fitur wajah terlihat jelas. Alur normal pada sistem besar ini dikerjakan penulis lain sehingga tidak dibahas pada tugas akhir ini. Tugas akhir ini berfokus pada alur

saat input foto wajah blur. Tugas akhir ini menitikberatkan pada pengenalan wajah dengan input foto wajah yang blur sehingga perlu dilakukan perbaikan foto sebelum dilakukan pengenalan wajah dalam proses persetujuan pengajuan anggaran dana. Sedangkan proses detail pengenalan wajah dengan perbaikan foto blur disajikan dalam proses bisnis yang dapat dilihat pada Gambar 3.1.1.



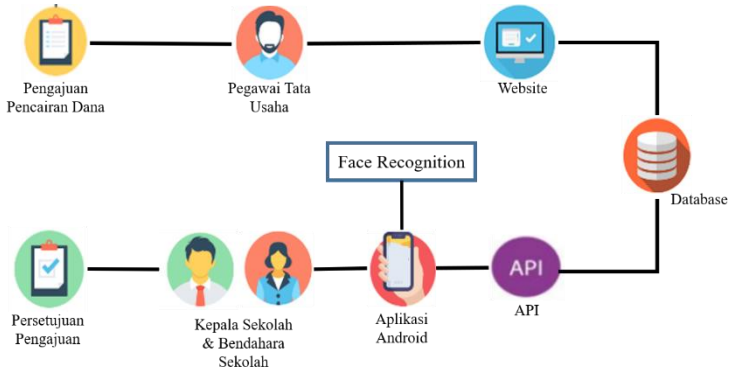
Gambar 3.1.1 Proses Bisnis dengan Pengenalan Wajah

Pada gambar 3.1.1 dapat dilihat alur proses bisnis yang terjadi di SIPKS. Proses bisnis ini melibatkan 3 aktor dan 1 sistem. Aktor yang terlibat yaitu Pegawai Tata Usaha, Kepala Sekolah, dan Bendahara Sekolah. Proses bisnin ini menggambarkan bagaimana alur persetujuan pengajuan diberikan.

3.1.2 Deskripsi Umum Sistem

Pada Tugas Akhir ini, system yang akan dibuat adalah perangkat lunak berupa aplikasi berbasis web untuk melakukan pengajuan pencairan anggaran dana sekolah dan android untuk melakukan persetujuan pengajuan anggaran dana sekolah, keduanya merupakan satu kesatuan aplikasi SIPKS atau Sistem Informasi Pengelolaan Keuangan Sekolah. Dengan adanya aplikasi SIPKS berbasis web, Karyawan Tata Usaha Sekolah dapat mengajukan pencairan dana anggaran untuk kegiatan sekolah. Pengajuan diajukan untuk disetujui oleh bendahara sekolah dan kepala sekolah. Dengan adanya aplikasi SIPKS berbasis android, bendahara sekolah dan kepala sekolah dapat memberikan persetujuan pengajuan dengan validasi menggunakan *face recognition*. Karyawan Tata Usaha Sekolah dapat menambahkan, mengedit, menghapus, dan melihat status progress pengajuan pada aplikasi web. Sedangkan Bendahara Sekolah dan Kepala Sekolah dapat melihat daftar pengajuan, detail setiap pengajuan, memberikan persetujuan atau penolakan pengajuan melalui aplikasi android. Validasi pemberian persetujuan dilakukan dengan menggunakan *face recognition* dengan sample wajah yang ditraining dari foto wajah yang diupload melalui aplikasi android.

Pada Gambar 3.1 menjelaskan bahwa pengajuan diajukan oleh Pegawai Tata Usaha melalui aplikasi website yang kemudian disimpan ke database. Sementara itu, persetujuan terhadap pengajuan dilakukan oleh bendahara dan kepala sekolah melalui aplikasi android. Aplikasi android menggunakan API untuk mengakses dan mengelola data di database. Jika hasil *face recognition* menyatakan wajah cocok dengan model yang ada, maka persetujuan berhasil disetujui. Begitu juga sebaliknya, jika hasil tidak cocok maka persetujuan gagal disetujui.



Gambar 3.2. Deskripsi Umum Sistem

3.1.3 Spesifikasi Perangkat Lunak

Subbab ini membahas spesifikasi kebutuhan perangkat lunak dari hasil analisis yang telah dilakukan. Bagian ini berisi kebutuhan perangkat lunak yang direpresentasikan dalam bentuk kebutuhan fungsional, kebutuhan non-fungsional, diagram kasus penggunaan, dan diagram aktivitas.

3.1.3.1 Aktor

Aktor adalah pihak-pihak yang terlibat dan berinteraksi langsung dengan sistem. Aktor yang berada pada sistem ini dibahas lebih rinci pada Tabel 3.1.

Tabel 3.1 Spesifikasi Karakter Aktor pada Sistem

No	Aktor	Tugas	Hak Akses
1	Kepala Sekolah	Melakukan persetujuan atau penolakan pengajuan melalui aplikasi android	• Melihat daftar pengajuan • Melihat detail pengajuan • Menyetujui/menolak pengajuan • Mengupload sample wajah

			• Melakukan training sample wajah
2	Bendahara Sekolah	Melakukan persetujuan atau penolakan pengajuan melalui aplikasi android	• Melihat daftar pengajuan • Melihat detail pengajuan • Menyetujui/menolak pengajuan • Mengupload sample wajah • Melakukan training sample wajah
3	Pegawai Tata Usaha	Mengelola pengajuan dana anggaran melalui aplikasi website	• Melihat daftar pengajuan dan detail pengaajuan • Melakukan edit data pengajuan dan detail pengajuan • Menghapus pengajuan dan detail pengajuan

3.1.3.2 Kebutuhan Fungsional

Kebutuhan fungsional mendefinisikan layanan yang harus dimiliki oleh perangkat lunak, reaksi terhadap suatu masukan, hasil yang dilakukan pada situasi khusus. Kebutuhan fungsional dari perangkat lunak untuk aplikasi ini dijelaskan pada Tabel 3.2.

Tabel 3.2. Kebutuhan Fungsional Sistem

Kode	Kebutuhan Fungsional	Deskripsi
FW01	Menangani kelola data pengajuan	Melakukan penambahan, perubahan, dan penghapusan

	dana anggara	data pengajuan serta menampilkan data pengajuan dana anggaran
FA01	Menangani kelola data sampel wajah	Melakukan penambahan data, proses training dan testing data, serta proses prediksi wajah
FA02	Melakukan persetujuan pengajuan anggaran dana	Melakukan verifikasi dan validasi wajah untuk menyetujui pengajuan dan aanggaran

3.1.3.3 Kebutuhan Non Fungsional

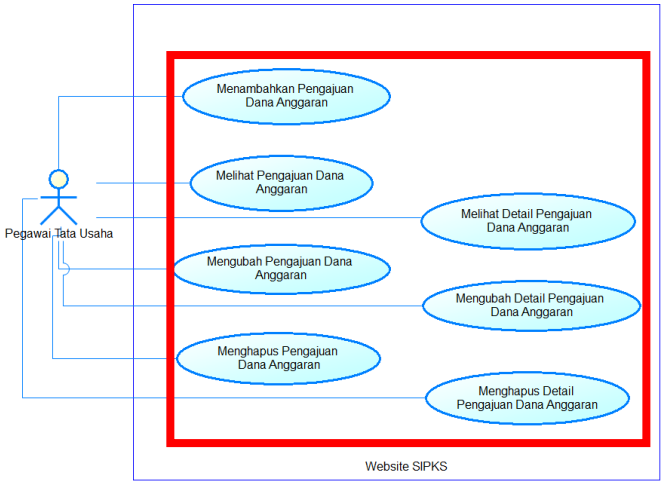
Kebutuhan non-fungsional yang harus dipenuhi sistem sebagai berikut:

1. Kebutuhan Performa
   Perangkat lunak akan berjalan dengan performa terbaik jika dijalankan di atas spesifikasi minimal.
2. Kebutuhan Operasional
   Sistem hanya dapat berjalan pada platform web untuk menginput pengajuan anggaran dana dan pada perangkat bergerak dengan sistem operasi Android untuk memberi konfirmasi persetujuan anggaran.

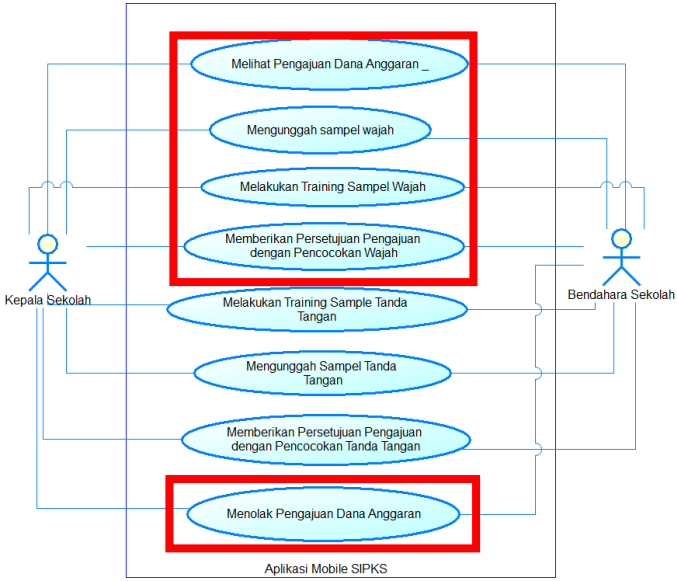
3.1.3.4 Diagram Kasus Penggunaan

Pada subbab ini dijelaskan secara rinci mengenai kasus penggunaan aplikasi SIPKS yang merupakan hasil analisis kebutuhan fungsionalitas perangkat lunak. Diagram kasus penggunaan aplikasi webite SIPKS dapat dilihat pada Gambar 3.3 dan diagram penggunaan aplikasi Android SIPK dapat dilihat pada Gambar 3.4.

Dari Gambar 3.3 dan Gambar 3.4 dapat diketahui bahwa kasus penggunaan yang akan dijelaskan pada tugas akhir ini adalah kasus diagram yang diberi penanda kotak merah.



Gambar 3.3. Diagram Kasus Penggunaan Website SIPKS



Gambar 3.4. Diagram Kasus Penggunaan Aplikasi Android SIPKS

3.1.3.4.1 Kasus Penggunaan Menambahkan Pengajuan Dana Anggaran

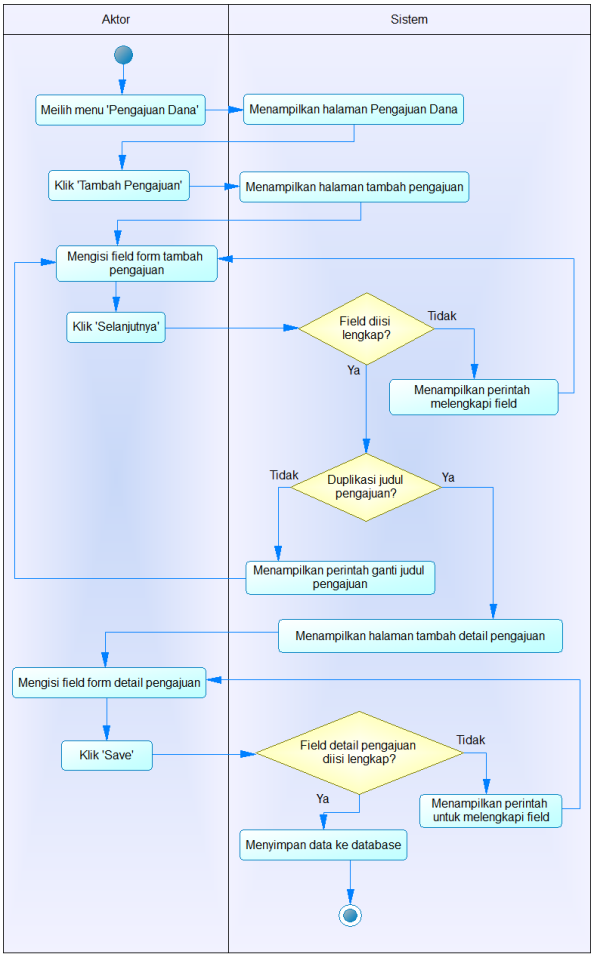
Kasus penggunaan ini diakses oleh Pegawai TU melalui website SIPKS. Pada kasus penggunaan ini, Pegawai TU dapat menambahkan pengajuan pencairan dana anggaran beserta detail penggunaan dana anggaran. Penjelasan skenario yang lebih rinci dapat dilihat pada Tabel 3.4.

Table 3.3 Spesifikasi Kasus Penggunaan Menambahkan Pengajuan Dana Anggaran

Nama Use Case	**Menambahkan Pengajuan Dana Anggaran**
Nomor	**WU-001**
Deskripsi	Aktor menambahkan pengajuan dana anggaran melalui website SIPKS
Aktor	Pegawai Tata Usaha
Kondisi Awal	Aktor Berhasil login ke website SIPKS
Kondisi Akhir	Aktor berhasil menambahkan Pengajuan Dana Anggaran beserta detail pengajuan ke database
Alur Normal	
Aktor	**Sistem**
1. Memilih menu Pengajuan Dana	1.1 Menampilkan halaman Pengajuan Dana
2. Klik tombol Tambah Pengajuan	2.1 Menampilkan halaman Tambah Pengajuan
3. Mengisi field pengajuan	
4. Klik tombol Selanjutnya	4.1 Menampilkan halaman tambah detail pengajuan
5. Mengisi field detail pengajuan	
6. Klik ‘Save’	6.1 Menyimpan data ke database
Alur Alternatif	
A1. Jika Aktor tidak mengisi satu atau beberapa field form	

pengajuan	
Aktor	Sistem
	A1.1 Sistem menampilkan perintah untuk melengkapi form A 1.2 Kembali ke alur 3
A2. Jika terjadi duplikasi judul pengajuan pada input dan data di database	
Aktor	Sistem
	A2.1 Sistem menampilkan peringatan duplikasi dan perintah mengganti judul pengajuan A2.2 Kembali ke alur 3
A3. Jika Aktor tidak mengisi form detail pengajuan dengan lengkap	
Aktor	Sistem
	A3.1 Sistem menampilkan perintah untuk melengkapi form A 1.2 Kembali ke alur 5

Skenario aktivitas selanjutnya digambarkan ke dalam diagram aktivitas. Gambar 3.5 menggambarkan diagram aktivitas dari skenario menambahkan pengajuan dana anggaran.



Gambar 3.5 Diagram Aktivitas Menambahkan Pengajuan Dana Anggaran

3.1.3.4.2 Kasus Penggunaan Melihat Pengajuan Dana Anggaran (Website)

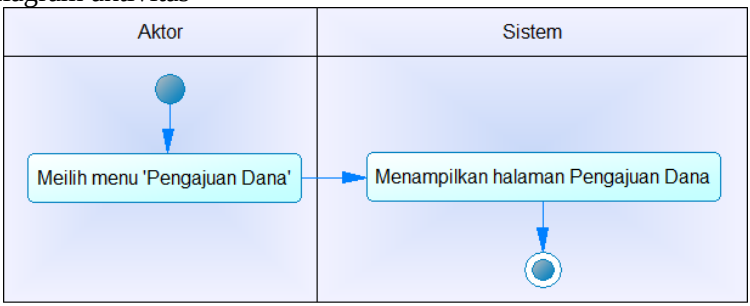
Kasus penggunaan ini diakses oleh Pegawai TU melalui website

SIPKS. Pada kasus penggunaan ini, Pegawai TU dapat melihat pengajuan pencairan dana anggaran dalam bentuk table. Table menampilkan judul pengajuan, waktu pembuatan pengajuan, nama sekolah, petugas yang menambahkan pengajuan, civitas yang meminta dana, nominal pengajuan, serta status pengajuan. Penjelasan skenario yang lebih rinci dapat dilihat pada Tabel 3.4.

Table 3.4 Spesifikasi Kasus Penggunaan Melihat Pengajuan Dana Anggaran

Nama Use Case	**Melihat Pengajuan Dana Anggaran**
Nomor	**WU-002**
Deskripsi	Aktor dapat melihat daftar pengajuan dana anggaran melalui website SIPKS
Aktor	Pegawai Tata Usaha
Kondisi Awal	Aktor Berhasil login ke website SIPKS
Kondisi Akhir	Aktor berhasil melihat daftar pengajuan dana melalui website SIPKS
Alur Normal	
Aktor	**Sistem**
1. Memilih menu Pengajuan Dana	1.1 Menampilkan halaman Pengajuan Dana

Skenario aktivitas selanjutnya digambarkan ke dalam diagram aktivitas



Gambar 3.6 Diagram Aktivitas Melihat Pengajuan Dana Anggaran

Gambar 3.6 merupakan diagram aktivitas dari skenario melihat pengajuan dana anggaran.

3.1.3.4.3 Kasus Penggunaan Mengubah Pengajuan Dana Anggaran

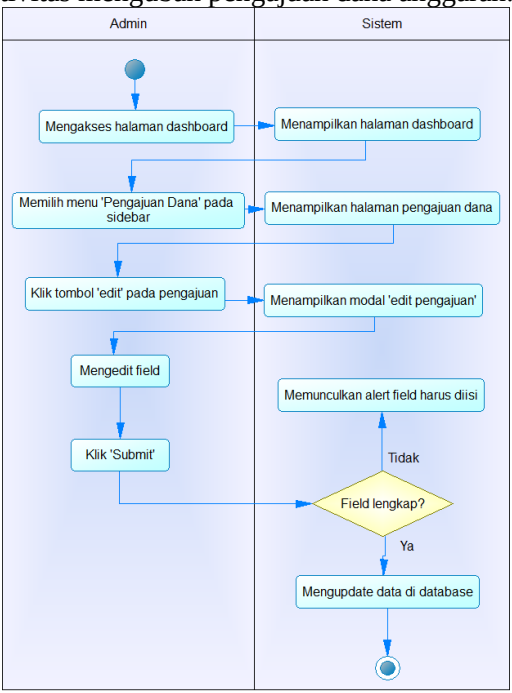
Kasus penggunaan ini diakses oleh Pegawai TU melalui website SIPKS. Pada kasus penggunaan ini, Pegawai TU dapat melakukan perubahan judul dan deskripsi pengajuan dana anggaran. Penjelasan skenario yang lebih rinci dapat dilihat pada Tabel 3.5.

Table 3.5. Spesifikasi Kasus Penggunaan Mengubah Pengajuan Dana Anggaran

Nama Use Case	**Mengubah Pengajuan Dana Anggaran**
Nomor	**WU-003**
Deskripsi	Aktor dapat mengubah judul dan deskripsi pengajuan dana anggaran
Aktor	Pegawai Tata Usaha
Kondisi Awal	Aktor Berhasil login ke website SIPKS
Kondisi Akhir	Aktor berhasil mengubah judul dan/atau deskripsi pengajuan dana anggaran
Alur Normal	
Aktor	**Sistem**
1. Mengakses halaman dashboard	1.1 Menampilkan halaman dashboard
2. Memilih menu Pengajuan Dana pada sidebar	2.1 Menampilkan halaman pengajuan dana
3. Klik tombol edit pada pengajuan yang akan diubah	3.1 Menampilkan modal edit pengajuan
4. Mengubah field judul dan/atau deskripsi pengajuan	
5. Klik Submit	5.1 Mengupdate data pada database

Alur Alternatif	
A1. Jika Aktor tidak mengisi satu atau beberapa field form pada modal edit	
Aktor	Sistem
	A1.1 Sistem menampilkan perintah untuk melengkapi form A 1.2 Kembali ke alur 4

Skenario aktivitas selanjutnya digambarkan ke dalam diagram aktivitas Gambar 3.7. di bawah ini menggambarkan diagram aktivitas mengubah pengajuan dana anggaran.



Gambar 3.7. Diagram Aktivitas Mengubah Pengajuan Dana Anggaran

3.1.3.4.4 Kasus Penggunaan Menghapus Pengajuan Dana Anggaran

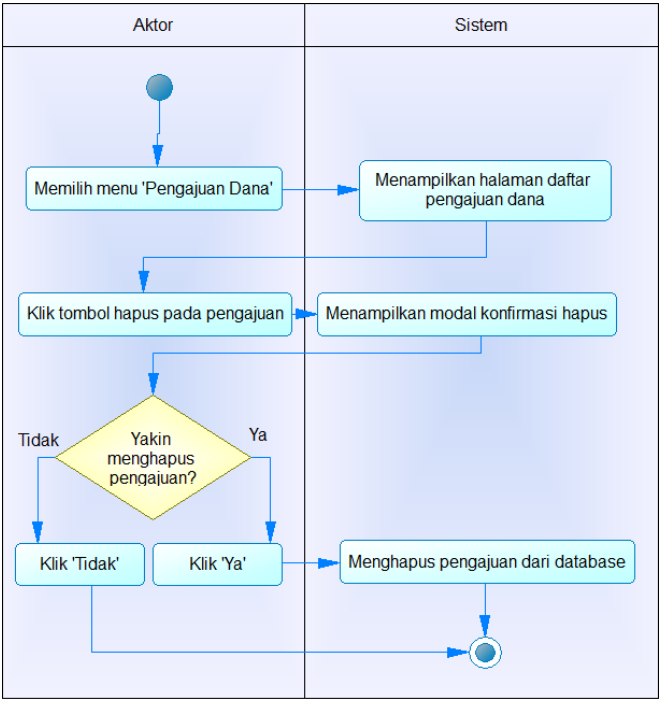
Kasus penggunaan ini diakses oleh Pegawai TU melalui website SIPKS. Pada kasus penggunaan ini, Pegawai TU dapat menghapus pengajuan dana anggaran. Pengajuan yang telah dihapus akan hilang data tampilan. Penjelasan skenario yang lebih rinci dapat dilihat pada Tabel 3.6.

Table 3.6. Spesifikasi Kasus Penggunaan Menghapus Pengajuan Dana Anggaran

Nama Use Case	**Menghapus Pengajuan Dana Anggaran**
Nomor	**WU-004**
Deskripsi	Aktor dapat menghapus pengajuan dana anggaran
Aktor	Pegawai Tata Usaha
Kondisi Awal	Aktor Berhasil login dan mengakses dashboard website SIPKS
Kondisi Akhir	Aktor berhasil menghapus pengajuan dana dari tampilan dan database
Alur Normal	
Aktor	**Sistem**
1. Memilih menu Pengajuan Dana pada sidebar	1.1 Menampilkan halaman pengajuan dana
2. Klik tombol hapus pada pengajuan yang ingin dihapus	2.1 Menampilkan modal konfirmasi hapus pengajuan
3. Klik ‘Ya’	3.1 Menghapus pengajuan dari database
Alur Alternatif	
A1. Jika Aktor memilih ‘Tidak’ pada modal konfirmasi hapus pengajuan	
Aktor	Sistem
	A1.1 Kembali ke alur 1.1

Skenario aktivitas selanjutnya digambarkan ke dalam

diagram aktivitas Gambar 3.8. di bawah ini menggambarkan diagram aktivitas menghapus pengajuan dana anggaran.



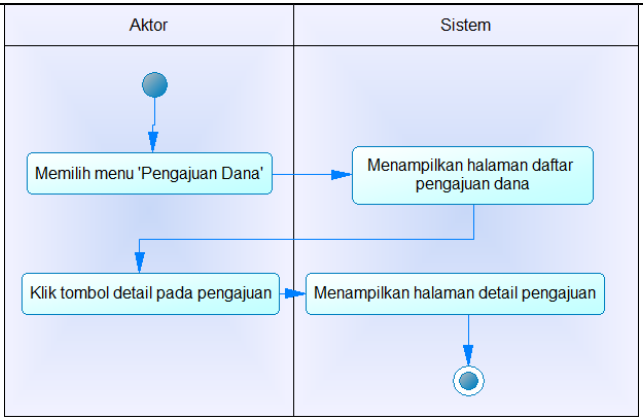
Gambar 3.8 Diagram Aktivitas Menghapus Pengajuan Dana Anggaran

3.1.3.4.5 Kasus Penggunaan Melihat Detail Pengajuan Dana Anggaran

Kasus penggunaan ini diakses oleh Pegawai TU melalui website SIPKS. Pegawai TU dapat melihat detail penggunaan dana anggaran, meliputi nama detail, jumlah, harga satuan, dan sub total dari detail pengajuan. Penjelasan skenario yang lebih rinci dapat dilihat pada Tabel 3.7. Skenario aktivitas selanjutnya digambarkan ke dalam diagram aktivitas pada Gambar 3.9.

Table 3.7 Spesifikasi Kasus Penggunaan Melihat Detail Pengajuan Dana Anggaran

Nama Use Case	**Melihat Detail Pengajuan Dana Anggaran**
Nomor	**WU-005**
Deskripsi	Aktor dapat melihat informasi detail penggunaan dana anggaran
Aktor	Pegawai Tata Usaha
Kondisi Awal	Aktor Berhasil login dan mengakses dashboard website SIPKS
Kondisi Akhir	Aktor berhasil melihat detail penggunaan dana pada pengajuan yang telah dibuat
Alur Normal	
Aktor	**Sistem**
1. Memilih menu Pengajuan Dana 2. Klik tombol detail pada pengajuan	1.2 Menampilkan halaman pengajuan dana 2.1 Menampilkan halaman detail pengajuan
Alur Alternatif	
-	



Gambar 3.9 Diagram Aktivitas Melihat Detail Pengajuan Dana Anggaran

3.1.3.4.6 Kasus Penggunaan Mengubah Detail Pengajuan Dana Anggaran

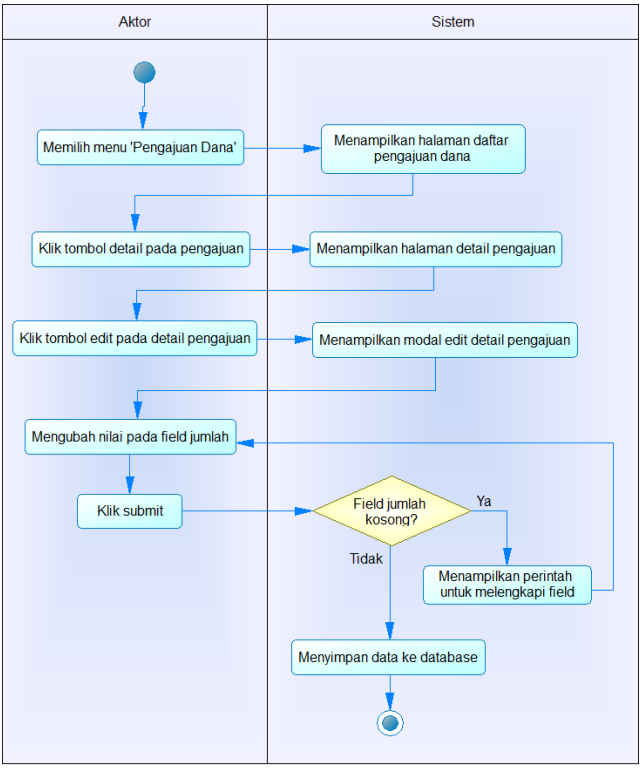
Kasus penggunaan ini diakses oleh Pegawai TU melalui website SIPKS. Pada kasus penggunaan ini, Pegawai TU dapat mengubah detail penggunaan dana anggaran pada pengajuan yang dipilih. Data yang dapat diubah dari masing-masing detail pengajuan adalah jumlahnya. Penjelasan skenario yang lebih rinci dapat dilihat pada Tabel 3.8.

Table 3.8. Spesifikasi Kasus Penggunaan Mengubah Detail Pengajuan Dana Anggaran

Nama Use Case	**Mengubah Detail Pengajuan Dana Anggaran**
Nomor	**WU-006**
Deskripsi	Aktor dapat melakukan perubahan jumlah detail pengajuan pada pengajuan yang dipilih
Aktor	Pegawai Tata Usaha
Kondisi Awal	Aktor Berhasil login dan mengakses dashboard website SIPKS
Kondisi Akhir	Aktor berhasil mengubah detail penggunaan dana pada pengajuan yang telah dibuat
Alur Normal	
Aktor	**Sistem**
1. Memilih menu Pengajuan Dana	1.1 Menampilkan halaman pengajuan dana
2. Klik tombol detail pada pengajuan	2.1 Menampilkan halaman detail pengajuan
3. Klik tombol edit pada detail pengajuan	3.1 Menampilkan modal edit detail pengajuan
4. Mengubah nilai field pada form	
5. Klik Submit	5.1 Menyimpan data ke database
Alur Alternatif	

A1. Jika Aktor tidak mengisi field jumlah pada form modal edit	
Aktor	Sistem
	A1.1 Sistem menampilkan perintah untuk melengkapi form A 1.2 Kembali ke alur 4

Skenario aktivitas selanjutnya digambarkan ke dalam diagram aktivitas Gambar 3.10. di bawah ini menggambarkan diagram aktivitas melihat detail pengajuan dana anggaran.



Gambar 3.10. Diagram Aktivitas Mengubah Detail Pengajuan Dana Anggaran

3.1.3.4.7 Kasus Penggunaan Menghapus Detail Pengajuan Dana Anggaran

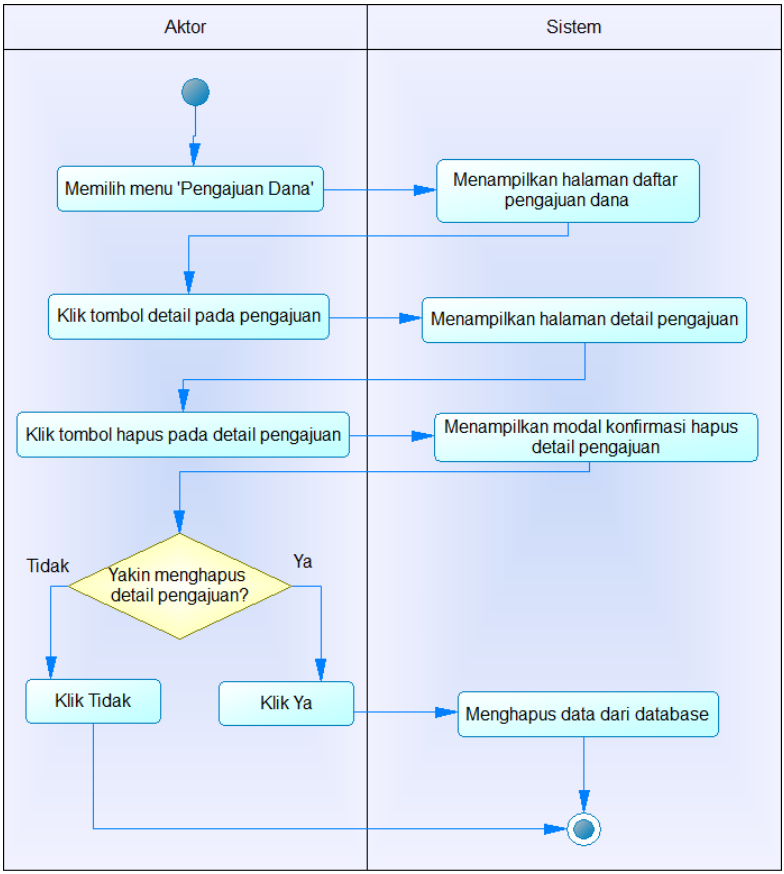
Kasus penggunaan ini diakses oleh Pegawai TU melalui website SIPKS. Pada kasus penggunaan ini, Pegawai TU dapat menghapus detail penggunaan dana anggaran pada pengajuan yang dipilih. Detail yang dihapus akan menghilang dari database dan tampilan Penjelasan skenario yang lebih rinci dapat dilihat pada Tabel 3.9.

Table 3.9 Spesifikasi Kasus Penggunaan Menghapus Detail Pengajuan Dana Anggaran

Nama Use Case	**Menghapus Detail Pengajuan Dana Anggaran**
Nomor	**WU-007**
Deskripsi	Aktor dapat menghapus detail pengajuan pada pengajuan yang dipilih
Aktor	Pegawai Tata Usaha
Kondisi Awal	Aktor Berhasil login dan mengakses dashboard website SIPKS
Kondisi Akhir	Aktor berhasil menghapus detail pada pengajuan yang telah dibuat
Alur Normal	
Aktor	**Sistem**
1. Memilih menu Pengajuan Dana	1.1 Menampilkan halaman pengajuan dana
2. Klik tombol detail pada pengajuan	2.1 Menampilkan halaman detail pengajuan
3. Klik tombol hapus pada detail pengajuan	3.1 Menampilkan modal konfirmasi hapus detail pengajuan
4. Klik ‘Ya’	4.1 Menghapus data dari database
Alur Alternatif	
A1. Jika Aktor memililh ‘Tidak’ mengisi field jumlah pada form modal edit	
Aktor	Sistem

	A1.1 Kembali ke alur 2.1

Skenario aktivitas selanjutnya digambarkan ke dalam diagram aktivitas Gambar 3.11. di bawah ini menggambarkan diagram aktivitas menghapus detail pengajuan dana anggaran.



Gambar 3.11 Diagram Aktivitas Menghapus Detail Pengajuan Dana Anggaran

3.1.3.4.8 Kasus Penggunaan Melihat Pengajuan Dana Anggaran (Mobile)

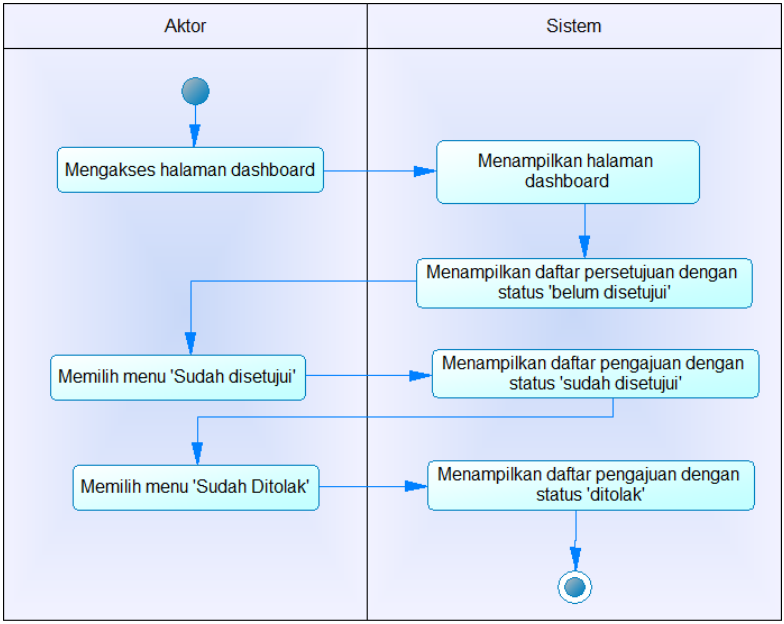
Kasus penggunaan ini diakses oleh Bendahara Sekolah dan Kepala Sekolah melalui aplikasi mobile. Pada kasus penggunaan ini, Bendahara Sekolah dan Kepala Sekolah dapat melihat daftar pengajuan yang statusnya belum disetujui, sudah disetujui, dan ditolak. Penjelasan skenario yang lebih rinci dapat dilihat pada Tabel 3.10.

Table 3.10. Spesifikasi Kasus Penggunaan Melihat Pengajuan Dana Anggaran (Mobile)

Nama Use Case	**Melihat Pengajuan Dana Anggaran (Mobile)**
Nomor	**AU-001**
Deskripsi	Aktor dapat melihat daftar pengajuan dana
Aktor	Bendahara Sekolah,Kepala Sekolah
Kondisi Awal	Aktor Berhasil login ke aplikasi mobile SIPKS
Kondisi Akhir	Aktor berhasil melihat daftar pengajuan dana anggaran pada aplikasi mobile berdasarkan statusnya
Alur Normal	
Aktor	**Sistem**
1. Mengakses halaman dashboard	1.1 Menampilkan halaman dashboard 1.2 Menampilkan daftar pengajuan dengan status ‘belum disetujui’
2. Memilih menu ‘Sudah disetujui’	2.1 Menampilkan daftar pengajuan dengan status ‘sudah disetujui’
3. Memilih menu ‘Sudah Ditolak’	3.1 Menampilkan daftar pengajuan dengan status ‘belum disetujui’

Alur Alternatif
-

Skenario aktivitas selanjutnya digambarkan ke dalam diagram aktivitas Gambar 3.12. di bawah ini menggambarkan diagram aktivitas melihat pengajuan dana anggaran pada aplikasi mobile.



Gambar 3.12 Diagram Aktivitas Melihat Pengajuan Dana Anggaran (Mobile)

3.1.3.4.9 Kasus Penggunaan Mengunggah Sampel Wajah

Kasus penggunaan ini diakses oleh Bendahara Sekolah dan Kepala Sekolah melalui aplikasi mobile. Pada kasus penggunaan ini, Bendahara Sekolah dan Kepala Sekolah melakukan upload sampel wajah yang akan digunakan dalam proses *face recognition* untuk keperluan pemberian persetujuan pengajuan dana anggaran. Sampel wajah selanjutnya akan

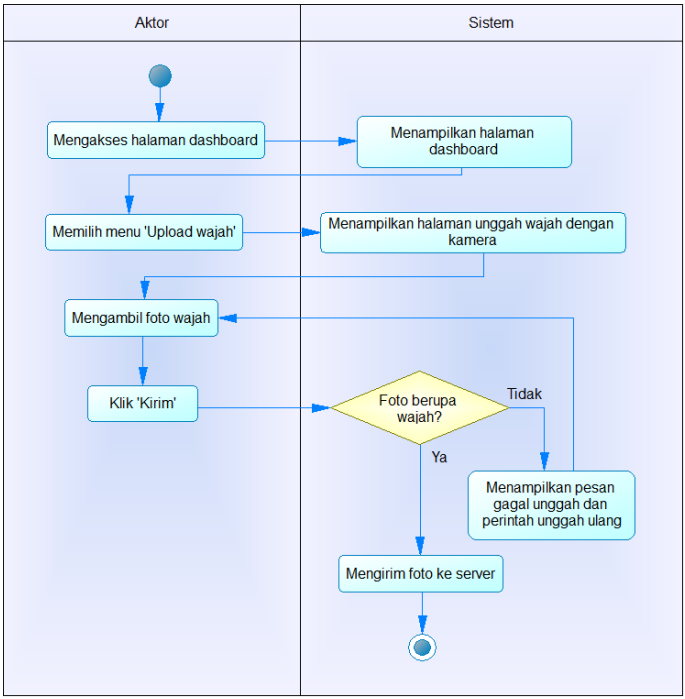
digunakan dalam proses *training* untuk menghasilkan model yang digunakan pada proses *face recognition*. Penjelasan skenario yang lebih rinci dapat dilihat pada Tabel 3.11.

Table 3.11 Spesifikasi Kasus Penggunaan Mengunggah Sampel Wajah

Nama Use Case	**Mengupload Sampel Wajah**
Nomor	**AU-002**
Deskripsi	Aktor dapat mengupload sampel wajah melalui aplikasi mobile dengan menggunakan kamera pada *smartphone*. Pengambilan dan pengunggahan gambar dilakukan sebanyak 6 kali.
Aktor	Bendahara Sekolah,Kepala Sekolah
Kondisi Awal	Aktor Berhasil login ke aplikasi mobile SIPKS
Kondisi Akhir	Aktor berhasil mengunggah sampel wajah
Alur Normal	
Aktor	**Sistem**
1. Mengakses halaman dashboard 2. Memilih menu 'Upload Wajah' 3. Mengambil foto wajah 4. Klik 'Kirim'	1.1 Menampilkan halaman dashboard 2.1 Menampilkan kamera pada halaman unggah sampel wajah 4.1 Menjalankan *face detection* 4.2 Mengirim gambar ke server
Alur Alternatif	
A1. Jika Aktor mengunggah foto bukan wajah	
Aktor	**Sistem**
	A1.1 Menampilkan pesan gagal unggah dan perintah

	unggahulang A1.2 Kembali ke alur 3

Skenario aktivitas selanjutnya digambarkan ke dalam diagram aktivitas Gambar 3.13 di bawah ini menggambarkan diagram aktivitas mengunggah sampel wajah pada aplikasi mobile.



Gambar 3.13 Diagram Aktivitas Mengunggah Sampel Wajah

3.1.3.4.10 Kasus Penggunaan Melakukan Training Sampel Wajah

Kasus penggunaan ini diakses oleh Bendahara Sekolah dan Kepala Sekolah melalui aplikasi mobile. Pada kasus penggunaan ini, Bendahara Sekolah dan Kepala Sekolah melakukan *training* sampel wajah yang sudah berhasil diunggah ke server. Proses *training* menggunakan metode *Generative*

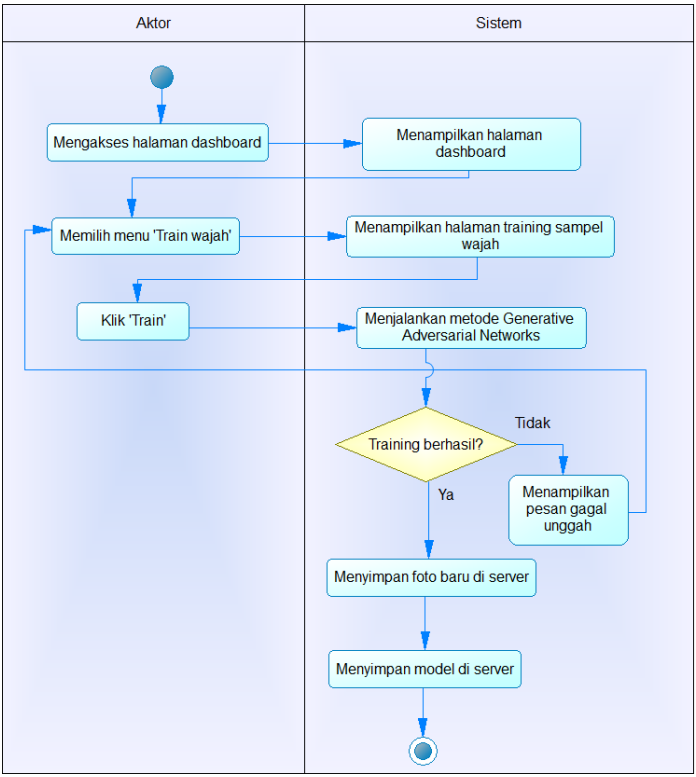
Adversarial Networks, menghasilkan foto baru untuk tambahan dataset serta menghasilkan model yang akan digunakan pada proses pencocokan wajah. Penjelasan skenario yang lebih rinci dapat dilihat pada Tabel 3.12.

Table 3.12 Spesifikasi Kasus Penggunaan Melakukan Training Sampel Wajah

Nama Use Case	**Melakukan *Training* Sampel Wajah**
Nomor	**AU-003**
Deskripsi	Aktor dapat melakukan *training* sampel wajahnya yang berhasil terupload ke server. Proses *training* hanya bisa dilakukakan sebanyak 1 kali.
Aktor	Bendahara Sekolah,Kepala Sekolah
Kondisi Awal	Aktor Berhasil login ke aplikasi mobile SIPKS dan sampel wajah sudah disimpan di server
Kondisi Akhir	Aktor berhasil melakukan *training* sampel wajah
Alur Normal	
Aktor	**Sistem**
1. Mengakses halaman dashboard 2. Memilih menu ‘Train Wajah’ 3. Klik ‘Train’	1.2 Menampilkan halaman dashboard 2.2 Menampilkan halaman *train* wajah 3.1 Menjalankan *Generative Adversarial Networks* 3.2 Menyimpan foto baru di server 3.3 Menyimpan model di server
Alur Alternatif	
A1. Jika *training* sampel wajah gagal dilakukan	
Aktor	**Sistem**

	A1.1 Menampilkan pesan gagal A1.2 Kembali ke alur 2

Skenario aktivitas selanjutnya digambarkan ke dalam diagram aktivitas Gambar 3.14 di bawah ini menggambarkan diagram aktivitas melakukan *training* sampel wajah pada aplikasi mobile.



Gambar 3.14 Diagram Aktivitas Melakukan *Training* Sampel Wajah

3.1.3.4.11 Kasus Penggunaan Memberikan Persetujuan Pengajuan dengan Pengenalan Wajah

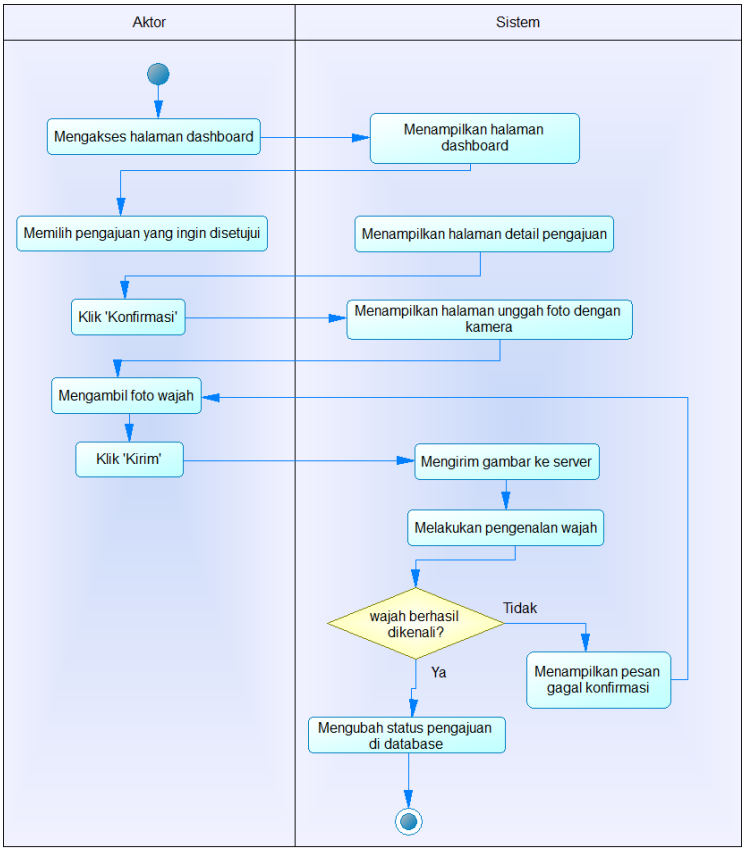
Kasus penggunaan ini diakses oleh Bendahara Sekolah dan Kepala Sekolah melalui aplikasi mobile. Pada kasus penggunaan ini, Bendahara Sekolah dan Kepala Sekolah melakukan persetujuan terhadap pengajuan dana anggaran. Persetujuan dilakukan dengan pengenalan wajah. Jika wajah dikenali dan valid, maka pengajuan berhasil disetujui. Penjelasan skenario yang lebih rinci dapat dilihat pada Tabel 3.13.

Table 3.13 Spesifikasi Kasus Penggunaan Memberikan Persetujuan Pengajuan dengan Pengenalan Wajah

Nama Use Case	**Memberikan Persetujuan Pengajuan dengan PengenalanWajah**
Nomor	**AU-004**
Deskripsi	Aktor melakukan persetujuan pengajuan dana anggaran dengan pengenalan wajah
Aktor	Bendahara Sekolah,Kepala Sekolah
Kondisi Awal	Aktor Berhasil login ke aplikasi mobile SIPKS dan berhasil melakukan *training* sampel wajahnya
Kondisi Akhir	Aktor berhasil melakukan persetujuan dan status pengajuan berubah menjadi disetujui
Alur Normal	
Aktor	**Sistem**
1. Mengakses halaman dashboard	1.3 Menampilkan halaman dashboard
2. Memilih pengajuan yang belum disetujui	2.3 Menampilkan halaman detail pengajuan dana anggaran
3. Klik ‘Konfirmasi’	3.1 Menampilkan halaman unggah foto dengan kamera
4. Mengambil foto wajah	
5. Klik ‘Kirim’	

	5.1 Mengirim foto ke server 5.2 Melakukan pengenalan wajah 5.3 Mengubah status pengajuan di database 5.4 Menampilkan pesan pengajuan berhasil dikonfirmasi
Alur Alternatif	
A1. Jika foto wajah yang diunggah tidak dapat dikenali sebagai aktor	
Aktor	**Sistem**
	A1.1 Menampilkan pesan gagal A1.2 Kembali ke alur 4

Skenario aktivitas selanjutnya digambarkan ke dalam diagram aktivitas Gambar 3.15 di bawah ini menggambarkan diagram aktivitas memberikan persetujuan pengajuan dengan pencocokan wajah pada aplikasi mobile.



Gambar 3.15 Diagram Aktivitas Memberikan Persetujuan Pengajuan dengan Pengenalan Wajah

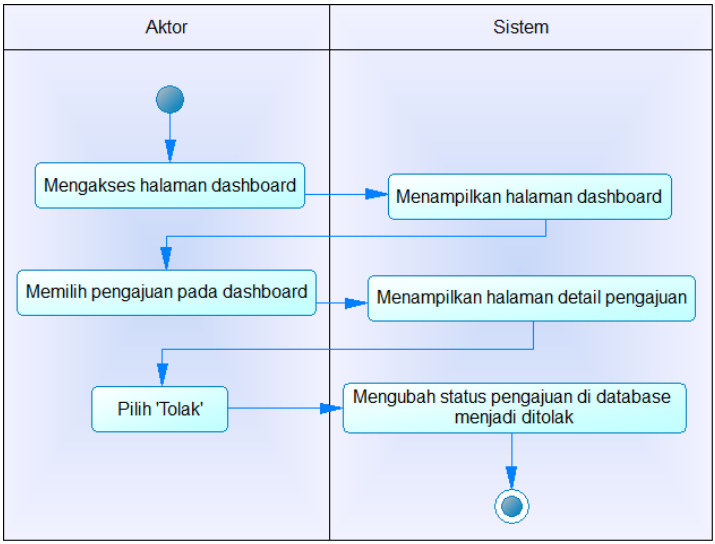
3.1.3.4.12 Kasus Penggunaan Menolak Pengajuan Dana Anggaran

Kasus penggunaan ini diakses oleh Bendahara Sekolah dan Kepala Sekolah melalui aplikasi mobile. Pada kasus penggunaan ini, Bendahara Sekolah dan Kepala Sekolah dapat melakukan penolakan terhadap pengajuan dana anggaran. Penjelasan skenario yang lebih rinci dapat dilihat pada Tabel 3.14.

Table 3.14 Spesifikasi Kasus Penggunaan Menolak Pengajuan Dana Anggaran

Nama Use Case	**Menolak Pengajuan Dana Anggaran**
Nomor	**AU-005**
Deskripsi	Aktor melakukan penolakan terhadap pengajuan dana anggaran
Aktor	Bendahara Sekolah,Kepala Sekolah
Kondisi Awal	Aktor Berhasil login ke aplikasi mobile SIPKS
Kondisi Akhir	Aktor berhasil melakukan penolakan dan status pengajuan menjadi ditolak
Alur Normal	
Aktor	**Sistem**
1. Mengakses halaman dashboard 2. Memilih pengajuan yang akan ditolak 3. Klik ‘Tolak’	1.1 Menampilkan halaman dashboard 2.1 Menampilkan halaman detail pengajuan dana anggaran 3.1 Mengupdate status pengajuan di database menjadi ditolak
Alur Alternatif	
-	

Skenario aktivitas selanjutnya digambarkan ke dalam diagram aktivitas Gambar 3.16 di bawah ini menggambarkan diagram aktivitas menolak pengajuan anggaran dana pada aplikasi mobile.



Gambar 3.16 Diagram Aktivitas Menolak Pengajuan Anggaran Dana

3.2 Perancangan

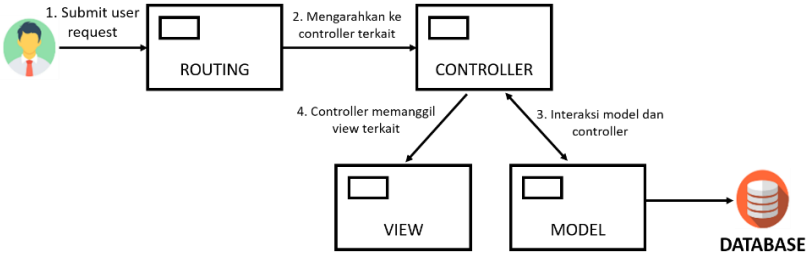
Pada subbab perancangan akan dijelaskan mengenai arsitektur sistem yang digunakan, perancangan proses, perancangan basis data, dan perancangan antarmuka pengguna.

3.2.1 Perancangan Arsitektur Sistem

Arsitektur sistem yang digunakan pada *platform* web pada perangkat lunak ini menggunakan kerangka kerja (*framework)* Laravel dengan pola *Model-View-Controller (MVC).* Gambar xx merupakan ilustrasi arsitektur sistem *platform* web.

Client melakukan *request* data kemudian *Routing* akan menerima *request* dan mengarahkannya kepada *Controller* yang terkait sesuai dengan *Controller* yang terdaftar pada *Routing. Controller* akan berinteraksi dengan model yang dibutuhkan. *Model* akan melakukan *query* terhadap *database.* Hasil *query* yang dilakukan *Model* akan dikembalikan ke *Controller*. Kemudian,

Controller akan memanggil *view* terkait untuk menampilkan hasil yang diminta oleh *client.*



Gambar 3.17 Ilustrasi Arsitektur *Platform* Website

Pada arsitektur sistem *platform mobile* akan memanggil API untuk menghubungkan *platform mobile* dengan *platform* web. Aplikasi *mobile* akan memanggil mengirim *request* melalui API. Data yang dikirim oleh aplikasi *mobile* akan diolah pada *platform* website kemudian hasilnya akan diteruskan ke *plarform mobile* untuk ditampilkan. Ilustrasi arsitektur sistem aplikasi *mobile* dapat dilihat pada Gambar 3.18.

Gambar 3.18 Ilustrasi Arsitektur *Platform Mobile*

3.2.2 Perancangan Basis Data

Perancangan basis data diperlukan dalam membuat suatu aplikasi sistem informasi. MySQL dipilih menjadi aplikasi basis data karena penggunaannya yang mudah. Rancangan basis data ditampilkan dalam bentuk Conceptual Data Model (CDM) dan Physical Data Modal (PDM). Struktur CDM dan PDM disertakan dalam Lampiran CDM dan PDM.

3.2.2.1 Penjelasan Tabel

Penjelasan dan deskripsi rinci table disertakan dalam Lampiran Daftar Rinci Tabel.

3.2.2.1.1 Tabel User

Tabel User digunakan untuk menyimpan data pengguna berupa id, nip, id_sekolah, id_jabatan, alamat_user, no_telp_user, role_user, nama, email, password, dan create_time.

3.2.2.1.2 Tabel Jabatan

Tabel Jabatan digunakan untuk menyimpan data jabatan berupa id_jabatan, nama_jabatan, dan create_time.

3.2.2.1.3 Tabel Sekolah

Tabel Sekolah digunakan untuk menyimpan data sekolah berupa id_sekolah, nama_sekolah, alamat_sekolah, email_sekolah, jenjang_sekolah, no_rek_sekolah, bank_norek_sekolah, nama_norek_sekolah, no_telp_sekolah, dan create_time.

3.2.2.1.4 Tabel Pengajuan

Tabel pengajuan digunakan untuk menyimpan data pengajuan berupa id_pengajuan, id_sekolah, id_akun, judul_pengajuan, deskripsi_pengajuan, nama_peminta_pengajuan, jumlah_pengajuan, status_pengajuan, dan create_time.

3.2.2.1.5 Tabel Detail Pengajuan

Tabel Detail Pengajuan digunakan untuk menyimpan data detail pengajuan berupa id_detail, nama_detail, satuan, harga_satuan, dan create_time.

3.2.2.1.6 Tabel Mapping Pengajuan Detail

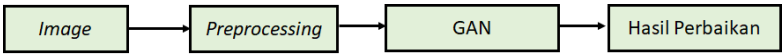
Tabel Mapping Pengajuan Detail digunakan untuk menyimpan data mapping pengajuan dengan detail pengajuan yang dimiliki. Tabel ini terdiri dari id_mapping_pengajua_detail, id_pengajuan, id_detail, jumlah_detail, sub_total, create_time.

3.2.3 Perancangan Proses

Pada Tugas akhir ini akan dilakukan proses *generate* perbaikan foto dan proses pengenalan wajah.

3.2.3.1 Perancangan Proses *Generate* Perbaikan Foto

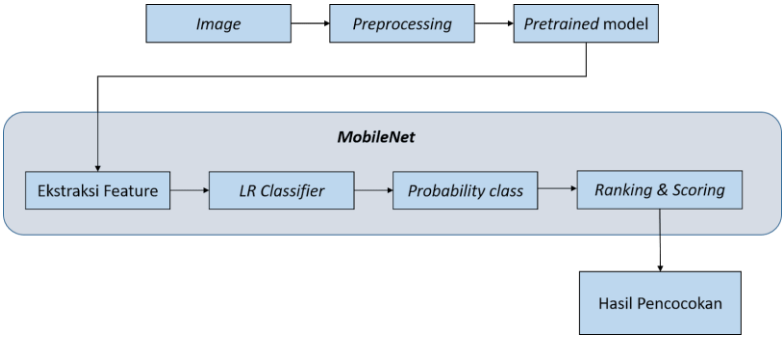
Pada proses ini terdapat beberapa tahap yang dijelaskan pada Gambar 3.19.



Gambar 3.19. Proses *Generate* Perbaikan Foto

Pada Gambar 3.19 dapat dilihat terdapat proses preprocessing. Preprocessing dilakukan untuk mempersiapkan data sebelum proses perbaikan. Tujuan preprocessing adalah untuk menyeragamkan data. Proses *preprocessing* yang dilakukan yaitu *align* foto atau *cropping* foto. Selanjutnya, digunakan *generative adversarial network* yang terdiri dari *discriminator* dan *generator*. Setelah *discriminator* dan *generator* selesai melatih dataset, maka akan dihasilkan foto baru hasil perbaikan dari dataset yang blur.

3.2.3.2 Perancangan Proses Pengenalan Wajah



Gambar 3.20. Proses Pengenalan Wajah

Gambar 3.20 menjelaskan tentang bagaimana proses pengenalan wajah dengan menggunakan *transfer learning* MobileNet. Input yang digunakan merupakan hasil perbaikan gambar menggunakan *Generative Adversarial Networks.* MobileNet dipilih karena tidak perlu membangun model dari awal. Model yang digunakan adalah *pretrained model* yang dihasilkan dari penelitian yang sudah ada dan siap pakai. Dengan begitu, waktu komputasinya bisa lebih cepat. MobileNet yang digunakan dimodifikasi menggunakan *Logistic Regression Classifier*. Dari proses tersebut didapatkan *probability class* yang merupakan

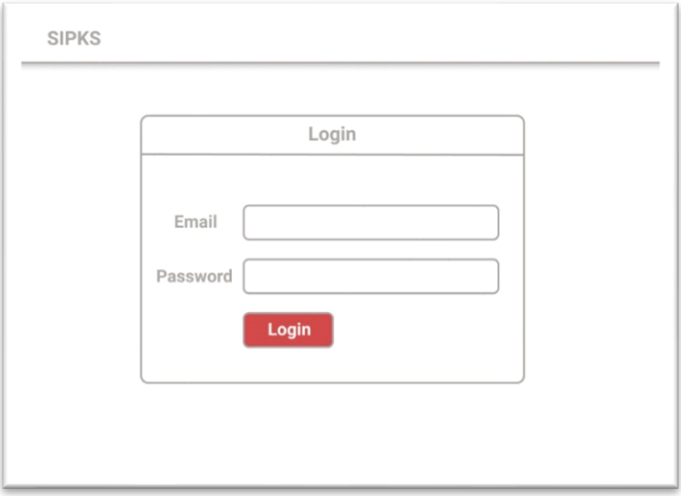
probabilitas kemiripan fitur dari input dan data yang sudah dilatih sebelumnya. Dari *probability class* didapatkan *score* dan *ranking* yang kemudian digunakan untuk menentukan hasil pencocokan.

3.2.4 Perancangan Antarmuka Pengguna

Perancangan antarmuka pengguna merupakan hal yang penting dan perlu diperhatikan dalam perancangan perangkat lunak. Antarmuka pengguna berhubungan langsung dengan aktor, sehingaa harus memiliki tampilan yang menarik. Selain itu, antarmuka pengguna juga harus memberikan kemudahan dalam penggunaan perangkat lunak, sehingga pengguna tidak kebingungan saat menggunakan perangkat lunak. Pada aplikasi ini terdapat antarmuka pengguna pada dua platform yaitu mobile dan web. Pada platform web, pengguna yang memiliki hak akses, yaitu Pegawai Tata Usaha. Sedangkan, pada platform mobile, terdapat dua hak akses pengguna yaitu Kepala Sekolah dan Bendahara Sekolah. Pegawai Tata Usaha dapat mengakses aplikasi website menggunakan semua browser dan menggunakan keyboard serta mouse. Sementara aplikasi mobile dapat diakses dengan minimal Android 6.0.

3.2.4.1 Rancangan Antarmuka Halaman *Login* Website

Halaman ini merupakan halaman awal ketika aktor mengakses website. Pada halaman ini terdapat form login yang berisi field email dan password. Rancangan halaman dapat dilihat pada Gambar 3.21 . Atribut antarmuka dapat dilihat pada Tabel 3.15.



Gambar 3.21 Rancangan Halaman Antarmuka *Login* Website

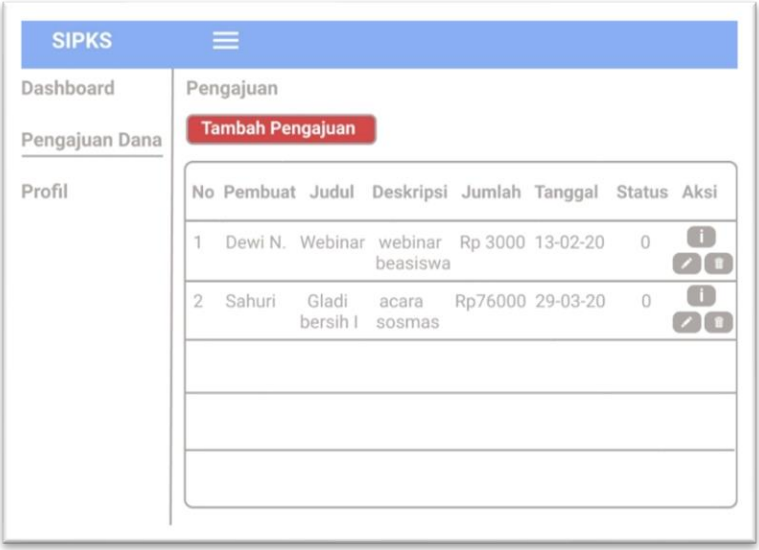
Tabel 3.15 Atribut Antarmuka Halaman *Login* Website

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan/ Keluaran
1	Email Akun	*Textfield*	Field mengisi email pengguna web (pegawai TU)	*String*
2	Password Akun	*Textfield*	Field mengisi password pengguna web (pegawai TU)	*String*
3	*LoginButton*	*Button*	Tombol untuk login akun menuju halaman dashboard	*Button Clicked*

3.2.4.2 Rancangan Antarmuka Halaman List Data Pengajuan

Pada halaman ini ditampilkan semua pengajuan yang dibuat oleh aktor dalam bentuk table. Halaman ini akan digunakan pada kasus penggunaan melihat pengajuan dana anggaran

(website). Rancangan halaman dapat dilihat pada Gambar 3.22. Atribut antarmuka dapat dilihat pada Tabel 3.16.



Gambar 3.22 Rancangan Antarmuka Halaman List Data Pengajuan

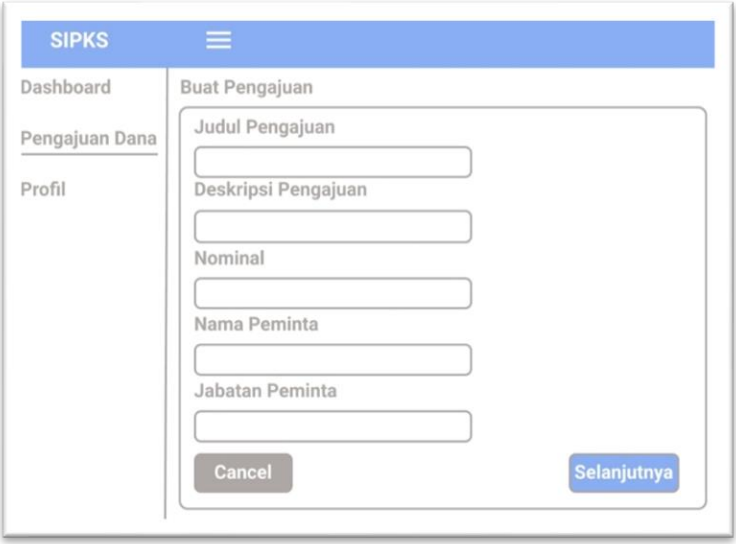
Tabel 3.16 Atribut Antarmuka Halaman List Data Pengajuan

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan/ Keluaran
1	Dashboard	*Link*	Link menuju halaman dashboard	*Link Clicked*
2	Pengajuan Dana	*Link*	Link menuju halaman pengajuan dana	*Link Clicked*
3	Profil	*Link*	Link menuju halaman profil	*Link Clicked*
4	*buttonTambahPengajuan*	*Button*	Tombol Tambah	*Button Clicked*

			Pengajuan menuju halaman Tambah Pengajuan	
5	*buttonDetail*	*Button*	Tombol detail menuju ke halaman detail pengajuan	*Button Clicked*
6	*buttonEdit*	*Button*	Tombol edit menuju ke modal edit pengajuan	*Button Clicked*
7	*buttonDelete*	*Button*	Tombol delete menuju ke modal hapus pengajuan	*Button Clicked*

3.2.4.3 Rancangan Antarmuka Halaman Tambah Pengajuan

Pada halaman ini aktor dapat menambahkan pengajuan. Halaman ini berisi form tambah pengajuan. Halaman ini akan digunakan pada kasus penggunaan menambahkan pengajuan dana anggaran. Rancangan halaman dapat dilihat pada Gambar 3.23. Atribut antarmuka dapat dilihat pada Tabel 3.17.



Gambar 3.23 Rancangan Antarmuka Halaman Tambah Pengajuan

Tabel 3.17 Atribut Antarmuka Halaman Tambah Pengajuan

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan/Keluaran
1	Dashboard	*Link*	Link menuju halaman dashboard	*Link Clicked*
2	Pengajuan Dana	*Link*	Link menuju halaman pengajuan dana	*Link Clicked*
3	Profil	*Link*	Link menuju	*Link Clicked*

			halaman profil	
4	*inputJudulPengajuan*	*Input Text*	Field input untuk mengisikan judul pengajuan	*String*
5	*inputDeskripsiPengajuan*	*Input Text*	Field input untuk mengisikan deskripsi pengajuan	*String*
6	*inputNominal*	*Input Text*	Field input untuk mengisikan nominal pengajuan	*Integer*
7	*inputNamaPeminta*	*Input Text*	Field input untuk mengisikan nama peminta pengajuan	*String*
8	*inputJabatanPeminta*	*Input Text*	Field input untuk mengisikan jabatan peminta pengajuan	*String*

9	*buttonCancel*	*Button*	Tombol cancel untuk membatalkan pembuatan pengajuan, menuju ke halaman daftar pengajuan	*Button Clicked*
10	*buttonSelanjutnya*	*Button*	Tombol selanjutnya menuju halaman Tambah Detail	*Button Clicked*

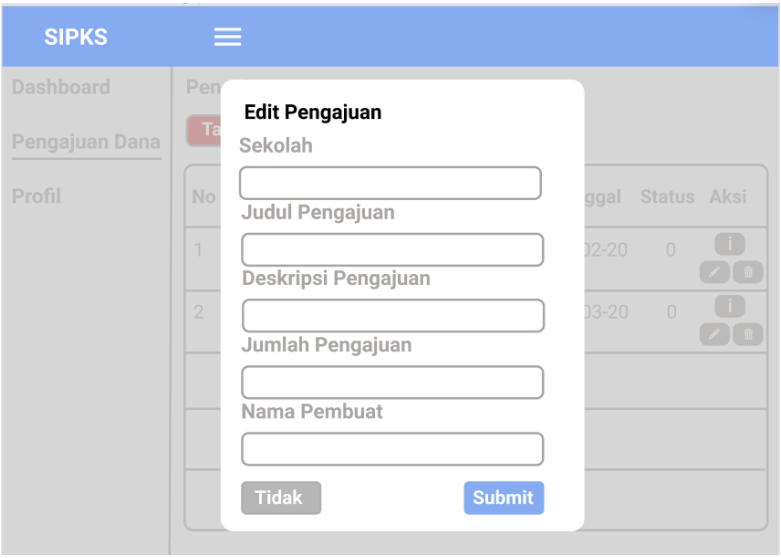
3.2.4.4 Rancangan Antarmuka Halaman Edit Pengajuan

Pada halaman ini ditampilkan modal edit data pengajuan. Halaman ini akan digunakan pada kasus penggunaan mengubahpengajuan dana anggaran. Rancangan halaman dapat dilihat pada Gambar 3.24. Atribut antarmuka dapat dilihat pada Tabel 3.18.

Tabel 3.18 Atribut Antarmuka Halaman Edit Pengajuan

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan/ Keluaran
1	*inputSekolah*	*Input Text*	Field input untuk	*String*

			mengisikan nama sekolah	
2	*inputJudulPengajuan*	*Input Text*	Field input untuk mengisikan judul pengajuan	*String*
3	*inputDeskripsiPengajuan*	*Input Text*	Field input untuk mengisikan deskripsi pengajuan	*String*
4	*inputNominal*	*Input Text*	Field input untuk mengisikan nominal pengajuan	*Integer*
5	*inputNamaPeminta*	*Input Text*	Field input untuk mengisikan nama peminta pengajuan	*String*
6	*buttonTidak*	*Button*	Tombol untuk membatalkan edit pengajuan	*Button Clicked*
7	*buttonSubmit*	*Button*	Tombol untuk menyimpan perubahan data pengajuan	*Button Clicked*



Gambar 3.24 Rancangan Antarmuka Halaman Edit Pengajuan

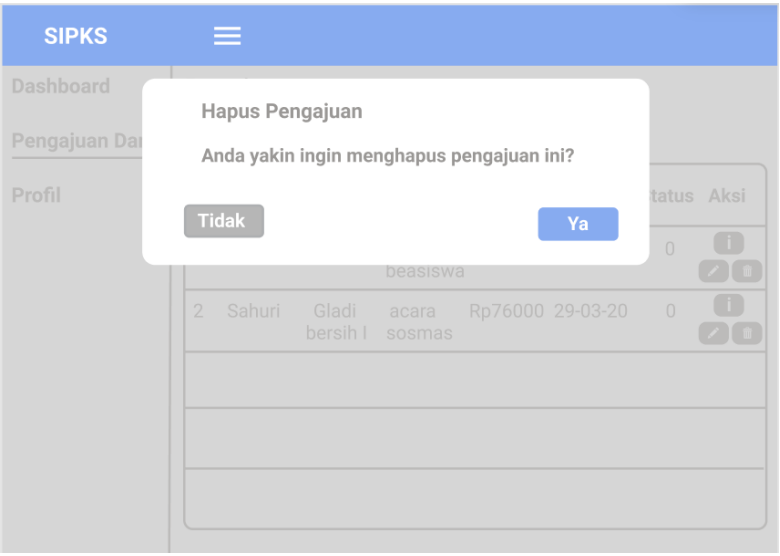
3.2.4.5 Rancangan Antarmuka Halaman Hapus Pengajuan

Pada halaman ini aktor dapat menghapus pengajuan yang telah dibuat. Halaman ini berisi modal konfirmasi hapus pengajuan. Halaman ini akan digunakan pada kasus penggunaan menghapus pengajuan dana anggaran. Rancangan halaman dapat dilihat pada Gambar 3.25. Atribut antarmuka dapat dilihat pada Tabel 3.19.

Tabel 3.19 Atribut Antarmuka Halaman Hapus Pengajuan

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan/ Keluaran
1	*buttonTidak*	*Button*	Tombol untuk membatalkan hapus pengajuan	*Button Clicked*

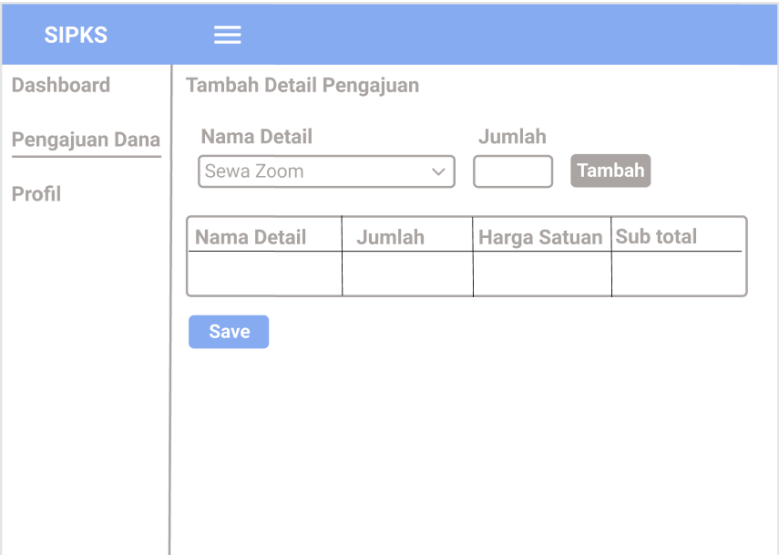
2	*buttonYa*	*Button*	Tombol untuk menghapus data pengajuan	*Button Clicked*



Gambar 3.25 Rancangan Antarmuka Halaman Hapus Pengajuan

3.2.4.6 Rancangan Antarmuka Halaman Tambah Detail Pengajuan

Pada halaman ini aktor dapat menambahkan detail pengajuan yang telah dibuat. Halaman ini berisi form tambah detail pengajuan. Halaman ini akan digunakan pada kasus penggunaan menambahkan detail pengajuan dana anggaran. Rancangan halaman dapat dilihat pada Gambar 3.26. Atribut antarmuka dapat dilihat pada Tabel 3.20.



Gambar 3.26 Rancangan Antarmuka Halaman Tambah Detail Pengajuan

Tabel 3.20 Atribut Antarmuka Halaman Tambah Detail Pengajuan

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan/ Keluaran
1	Dashboard	*Link*	Link menuju halaman dashboard	*Link Clicked*
2	Pengajuan Dana	*Link*	Link menuju halaman pengajuan dana	*Link Clicked*
3	Profil	*Link*	Link menuju halaman profil	*Link Clicked*
4	*inputNamaDetail*	*Select Option*	Field select untuk memilih	*String*

			nama detail pengajuan	
5	*inputJumlah Detail*	*Input Text*	Field input untuk jumlah setiap detail pengajuan	*Integer*
6	*inputNominal*	*Input Text*	Field input untuk mengisikan nominal pengajuan	*Integer*
7	*inputHargaSatuanDetail*	*Input Text*	Field input untuk mengisikan harga satuan setiap detail pengajuan	*Integer*
8	*inputSubTotal*	*Input Text*	Field input untuk mengisikan sub total setiap detail pengajuan	*Integer*
9	*buttonTambahDetail*	*Button*	Tombol tambah detail untuk menambahkan row input detail	*Button Clicked*
10	*buttonDeleteInputDetail*	*Button*	Tombol hapus untuk menghilangkan row input detail	*Button Clicked*

11	*buttonSaveDetail*	*Button*	Tombol save untuk menyimpan input detail pengajuan	*Button Clicked*

3.2.4.7 Rancangan Antarmuka Halaman List Detail Pengajuan

Pada halaman ini aktor dapat melihat daftar detail pengajuan. Halaman ini berisi table detail pengajuan dari pengajuan yang dipilih. Halaman ini akan digunakan pada kasus penggunaan melihat detail pengajuan dana anggaran. Rancangan halaman dapat dilihat pada Gambar 3.27. Atribut antarmuka dapat dilihat pada Tabel 3.21.

Tabel 3.21 Atribut Antarmuka Halaman List Detail Pengajuan

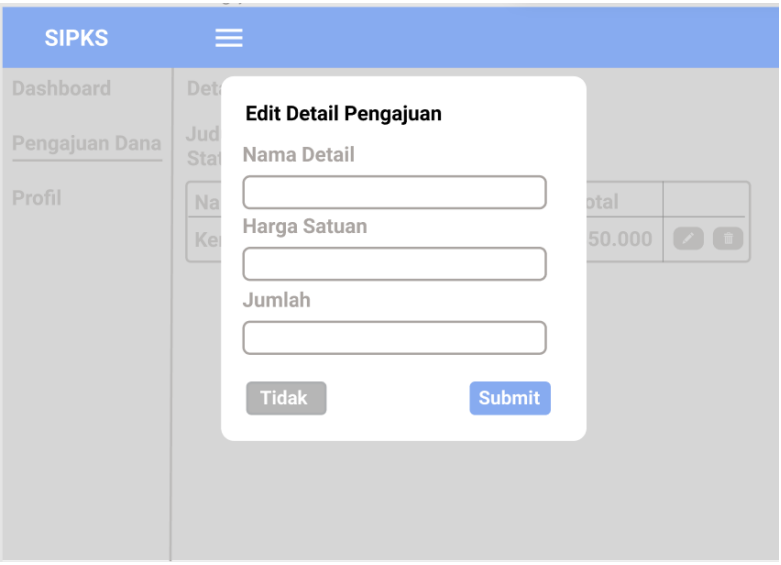
No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan/ Keluaran
1	Dashboard	*Link*	Link menuju halaman dashboard	*Link Clicked*
2	Pengajuan Dana	*Link*	Link menuju halaman pengajuan dana	*Link Clicked*
3	Profil	*Link*	Link menuju halaman profil	*Link Clicked*
4	*buttonEditDetail*	*Button*	Tombol edit detail menuju ke modal edit detail pengajuan	*Button Clicked*

5	*buttonDelete Detail*	*Button*	Tombol delete detail menuju ke modal hapus detail pengajuan	*Button Clicked*

Gambar 3.27 Rancangan Antarmuka Halaman List Detail Pengajuan

3.2.4.8 Rancangan Antarmuka Halaman Edit Detail Pengajuan

Pada halaman ini aktor dapat mengubah detail pengajuan. Halaman ini berisi form edit detail pengajuan. Halaman ini akan digunakan pada kasus penggunaan mengubah detail pengajuan dana anggaran. Rancangan halaman dapat dilihat pada Gambar 3.28. Atribut antarmuka dapat dilihat pada Tabel 3.22.



Gambar 3.28 Rancangan Antarmuka Halaman Edit Detail Pengajuan

Tabel 3.22 Atribut Antarmuka Halaman Edit Detail Pengajuan

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan/ Keluaran
1	*inputNamaDetail*	*Input Text*	Field input untuk mengisikan nama detail	*String*
2	*inputHargaSatuan*	*Input Text*	Field input untuk mengisikan harga satuan	*String*
3	*inputJumlah*	*Input Text*	Field input untuk mengisikan nominal pengajuan	*Integer*

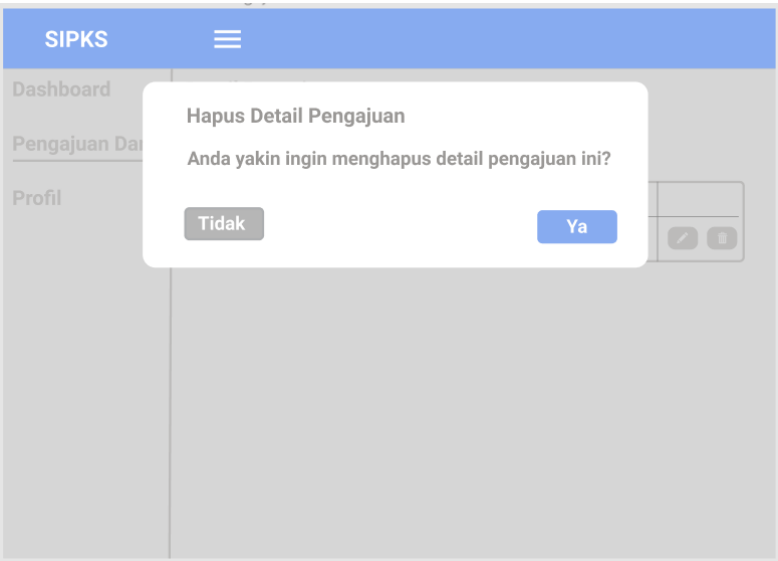
4	*buttonTidak*	*Button*	Tombol untuk membatalkan edit detail pengajuan	*Button Clicked*
5	*buttonSubmit*	*Button*	Tombol untuk menyimpan perubahan detail pengajuan	*Button Clicked*

3.2.4.9 Rancangan Antarmuka Halaman Hapus Detail Pengajuan

Pada halaman ini aktor dapat menghapus detail pengajuan dari pengajuan tertentu. Halaman ini berisi modal konfirmasi hapus detail pengajuan. Halaman ini akan digunakan pada kasus penggunaan menghapus detail pengajuan dana anggaran. Rancangan halaman dapat dilihat pada Gambar 3.29. Atribut antarmuka dapat dilihat pada Tabel 3.23.

Tabel 3.23 Atribut Antarmuka Halaman Hapus Detail Pengajuan

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan/ Keluaran
1	*buttonTidak*	*Button*	Tombol untuk membatalkan hapus pengajuan	*Button Clicked*
2	*buttonYa*	*Button*	Tombol untuk menghapus data pengajuan	*Button Clicked*



Gambar 3.29. Rancangan Antarmuka Halaman Hapus Detail Pengajuan

3.2.4.10 Rancangan Antarmuka Halaman *Start Page* Android

Halaman ini merupakan halaman awal ketika aktor membuka aplikasi android SIPKS. Pada halaman ini terdapat gambar logo SIPKS beserta satu tombol untuk menuju halaman login. Rancangan halaman dapat dilihat pada Gambar 3.30. Atribut antarmuka dapat dilihat pada Tabel 3.24.

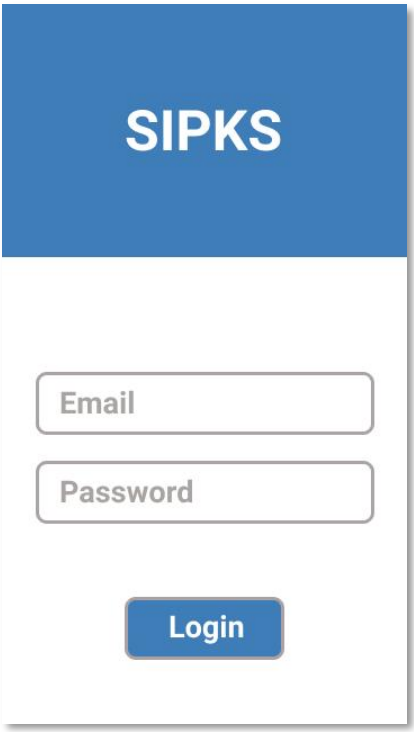
Gambar 3.30. Rancangan Antarmuka Halaman *Start Page* Android

Tabel 3.24 Atribut Antarmuka Halaman *Start Page* Android

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan/Keluaran
1	*ButtonLanjutkan*	*Button*	Tombol lanjutkan menuju halaman login	*Button Clicked*

3.2.4.11Rancangan Antarmuka Halaman *Login* Android

Halaman ini merupakan halaman yang digunakan aktor untuk masuk ke aplikasi android SIPKS. Halaman ini ditampilkan setelah aktor menekan tombol 'Lanjutkan' pada halaman *start page*. Rancangan halaman dapat dilihat pada Gambar 3.31. Atribut antarmuka dapat dilihat pada Tabel 3.25.



Gambar 3.31. Rancangan Antarmuka Halaman *Login* Android

Tabel 3.25 Atribut Antarmuka Halaman *Login* Android

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan/ Keluaran

1	*inputEmail*	*Edit Text*	Input untuk mengisikan email pengguna android (bendahara maupun kepala sekolah)	*String*
2	*inputPassword*	*Edit Text*	Input untuk mengisikan password pengguna android (bendahara maupun kepala sekolah)	*String*
3	*ButtonLogin*	*Button*	Tombol login akun menuju halaman dashboard	*Button Clicked*

3.2.4.12Rancangan Antarmuka Halaman Dashboard Android

Halaman ini merupakan halaman yang ditampilkan setelah aktor berhasil melakukan login ke aplikasi android. Rancangan halaman dapat dilihat pada Gambar 3.32. Atribut antarmuka dapat dilihat pada Tabel 3.26.

Tabel 3.26 Atribut Antarmuka Halaman Dashboard Android

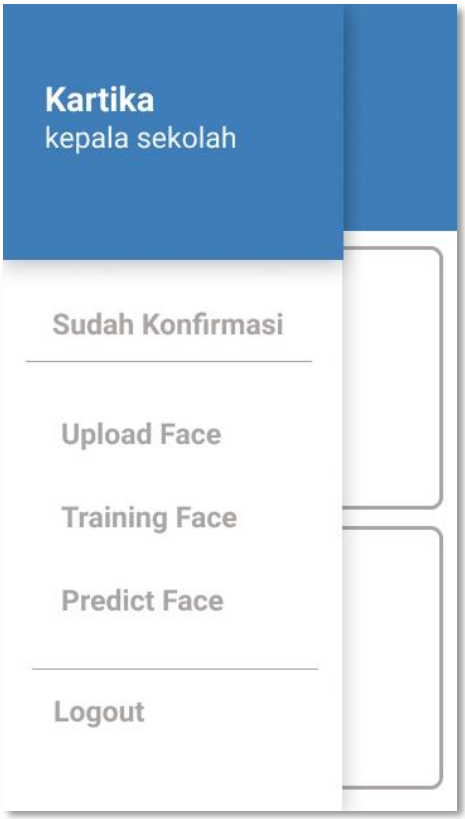
No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan /Keluaran
1	*nama*	*Text View*	Menampilkan nama akun sesuai *session*	*String*

2	*jabatan*	*Text View*	Menampilkan jabatan akun sesuai *session*	*String*
3	*judulPengajuan*	*Recycler View*	Menampilkan daftar judul pengajuan yang belum di setujui	*String*
4	*JumlahPengajuan*	*Recycler View*	Menampilkan daftar jumlah pengajuan yang belum disetujui	*Integer*
5.	*Waktu Pengajuan*	*Recycler View*	Menampilkan waktu pengajuan yang belum disetujui	*String*

Gambar 3.32. Rancangan Antarmuka Halaman Dashboard Android

3.2.4.13 Rancangan Antarmuka Halaman Menu Sidebar

Halaman ini merupakan halaman list menu dalam bentuk sidebar. Menu bisa diakses oleh Kepala Sekolah dan Bendahara Sekolah setelah berhasil login ke aplikasi android. Rancangan halaman dapat dilihat pada Gambar 3.33. Atribut antarmuka dapat dilihat pada Tabel 3.27.



Gambar 3.33 Rancangan Antarmuka Halaman Menu Sidebar

Tabel 3.27 Atribut Antarmuka Halaman Menu Sidebar

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan/Keluaran
1	*nama*	*Text View*	Menampilkan nama akun sesuai *session*	*String*
2	*jabatan*	*Text View*	Menampilkan jabatan akun	*String*

			sesuai *session*	
3	*navSudahKonfirmasi*	*Button*	Tombol sudah konfirmasi menuju halaman daftar pengajuan yang sudah disetujui	*Button Clicked*
4	*navUploadFace*	*Button*	Tombol upload wajah menuju halaman upload wajah	*Button Clicked*
5	*navTrainingFace*	*Button*	Tombol training wajah menuju halaman training wajah	*Button Clicked*
6	*navPredictFace*	*Button*	Tombol predict wajah menuju halaman prediksi wajah	*Button Clicked*
7	*navLogout*	*Button*	Tombol logout untuk logout dari session.	*Button Clicked*

3.2.4.14 Rancangan Antarmuka Halaman Pengajuan Belum Disetujui

Halaman ini berisi pengajuan yang statusnya belum disetujui berserta detail atau rincian penggunaan dana anggaran. Halaman ini akan digunakan pada kasus memberikan penolakan pengajuan dana anggaran dengan menekan tombol tolak, dan atau memberikan persetujuan pencairan dana anggaran dengan menekan tombol konfirmasi dan akan dilakukan pencocokan wajah. Rancangan halaman dapat dilihat pada Gambar 3.34. Atribut antarmuka dapat dilihat pada Tabel 3.28.

Gambar 3.34. Rancangan Antarmuka Halaman Pengajuan Belum Disetujui

Tabel 3.28 Atribut Antarmuka Halaman Pengajuan Belum Disetujui

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan/Keluaran
1	*judulPengajuan*	*Text View*	Menampilkan judul pengajuan	*String*
2	*tanggalPengajuan*	*Text View*	Menampilkan waktu dibuatnya pengajuan	*String*
3	*jumlahPengajuan*	*Text View*	Menampilkan total dana yang diajukan	*String*
4	*statusPengajuan*	*Text View*	Menampilkan status persetujuan pengajuan	*String*
5	*namaDetailBarang*	*Recycler View*	Menampilkan daftar nama setiap jennis item	*String*
6	*hargaSatuanBarang*	*Recycler View*	Menampilkan daftar jumlah satuan setiap jenis item	*String*

7	*totalBarang*	*Recycl er View*	Menampil kan total harga setiap jenis item	*String*
8	*ButtonTolak*	*Button*	Tombol untuk menolak pengajuan dana anggaran	*Button Clicked*
9	*ButtonKonfirma si*	*Button*	Tombol untuk melakukan konfirmasi menuju halaman upload tanda tangan	*Button Clicked*

3.2.4.15 Rancangan Antarmuka Halaman Upload Wajah

Pada halaman ini aktor dapat melakukan upload wajah menggunakan kamera pada *smartphone*. Foto wajah yang diupload akan digunakan untuk proses *training* dan validasi persetujuan pengajuan dana anggaran. Halaman ini berisi satu tombol dan fragmen kamera. Halaman ini akan digunakan pada kasus mengunggah sampel wajah dan memberikan persetujuan pencairan dana anggaran. Rancangan halaman dapat dilihat pada Gambar 3.35. Atribut antarmuka dapat dilihat pada Tabel 3.29.

Tabel 3.29 Atribut Antarmuka Halaman Upload Wajah

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan/ Keluaran

1	*cameraView*	*Camera View*	Menampilkan fragmen kamera	*Image*
2	*gambar*	*Image View*	Menampilkan gambar hasil kamera	*Image*
3	*buttonKirim*	*Button*	Tombol untuk menangkap gambar dan mengirim gambar	*Button Clicked*

Gambar 3.35. Rancangan Antarmuka Halaman Upload Wajah

3.2.4.16 Rancangan Antarmuka Halaman *Training* Wajah

Pada halaman ini terdapat tombol untuk melakuka proses *training* atau pelatihan data. Halaman ini akan digunakan pada kasus melakukan *training* sampel wajah Rancangan halaman dapat dilihat pada Gambar 2.36. Atribut antarmuka dapat dilihat pada Tabel 3.30.

Gambar 3.36 Rancangan Antarmuka Halaman *Training* Wajah

Tabel 3.30 Atribut Antarmuka Halaman *Training* Wajah

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan/ Keluaran
1	*buttonTrain*	*Button*	Tombol untuk melakukan proses *training* wajah	*Button Clicked*

3.2.4.17 Rancangan Antarmuka Halaman Pengajuan Sudah Disetujui

Halaman ini berisi pengajuan yang statusnya sudah disetujui berserta detail atau rincian penggunaan dana anggaran. Rancangan halaman dapat dilihat pada Gambar 3.37. Atribut antarmuka dapat dilihat pada Tabel 3.31.

Gambar 3.37. Rancangan Antarmuka Halaman Pengajuan Sudah Disetujui

Tabel 3.31 Atribut Antarmuka Halaman Pengajuan Sudah Disetujui

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan/ Keluaran
1	*judulPengajuan*	*Text View*	Menampilkan judul pengajuan	*String*

2	*tanggalPengajuan*	*Text View*	Menampilkan waktu dibuatnya pengajuan	*String*
3	*jumlahPengajuan*	*Text View*	Menampilkan total dana yang diajukan	*String*
4	*statusPengajuan*	*Text View*	Menampilkan status persetujuan pengajuan	*String*
5	*namaDetailBarang*	*Recycler View*	Menampilkan daftar nama setiap jennis item	*String*
6	*hargaSatuanBarang*	*Recycler View*	Menampilkan daftar jumlah satuan setiap jenis item	*String*
7	*totalBarang*	*Recycler View*	Menampilkan total harga setiap jenis item	*String*

3.2.4.18Rancangan Antarmuka Halaman Pengajuan Ditolak

Halaman ini berisi pengajuan yang statusnya ditolak beserta detail atau rincian penggunaan dana anggaran. Rancangan halaman dapat dilihat pada Gambar 3.38. Atribut antarmuka dapat dilihat pada Tabel 3.32.

Gambar 3.38. Rancangan Antarmuka Halaman Pengajuan Ditolak

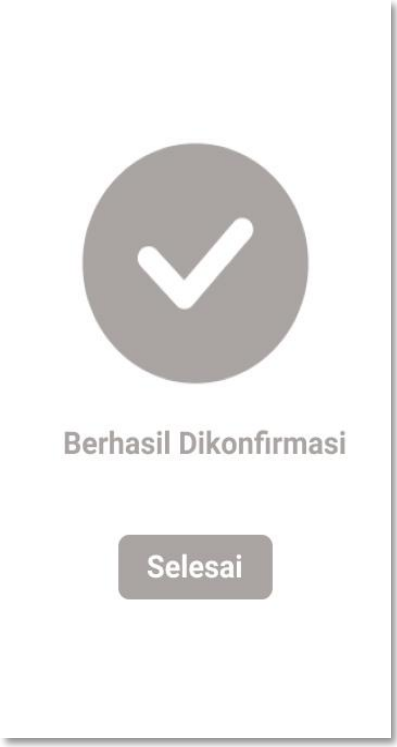
Tabel 3.32 Atribut Antarmuka Halaman Pengajuan Ditolak

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan/ Keluaran

1	*judulPengajuan*	*Text View*	Menampilkan judul pengajuan	*String*
2	*tanggalPengajuan*	*Text View*	Menampilkan waktu dibuatnya pengajuan	*String*
3	*jumlahPengajuan*	*Text View*	Menampilkan total dana yang diajukan	*String*
4	*statusPengajuan*	*Text View*	Menampilkan status persetujuan pengajuan	*String*
5	*namaDetailBarang*	*Recycler View*	Menampilkan daftar nama setiap jennis item	*String*
6	*hargaSatuanBarang*	*Recycler View*	Menampilkan daftar jumlah satuan setiap jenis item	*String*
7	*totalBarang*	*Recycler View*	Menampilkan total harga setiap jenis item	*String*

3.2.4.19Rancangan Antarmuka Halaman Berhasil Disetujui

Halaman ini ditampilkan setelah proses persetujuan berhasil dilakukan, dimana proses *face recognition* menyatakan wajah yang diupload berhasil dikenali sebagai aktor yang memiliki akun tersebut. Rancangan halaman dapat dilihat pada Gambar 3.39. Atribut antarmuka dapat dilihat pada Tabel 3.33.



Gambar 3.39. Rancangan Antarmuka Halaman Berhasil Disetujui

Tabel 3.33 Atribut Antarmuka Halaman Berhasil Disetujui

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan/ Keluaran
1	*statusKonfirmasi*	Text View	Menampilkan status konfirmasi	*String*
2	*ButtonSelesai*	*Button*	Tombol Selesai menuju halaman detail sudah dikonfirmasi	*Button Clicked*

3.2.5 Perancangan API *Endpoint*

Pada Tugas Akhir ini dibuat *endpoint* yang digunakan pada aplikasi *mobile* untuk berkomunikasi dengan aplikasi web. Perancangan API *endpoint* pada Tugas Akhir ini dijelaskan pada Tabel 3.34.

Tabel 3.34 API yang Akan Digunakan

No	Endpoint	Method	Penggunaan dalam Aplikasi Android
1	/login	POST	Login ke aplikasi
2	/profile	GET	Menampilkan profile akun
3	/pengajuan	GET	Menampilkan daftar pengajuan
4	/pengajuan/ detail/{id}	GET	Menampilkan detail pengajuan dari id pengajuan yang dipilih
5	/trainfacegan	GET	Menjalankan proses training foto wajah
6	/sendface	POST	Mengirimkan data foto wajah ke server
7	/predictface/ {id}	POST	Melakukam pencocokan wajah dan persetujuan pengajuan

3.2.5.1 Struktur JSON API *Endpoint*

Struktur API yang dirancang pada Tugas Akhir ini menggunakan JSON. Parameter-parameter pada proses *request* dan *response* yang akan digunakan oleh API pada Tugas Akhir ini disertakan pada Lampiran Struktur JSON API Endpoint.

BAB IV

IMPLEMENTASI

Pada Bab ini akan dijelaskan mengenai implementasi yang dilakukan berdasarkan rancangan yang telah dijabarkan pada bab sebelumnya. Sebelum penje;asan implementasi akan ditunjukkan terlebih dahulu lingkungan untuk melakukan implementasi. Pada bagian implementasi juga akan dijelaskan mengenai fungsi-fungsi yang digunakan dalam program Tugas Akhir ini dan disertai dengan kode semu masing-masing fungsi utama.

4.1. Lingkungan Implementasi

Lingkungan implementasi sistem yang digunakan untuk mengembangkan Tugas Akhir memiliki spesifikasi perang seperti yang ditampilkan pada Tabel 4.1.

Tabel 4.1 Lingkungan Implementasi Perangkat Lunak

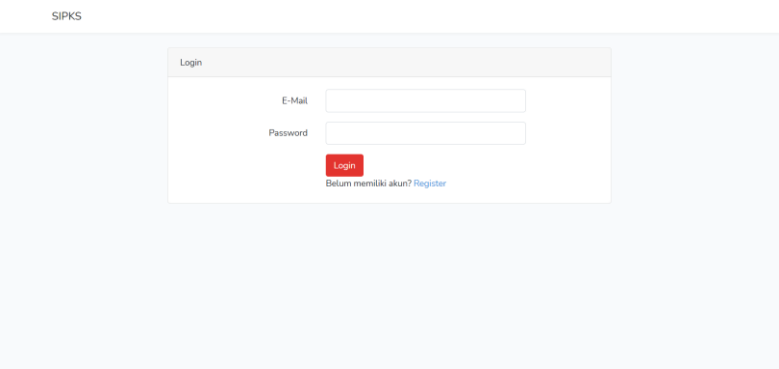
Perangkat	Spesifikasi
Perangkat keras	Prosesor: Intel® Core™ i7-8550U CPU @1.80GHz (8CPUs) , ~2.0GHz Memori: 8 GB
Perangkat lunak	Sistem Operasi : Microsoft Windows 10 64-bit
	Perangkat Pengembang : Android Studio 4.0.1, Microsoft Visual Studio Code, Postman
	Perangkat Perancang Diagram : Sybase Power Designer 16, Figma
	Perangkat Database : MySQL

4.2. Implementasi Antarmuka Sistem

Berikut ini akan dijelaskan mengenai implementasi antarmuka sistem yang sudah direalisasikan.

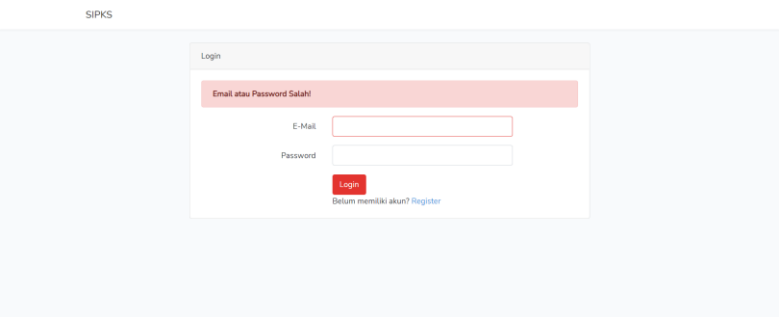
4.2.1. Implementasi Antarmuka Halaman *Login* Website

Halaman ini merupakan halaman pertama yang muncul ketika mengakses website SIPKS. Pada halaman ini terdapat form login dnegan field email, password , dan tombol login. Implementasi antarmuka halaman *login* dapat dilihat pada Gambar 4.1 dan Gambar 4.2.



Gambar 4.1 Implementasi Antarmuka *Login* Website

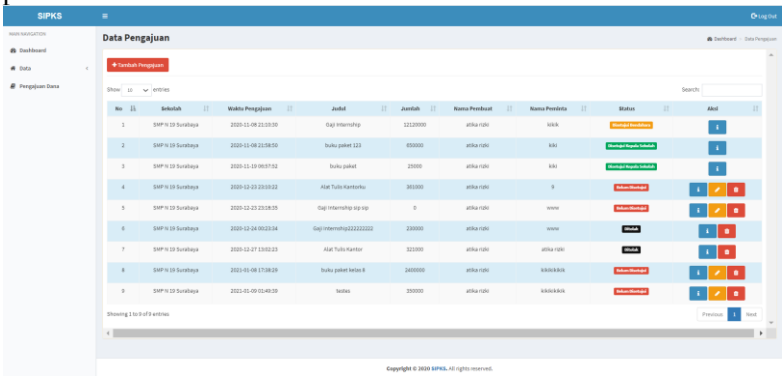
Berikut implementasi antarmuka jika email dan/atau password yang dimasukkan salah.



Gambar 4.2 Implementasi Antarmuka *Login* Website Error

4.2.2. Implementasi Antarmuka Halaman Data Pengajuan

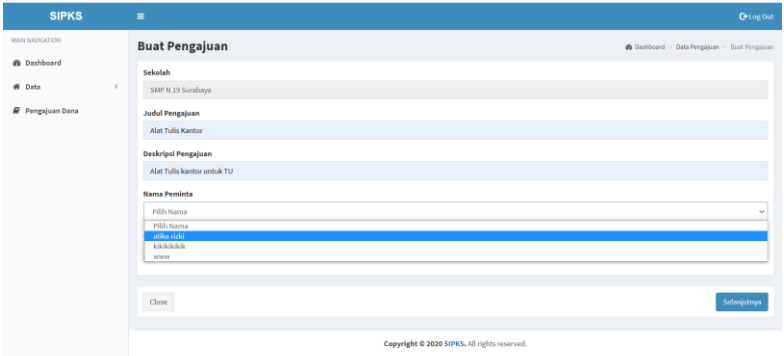
Halaman ini menampilkan data pengajuan yang pernah ditambahkan oleh aktor. Halaman ini digunakan pada kasus penggunaan WU-002 yaitu melihat pengajuan dana anggaran. Implementasi antarmuka halaman list data pengajuan dapat dilihat pada Gambar 4.3.



Gambar 4.3. Implementasi Antarmuka Halaman Data Pengajuan

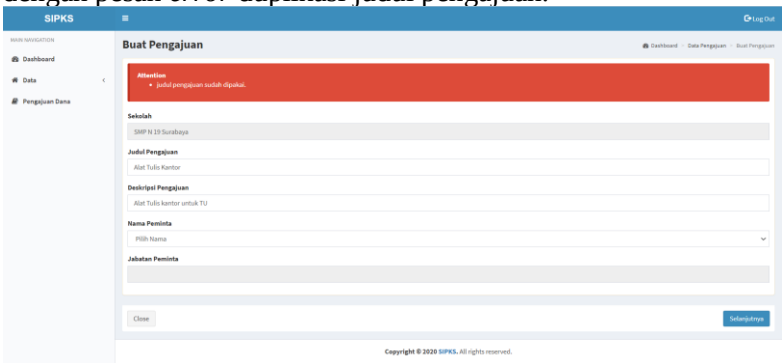
4.2.3. Implementasi Antarmuka Halaman Tambah Pengajuan

Halaman ini menampilkan form tambah pengajuan. Terdapat *drop-down* untuk mempermudah pengisian form. Halaman ini digunakan pada kasus penggunaan WU-001 yaitu menambahkan data pengajuan. Implementasi antarmuka halaman tambah pengajuan dapat dilihat pada Gambar 4.4 dan Gambar 4.5.



Gambar 4.4. Implementasi Antarmuka Halaman Tambah Pengajuan

Berikut implementasi dari halaman tambah pengajuan dengan pesan *error* duplikasi judul pengajuan.

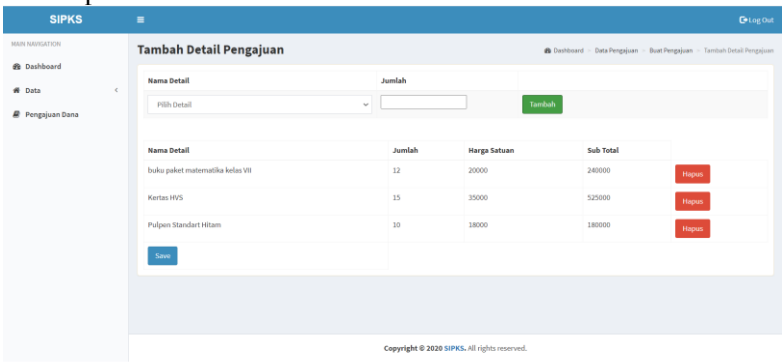


Gambar 4.5. Implementasi Antarmuka Halaman Tambah Pengajuan *Error*

4.2.4. Implementasi Antarmuka Halaman Tambah Detail Pengajuan

Halaman ini digunakan pada kasus penggunaan WU-001 yaitu menambahkan data pengajuan. Halaman ini muncul setelah pengajuan berhasil dibuat. Terdapat *drop-down* dan input *field* dinamis yang dapat disesuaikan dengan kebutuhan aktor.

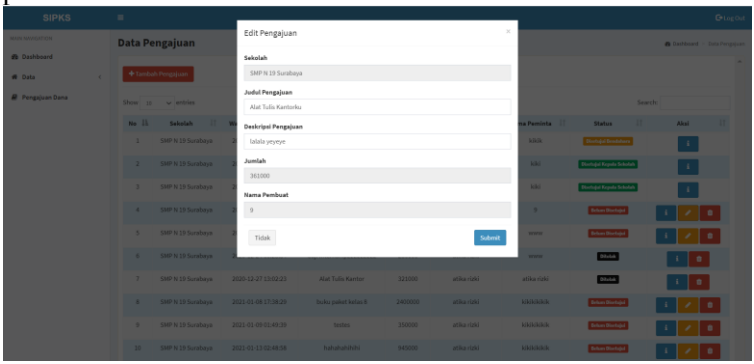
Implementasi antarmuka halaman tambah detail pengajuan dapat dilihat pada Gambar 4.6.



Gambar 4.6. Implementasi Antarmuka Halaman Tambah Detail Pengajuan

4.2.5. Implementasi Antarmuka Halaman Edit Pengajuan

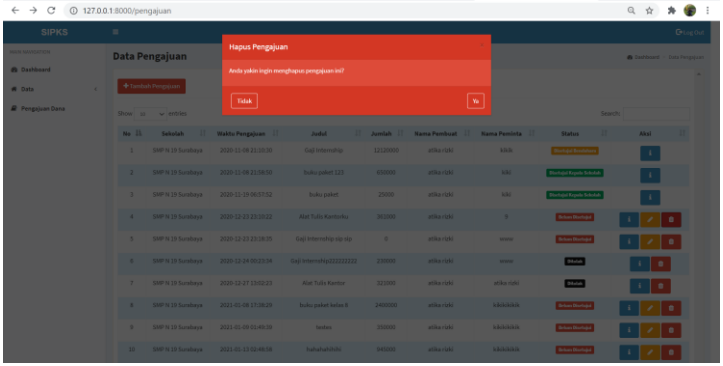
Halaman ini digunakan pada kasus penggunaan WU-003 yaitu mengubah data pengajuan. Halaman ini muncul setelah tombol edit pengajuan diklik. Terdapat modal berisi form edit pengajuan. Implementasi antarmuka halaman edit pengajuan dapat dilihat pada Gambar 4.7.



Gambar 4.7. Implementasi Antarmuka Halaman Edit Pengajuan

4.2.6. Implementasi Antarmuka Halaman Hapus Pengajuan

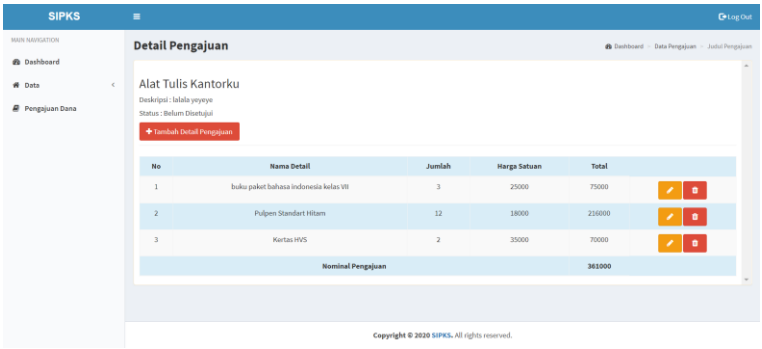
Halaman ini digunakan pada kasus penggunaan WU-004 yaitu menghapus data pengajuan. Halaman ini menampilakan modal konfirmasi hapus pengajuan. Implementasi antarmuka halaman hapus pengajuan dapat dilihat pada Gambar 4.8.



Gambar 4.8. Implementasi Antarmuka Halaman Hapus Pengajuan

4.2.7. Implementasi Antarmuka Halaman List Detail Pengajuan

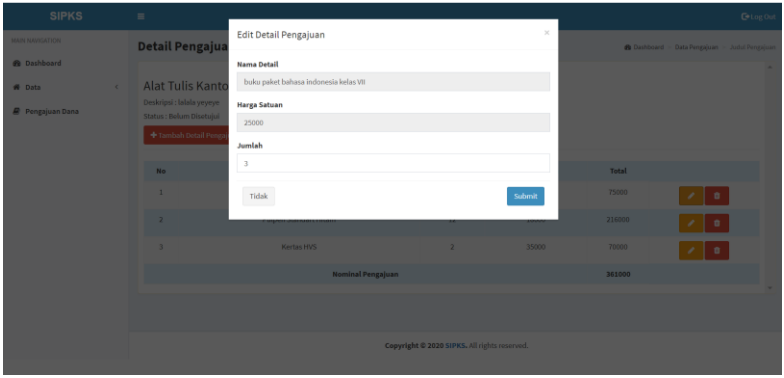
Halaman ini digunakan dalam kasus penggunaan WU-005 yaitu melihat detail pengajuan dana anggaran. Implementasi antarmuka halaman List detail pengajuan dapat dilihat pada Gambar 4.9.



Gambar 4.9. Implementasi Antarmuka Halaman List Detail Pengajuan

4.2.8. Implementasi Antarmuka Halaman Edit Detail Pengajuan

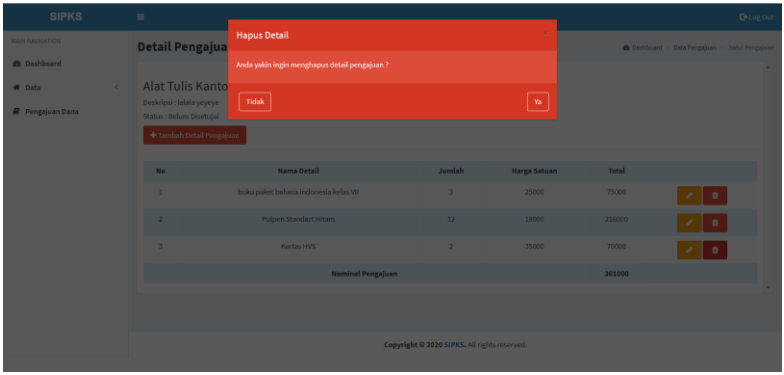
Halaman ini digunakan dalam kasus penggunaan WU-006 yaitu mengubah detail pengajuan dana anggaran. Implementasi antarmuka halaman edit detail pengajuan dapat dilihat pada Gambar 4.10.



Gambar 4.10. Implementasi Antarmuka Halaman Edit Detail Pengajuan

4.2.9. Implementasi Antarmuka Halaman Hapus Detail Pengajuan

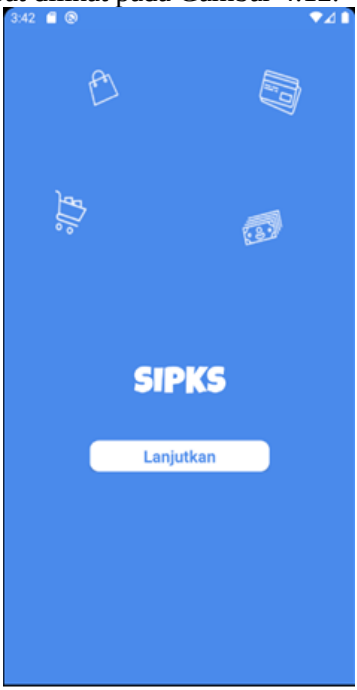
Halaman ini digunakan dalam kasus penggunaan WU-007 yaitu menghapus detail pengajuan dana anggaran. Implementasi antarmuka halaman hapus detail pengajuan dapat dilihat pada Gambar 4.11.



Gambar 4.11. Implementasi Antarmuka Halaman Hapus Detail Pengajuan

4.2.10. Implementasi Antarmuka Halaman *Start Page* Android

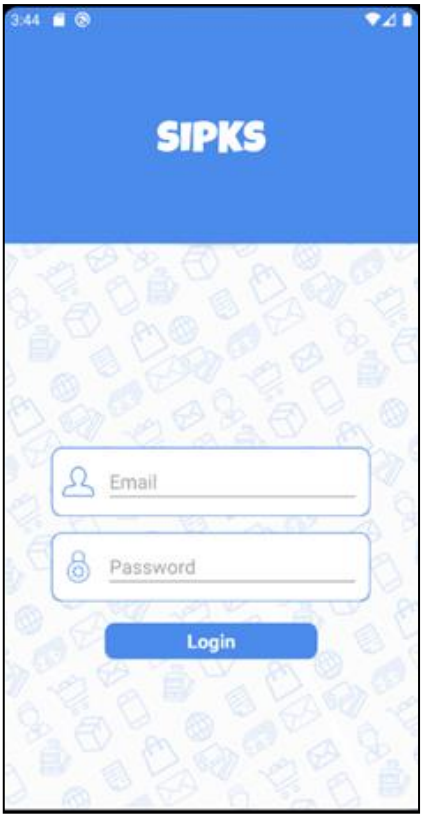
Halaman ini merupakan *start page* untuk aplikasi android SIPKS. Halaman ini muncul saat aktor pertama kali membuka aplikasi android SIPKS. Implementasi antarmuka halaman *Start Page* Android dapat dilihat pada Gambar 4.12.



Gambar 4.12. Implementasi Antarmuka *Start Page* Android

4.2.11. Implementasi Antarmuka Halaman *Login* Android

Halaman ini muncul setelah tombol Lanjutkan pada *Start Page* android diklik. Pada halaman ini terdapat form login dengan *field* email dan password serta tombol Login. Implementasi halaman antarmuka *login* android dapat dilihat pada Gambar4.13.



Gambar 4.13. Implementasi Antarmuka *Login* Android

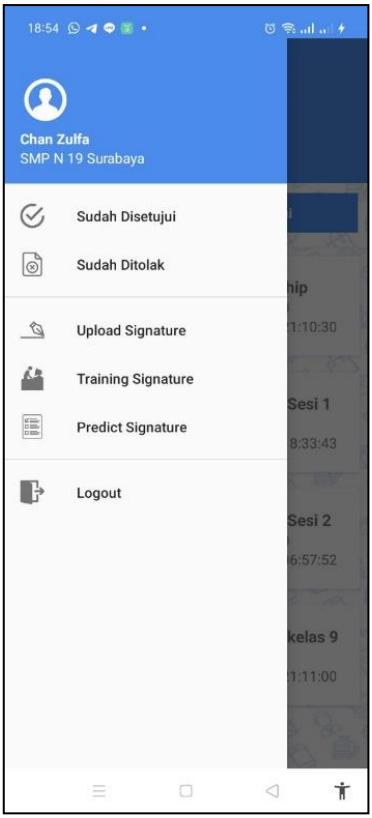
4.2.12. Implementasi Antarmuka Halaman Dashboard Android

Halaman ini muncul ketika aktor berhasil melakukan *login*. Pada halaman ini terdapat nama, jabatan, dan sekolah aktor. Selain itu juga terdapat *list view* dari pengajuan yang perlu disetujui oleh aktor dan tombol untuk menampilkan menu sidebar. Halaman ini digunakan dalam kasus penggunaan AU-001 yaitu melihat pengajuan anggaran dana (*mobile).* Implementasi antarmuka halaman dashboard android dapat dililhat pada Gambar 4.14.

Gambar 4.14. Implementasi Antarmuka *Dashboard* Android

4.2.13. Implementasi Antarmuka Halaman Menu Sidebar

Halaman ini muncul ketika tombol sidebar pada halaman dashboard di klik. Halaman ini berisi menu yang dapat digunakan aktor. Selain itu juga terdapat nama dan sekolah aktor. Implementasi antarmuka halaman Menu Sidebar dapat dilihat pada Gambar 4.15.



Gambar 4.15. Implementasi Antarmuka Menu *Sidebar* Android

4.2.14. Implementasi Antarmuka Halaman Pengajuan Belum Disetujui

Halaman ini menampilkan detail dari pengajuan yang memiliki status belum disetujui. Pada halaman ini terdapat judul, waktu, nominal, dan detail penggunaan dana anggaran pengajuan yang dipilih. Selain itu juga terdapat tombol untuk memberikan persetujuan dan tombol untuk memberikan penolakan terhadap pengajuan. Halaman ini digunakan dalam kasus penggunaan AU-004 yaitu memberikan persetujuan pengajuan dana anggaran dan

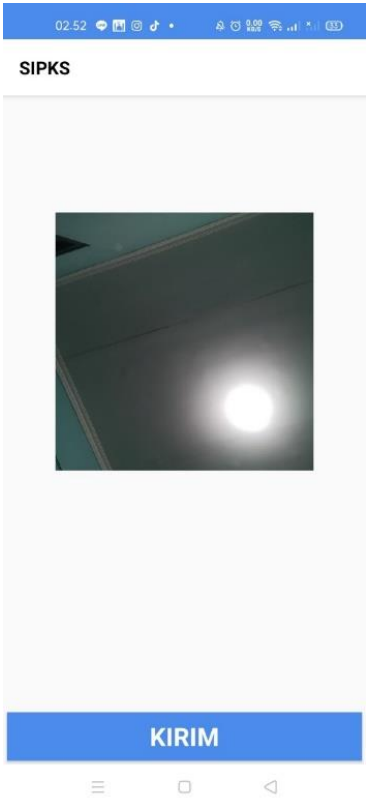
AU-005 yaitu memberikan penolakan pengajuan dana anggaran. Implementasi antarmuka halaman Pengajuan Belum Disetujui dapat dilihat pada Gambar 4.16.

Gambar 4.16. Implementasi Antarmuka *Start Page* Android

4.2.15. Implementasi Antarmuka Halaman Upload Wajah

Pada halaman ini terdapat *camera view* yang digunakan untuk menangkap foto wajah aktor. Selain itu, terdapat tombol kirim untuk mengirim gambar ke server. Halaman ini digunakan pada kasus penggunaan AU-002 mengunggah sampel wajah dan AU-004 yaitu memberikan persetujuan pengajuan dana anggaran

dengan pengenalan wajah. Implementasi antarmuka halaman upload wajah dapat dilihat pada Gambar 4.17.



Gambar 4.17. Implementasi Antarmuka Halaman Upload Wajah

4.2.16. Implementasi Antarmuka Halaman *Training* Wajah

Pada halaman ini terdapat tombol train. Selain itu juga terdapat teks hint berisi petunjuk dari proses *training* yang akan berjalan setelah tombol *Train* diklik. Halaman ini digunakan pada kasus penggunaan AU-003 yaitu melakukan *training* sampel

wajah. Implementasi antarmuka halaman *training* wajah dapat dilihat pada Gambar 4.18.

Gambar 4.18. Implementasi Antarmuka Halaman *Training* Wajah

4.2.17. Rancangan Antarmuka Halaman List Pengajuan Sudah Disetujui

Halaman ini menampilkan daftar pengajuan yang memiliki status sudah disetujui. Halaman ini digunakan dalam kasus penggunaan AU-001 yaitu melihat pengajuan dana anggaran.

Implementasi halaman list pengajuan sudah disetujui dapat dilihat pada Gambar 4.19.

Gambar 4.19. Implementasi Antarmuka Halaman List Pengajuan Sudah Disetujui

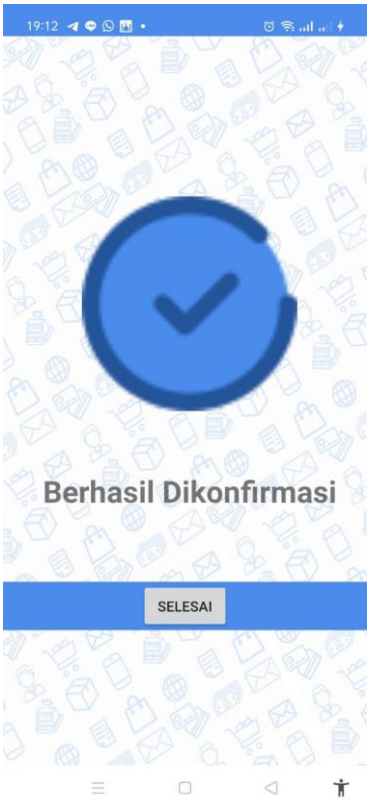
4.2.18. Implementasi Antarmuka Halaman List Pengajuan Ditolak

Halaman ini menampilkan daftar pengajuan yang memiliki status ditolak. Halaman ini digunakan dalam kasus penggunaan AU-001 yaitu melihat pengajuan dana anggaran. Implementasi halaman list pengajuan ditolak dapat dilihat pada Gambar 4.20.

Gambar 4.20. Implementasi Antarmuka Halaman List Pengajuan Ditolak

4.2.19. Implementasi Antarmuka Halaman Berhasil Disetujui

Halaman ini muncul ketika pengajuan berhasil disetujui, artinya wajah aktor berhasil dikenali sebagai aktor itu sendiri. Pada halaman ini terdapat gambar dan teks yang memberikan info bahwa pengajuan berhasil dikonfirmasi. Selain itu juga terdapat tombol selesai yang jika diklik akan Kembali ke halaman dashboard android. Implementasi antarmuka halaman berhasil disetujui dapat dilihat pada Gambar 4.21.



Gambar 4.21. Implementasi Antarmuka Halaman Berhasil Disetujui

4.3. Implementasi Kasus Penggunaan Aplikasi

Implementasi kasus penggunaan aplikasi berbasis web dibangun menggunakan Bahasa pemrograman PHP dengan kerangka kerja Laravel. Sementara implementasi kasus penggunaan berbasis *mobile* dibangun menggunakan Bahasa pemrograman Java yang dibangun pada lingkungan pengembangan Android Studio. Disisi lain pengembangan memanfaatkan web service untuk mengirimkan string request dan aplikasi akan menerima JSON post response. Pada subbab

ini akan menjelaskan dan menampilkan kode sumber pada sisi aplikasi pada beberapa kasus penggunaan.

4.3.1. Implementasi Kasus Penggunaan Menambahkan Pengajuan Dana Anggaran

Kasus penggunaan ini digunakan aktor Pegawai Tata usaha yang berhasil *login* ke aplikasi website untuk menambahkan pengajuan baru beserta detailnya. Halaman kasus penggunaan ini menampilkan form tambah pengajuan dilanjutkan dengan halaman form tambah detail pengajuan. Implementasi kasus penggunaan menambahkan pengajuan dana anggaran dapat dilihat pada kode sumber 4.1 dan kode sumber 4.2 serta penjelasan dari implementasi dapat dilihat pada table 4.2 dan table 4.3.

```
1   public function postPengajuan(Request $request){
2               $this->validate($request,[
3                   'judul_pengajuan' => 'unique:pengajuan'
4               ]);
5
6                $newPengajuan = new Pengajuan();
7                $newPengajuan->id_sekolah = $request->id_sekolah
                     ;
8                $newPengajuan->id_akun = Auth::user()->id;
9                $newPengajuan->judul_pengajuan = $request
                     ->judul_pengajuan;
10               $newPengajuan->deskripsi_pengajuan = $request
                     ->deskripsi_pengajuan;
11               $newPengajuan->jumlah_pengajuan = 0  ;
12               $newPengajuan->nama_pembuat_pengajuan = $request
                     ->nama_pembuat_pengajuan;
13               $newPengajuan->status_pengajuan = 0;
14              $status = $newPengajuan->save();
15              if($status){
16                  return redirect()->route('tambahDetail'
                         ,[$newPengajuan->id_pengajuan]);
17              }
18              else{
19                  return redirect('/pengajuan/tambah')->with
                         ('error','Judul Pengajuan sudah dipakai'
                         );
20              }
21      }
```

Kode sumber 4.1 Fungsi Menambahkan Pengajuan Dana Anggaran

Table 4.2. Penjelasan Fungsi Menambahkan Pengajuan Dana Anggaran

No Baris	Kegunaan
2-4	Memvalidasi judul pengajuan harus unique (tidak boleh sama)
6-13	Mengambil data pengajuan melalui data input yang dituliskan user melalui form tambah pengajuan
14	Menyimpan data ke database table pengajuan
15-20	Meredirect ke halaman tambah detail pengajuan jika pengajuan berhasil dibuat dan jika tidak mengeluarkan pesan error

Table 4.3. Penjelasan Fungsi Menambahkan Detail Pengajuan Dana Anggaran

No Baris	Kegunaan
2-14	Mengambil data detail pengajuan melalui data input yang dituliskan user melalui form tambah detail pengajuan
16	Menyimpan data ke database table mapping detail pengajuan
19-24	Melakukan *update* jumlah pengajuan pada table pengajuan
25	*Redirect* ke halaman detail pengajuan

```
1   public function postDetail(Request $request){
2               $id = $request->id_pengajuan;
3               $arr = $request->arr;
4               $val = [];
5               $sql = [];
6               $jumlah = 0;
7                $arr_split = array_chunk($arr,3);
8               for($i=0;$i<count($arr_split);$i++){
9                   array_unshift($arr_split[$i], $id);
10                  $detail = new MappingDetailPengajuan();
11                  $detail->id_pengajuan = $arr_split[$i][0];
12                  $detail->id_detail = $arr_split[$i][1];
13                  $detail->jumlah_detail = $arr_split[$i][2];
14                  $detail->sub_total = $arr_split[$i][3];
15
16                  $detail->save();
17                  $jumlah+=$detail->sub_total;
18              }
19              $pengajuan = Pengajuan::find($id);
20
21              if($pengajuan) {
22                  $pengajuan->jumlah_pengajuan += $jumlah;
23                  $pengajuan->save();
24              }
25              return redirect()->route('detailawal', [$request
                    ->id_pengajuan]);
26          }
27
```

Kode sumber 4.2 Fungsi Menambahkan Detail Pengajuan Dana Anggaran

4.3.2. Implementasi Kasus Penggunaan Melihat Pengajuan Dana Anggaran (Website)

Kasus penggunaan ini digunakan aktor Pegawai Tata usaha yang berhasil *login* ke aplikasi website. Halaman kasus penggunaan ini menampilkan seluruh data pengajuan yang pernah ditambahakan oleh aktor. Implementasi kasus penggunaan melihat pengajuan dana anggaran (website) dapat dilihat pada kode sumber 4.3 dan penjelasan dari implementasi dapat dilihat pada table 4.4.

Table 4.4. Penjelasan Fungsi Melihat Pengajuan Dana Anggaran (Website)

No Baris	Kegunaan

2-7	Mengambil semua data pada table pengajuan, user, dan sekolah yang dimiliki auth user
8	Memanggil view untuk menampilkan data-data yang sudah disimpan

```
1  public function awal(){
2          $data['pengajuans'] = DB::table('Pengajuan')
3                                          ->join('Sekolah',
                                   'sekolah.id_sekolah', '=',
                                   'pengajuan.id_sekolah')
4                                          ->join('Users', 'Users
                                   .id','=','pengajuan.id_akun')
5                                          ->where('Pengajuan
                                   .id_sekolah','=', Auth::user
                                   ()->id_sekolah)
6                                          ->get();
7          return view('directory.pengajuan', $data);
8      }
9
```

Kode sumber 4.3 Fungsi Melihat Pengajuan Dana Anggaran (Website)

4.3.3. Implementasi Kasus Penggunaan Mengubah Pengajuan Dana Anggaran

Kasus penggunaan ini digunakan aktor Pegawai Tata usaha yang berhasil login ke aplikasi website dan ingin mengubah info pada pengajuan. Halaman kasus penggunaan ini menampilkan modal edit data pengajuan. Implementasi kasus penggunaan mengubah pengajuan dana anggaran dapat dilihat pada kode sumber 4.4 dan penjelasan dari implementasi dapat dilihat pada table 4.5.

Table 4.5. Penjelasan Fungsi Mengubah Pengajuan Dana Anggaran

No Baris	Kegunaan
2	Mengambil data pengajuan dari table pengajuan berdasarkan id pengajuan

3-4	Mengambil hasil edit melalui form pengajuan
5	Menyimpan hasil edit ke database table pengajuan
6	*Redirect* ke halaman pengajuan

```
1    public function editPengajuan(Request $request, $id){
2        $updatePengajuan = Pengajuan::find($id);
3        $updatePengajuan->judul_pengajuan = $request
             ->judul_pengajuan;
4        $updatePengajuan->deskripsi_pengajuan = $request
             ->deskripsi_pengajuan;
5        $updatePengajuan->save();
6        return redirect('/pengajuan');
7     }
```

Kode sumber 4.4 Fungsi Mengubah Pengajuan Dana Anggaran

4.3.4. Implementasi Kasus Penggunaan Menghapus Pengajuan Dana Anggaran

Kasus penggunaan ini digunakan aktor Pegawai Tata usaha yang berhasil login ke aplikasi website dan ingin menghapus pengajuan. Halaman kasus penggunaan ini menampilkan modal konfirmasi hapus pengajuan. Implementasi kasus penggunaan menghapus pengajuan dana anggaran dapat dilihat pada kode sumber 4.5 dan penjelasan dari implementasi dapat dilihat pada table 4.6.

```
1   public function hapusPengajuan($id){
2        MappingDetailPengajuan::where('id_pengajuan',$id
             )->delete();
3        Pengajuan::destroy($id);
4        return redirect('/pengajuan');
5     }
```

Kode sumber 4.5 Fungsi Menghapus Pengajuan Dana Anggaran

Table 4.6. Penjelasan Fungsi Menghapus Pengajuan Dana Anggaran

No Baris	Kegunaan
2	Menghapus detail pengajuan dari table mapping detail pengajuan sesuai dengan id pengajuan yang dihapus
3	Menghapus pengajuan dari table pengajuan sesuai dengan id pengajuan yang dihapus
4	*Redirect* ke halaman pengajuan

4.3.5. Implementasi Kasus Penggunaan Melihat Detail Pengajuan Dana Anggaran

Kasus penggunaan ini digunakan aktor Pegawai Tata usaha yang berhasil login ke aplikasi website dan akan melihat detail pengajuan dari pengajuan yang dipilih. Halaman kasus penggunaan ini menampilkan daftar detail penggunaan dana anggaran dari pengajuan yang dipilih. Implementasi kasus penggunaan melihat detail pengajuan dana anggaran dapat dilihat pada kode sumber 4.6 dan penjelasan dari implementasi dapat dilihat pada table 4.7.

Table 4.7. Penjelasan Fungsi Melihat Detail Pengajuan Dana Anggaran

No Baris	Kegunaan
2	Mendapatkan data dari table detail pengajuan
3	Mendapatkan data pengajuan berdasarkan id pengajuan dari table pengajuan
4-8	Mengambil semua data pada table pengajuan, detail pengajuan, dan mapping detail pengajuan

10	Menampilkan hasil data ke *view*

```
1   public function detailPengajuan($id){
2         $data['details'] = DetailPengajuan::get();
3         $data['judul']= Pengajuan::where('id_pengajuan',$id
              )->get();
4         $data['detail_pengajuans'] = DB::table('Pengajuan')
5                                         ->join
                                    ('mapping_pengajuan_detail'
                                    ,'pengajuan.id_pengajuan','=',
                                    'mapping_pengajuan_detail
                                    .id_pengajuan')
6                                         ->join('detail_pengajuan',
                                    'detail_pengajuan.id_detail', '
                                    =', 'mapping_pengajuan_detail
                                    .id_detail')
7                                         ->where
                                    ('mapping_pengajuan_detail
                                    .id_pengajuan',$id)
8                                         ->get();
9         //  dd($data);
10         return view('directory.detailPengajuan', $data);
11    }
```

Kode sumber 4.6 Fungsi Melihat Detail Pengajuan Dana Anggaran

4.3.6. Implementasi Kasus Penggunaan Mengubah Detail Pengajuan Dana Anggaran

Kasus penggunaan ini digunakan aktor Pegawai Tata usaha yang berhasil login ke aplikasi website dan ingin mengubah info pada detail pengajuan. Halaman kasus penggunaan ini menampilkan modal edit data detail pengajuan. Implementasi kasus penggunaan mengubah detail pengajuan dana anggaran dapat dilihat pada kode sumber 4.7 dan penjelasan dari implementasi dapat dilihat pada table 4.8.

```
1  public function editdetailPengajuan(Request $request){
2          $updateDetail = MappingDetailPengajuan::find($request
               ->id_mapping_pengajuan_detail);
3          $updateDetail->jumlah_detail = $request->jumlah_detail
               ;
4          $updateDetail->sub_total = $request->jumlah_detail
               *$request->harga_satuan_detail;
5          $updateDetail->update();
6          $nominal=0;
7          $data['nominal_pengajuan'] = MappingDetailPengajuan
               ::where('id_pengajuan',$request->id_pengajuan
               )->select('sub_total')->get();
8          foreach($data['nominal_pengajuan'] as $item){
9              $nominal+=$item->sub_total;
10         }
11         $pengajuan = Pengajuan::find($request->id_pengajuan);
12         $pengajuan->jumlah_pengajuan = $nominal;
13         $pengajuan->save();
14         return redirect()->route('detailawal', [$request
               ->id_pengajuan]);
15     }
16
```

Kode sumber 4.7 Fungsi Mengubah Detail Pengajuan Dana Anggaran

Table 4.8. Penjelasan Fungsi Mengubah Pengajuan Dana Anggaran

No Baris	Kegunaan
2	Mengambil data mapping detail pengajuan dari table meping detail pengajuan berdasarkan id nya
3-4	Mengambil hasil edit melalui form edit detail pengajuan
5	Menyimpan hasil edit ke database table mapping detail pengajuan
6-13	Melakukan *update* nominal pengajuan pada table pengajuan sesuai dengan id pengajuan
14	*Redirect* ke halaman detail pengajuan

4.3.7. Implementasi Kasus Penggunaan Menghapus Detail Pengajuan Dana Anggaran

Kasus penggunaan ini digunakan aktor Pegawai Tata usaha yang berhasil login ke aplikasi website dan ingin menghapus pengajuan. Halaman kasus penggunaan ini menampilkan modal konfirmasi hapus pengajuan. Implementasi kasus penggunaan menghapus pengajuan dana anggaran dapat dilihat pada kode sumber 4.8 dan penjelasan dari implementasi dapat dilihat pada table 4.9.

```
1   public function hapusdetailPengajuan($id_pengajuan
        ,$id_mapping_pengajuan_detail){
2       $mapping_detail = MappingDetailPengajuan::find
            ($id_mapping_pengajuan_detail);
3       $sub_total = $mapping_detail->sub_total;
4       $pengajuan = Pengajuan::find($id_pengajuan);
5           if($pengajuan) {
6               $pengajuan->jumlah_pengajuan -= $sub_total;
7               $pengajuan->save();
8           }
9       $mapping_detail->destroy($id_mapping_pengajuan_detail
            );
10      return redirect()->route('detailawal', [$id_pengajuan]
            );
11  }
```

Kode sumber 4.8 Fungsi Menghapus Detail Pengajuan Dana Anggaran

Table 4.9. Penjelasan Fungsi Menghapus Detail Pengajuan Dana Anggaran

No Baris	Kegunaan
2	Mengambil data detail pengajuan dari table mapping detail pengajuan berdasarkan id pengajuan
4	Mengambil data pengajuan dari tabel pengajuan berdasarkan id pengajuan

3, 6-7	Mengupdate jumlah pengajuan sesuai dengan id pengajuan
9	Menghapus detail dari table mapping pengajuan detail
10	*Redirect* ke halaman detail pengajuan

4.3.8. Implementasi Kasus Penggunaan Melihat Pengajuan Dana Anggaran (Mobile)

Kasus penggunaan ini digunakan aktor kepala sekolah dan bendahara sekolah yang berhasil *login* ke aplikasi android. Halaman kasus penggunaan ini menampilkan seluruh data pengajuan dengan id sekolah yang sama dengan id sekolah aktor. Implementasi kasus penggunaan melihat pengajuan dana anggaran (android) dapat dilihat pada kode sumber 4.9 dan penjelasan dari implementasi dapat dilihat pada table 4.10.

```
1   private void getList() {
2
3   final Call <ResponsePengajuan> call = api.getpengajuan(0);
4   call.enqueue(new Callback
5       <ResponsePengajuan>() {
6
7   @Override
8   public void onResponse(Call
9           <ResponsePengajuan> call, Response
10              <ResponsePengajuan> response) {
11  if(response.isSuccessful()){
12  ResponsePengajuan get = response.body();
13  adapter = new ListAdapter(MainActivity.this,get.getPengajuan())
14  recyclerView.setAdapter(adapter);
15  return;
16  }
17
18  @Override
19  public void onFailure(Call
20                  <ResponsePengajuan> call,Throwable t){
21  System.out.println("failed" + t.toString()); }});
22  }
23
```

Kode sumber 4.9 Fungsi Melihat Pengajuan Dana Anggaran (Mobile)

Table 4.10. Penjelasan Fungsi Melihat Pengajuan Dana Anggaran (Mobile)

No Baris	Kegunaan
3	Memanggil Response yang telah dibuat dan memanggil data dari database
5-16	Menampilkan data pengajuan jika program berhasil
19-21	Menampilkan string ‘failed’ jika program gagal menampilkan pengajuan

4.3.9. Implementasi Kasus Penggunaan Mengunggah Sampel Wajah

Kasus penggunaan ini digunakan aktor kepala sekolah dan bendahara sekolah yang berhasil *login* ke aplikasi android. Aktor mengunggah sampel wajah sebanyak 5 kali. Sampel wajah kemudian dikirim ke server untuk disimpan. Sebelum disimpan, akan dialkukan pengecekan apakah foto yang diunggah benar wajah atau bukan wajah. Implementasi kasus penggunaan mengunggah sampel wajah dapat dilihat pada kode sumber 4.10 dan penjelasan dari implementasi dapat dilihat pada table 4.11.

```
1  public function sendFace(Request $request)
2      {
3          $base64_string = $request->file('image');
4          $id_usr = Auth::user()->id;
5          $image_name = 'uploadFace/'.$id_usr;
6
7          if (!file_exists($image_name)) {
8           if (!mkdir($image_name)) {
9              $m=array('msg' => "REJECTED, cant create folder"
                   );
10             echo json_encode($m);
11             return;}
12         }
13
14         if ($files !== false) {
15                 $filecount = count($files);
16         }
17         if($filecount==5){
18             echo json_encode(array('msg' => "Upload Wajah
                   Selesai, Data Tersimpan"));
19             return;
20         }
21
22         $fullName = $id_usr."_".$filecount."_". date
               ("YmdHis") ".".png";
23         $ifp = $base64_string->move($image_name,$fullName);
24         if (!$ifp){
25             $m=array('msg' => "REJECTED, ".$fullName."not
                   saved");
26             echo json_encode($m);
27             return;
28         }
29
30         $command = escapeshellcmd("python ".public_path
               ("code/checkFace.py")." ". public_path
               ("uploadFace/".$id_usr."/".$fullName));
31         $output = shell_exec($command);
```

Kode sumber 4.10 Fungsi Mengunggah Sampel Wajah

Table 4.11. Penjelasan Fungsi Mengunggah Sampel Wajah

No Baris	Kegunaan
3	Menyimpan *request image* ke dalam variabel
4	Menyimpan *auth id* milik user ke dalam variabel

5	Deklarasi path direktori untuk menampung *image* hasil *upload*
7-12	Proses pengecekan adanya direktori image yang sudah dideklarasikan sebelumnya
14-20	Menghitung jumlah file yang diupload dalam satu direktori yang berhasil dibuat sebelumnya
22-27	Membuat nama foto setelah diupload dan memindahkan foto ke dalam direktori yang berhasil dibuat sebelumnya
30	Melakukan pemanggilan kode sumber untuk melakukan pengecekan foto yang diupload berupa wajah atau bukan
31	Mengembalikan output dari pemanggilan kode sumber pengecekan wajah

4.3.10. Implementasi Kasus Penggunaan Melakukan *Training* Sampel Wajah

Kasus penggunaan ini digunakan aktor kepala sekolah dan bendahara sekolah yang berhasil *login* ke aplikasi android. Aktor melakukan *training* foto wajah yang sudah berhasil diunggah ke server. Implementasi kasus penggunaan melakukan *training* sampel wajah dapat dilihat pada kode sumber 4.11 dan penjelasan dari implementasi dapat dilihat pada table 4.12.

Table 4.12. Penjelasan Fungsi Melakukan *Training* Sampel Wajah

No Baris	Kegunaan
2	Menyimpan auth id milik user kedalam variable
3	Deklarasi path direktori berisi menampung *image* yang akan di*training*

5-9	Proses pengecekan adanya direktori image yang sudah dideklarasikan sebelumnya
18-23	Memanggil kode sumber untuk melakukan proses *training*
24-29	Mengembalikan output dari pemanggilan kode sumber melakukan proses *training*

```
1   public function trainGAN(){
2           $id_usr = Auth::user()->id;
3           $image_name = 'uploadFace/'.$id_usr;
4
5           if (!file_exists($image_name))
6           {
7               $m = array('msg' => "REJECTED,no data to
                    train");
8               echo json_encode($m);
9               return;
10          }
11
12          $files = File::files(public_path($image_name));
13            $filecount = 0;
14
15            if ($files !== false) {
16                $filecount = count($files);
17            }
18          $command = escapeshellcmd("python ".public_path
                ("code/gan/run_model.py")." --input_path=".
                public_path("uploadFace/".$id_usr."")."
                --output_path=".public_path("uploadFace/"
                .$id_usr.""));
19          set_time_limit(1000);
20          $output = shell_exec($command);
21          if($output='Done'){
22              $command = escapeshellcmd("python "
                    .public_path("code/doTrainMobileNet.py"
                    )." ". $id_usr);
23              $trainMN = shell_exec($command);
24              $m = array('msg' => $trainMN);
25          }
26          else{
27              $m = array('msg' => $output);
28          }
29          echo json_encode($m);
30      }
31
```

Kode sumber 4.11 Fungsi Melakukan *Training* Sampel Wajah

4.3.11. Implementasi Kasus Penggunaan Memberikan Persetujuan Pengajuan dengan Pengenalan Wajah

Kasus penggunaan ini digunakan aktor kepala sekolah dan bendahara sekolah yang berhasil *login* ke aplikasi android dan ingin menyetujui pengajuan. Aktor mengunggah foto wajah yang akan diprediksi dan dikenali oleh sistem menggunakan pengenalan wajah. Jika wajah berhasil dikenali dan benar, maka pengajuan berhasil disetujui. Jika gagal, maka pengajuan tidak dapat disetujui dan aktor harus mengulangi ambil gambar jika ingin menyetujui Kembali. Implementasi kasus penggunaan memberikan persetujuan dengan pengenalan wajah dapat dilihat pada kode sumber 4.12 dan penjelasan dari implementasi dapat dilihat pada table 4.13.

Table 4.13. Penjelasan Fungsi Memberikan Persetujuan Pengajuan dengan Pengenalan Wajah

No Baris	Kegunaan
3	Menyimpan *request image* ke dalam variabel
4	Menyimpan *auth id* milik user ke dalam variabel
5	Deklarasi path direktori untuk menampung *image* hasil *upload* untuk pengenalan wajah
8-12	Proses pengecekan adanya direktori image yang sudah dideklarasikan sebelumnya
16-20	Menghitung jumlah file dalam direktori yang sudah dideklarasikan sebelumnya
24-31	Membuat nama untuk foto yang di upload kemudian memindahkannya ke direktori yang sudah dideklarasikan sebelumnya
32-33	Memanggil kode sumber pengenalan wajah

34-37	Mengubah status pengajuan di database menjadi disetujui
39-40	Mengembalikan output dari pemanggilan kode sumber pengenalan wajah

```
1  public function predictFace(Request $request)
2      {
3          $base64_string = $request->file('image');
4
5          $id_usr = Auth::user()->id;
6          $image_name = 'predictFace/'.$id_usr;
7
8          if (!file_exists($image_name)) {
9           if (!mkdir($image_name)) {
10             $m=array('msg' => "REJECTED, cant create folder");
11             echo json_encode($m);
12             return;}
13         }
14
15
16         $files = File::files(public_path($image_name));
17           $filecount = 0;
18
19           if ($files !== false) {
20               $filecount = count($files)+1;
21           }
22
23
24         $fullName = $id_usr."_".$filecount."_". date("YmdHis") .".png";
25
26         $ifp = $base64_string->move($image_name,$fullName);
27
28         if (!$ifp){
29             $m=array('msg' => "REJECTED, ".$fullName."not saved");
30             echo json_encode($m);
31             return;}
32         $command = escapeshellcmd("python ".public_path("code/doPredict.py")." ".$id_usr. " ".public_path("predictFace/".$id_usr."/".$fullName));
33         $output = shell_exec($command);
34         if(strpos($output,"ACCEPTED") && $request->id != 0){
35             $pengajuan = Pengajuan::where("id_pengajuan", $request->id)->first();
36             $pengajuan->status_pengajuan = (Auth::user()->role_akun == "1")?"2":"1";
37             $pengajuan->save();
38         }
39         $m = array('msg' => $output);
40         echo json_encode($m);
41     }
```

Kode sumber 4.12 Fungsi Memberikan Persetujuan Pengajuan dengan Pengenalan Wajah

4.3.12. Implementasi Kasus Penggunaan Menolak Pengajuan Dana Anggaran

Kasus penggunaan ini terjadi Ketika Kepala Sekolah maupun Bendahara Sekolah mengklik tombol ‘Tolak’ pada pengajuan yang belum disetujui. Implementasi kasus penggunaan memberikan penolakan pengajuan dana anggaran dapat dilihat pada kode sumber 4.13 dan penjelasan dari implementasi dapat dilihat pada table 4.14.

```
1  tolakKonfirmasi.setOnClickListener(new View.OnClickListener(){
2
3  @Override
4
5  public void onClick(View v){
6  final SweetAlertDialog pDialog=newSweetAlertDialog(DetailPengajuanActivity.this,SweetAlertDialog.PROGRESS_TYPE);
7  pDialog.getProgressHelper().setBarColor(Color.parseColor("#A5DC86"));
8      pDialog.setTitleText("Loading");
9      pDialog.setCancelable(false);
10     pDialog.show();
11     ApiClientAttendance api = ServerAttendance.createServiceWithAuth(ApiClientAttendance.class,tokenManager);
12 Call
13 <OKResponse> call = api.postdetailpengajuan(id,"3");
14 call.enqueue(new Callback
15     <OKResponse>(){
16 @Override
17 public void onResponse(Call
18         <OKResponse> call, Response
19             <OKResponse> response) {
20 if(response.isSuccessful()){
21 pDialog.dismiss();
22 new SweetAlertDialog(DetailPengajuanActivity.this,SweetAlertDialog.SUCCESS_TYPE).setTitleText("Hasil")
23 .setContentText("Berhasil Ditolak")
24 .setConfirmText("OK").setConfirmClickListener(new SweetAlertDialog.OnSweetClickListener(){
25 @Override
26 public void onClick(SweetAlertDialog sDialog){
27 sDialog.dismissWithAnimation();}
28 }).show();
29 Intent intent = new Intent(DetailPengajuanActivity.this,TolakActivity.class);
30 intent.putExtra("id",id);
31 startActivity(intent);
32 }
33 }
34 @Override
35 public void onFailure(Call
36             <OKResponse> call, Throwable t){  }
37 });
38 }
39 });
40
```

Kode sumber 4.13 Fungsi Menolak Pengajuan Dana Anggaran

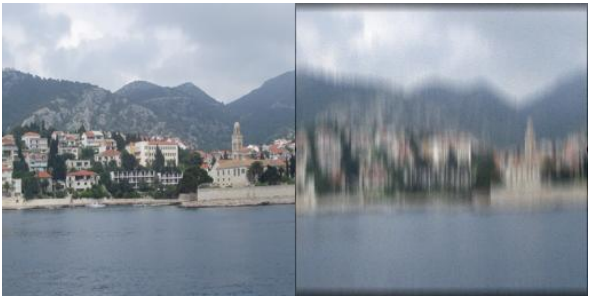
Table 4.14. Penjelasan Fungsi Menolak Pengajuan Dana Anggaran

No Baris	Kegunaan
6-10	Membuat alert dialog
11	Inisiasi `createService` untuk menyimpan `session auth user`

12-13	Memanggil Response yang telah dibuat dan memanggil data dari database
14-24	Memunculkan open dialog berhasil ditolak jika program berhasil
26-31	Mengintent ke activity lain dengan membawa id pengajuan yang ditolak
35	Fungsi yang dijalankan jika program gagal

4.4. Implementasi Proses Perbaikan Foto

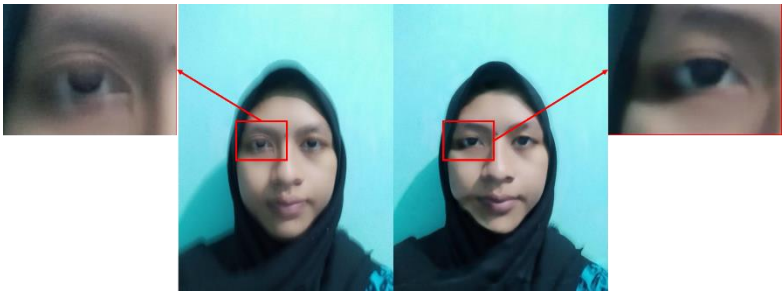
Proses perbaikan foto dilakukan penulis dengan menggunakan pengembangan *Generative Adversarial Networks* khusus untuk *deblurring image* yaitu *DeblurGAN*. Pada penelitian Kupyn Orest[14], *DeblurGAN* dapat memperbaiki kualitas foto blur menjadi lebih tajam. Penelitian ini menggunakan dataset GOPRO. *Preprocessing* yang dilakukan yaitu memisahkan foto menjadi dua kategori berdasarkan kualitas fokus, yaitu blur (kanan) dan sharp (kiri).



Gambar 4.22. Contoh Hasil *Preprocessing*

Selanjutnya dilakukakan pemanggilan kode sumber untuk *discriminator* dan *generator DeblurGAN*. Pada *generator*

digunakan metode ResNet Block untuk mempertajam gambar. Kode sumber tersebut dapat dilihat pada Lampiran Kode Sumber *DeblurGAN*. Dari pemanggilan kode sumber tadi akan mengupdate model yang ada yang nantinya akan digunakan pada proses testing data. Perlu diketahui bahwa pada pengerjaan tugas akhir ini penulis menggunakan model dari hasil penelitian sebelumnya tentang DeblurGAN[14] sehingga penulis tidak membuat model dari awal.



Gambar 4.23. Hasil Proses Perbaikan Foto

Pada gambar 4.23. dapat dilihat bahwa penerapan *DeblurGAN* dapat memperbaiki bagian wajah yang blur. Bagian mata yang terlihat berbayang menjadi fokus setelah diperbaiki. Fitur mata dapat dikenali dengan jelas.

4.5. Implementasi Proses Pengenalan Wajah

Pada implementasi proses pengenalan wajah digunakan *transfer learning* MobileNet. Implementasi dilakukan pada dua macam input, yaitu foto blur dan foto hasil perbaikan foto blur. Hasil pengenalan wajah dengan MobileNet dapat dilihat pada Gambar 4.24.

Sebelum diperbaiki (Score : 17,84)
REJECTED

Setelah diperbaiki (Score : 96,01)
ACCEPTED

Gambar 4.24. Hasil Pengenalan Wajah

Pada Gambar 4.24. dapat dilihat foto yang digunakan sebagai input pengenalan wajah. Sebelah kiri merupakan foto blur, sedangkan foto di sebelah kanan merupakan hasil perbaikan foto menggunakan *DeblurGAN*. Pengenalan wajah pada foto blur mendapati hasil *rejected* yang artinya tidak berhasil dikenali. Akurasi yang dimiliki juga rendah yaitu 17,84. Fitur wajah seperti mata, hidung dan mulut terlihat namun tidak jelas. Sedangkan pengenalan wajah pada foto hasil perbaikan memiliki hasil *accepted* yang berarti wajah berhasil dikenali dan cocok. Akurasi yang dimiliki juga terbilang mendekati 100% yaitu 96,01. Fitur wajah seperti mata, hidung, dan mulut terlihat jelas. Akan tetapi, terdapat kekurangan pada foto hasil perbaikan yaitu terdapat beberapa bagian wajah yang terlihat kurang halus dan tidak rata.

4.6. Penggunaan API *Endpoint*

Dalam aplikasi Android SIPKS, dibutuhkan API Endpoint agar platform aplikasi Android ini dapat berkomunikasi dengan aplikasi platfrom website.

4.6.1. Implementasi API *Endpoint Login*

API *Endpoint Login* digunakan untuk memberikan akses masuk Kepala Sekolah dan Bendahara Sekolah ke aplikasi android. Metode API *Endpoint Login* yang digunakan adalah POST. Kode

sumber implementasi untuk API *Endpoint Login* dapat dilihat pada kode sumber 4.14.

```
1   public function login(Request $request){
2             $request->validate([
3               'email' => 'required',
4               'password' => 'required'
5             ]);
6             $params = [
7               'grant_type' => 'password',
8               'client_id' => $this->client->id,
9               'client_secret' => $this->client->secret,
10              'username' => $request->email,
11              'password' => $request->password,
12              'scope' => '*'
13            ];
14            $request->request->add($params);
15            $proxy = Request::create('oauth/token','POST');
16            return Route::dispatch($proxy);
17          }
```

Kode sumber 4.14 Implementasi API *Endpoint Login* dengan metode POST

4.6.2. Implementasi API *Endpoint Profile*

API *Endpoint Profile* digunakan untuk mendapatkan data akun Kepala Sekolah dan Bendahara Sekolah yang berhasil *login* ke aplikasi *mobile*. Metode API *Endpoint Profile* yang digunakan adalah GET. Kode sumber implementasi untuk API *Endpoint Profile* dapat dilihat pada kode sumber 4.15.

```
1   public function getprofile () {
2             $data['profile'] = DB::table('users')->join
                  ('sekolah', function($join)
3             {
4               $join->on('users.id_sekolah', '=', 'sekolah
                    .id_sekolah')
5                     ->where('users.id', Auth::user()->id);
6             })->get();
7             return json_encode($data);
8           }
```

Kode sumber 4.15 Implementasi API *Endpoint Profile* dengan metode GET

4.6.3. Implementasi API *Endpoint* Pengajuan

API *Endpoint* Pengajuan digunakan untuk mendapatkan data pengajuan sesuai dengan sekolah pada data akun Kepala Sekolah dan Bendahara Sekolah yang berhasil *login* ke aplikasi *mobile*. Metode API *Endpoint* Pengajuan yang digunakan adalah GET. Kode sumber implementasi untuk API *Endpoint* Pengajuan dapat dilihat pada kode sumber 4.16.

```
1   public function getpengajuan (Request $request){
2        $data['pengajuan'] = Auth::user();
3       if(Auth::user()->role_akun=='1')
4       {
5           if($request->status==0)
6           {
7               $data['pengajuan'] = DB::table('Pengajuan'
                    )->join('users', function ($join)
8               {
9                   $join->on('Pengajuan.id_sekolah', '=',
                        'Users.id_sekolah')
10                      ->where ('users.id', Auth::user()->id
                            );
11              })->where('status_pengajuan','0')->orWhere
                    ('status_pengajuan','1')->get();
12
13          }
14          else if($request->status==3){
15           $data['pengajuan'] = DB::table('Pengajuan')->join
                ('users', function ($join)
16               {
17                  $join->on('Pengajuan.id_sekolah', '=',
                        'Users.id_sekolah')
18                      ->where ('users.id', Auth::user
                            ()->id);
19
20              })->where('status_pengajuan','3')->get();
21              }
22              else{
23                  $data['pengajuan'] = DB::table
                        ('Pengajuan')->join('users', function
                        ($join)
24                  {
25                      $join->on('Pengajuan.id_sekolah', '=',
                            'Users.id_sekolah')
26                          ->where ('users.id', Auth::user
                                ()->id);
```

```
27
28                  })->where('status_pengajuan','2')->get();
29              }
30          }
31          else
32          {
33              if($request->status==0)
34              {
35                  $data['pengajuan'] = DB::table('Pengajuan'
                        )->join('users', function ($join)
36                  {
37                      $join->on('Pengajuan.id_sekolah', '=',
                            'Users.id_sekolah')
38                          ->where ('users.id', Auth::user
                                ()->id);
39
40                  })->where('status_pengajuan','0')->get();
41              }
42              else if($request->status==3)
43              {
44                  $data['pengajuan'] = DB::table('Pengajuan'
                        )->join('users', function ($join)
45                  {
46                      $join->on('Pengajuan.id_sekolah', '=',
                            'Users.id_sekolah')
47                          ->where ('users.id', Auth::user
                                ()->id);
48
49                  })->where('status_pengajuan','3')->get();
50              }
51              else{
52                  $data['pengajuan'] = DB::table
                        ('Pengajuan')->join('users', function
                        ($join)
53                  {
54                      $join->on('Pengajuan.id_sekolah', '=',
                            'Users.id_sekolah')
55                          ->where ('users.id', Auth::user
                                ()->id);
56
57                  })->where('status_pengajuan','1')->orWhere
                        ('status_pengajuan','2')->get();
58              }
59          }
60          return json_encode($data);
61      }
```

Kode sumber 4.16 Implementasi API *Endpoint* Pengajuan dengan metode GET

4.6.4. Implementasi API *Endpoint* Detail Pengajuan

API *Endpoint* Detail Pengajuan digunakan untuk mendapatkan data detail pengajuan sesuai id pengajuan yang dipilih. Metode API *Endpoint* Detail Pengajuan yang digunakan adalah GET. Kode sumber implementasi untuk API *Endpoint* Detail Pengajuan dapat dilihat pada kode sumber 4.17.

```
1   public function getdetail($id){
2
3       if(Auth::user()->role_akun=='1')
4       {
5       $data['detailpeng'] = DB::table('Pengajuan')
6           ->join('mapping_pengajuan_detail','pengajuan
                .id_pengajuan','=', 'mapping_pengajuan_detail
                .id_pengajuan')
7           ->join('detail_pengajuan', 'detail_pengajuan
                .id_detail', '=', 'mapping_pengajuan_detail
                .id_detail')
8           ->where('mapping_pengajuan_detail.id_pengajuan'
                ,$id)
9           ->get();
10            foreach($data['detailpeng'] as $item){
11                if($item->status_pengajuan == "1"){
12                    $item->msg = "1";
13                }
14                else{
15                    $item->msg = "0";
16                }
17            }
18            return json_encode($data);
19          }
20      else
21      {
22      $data['detailpeng'] = DB::table('Pengajuan')
23          ->join('mapping_pengajuan_detail','pengajuan
                .id_pengajuan','=', 'mapping_pengajuan_detail
                .id_pengajuan')
24          ->join('detail_pengajuan', 'detail_pengajuan
                .id_detail', '=', 'mapping_pengajuan_detail
                .id_detail')
25          ->where('mapping_pengajuan_detail.id_pengajuan',$id
                )->get();
26            foreach($data['detailpeng'] as $item){
27                if($item->status_pengajuan == "0"){
28                    $item->msg = "1";
29                    }
30                else {
31                    $item->msg = "0";
32                    }
33            }
34            return json_encode($data);
35          }
36      }
```

Kode sumber 4.17 Implementasi API *Endpoint* Detail Pengajuan dengan metode GET

4.6.5. Implementasi API *Endpoint* Mengirim Foto Wajah

API *Endpoint* Mengirim Foto Wajah digunakan untuk mengirim foto wajah yang diambil melalu aplikasi Android ke

server. Metode API *Endpoint* Mengirim Foto Wajah yang digunakan adalah POST. Kode sumber implementasi untuk API *Endpoint* Mengirim Foto Wajah Pengajuan dapat dilihat pada kode sumber 4.18.

```
1   public function sendFace(Request $request)
2       {
3           $base64_string = $request->file('image');
4           $id_usr = Auth::user()->id;
5           $image_name = 'uploadFace/'.$id_usr;
6
7           if (!file_exists($image_name)) {
8            if (!mkdir($image_name)) {
9               $m=array('msg' => "REJECTED, cant create folder"
                    );
10              echo json_encode($m);
11              return;}
12          }
13
14          if ($files !== false) {
15                  $filecount = count($files);
16          }
17          if($filecount==5){
18              echo json_encode(array('msg' => "Upload Wajah
                    Selesai, Data Tersimpan"));
19              return;
20          }
21
22          $fullName = $id_usr."_".$filecount."_". date
                ("YmdHis") .".png";
23          $ifp = $base64_string->move($image_name,$fullName);
24          if (!$ifp){
25              $m=array('msg' => "REJECTED, ".$fullName."not
                    saved");
26              echo json_encode($m);
27              return;
28          }
29
30          $command = escapeshellcmd("python ".public_path
                ("code/checkFace.py")." ". public_path
                ("uploadFace/".$id_usr."/".$fullName));
31          $output = shell_exec($command);
```

Kode sumber 4.18 Implementasi API *Endpoint* Mengirim Foto Wajah dengan metode POST

4.6.6. Implementasi API *Endpoint Training* Foto Wajah

API *Endpoint Training* Foto Wajah digunakan untuk menjalankan proses latih data dengan algoritma DeblurGAN yang akan menghasilkan foto baru. Metode API *Endpoint Training* Foto Wajah yang digunakan adalah GET. Kode sumber implementasi untuk API *Endpoint Training* Foto Wajah Pengajuan dapat dilihat pada kode sumber 4.19.

```
1  tolakKonfirmasi.setOnClickListener(new View.OnClickListener(){
2
3  @Override
4
5  public void onClick(View v){
6  final SweetAlertDialog pDialog=newSweetAlertDialog(DetailPengajuanActivity.this,SweetAlertDialog.PROGRESS_TYPE);
7  pDialog.getProgressHelper().setBarColor(Color.parseColor("#A5DC86"));
8      pDialog.setTitleText("Loading");
9      pDialog.setCancelable(false);
10     pDialog.show();
11     ApiClientAttendance api = ServerAttendance.createServiceWithAuth(ApiClientAttendance.class,tokenManager);
12 Call
13 <OKResponse> call = api.postdetailpengajuan(id,"3");
14 call.enqueue(new Callback
15     <OKResponse>(){
16 @Override
17 public void onResponse(Call
18         <OKResponse> call, Response
19             <OKResponse> response) {
20 if(response.isSuccessful()){
21 pDialog.dismiss();
22 new SweetAlertDialog(DetailPengajuanActivity.this,SweetAlertDialog.SUCCESS_TYPE).setTitleText("Hasil")
23 .setContentText("Berhasil Ditolak")
24 .setConfirmText("OK").setConfirmClickListener(new SweetAlertDialog.OnSweetClickListener(){
25 @Override
26 public void onClick(SweetAlertDialog sDialog){
27 sDialog.dismissWithAnimation();}
28 }).show();
29 Intent intent = new Intent(DetailPengajuanActivity.this,TolakActivity.class);
30 intent.putExtra("id",id);
31 startActivity(intent);
32 }
```

Kode sumber 4.19 Implementasi API *Endpoint Training* Foto Wajah dengan metode GET

4.6.7. Implementasi API *Endpoint* Menyetujui Pengajuan

API *Endpoint* Menyetujui Pengajuan digunakan untuk mengirim foto wajah kemudian menjalankan proses pengenalan wajah dengan MobileNET guna memberikan persetujuan pengajuan. Metode API *Endpoint* Menyetujui Pengajuan yang digunakan adalah POST. Kode sumber implementasi untuk API *Endpoint* Menyetujui Pengajuan Pengajuan dapat dilihat pada kode sumber 4.20.

```
1  public function predictFace(Request $request)
2      {
3          $base64_string = $request->file('image');
4
5          $id_usr = Auth::user()->id;
6          $image_name = 'predictFace/'.$id_usr;
7
8          if (!file_exists($image_name)) {
9           if (!mkdir($image_name)) {
10              $m=array('msg' => "REJECTED, cant create folder");
11              echo json_encode($m);
12              return;}
13         }
14
15
16         $files = File::files(public_path($image_name));
17           $filecount = 0;
18
19           if ($files !== false) {
20               $filecount = count($files)+1;
21           }
22
23
24         $fullName = $id_usr."_".$filecount."_". date("YmdHis") .".png";
25
26         $ifp = $base64_string->move($image_name,$fullName);
27
28         if (!$ifp){
29             $m=array('msg' => "REJECTED, ".$fullName."not saved");
30             echo json_encode($m);
31             return;}
32         $command = escapeshellcmd("python ".public_path("code/doPredict.py")." ".$id_usr. " ".public_path("predictFace/".$id_usr."/".$fullName));
33         $output = shell_exec($command);
34         if(strpos($output,"ACCEPTED") && $request->id != 0){
35             $pengajuan = Pengajuan::where("id_pengajuan", $request->id)->first();
36             $pengajuan->status_pengajuan = (Auth::user()->role_akun == "1")?"2":"1";
37             $pengajuan->save();
38         }
39         $m = array('msg' => $output);
40         echo json_encode($m);
41     }
```

Kode sumber 4.20 Implementasi API *Endpoint* Menyetujui Pengajuan dengan metode POST

BAB V

UJI COBA DAN EVALUASI

Bab ini membahas uji coba dan evaluasi terhadap perangkat lunak yang telah dikembangkan dari implementasi kasus penggunaan, pengelolaan sampel, dan API *Endpoint*.

5.1 Lingkungan Uji Coba

Pada proses pengujian perangkat lunak, dibutuhkan suatu lingkungan pengujian yang sesuai dengan standar kebutuhan. Lingkungan pengujian dalam Tugas Akhir ini dilakukan pada setiap kasus penggunaan. Spesifikasi masing-masing lingkungan pengujian dijabarkan pada Tabel 5.1 dan Tabel 5.2

Tabel 5.1 Lingkungan Pengujian Aplikasi Web

Spesifikasi	Deskripsi
CPU	Prosesor: Intel® Core™ i7-8550U CPU @1.80GHz (8CPUs) , ~2.0GHz
RAM	8.00 GB
Sistem Operasi	Windows 10 Education 64-bit
Browser	Google Chrome versi 79

Tabel 5.2 Lingkungan Pengujian Aplikasi Android

Spesifikasi	Deskripsi
Jenis Perangkat	*Smartphone*
Tipe perangkat	OPPO A53
Sistem Operasi	Android 10
RAM	4.00 GB

Tabel 5.3 Lingkungan Server Aplikasi

Spesifikasi	Deskripsi
CPU	Prosesor: Intel® Core™ i7-8550U CPU @1.80GHz (8CPUs) , ~2.0GHz
RAM	8.00 GB
Sistem Operasi	Windows 10 Education 64-bit
Server Web	Apache 2.4.25

5.2 Pengujian Kasus Penggunaan

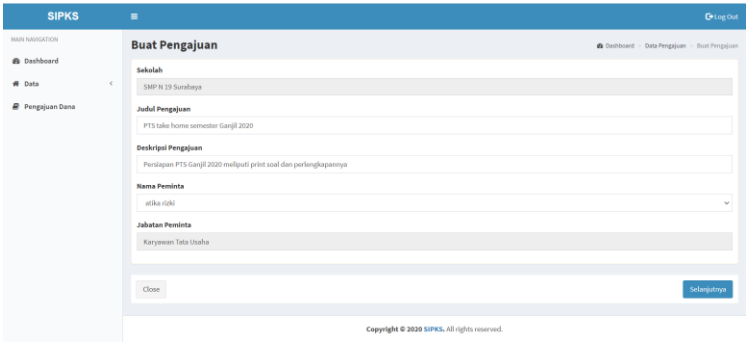
Pada subbab ini akan dibahas mengenai uji coba pada kasus penggunaan yang sudah diimplementasikan. Pengujian dilakukan dengan metode black boxuntuk menguji masing-masing fungsionalitas yang sudah dirancang pada sistem. Metode black boxmerupakan metode pengujianperangkat lunak yang memeriksa fungsionalitas dari suatu perangkat lunak tanpa memandang struktur internalnya.Pada proses uji coba, pengujian dilakukan dengan menjalankanserangkaian perintah terhadap sistem yang selanjutnya akan disebut sebagai kasus pengujian. Kasus pengujian ini berkorelasi dengan kasus-kasus penggunaan dan kebutuhan fungsional yang sebelumnya sudah dirancang dan dijelaskan pada Bab III.

5.2.1 Pengujian Menambahkan Pengajuan Dana Anggaran

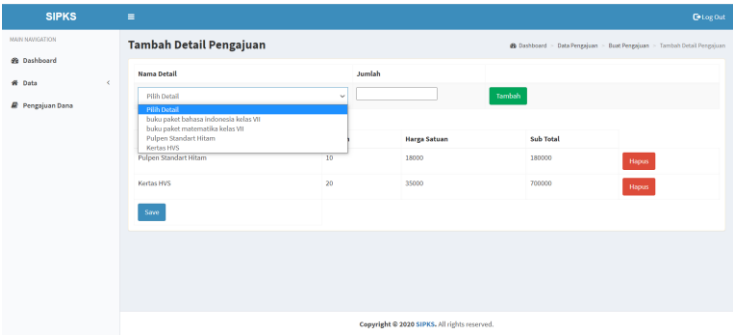
Tabel 5.4 Kasus Pengujian Menambahkan Pengajuan Dana Anggaran

Menambahkan Pengajuan Dana Anggaran	
No. Pengujian	**PWU-001**

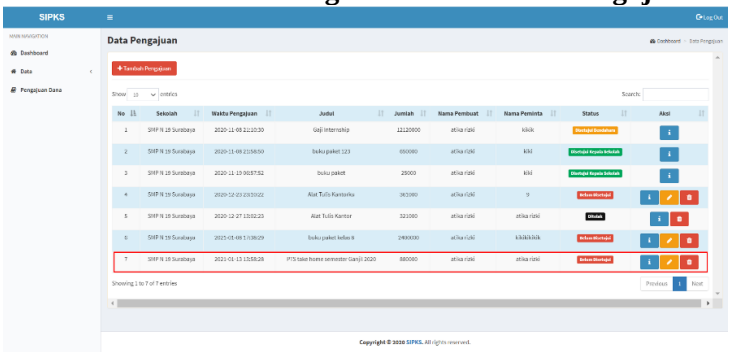
Nama	Menambahkan Pengajuan Dana Anggaran
Kondisi Awal	Penguji sudah login melalui aplikasi web SIPKS
Tujuan Pengujian	Menguji apakah sistem sudah mampu menambahkan pengajuan dana baru
Langkah Pengujian	1. Penguji mengakses halaman dashboard 2. Penguji menu 'Pengajuan dana' pada *sidebar* 3. Penguji mengklik tombol 'Tambah Pengajuan' 4. Penguji mengisi semua field form 5. Penguji mengklik tombol 'selanjutnya' dan mengisi field form detail pengajuan 6. Penguji mengklik tombol 'Save'
Hasil yang Diharapkan	Data pengajuan yang diinputkan berhasil ditambahkan pada database dan dapat dilihat pada table list pengajuan
Hasil yang Diperoleh	Data pengajuan yang diinputkan berhasil ditambahkan pada database dan dapat dilihat pada table list pengajuan
Hasil Pengujian	Berhasil



Gambar 5.1 Proses Pengisian Form Pengajuan



Gambar 5.2 Proses Pengisian Form Detail Pengajuan

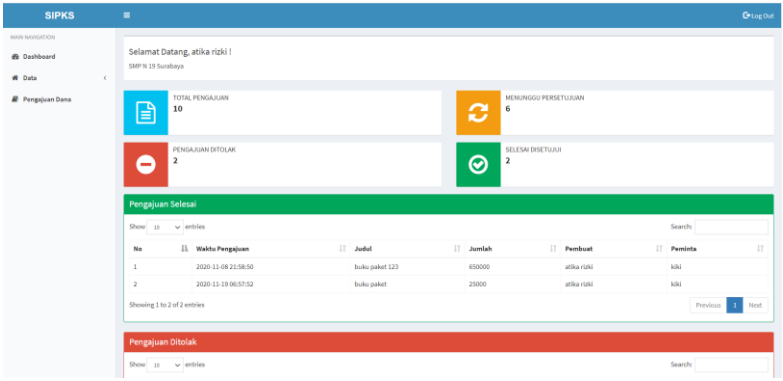


Gambar 5.3 Hasil Pengujian PWU-001 Pengajuan yang Ditambahkan Berhasil Di Tampilkan

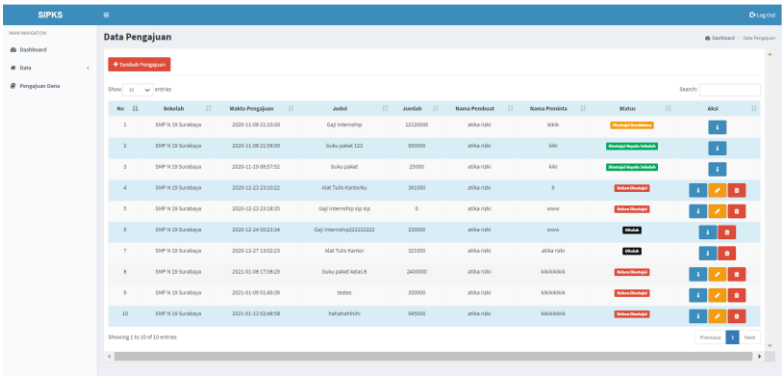
5.2.2 Pengujian Melihat Pengajuan Dana Anggaran (Website)

Tabel 5.5 Kasus Pengujian Melihat Pengajuan Dana Anggaran (Website)

Melihat Pengajuan Dana Anggaran (Website)	
No. Pengujian	**PWU-002**
Nama	Melihat Pengajuan Dana Anggaran
Kondisi Awal	Penguji sudah login melalui aplikasi web SIPKS
Tujuan Pengujian	Menguji apakah sistem sudah mampu menampilkan list pengajuan dana
Langkah Pengujian	1. Penguji mengakses halaman dashboard 2. Penguji memilih menu 'Pengajuan dana' pada *sidebar*
Hasil yang Diharapkan	List tabel pengajuan ditampilkan
Hasil yang Diperoleh	List tabel pengajuan ditampilkan
Hasil Pengujian	Berhasil



Gambar 5.4 Hasil Pengaksesan Halaman Dashboard



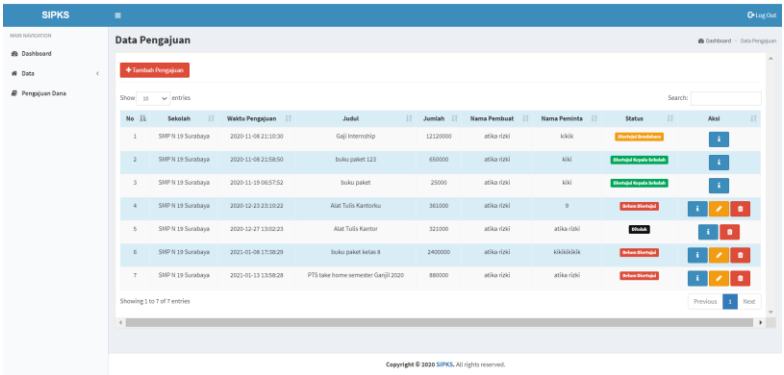
Gambar 5.5. Hasil Pengujian PWU-002 List Tabel Pengajuan Berhasil Di Tampilkan

5.2.3 Pengujian Mengubah Pengajuan Dana Anggaran

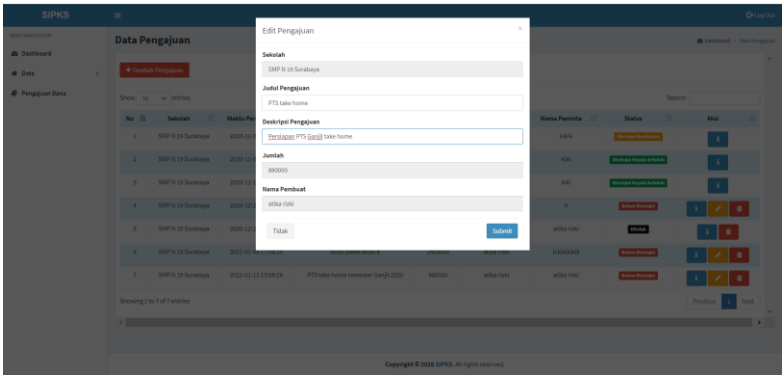
Tabel 5.6 Kasus Pengujian Mengubah Pengajuan Dana Anggaran

Mengubah Pengajuan Dana Anggaran

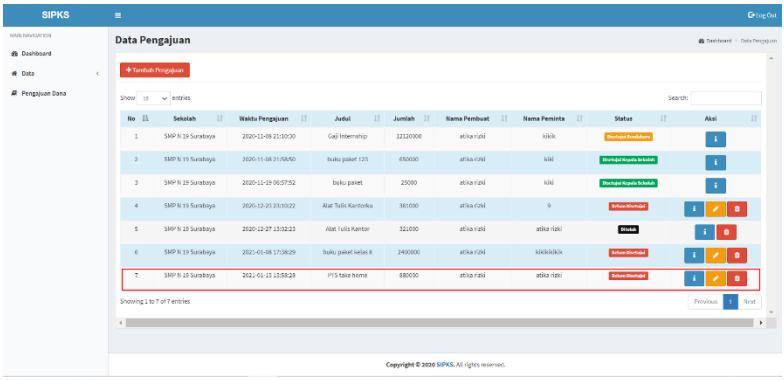
No. Pengujian	**PWU-003**
Nama	Mengubah Pengajuan Dana Anggaran
Kondisi Awal	Pengguna sudah login melalui aplikasi web SIPKS
Tujuan Pengujian	Menguji apakah sistem sudah mampu mengedit field dari pengajuan yang sudah ada
Langkah Pengujian	1. Penguji mengakses halaman dashboard 2. Penguji memilih menu 'Pengajuan dana' pada *sidebar* 3. Penguji mengklik tombol edit dengan icon 'pencil' pada pengajuan yang dipilih 4. Penguji mengubah field yang ingin diubah 5. Penguji mengklik tombol 'Submit'
Hasil yang Diharapkan	Pengajuan yang diubah berhasil di modifikasi di database dan ditampilkan pada table list pengajuan
Hasil yang Diperoleh	Pengajuan yang diubah berhasil di modifikasi di database dan ditampilkan pada table list pengajuan
Hasil Pengujian	Berhasil



Gambar 5.6. Proses Pemilihan Pengajuan yang Ingin di Edit



Gambar 5.7. Proses Pengeditan pada Form Edit Pengajuan



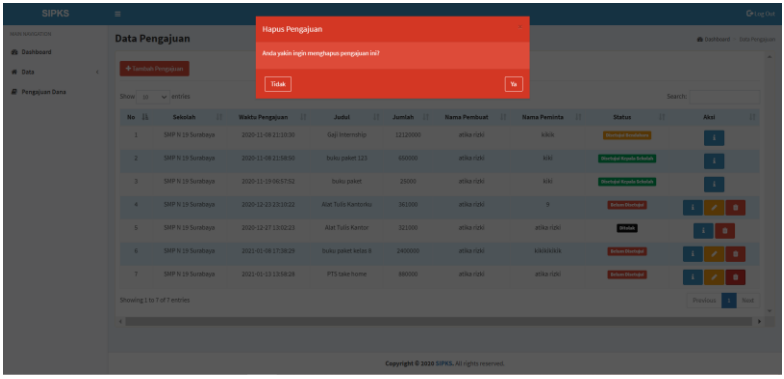
Gambar 5.8 Hasil Pengujian PWU-003 Pengajuan yang Telah Di Edit Berhasil Di Tampilkan

5.2.4 Pengujian Menghapus Pengajuan Dana Anggaran

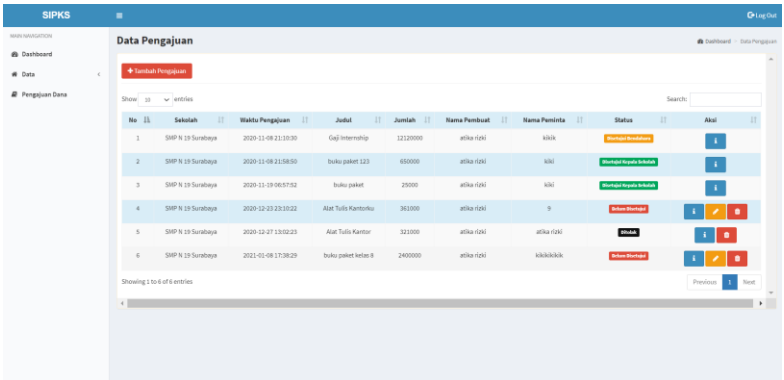
Tabel 5.7 Kasus Pengujian Menghapus Pengajuan Dana Anggaran

Menghapus Pengajuan Dana Anggaran	
No. Pengujian	**PWU-004**
Nama	Menghapus Pengajuan Dana Anggaran
Kondisi Awal	Pengguna sudah login melalui aplikasi web SIPKS
Tujuan Pengujian	Menguji apakah sistem sudah mampu menghapus pengajuan yang sudah ada
Langkah Pengujian	1. Penguji mengakses halaman dashboard 2. Penguji memilih menu ‘Pengajuan dana’ pada *sidebar* 3. Penguji mengklik tombol icon ‘sampah’ pada pengajuan yang dipilih

	4. Penguji mengubah field yang ingin diubah 5. Penguji mengklik tombol Ya’
Hasil yang Diharapkan	Pengajuan yang dihapus terhapus di database dan sudah tidak ada pada table list pengajuan
Hasil yang Diperoleh	Pengajuan yang dihapus terhapus di database dan sudah tidak ada pada table list pengajuan
Hasil Pengujian	Berhasil



Gambar 5.9. Proses Penghapusan Pengajuan yang Dipilih



Gambar 5.10 Hasil Pengujian PWU-004 Pengajuan yang Telah Dihapus Berhasil Terhapus dari List Pengajuan

5.2.5 Pengujian Melihat Pengajuan Dana Anggaran (mobile)

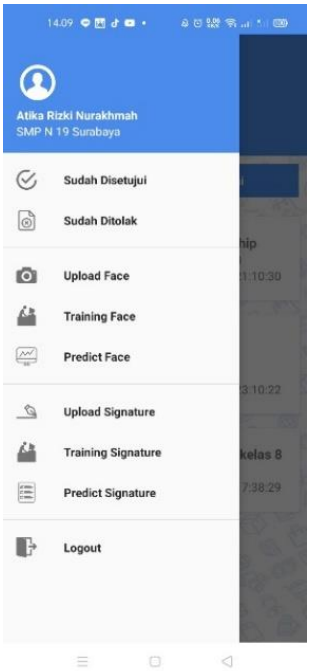
Tabel 5.8 Kasus Pengujian Melihat Pengajuan Dana Anggaran

Melihat Pengajuan Dana Anggaran	
No. Pengujian	**PAU-001**
Nama	Melihat Pengajuan Dana Anggaran
Kondisi Awal	Penguji sudah login melalui aplikasi mobile SIPKS
Tujuan Pengujian	Menguji apakah sistem sudah mampu menampilkan list pengajuan dana maupun detailnya
Langkah Pengujian	1. Penguji mengakses halaman utama 2. Aplikasi menampilkan list pengajuan 'Belum dikonfimasi' 3. Penguji memilih menu 'Sudah Disetujui' pada sidebar

	4. Penguji memilih menu ‘Sudah Ditolak’ pada sidebar
Hasil yang Diharapkan	Data pengajuan yang belum dikonfirmasi, sudah dikonfirmasi, maupun ditolak dapat ditampilkan
Hasil yang Diperoleh	Data pengajuan yang belum dikonfirmasi, sudah dikonfirmasi, maupun ditolak dapat ditampilkan
Hasil Pengujian	Berhasil

Gambar 5.11 Proses *login* ke aplikasi android

Gambar 5.12 Proses mengakses halaman *dashboard* android



Gambar 5.13 Proses mengakses menu sidebar untuk memilih status pengajuan

Gambar 5.14 Hasil pengujian PAU-001 untuk status pengajuan disetujui berhasil ditampilkan

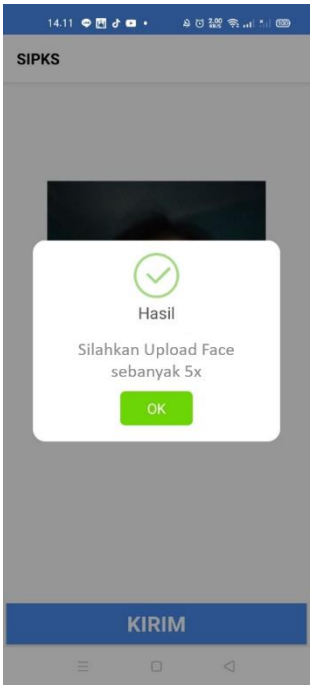
Gambar 5.15 Hasil pengujian PAU-001 untuk status pengajuan ditolak berhasil ditampilkan

5.2.6 Pengujian Mengunggah Sampel Wajah

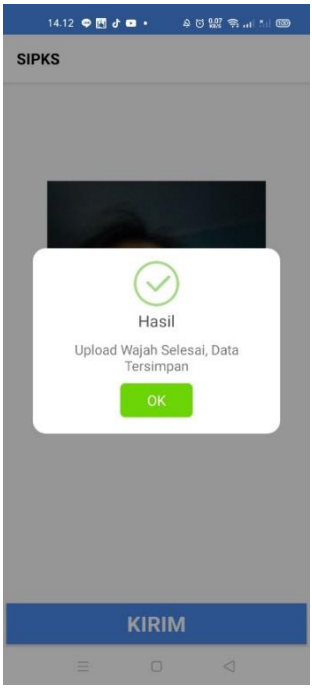
Tabel 5.9 Kasus Pengujian Mengunggah Sampel Wajah

Mengunggah Sampel Wajah	
No. Pengujian	**PMU-002**
Nama	Mengunggah sampel wajah
Kondisi Awal	Penguji sudah login melalui aplikasi mobile SIPKS

Tujuan Pengujian	Menguji apakah sistem sudah mampu mengirim atau mengupload foto wajah untuk disimpan ke server
Langkah Pengujian	1. Penguji mengakses halaman utama 2. Penguji memilih menu '*Upload Face*' pada *sidebar* 3. Penguji mengambil gambar wajah dengan kamera yang tersedia 4. Penguji mengklik tombol 'Kirim' 5. Penguji mengambil foto Kembali sebanyak 5x, lalu mengklik 'Kirim' Kembali
Hasil yang Diharapkan	Sampel wajah berhasil disimpan ke server sesuai id penguji
Hasil yang Diperoleh	Sampel wajah berhasil disimpan ke server sesuai id penguji
Hasil Pengujian	Berhasil



Gambar 5.16 Hasil Pengujian PAU-002 Upload Wajah sebanyak 1 kali



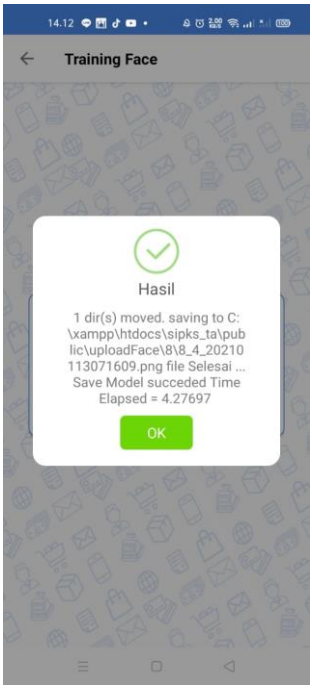
Gambar 5.17 Hasil Pengujian PAU-002 Upload Wajah sebanyak 5 kali

5.2.7 Pengujian Melakukan *Training* Foto Wajah

Tabel 5.10 Kasus Pengujian Melakukan Training Foto Wajah

Melakukan Training Foto Wajah	
No. Pengujian	**PMU-003**

Nama	Melakukan training foto wajah
Kondisi Awal	Sampel wajah penguji telah terdaftar pada aplikasi mobile SIPKS
Tujuan Pengujian	Menguji apakah sistem sudah mampu melakukan training foto wajah penguji yang sebelumnya sudah terdaftar
Langkah Pengujian	1. Penguji mengakses halaman utama 2. Penguji memilih menu *'Train Face'* pada *sidebar* 3. Penguji mengklik tombol 'train'
Hasil yang Diharapkan	Data foto wajah penguji berhasil ditrain, menghasilkan foto baru dan model tersimpan
Hasil yang Diperoleh	Data tanda tangan penguji berhasil ditrain, menghasilkan foto baru dan model tersimpan
Hasil Pengujian	Berhasil



Gambar 5.18 Proses mengakse halaman *training* wajah

Gambar 5.19 Hasil Pengujian PAU-003 Sampel Wajah berhasil di*training*

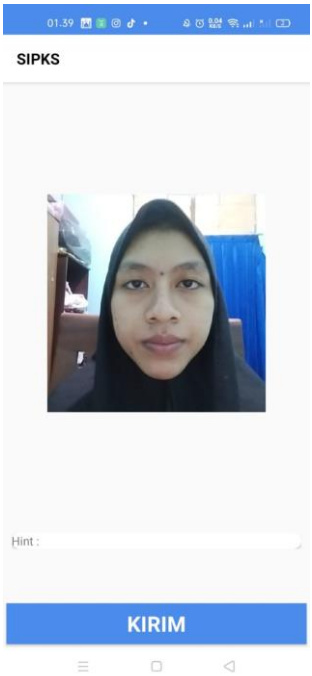
5.2.8 Pengujian Memberikan Persetujuan dengan Pengenalan Wajah

Tabel 5.11 Kasus Pengujian Memberikan Persetujuan dengan Pengenalan Wajah

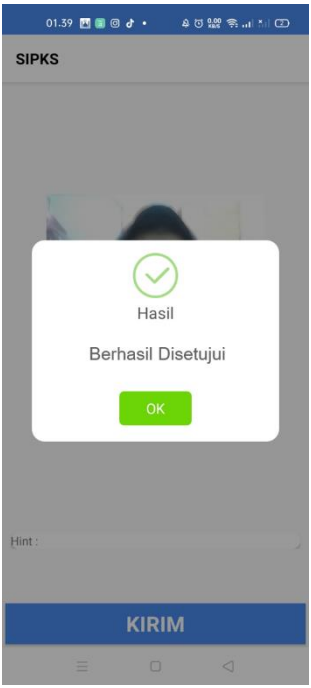
Memberikan Persetujuan dengan Pengenalan Wajah	
No. Pengujian	**PMU-004**

Nama	Memberikan persetujuaan dengan pengenalan wajah
Kondisi Awal	Sampel wajah penguji telah terdaftar dan sudah di train pada aplikasi mobile SIPKS
Tujuan Pengujian	Menguji apakah sistem sudah mampu melakukan pengenalan wajah untuk menyetujui pengajuan anggaran dana
Langkah Pengujian	1. Penguji mengakses halaman utama 2. Penguji memilih salah satu pengajuan yang belum disetujui dengan cara mengklik pengajuan tersebut 3. Aplikasi menampilkan halaman detail pengajuan 4. Penguji mengklik tombol 'konfirmasi' 5. Penguji mengambil gambar wajah dengan kamera yang tersedia 6. Penguji mengklik tombol 'Kirim'
Hasil yang Diharapkan	Status pengajuan anggaran dana menjadi 'Sudah Disetujui'
Hasil yang Diperoleh	Status pengajuan anggaran dana menjadi 'Sudah Disetujui'
Hasil Pengujian	Berhasil

Gambar 5.20 Proses mengakses halaman detail pengajuan yang belum disetujui



Gambar 5.21 Proses mengakses halaman ambil foto wajah



Gambar 5.22 Hasil Pengujian PAU-004 Pengajuan Berhasil Disetujui Dengan Pengenalan Wajah

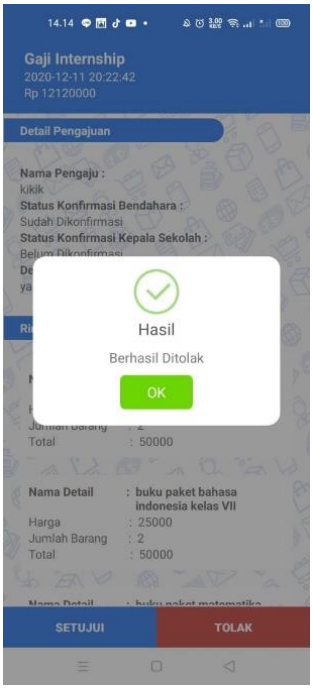
Gambar 5.23 Hasil Pengujian PAU-004 Daftar Pengajuan Sudah Disetujui

5.2.9 Pengujian Memberikan Penolakan Pengajuan Dana Anggaran

Tabel 5.12 Kasus Pengujian Memberikan Penolakan Pengajuan Dana Anggaran

Memberikan Penolakan Pengajuan Dana Anggaran	
No. Pengujian	**PAU-005**
Nama	Memberikan penolakan pengajuan dana anggaran
Kondisi Awal	Pengguna sudah login melalui aplikasi mobile SIPKS
Tujuan Pengujian	Menguji apakah sistem sudah mampu menolak pengajuan anggaran dana yang masuk
Langkah Pengujian	1. Penguji mengakses halaman utama 2. Penguji memilih salah satu pengajuan yang belum disetujui dengan cara mengklik pengajuan tersebut 3. Aplikasi menampilkan halaman detail pengajuan 4. Penguji mengklik tombol ‘Tolak’
Hasil yang Diharapkan	Status pengajuan dana anggaran ‘Ditolak’
Hasil yang Diperoleh	Status pengajuan dana anggaran ‘Ditolak’
Hasil Pengujian	Berhasil

Gambar 5.24 Proses mengakses halaman detail pengajuan yang belum disetujui

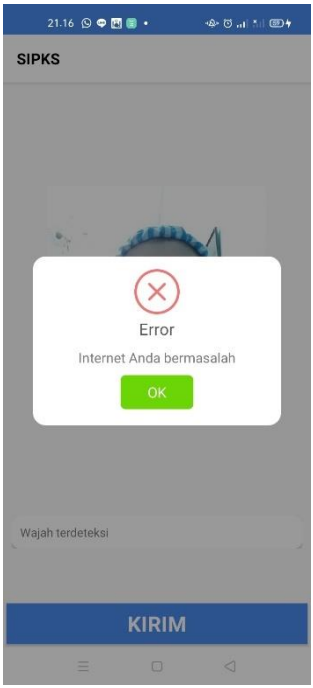


Gambar 5.25 Hasil Pengujian PAU-005 Persetujuan berhasil ditolak

Gambar 5.26 Hasil Pengujian PAU-005 Persetujuan yang ditolak berhasil masuk ke daftar pengajuan ditolak

5.3 Pengujian Koneksi Internet

Pada pengujian ini, aplikasi diuji dengan kondisi koneksi internet terputus di tengah proses persetujuan berlangsung. Pada kondisi ini, proses pengenalan wajah akan berhenti dan persetujuan dinyatakan gagal sehingga user harus melakukan persetujuan ulang dengan mengambil foto wajah lagi melalui kamera. Aplikasi akan memberikan *alert* tentang kesalahan koneksi internet dan akan *redirect* ke halaman *dashboard.* Hasil pengujian dapat dilihat pada Gambar 5.27 dan Gambar 5.28.



Gambar 5.27 *Alert* saat koneksi internet terputus

Gambar 5.8 *Redirect* ke halaman *dashboard*

5.4 Pengujian Kasus Pengenalan Wajah

Pada uji kasus pengenalan wajah, digunakan wajah dengan berbagai kondisi, seperti terdapat lebih dari 1 wajah, wajah diambil dengan kondisi sekitar yang gelap, foto wajah diambil dengan *background* yang tidak polos, dan pengenalan wajah menggunakan foto cetak. Data yang digunakan pada uji kasus ini terdapat pada Lampiran. Hasil uji kasus pengenalan wajah dapat dilihat pada Tabel 5.13.

Tabel 5.13 Hasil Uji Kasus Pengenalan Wajah

No	Kasus	Tanpa GAN		Dengan GAN	
		Akurasi	Waktu komputasi (second)	Akurasi	Waktu komputasi (second)
1.	*Background* tidak polos	48.41	4.12148	80.06	87
		Rejected		**Accepted**	
2.	Terdapat lebih dari 1 wajah, namun dominan wajah yang benar	39.85	4.57985	75.84	94
		Rejected		**Accepted**	
3.	Menggunakan foto cetak	82.28	4.17049	88.69	90
		Accepted		**Accepted**	
4.	Kondisi gelap	14.66	4.18455	19.51	91
		Rejected		**Rejected**	

Dari tabel 5.13. dapat dilihat bahwa dari 4 kasus yang diujikan pada pengenalan wajah dengan GAN dan tanpa menggunakan GAN. Penulis menggunakan batas minimal akurasi sebesar 60 untuk menentukan apakah wajah berhasil dikenali (Accepted) atau tidak berhasil dikenali (Rejected). Penulis menggunakan 5 foto untuk melakukan *training* dan 1 foto untuk prediksi. Foto yang digunakan pada prediksi merupakan foto blur. Pada pengenalan wajah tanpa GAN, tidak dilakukan proses perbaikan gambar blur terlebih dahulu. Prediksi langsung dilakukan menggunakan *transfer learning* MobileNet. Sementara pada pengenalan wajah dengan GAN, dilakukukan perbaikan

gambar blur terlebih dahulu sebelum dilakukan prediksi menggunakan *transfer learning* MobileNet. Dari 4 kasus yang diujikan, terdapat 3 kasus yang memiliki hasil *rejected* atau wajah tidak berhasil dikenali pada pengenalan tanpa GAN yaitu ketika foto diambil dengan *background* yang tidak polos, terdapat lebih dari satu wajah pada foto, dan saat foto diambil dalam kondisi gelap. Sementara hasil uji kasus pengenalan wajah tanpa GAN dengan input berupa foto cetak adalah *accepted* yang berarti wajah dapat dikenali dan cocok. Pada pengenalan wajah dengan GAN, 3 kasus berhasil mendapatkan hasil *accepted* yaitu pengenalan wajah dengan input foto cetak, foto yang diambil dengan *background* yang tidak polos (*noise*), dan foto dengan wajah lebih dari satu namun memiliki wajah dominan. Sementara foto yang diambil dalam kondisi gelap tidak dapat dikenali pada pengenalan wajah dengan GAN. Secara keseluruhan, dapat diketahui bahwa pengenalan wajah dengan input yang diperbaiki menggunakan GAN memiliki akurasi yang lebih tinggi daripada input yang langsung diprediksi tanpa dilakukan perbaikan gambar sebelumnya. Namun, waktu komputasi yang dibutuhkan pada proses pengenalan wajah dengan GAN jauh lebih lama dibandingkan tanpa GAN. Hal ini disebabkan karena ada proses perbaikan gambar menggunakan DeblurGAN yang strukturnya terdiri dari banyak *layer* dan menggunakan *Convolutional Neural Network* dengan *Residual Block*.

[Halaman ini sengaja dikosongkan]

BAB VI

KESIMPULAN DAN SARAN

Bab ini berisi tentang kesimpulan yang diperoleh selama pengerjaan Tugas Akhir ini berdasarkan hasil pengujian dan hal lainnya yang telah dilakukan. Selain itu juga terdapat beberapa saran terhadap Tugas Akhir yang penulis ajukan untuk pengembangan kedepannya.

6.1 Kesimpulan

Dari hasil pengamatan selama proses perancangan, implementasi, dan pengujian perangkat lunak yang dilakukan, dapat diambil kesimpulan sebagai berikut:

1. Proses pengajuan anggaran dana pada aplikasi SIPKS dilakukan oleh Pegawai Tata Usaha melalui website SIPKS.
2. Persetujuan pengajuan anggaran dana melalui aplikasi SIPKS dilalkukakn oleh Kepala Sekolah dan Bendahara Sekolah melalui aplikasi android SIPKS dengan melakukan pencocokan serta pengenalan wajah.
3. *DeblurGAN* dengan metode ResNet merupakan salah satu pengembangan dari *Generative Adversarial Networks* yang dapat menangani *image deblurring* untuk memperbaiki kualitas foto wajah yang blur menjadi lebih fokus. Selain itu, penggunaan *DeblurGAN* dapat meningkatkan akurasi pengenalan wajah. Akan tetapi, waktu komputasi dengan *DeblurGAN* lebih lama dibandingkan pengenalan wajah tanpa perbaikan foto blur.
4. Pengenalan wajah dilakukan dengan *transfer learning* MobileNet yang dimodifikasi menggunakan *Logistic Regression Classifier*.

6.2 Saran

Terdapat beberapa saran terkait pengembangan Tugas Akhir yang didasarkan pada hasil perancangan, implementasi, dan pengujian yang telah dilakukan. Saran-saran tersebut antara lain :

1. *Generative Adversarial Networks* dapat digunakan untuk memperbaiki kualitas foto pada aspek lainnya, tidak hanya blur. Hasil perbaikannya dapat menambah dataset sehingga lebih variative.
2. Pada Tugas Akhir ini, waktu komputasi dengan DeblurGAN masih cukup lama, dapat dilakukan pengembangan terkait optimalisasi waktu komputasi sehingga menjadi lebih cepat.
3. Pada Tugas Akhir ini, input berupa foto cetak dan wajah bukan foto belum bisa dibedakan. Untuk pengembangan selanjutnya dapat ditambahkan identifikasi apakah foto input berupa wajah asli (bukan foto cetak) atau wajah berupa foto yang dicetak.
4. Kepala Sekolah dan Bendahara Sekolah dapat mengetahui berapa jumlah anggaran yang sudah disetujui dan berapa sisa dana yang dimiliki sekolah.

Saran-saran ini didasarkan pada hasil perancangan, implementasi dan pengujian yang telah dilakukan.

DAFTAR PUSTAKA

[1] Direktorat Jenderal Pendidikan Anak Usia Dini, Pendidikan Dasar dan Pendidikan Menengah, Kementerian Pendidikan dan Kebudayaan, "Data Pokok Pendidikan Dasar dan Menengah", 2019.[Online].Available:https://dapo.dikdasmen.kemdikbud.go.id /sp/2/056000 [Diakses pada 22 April 2020]

[2] "Facial Recognition System" Wikipedia, 2020.[Online]. Available:https://en.wikipedia.org/wiki/Facial_recognition_system [Diakses pada 23 April 2020]

[3] Steve Symanovich, "How does facial recognition work?".[Online].Available: https://us.norton.com/internetsecurity-iot-how-facial-recognition-software-works.html [Diakses pada 23 April 2020]

[4] Andrew Heinzman, "How Does Facial Recognition Work?", 2019.[Online].Available:https://www.howtogeek.com/427897/how-does-facial-recognition-work/ [Diakses pada 23 April 2020]

[5] Jason Brownlee, "Generative Adversarial Networks with Python-Deep Learning Generative Models for Image Synthesis and Image Translation", edisi v1.5, 2019.

[6] Saibharat, "Generative Adversarial Networks (GANs)", 2019. Availbale: https://medium.com/@saibharath897/generative-adversarial-networks-gans-560c5c988128

[7] Raphaël Meudec, "GAN with Keras: Application to Image Deblurring," medium.com, 20 Maret 2018. [Online]. Available: https://medium.com/sicara/keras-generative-adversarial-networks-image-deblurring-45e3ab6977b5. [Diakses 20 Desember 2020].

[8] Luke Welling and Laura Thomson, "PHP and MYSQL Web Development – Developer's Library", fourth edition, 2009.

[9] "Laravel - The PHP Framework for Web Artisans" [Online]. Available: https://laravel.com/ [Diakses pada 20 Desember 2020]

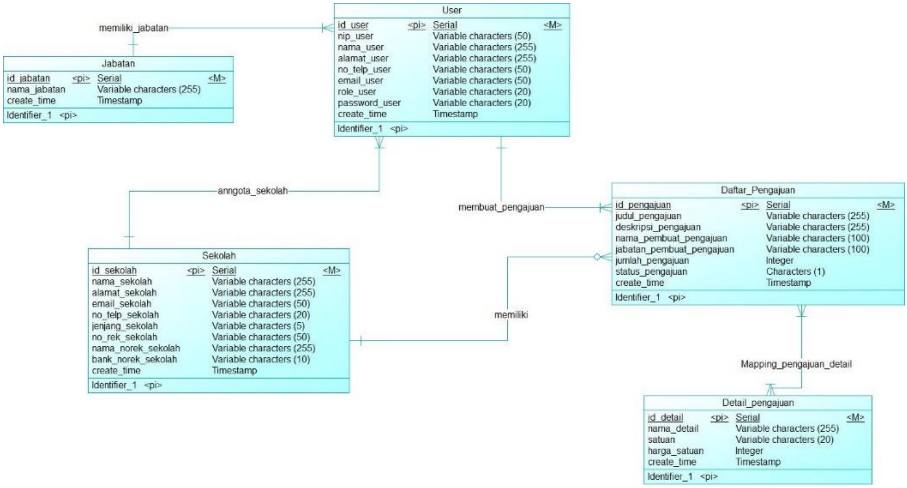
[10] "About Python," Python, [Online]. Available: https://www.python.org/about/. [Diakses 20 Desesmber 2020].

[11] Rizqi Okta Ekoputris, "MobileNet: Deteksi Objek pada

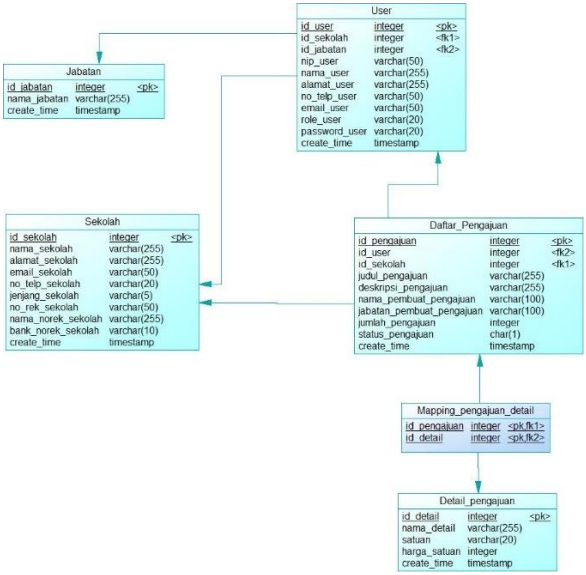
Platform Mobile," medium.com, 09 Mei 2018. [Online]. Available: https://medium.com/nodeflux/mobilenet-deteksi-objek-pada-platform-mobile-bbbf3806e4b3. [Diakses 20 Desember 2020].
[12] "Google's Android OS: Past, Present, and Future - PhoneArena." [Online]. Available: https://www.phonearena.com/news/Googles-Android-OSPast-Present-and-Future_id21273. [Diakses 20 Desember 2020].
[13] "What is Java?" [Online] Available: https://go.java/ [Diakses 20 Desember 2020].
[14]Orest Kupyn, Volodymyr Budzan, Mykola Mykhailych, Dmytro Mishkin, Jiˇri Matas, "DeblurGAN: Blind Motion Deblurring Using Conditional Adversarial Networks", arXiv:1711.07064v4 [cs.CV] 3 Apr 2018.

LAMPIRAN A : CDM DAN PDM

1. CDM



2. PDM



LAMPIRAN B : DAFTAR RINCI TABEL

1. Tabel User

No	Id Field	Deskripsi	Tipe & Length	Keterangan
1	id_user	Variable untuk ID user	integer	Primary key
2	Nip_user	Variable untuk nip user	Varchar (50)	-
3	Nama_user	Variable untuk nama user	Varchar (255)	-
4	Alamat_user	Variable untuk Alamat user	Varchar (255)	-
5	No_telp_user	Variable untuk nomor telepon user	Varchar (50)	-
6	Email_user	Variable untuk email user	Varchar (255)	-
7	Role_user	Variable untuk role user	Char(1)	-
8	Password_user	Variable untuk password user	Varchar (255)	-
9	Create_time	Variable untuk menyimpan waktu poembuatan akun	timesta mp	-

2. Tabel Jabatan

No	Id Field	Deskripsi	Tipe & Length	Keterangan
1	id_jabatan	Variable untuk ID jabatan	integer	Primary key

2	Nama_jabatan	Variable untuk nama jabatan	Varchar (255)	-
3	Create_time	Variabel untuk menyimpan waktu penginputan data	timestamp	-

3. Table Sekolah

No	Id Field	Deskripsi	Tipe & Length	Keterangan
1	id_sekolah	Variable untuk ID sekolah	integer	Primary key
2	nama_sekolah	Variable untuk nama sekolah	Varchar (255)	-
3	alamat_sekolah	Variabel untuk menyimpan alamat sekolah	Varchar (255)	-
4	email_sekolah	Variabel untuk menyimpan email sekolah	Varchar (255)	-
5	jenjang_sekolah	Variabel untuk menyimpan jenjang sekolah	Varchar (255)	-
6	no_rek_sekolah	Variabel untuk menyimpan nomor rekening	Varchar (255)	-
7	Nama_norek_sekolah	Variabel untuk menyimpan nama nomor rekening sekolah	Varchar (255)	-

8	bank_norek_sekolah	Variabel untuk menyimpan bank nomor rekening sekolah	Varchar (255)	-
9	Create_time	Variabel untuk menyimpan waktu penyimpanan data	timestamp	-
10	No_telp_sekolah	Variabel untuk menyimpan nomor telepon sekolah	Varchar (255)	-

4. Tabel Pengajuan

No	Id Field	Deskripsi	Tipe & Length	Keterangan
1	id_pengajuan	Variable untuk ID pengajuan	Integer	Primary key
2	id_sekolah	Variable untuk ID sekolah	Integer	Foreign Key
3	id_akun	Variable untuk ID akun	Integer	Foreign Key
4	judul_pengajuan	Variable untuk nama user	Varchar (255)	-
5	deskripsi_pengajuan	Variable untuk Alamat user	Varchar (255)	-
6	nama_pembuat_pengajuan	Variable untuk nomor telepon user	Varchar (50)	-
7	jabatan_pembuat_pengajuan	Variable untuk email user	Varchar (255)	-

8	jumlah_pengajuan	Variable untuk role user	Integer	-
9	status_pengajuan	Variable untuk password user	Char(1)	-
10	Create_time	Variable untuk menyimpan waktu poembuatan akun	timestamp	-

5. Tabel Detail Pengajuan

No	Id Field	Deskripsi	Tipe & Length	Keterangan
1	id_detail	Variable untuk ID detail pengajuan	Integer	Primary key
2	nama_detail	Variable untuk nama detail	Varchar(255)	-
3	satuan	Variable untuk satuan detail	Varchar(20)	-
4	harga_satuan	Variable untuk harga detail per satuan	Integer	-
5	Create_time	Variable untuk menyimpan waktu pembuatan detail pengajuan	timestamp	-

6. Tabel Mapping Pengajuan

No	Id Field	Deskripsi	Tipe & Length	Keterangan
1	id_mapping_detail_pengajuan	Variable untuk ID mapping pengajuan detail	Integer	Primary key
2	id_pengajuan	Variable untuk ID pengajuan	Integer	Foreign Key
3	id_detail	Variable untuk ID detail pengajuan	Integer	Foreign key
5	Create_time	Variable untuk menyimpan waktu pembuatan mapping pengajuan detail	timestamp	-

LAMPIRAN C : STRUKTUR JSON API ENDPOINT

1. API *Endpoint Login*

No	Object Main JSON	Key Object	Tipe Data
1	code	-	List
2	messagge	-	List
3	Data	Token	String

2. API *Endpoint Profile*

No	Object Main JSON	Key Object	Tipe Data
1	code	-	List
2	messagge	-	List
3	data	bankNorekSekolah	String
		NIPAkun	String
		jenjangSekolah	String
		alamatSekolah	String
		usernameAkun	String
		noTelpAkun	String
		alamatAkun	String
		idSekolah	Integer
		idAkun	Integer
		roleAkun	String
		password	String
		emailSekolah	String
		name	String
		namaSekolah	String
		email	String

3. API *Endpoint* Pengajuan

No	Object Main JSON	Key Object	Tipe Data
1	code	-	List
2	messagge	-	List
3	data	jumlah_pengajuan	String
		nIPAkun	String
		jabatanPembuatPengajuan	String
		createdAt	String
		usernameAkun	String
		judulPengajuan	String
		noTelpAkun	String
		alamatAkun	String
		idSekolah	Integer
		statusPengajuan	String
		namaPembuatPengajuan	String
		Name	String
		Id	Integer
		idAkun	Integer
		deskripsiPengajuan	String
		email	String

4. API *Endpoint* Detail Pengajuan

No	Object Main JSON	Key Object	Tipe Data
1	code	-	List
2	message	-	List
3	data	id_pengajuan	Integer
		id_detail	Integer
		id_sekolah	Integer
		id_akun	Integer
		id_mapping_pengajuan	Integer
		judul_pengajuan	String
		jumlah_detail	String
		harga_satuan	String
		satuan	Integer
		status_pengajuan	String
		nama_pembuat_pengajuan	String
		sub_total	String
		create_time_pengajuan	String
		deskripsi_pengajuan	String

5. API *Endpoint* Mengirim Foto Wajah

No	Object Main JSON	Key Object	Tipe Data
1	code	-	List

2	message	-	List
3	Data	id_usr	Integer
		Nama_file	String

6. API *Endpoint Training* Foto Wajah

No	Object Main JSON	Key Object	Tipe Data
1	code	-	List
2	message	-	List
3	Data	id_usr	Integer
		Nama_file	String

7. API *Endpoint* Menyetujui Persetujuan

No	Object Main JSON	Key Object	Tipe Data
1	code	-	List
2	message	-	List
3	Data	id_usr	Integer
		Nama_file	String

LAMPIRAN D : KODE SUMBER *DEBLURGAN*

```
def train(self):
        def get_optimizer(loss, global_step=None, var_list=None, is_gradient_clip=False):
            train_op = tf.train.AdamOptimizer(self.lr)
            if is_gradient_clip:
                grads_and_vars = train_op.compute_gradients(loss, var_list=var_list)
                unchanged_gvs = [(grad, var) for grad, var in grads_and_vars if not 'LSTM' in var.name]
                rnn_grad = [grad for grad, var in grads_and_vars if 'LSTM' in var.name]
                rnn_var = [var for grad, var in grads_and_vars if 'LSTM' in var.name]
                capped_grad, _ = tf.clip_by_global_norm(rnn_grad, clip_norm=3)
                capped_gvs = list(zip(capped_grad, rnn_var))
                train_op = train_op.apply_gradients(grads_and_vars=capped_gvs + unchanged_gvs, global_step=global_step)
            else:
                train_op = train_op.minimize(loss, global_step, var_list)
            return train_op

        global_step = tf.Variable(initial_value=0, dtype=tf.int32, trainable=False)
        self.global_step = global_step

        # build model
        self.build_model()

        # learning rate decay
        self.lr = tf.train.polynomial_decay(self.learning_rate, global_step, self.max_steps, end_learning_rate=0.0,
                                            power=0.3)
        tf.summary.scalar('learning_rate', self.lr)

         # training operators
         train_gnet = get_optimizer(self.loss_total, global_step, self.all_vars)

         # session and thread
         gpu_options = tf.GPUOptions(allow_growth=True)
         sess = tf.Session(config=tf.ConfigProto(gpu_options=gpu_options))
         self.sess = sess
         sess.run(tf.global_variables_initializer())
         self.saver = tf.train.Saver(max_to_keep=50, keep_checkpoint_every_n_hours=1)
         coord = tf.train.Coordinator()
         threads = tf.train.start_queue_runners(sess=sess, coord=coord)

         # training summary
         summary_op = tf.summary.merge_all()
         summary_writer = tf.summary.FileWriter(self.train_dir, sess.graph, flush_secs=30)

         for step in xrange(sess.run(global_step), self.max_steps + 1):

             start_time = time.time()

             # update G network
             _, loss_total_val = sess.run([train_gnet, self.loss_total])

            duration = time.time() - start_time
            # print loss_value
            assert not np.isnan(loss_total_val), 'Model diverged with loss = NaN'

            if step % 5 == 0:
                num_examples_per_step = self.batch_size
                examples_per_sec = num_examples_per_step / duration
                sec_per_batch = float(duration)

                format_str = ('%s: step %d, loss = (%.5f; %.5f, %.5f)(%.1f data/s; %.3f s/bch)')
                print(format_str % (datetime.now().strftime('%Y-%m-%d %H:%M:%S'), step, loss_total_val, 0.0,
                                    0.0, examples_per_sec, sec_per_batch))

            if step % 20 == 0:
                # summary_str = sess.run(summary_op, feed_dict={inputs:batch_input, gt:batch_gt})
                summary_str = sess.run(summary_op)
                summary_writer.add_summary(summary_str, global_step=step)

             # Save the model checkpoint periodically.
             if step % 1000 == 0 or step == self.max_steps:
                 checkpoint_path = os.path.join(self.train_dir, 'checkpoints')
                 self.save(sess, checkpoint_path, step)
```

```
1  def test(self, height, width, input_path, output_path):
2          if not os.path.exists(output_path):
3              os.makedirs(output_path)
4          imgsName = sorted(os.listdir(input_path))
5
6          H, W = height, width
7          inp_chns = 3 if self.args.model == 'color' else 1
8          self.batch_size = 1 if self.args.model == 'color' else 3
9          inputs = tf.placeholder(shape=[self.batch_size, H, W, inp_chns], dtype=tf.float32)
10         outputs = self.generator(inputs, reuse=False)
11
12         sess = tf.Session(config=tf.ConfigProto(gpu_options=tf.GPUOptions(allow_growth=True)))
13
14         self.saver = tf.train.Saver()
15         self.load(sess, self.train_dir, step=523000)
16
17         for imgName in imgsName:
18             blur = scipy.misc.imread(os.path.join(input_path, imgName))
19             h, w, c = blur.shape
20             # make sure the width is larger than the height
21             rot = False
22             if h > w:
23                 blur = np.transpose(blur, [1, 0, 2])
24                 rot = True
25             h = int(blur.shape[0])
26             w = int(blur.shape[1])
27             resize = False
28             if h > H or w > W:
29                 scale = min(1.0 * H / h, 1.0 * W / w)
30                 new_h = int(h * scale)
31                 new_w = int(w * scale)
32                 blur = scipy.misc.imresize(blur, [new_h, new_w], 'bicubic')
33                 resize = True
34                 blurPad = np.pad(blur, ((0, H - new_h), (0, W - new_w), (0, 0)), 'edge')
35             else:
36                 blurPad = np.pad(blur, ((0, H - h), (0, W - w), (0, 0)), 'edge')
37             blurPad = np.expand_dims(blurPad, 0)
38             if self.args.model != 'color':
39                 blurPad = np.transpose(blurPad, (3, 1, 2, 0))
40
41             start = time.time()
42             deblur = sess.run(outputs, feed_dict={inputs: blurPad / 255.0})
43             duration = time.time() - start
44             imageName = "gan_"+imgName
45             # print('Saving results: %s ... %4.3fs' % (os.path.join(output_path, imageName), duration))
46             res = deblur[-1]
47             if self.args.model != 'color':
48                 res = np.transpose(res, (3, 1, 2, 0))
49             res = im2uint8(res[0, :, :, :])
50             # crop the image into original size
51             if resize:
52                 res = res[:new_h, :new_w, :]
53                 res = scipy.misc.imresize(res, [h, w], 'bicubic')
54             else:
55                 res = res[:h, :w, :]
56
57             if rot:
58                 res = np.transpose(res, [1, 0, 2])
59             scipy.misc.imsave(os.path.join(output_path, imageName), res)
60         print('Success train gan')
```

LAMPIRAN E : FOTO INPUT PENGENALAN WAJAH

No	Kasus	Sebelum Perbaikan	Setelah Perbaikan
1.	Background tidak polos		
2.	Terdapat lebih dari 1 wajah, namun dominan wajah yang benar		
3.	Menggunakan foto cetak		
4.	Kondisi gelap		

BIODATA PENULIS

Atika Rizki Nurakhmah, anak pertama dari dua bersaudara yang lahir di Kebumen, 12 Februari 1999. Penulis telah menempuh pendidikan formal mulai dari SD Negeri Mangun Jaya 01 Tambun Selatan, SMP Negeri 12 Semarang, SMA Negeri 4 Semarang dan terakhir sebagai mahasiswa Departemen Teknik Informatika Institut Teknologi Sepuluh Nopember dengen rumpun mata kuliah Rekayasa Perangkat Lunak (2017- 2021). Penulis memiliki ketertarikan di bidang Pemrograman Web dan Android. Semasa Perkuliahan penulis aktif mengikuti organisasi dan kepanitiaan diantaranya menjadi Kepala Departemen Internal UKTK ITS 2019, Kepala Divisi Keamanan dan Perizinan Big Event IFLS 2019 K-Fest INOCHI, Bendahara 2 Schematics ITS 2019, dan Staff Ahli Departemen Sosial Masyarakat HMTC 2020. Selain itu penulis pernah menjadi *awardee* KEB Hana Bank *Scholarship* 2020. Penulis memiliki pengalaman mengikuti *internship* di Telkom Direktorat Digital Business (Telkom DDB) sebagai Quality Assurance Engineer (QA. Penulis juga berkesempatan menjadi Asisten Dosen di beberapa mata kuliah. Penulis dapat dihubungi lebih lanjut melalui surat elektronik atikarizki41@gmail.com.