SKRIPSI

PROTOTYPE DESAIN KEAMANAN LOGIN MENGGUNAKAN BIOMETRIK WAJAH DENGAN METODE EIGENFACE

Disusun Oleh :

Nama : Sugeng Widodo

Nim : 15111045

PROGRAM STUDI INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI

UNIVERSITAS MERCU BUANA YOGYAKARATA

TAHUN 2019

PROTOTYPE DESAIN KEAMANAN LOGIN MENGGUNAKAN BIOMETRIK WAJAH DENGAN METODE EIGENFACE

Disusun oleh

Sugeng Widodo

15111045

Skripsi ini diajukan untuk memenuhi persyaratan akademik sarjana pada Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Mercu Buana Yogyakarta.

Yogyakarta, 19 juli 2019

Menyetujui Pembimbing,

Supatman, ST., MT.

NIDN. 0509057202

HALAMAN PENGESAHAN

Prototype Desain Keamanan Login Menggunakan Biometrik Wajah dengan Metode Eigenface

Oleh:

Sugeng Widodo
15111045

Telah dipertanggungjawabkan dan
diterima oleh Tim Penguji pada
tanggal 1 Agustus 2019

Mengetahui
Dekan,

(Supatman, S.T.,MT.)
NIDN. 0509057202

Dosen Pembimbing,

(Supatman, S.T.,MT.)
NIDN. 0509057202

Dosen Penguji,

1. Imam Suharjo, ST., M.Eng.
NIDN. 0509057901

2. Anita Witanti, ST.,MT.
NIDN. 0422018102

3. Miftakhin Akbar, S.Kom., M.T.
NIDN. 0528078902

ii

HALAMAN PERNYATAAN PERSETUJUAN
PUBLIKASI SKRIPSI UNTUK KEPENTINGAN
AKADEMIS

Sebagai mahasiswa Universitas Mercu Buana Yogyakarta, saya yang bertanda tangan di bawah ini:

Nama	:	Sugeng Widodo
NIM	:	15111045
Program Studi	:	Informatika
Fakultas	:	Teknologi Informasi
Jenis Karya	:	Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Mercu Buana Yogyakarta Hak Bebas Royalti Non-eksklusif (*None-exclusive Royalty Free Right*) atas kerya ilmiah saya yang berjudul:

Prototype Desain Keamanan Login Menggunakan Biometrik Wajah dengan Metode Eigenface

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Royalti Non-eksklusif ini Universitas Mercu Buana Yogyakarta berhak menyimpan, mengalih-media-kan, mengelola dalam bentuk pangkalan data (database), merawat dan mempublikasikan skripsi saya selama tetap mencantumkan nama saya sebagai penulis dan sebagai pemilik hak cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : YOGYAKARTA
Pada Tanggal : 21 Juli 2019

Yang menyatakan,

(Sugeng Widodo)

iii

HALAMAN PERNYATAAN ORISINALITAS KARYA

Skripsi ini adalah hasil karya saya sendiri,
dan semua sumber baik yang dikutip maupun yang dirujuk,
telah saya nyatakan dengan benar.

Nama Lengkap	:	Sugeng Widodo
NIM	:	15111045
Judul Skripsi	:	Prototype Desain Keamanan Login Menggunakan Bioemetrik Wajah Dengan Menggunakan Metode Eigenface
Program Studi	:	Informatika
Tanggal	:	21 Juli 2019
Tanda Tangan	:	

HALAMAN PERSEMBAHAN

Puji syukur kehadirat Allah SWT saya dapat menyelesaikan skripsi saya.

1. Terima kasih kepada kedua Orang Tua dan saudara-saudaraku yang telah memberikan support dan Doa-nya Karena tanpanya saya tidak mungkin dapat menyelesaikan skripsi saya.
2. Terima kasih juga kepada teman-teman sekalian baik teman nongkrong maupun teman mancing yang selalu ada disaat membutuhkan, karenanya selalu membuat hari-hariku penuh warna, percayalah sensasi tarikan ikan besar yang tiada duanya akan sangat dirindukan jika tanpa dirimu.
3. Terima kasih kepada Supatman, ST., MT. yang telah membimbing saya dengan sabar sehingga saya dapat menyelesaikan skripsi ini.

HALAMAN MOTTO

“Don’t Loop Back Don’t Regret”

Jangan mengulangi kesalahan yang telah dilakukan dan jangan menyesali waktu yang telah terbuang sia-sia. Tetapi “**Keep Moving Forward**”. Dan apabila butuh waktu sejenak untuk berfikir, “PERGILAH MEMANCING”.

KATA PENGANTAR

Puji syukur kehadirat Allah SWT. yang melimpahkan segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan karya tulis yang berbentuk skripsi ini sesuai dengan waktu yang telah direncanakan. yang telah mendukung dan membantu proses penulisan ini.

Penulis menyadari bahwa dalam menyelesaikan skripsi ini tidak lepas dari peran berbagai pihak yang telah memberikan nasehat bimbingan dan bantuan khususnya kepada:

1. Bapak Supatman, S.T., M.T. selaku Dekan Fakultas Teknologi Informasi Universitas Mercu Buanan Yogyakarta dan Dosen Pembimbing Skripsi yang telah memberikan penjelasan selama penulisan skripsi.
2. Bapak Agus Sidiq Purnomo, S. Kom., M.Eng. selaku Ketua Prodi Informatika.
3. Bapak Imam Suharjo, S.T., M.Eng. selaku Dosen Pembimbing Akademik.
4. Bapak, Ibu, Adik, dan keluarga tersayang yang telah banyak memberikan doa, motivasi dan dorongan dalam penyelesaian skripsi ini.
5. Sahabat dan Teman-teman FTI 2015 terimakasih atas kebersamaan selama ini yang telah memberikan dorongan dan semangat besar dalam mengisi keseharian.

Penulis menyadari bahwa dalam penyusunan skripsi ini banyak kekurangannya. Oleh karena itu penulis mengharapkan saran dan kritik untuk menjadikan lebih baik.

Yogyakarta, 22 Juli 2019

Penulis,

(Sugeng Widodo)

PROTOTYPE DESAIN KEAMANAN LOGIN MENGGUNAKAN BIOMETRIK WAJAH DENGAN METODE EIGENFACE

ABSTRAK

Keamanan *login* menjadi masalah utama apabila menggunakan suatu perangkat yang terhubung dalam jaringan luar atau jaringan internet. Karenanya dilakukan suatu pengamanan data menggunakan biometrik wajah.

Eigenface adalah suatu metode pengenalan wajah yang berdasarkan pada algoritma *Principal Component Analysis* (PCA). Secara singkat prosesnya adalah citra direpresentasikan dalam sebuah gabungan *vector* yang dijadikan satu matriks tunggal. Dari matriks tunggal ini akan diekstraksi suatu ciri utama yang akan membedakan antara citra wajah satu dengan citra wajah lainnya.

Dalam penggunaan sistem *login* biometrik wajah ini adalah *user* mendaftar ke sistem kemudian wajah *user* akan di train sehingga *user* akan dikenali oleh sistem untuk keperluan *login*. Pada saat *user* melakukan *login* maka data wajah *user* akan diproses secara *real time* dan dicocokan dengan data yang ada pada *database* sehingga apabila data cocok maka *user* akan berhasil *login*.

Jarak wajah dengan webcam standarnya adalah 50-60 cm, sedangkan tingkat pencahayaan minimal wajah dapat dikenali adalah 5 *lux*, dan sudut kemiringan wajah yang masih dapat dikenali sistem adalah 40°.

Kata kunci: Biometrik wajah, *Eigenface, Principal Component Analysis* (PCA), *Real Time, Lux Meter*.

DAFTAR ISI

HALAMAN PENGESAHAN ii
HALAMAN PERNYATAAN PERSETUJUAN iii
HALAMAN PERNYATAAN ORSINILITAS KARYA iv
HALAMAN PERSEMBAHAN v
HALAMAN MOTTO vi
KATA PENGANTAR vii
ABSTRAK viii
DAFTAR ISI viii
DAFTAR TABEL xi
DAFTAR GAMBAR xii
DAFTAR LAPIRAN xiii
BAB I PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Rumusan Masalah 2
1.3 Tujuan 2
1.4 Manfaat 2
1.5 Batasan Masalah 2
BAB II TINJAUAN PUSTAKA DAN LANDASAN TEORI 3
2.1 Tinjauan Pustaka 3
2.2 Landasan Teori 5
2.2.1 Biometrik 5
2.2.2 Pengamanan dengan Biometrik 7
2.2.3 Jenis Biometrik 9
2.2.4 Klasifikasi Biometrika 11
2.2.5 Cara Kerja Sistem Biometrika 12
2.2.6 Pengukuran Kinerja Alat Biometrika 13
2.2.7 FAR, FRR dan EER 14
2.2.8 Biometrik Wajah 16
2.2.9 Teknik Identifikasi Wajah 17
2.2.10 Pengenalan Pola 19

2.2.11 Citra RGB 19
2.2.12 Algoritma Eigenface 20
2.2.12.1 Segmentasi Citra 20
2.2.12.2 Perhitungan Principal Component Analysis (PCA) 21
BAB III METODOLOGI 30
3.1 Bahan Penelitian 30
3.2 Alat Penelitian 30
3.3 Jalan Penelitian 30
3.3.1 Tahap Analisis 31
3.3.2 Tahap Desain 31
3.3.3 Tahap Implementasi dan Pengujian 31
3.3.4 Tahap Kesimpulan dan Saran 31
3.4 Perancangan Sistem 32
3.4.1 Tabel 32
3.4.2 Desain Sistem 32
3.4.3 Desain Algoritma Identifkasi Image 33
3.4.4 Desain Antar Muka 34
3.4.4.1 Rancangan halaman utama (Front End) 34
BAB IV ANALISIS DAN PEMBAHASAN 36
4.1 Hasil Penelitian 36
4.1.1 Hasil Implementasi Desain Program 36
4.1.1.1 Tampilan Halaman Awal (Front End) 36
4.2 Analisis dan Pembahasan 37
4.2.1 Uji Fungsionalitas 37
4.2.1.1 Proses Tambah Anggota 37
4.2.1.2 Proses Train Wajah 39
4.2.1.3 Proses Tes Login 42
4.2.1.4 Pengujian Aplikasi Halaman Utama (Front End) 46
BAB V PENUTUP 48
5.1 Kesimpulan 48
5.2 Saran 48

DAFTAR PUSTAKA 49
LAMPIRAN xiv

DAFTAR TABEL

Tabel 2. 1 Klasifikasi biometrika (Nugroho E. , 2009) 11
Tabel 3. 1 Alat Penelitian 30
Tabel 3. 2 Tabel Biodata 32
Tabel 4. 1 Pengujian Proses *Train* Wajah 40
Tabel 4. 2 Pengujian Proses *Train* Wajah (Lanjutan) 41
Tabel 4. 3 Jarak Standar 42
Tabel 4. 4 Tes Sudut Kemiringan Wajah 43
Tabel 4. 5 Tes *Gesture* Wajah dan *Accerssoris* 44
Tabel 4. 6 Tes Tingkat Pencahayaan 45

DAFTAR GAMBAR

Gambar 2. 1 Verifikasi / Otentikasi. (Nugroho E. , 2009) 8
Gambar 2. 2 Rekognisi/ Identifikasi, (Nugroho E. , 2009) 8
Gambar 2. 3 Pendaftaran (Nugroho E. , 2009)..... 12
Gambar 2. 4 Pemindaian (Nugroho E. , 2009)..... 13
Gambar 3. 1 Blok Diagram Perancangan Sistem *Login*..... 31
Gambar 3. 2 Flowchart Sistem Keseluruhan..... 33
Gambar 3. 3 Desain Algoritma Identifikasi Image dengan Eigenface..... 34
Gambar 3. 4 Halaman utama (*front end*)..... 35
Gambar 4. 1 Halaman Utama (Front End) 36
Gambar 4. 2 Proses tambah anggota 38
Gambar 4. 3 Data anggota disimpan pada *database* 38
Gambar 4. 4 Proses *Train* Wajah 39
Gambar 4. 5 ID pengguna terlihat dikamera 40

DAFTAR LAPIRAN

Lampiran A. Biodata xiv
Lampiran B. Data Train Wajah xv
Lampiran C. Source Code MainForm.cs xxvii
Lampiran D. Source Code Recognizer.cs xxxiv
Lampiran E. Scan Form Bimibingan Skripsi xxxviii
Lampiran F. Scan Form Revisi xxxix

BAB I
PENDAHULUAN

1.1 Latar Belakang

Masalah keamanan merupakan masalah utama yang harus di perhatikan apabila menggunakan suatu perangkat yang terhubung dalam jaringan luar atau jaringan internet. Untuk itu *user* sering sekali diminta untuk membuat *password* yang unik dan susah ditebak oleh orang lain.

Password adalah kumpulan karakter atau string yang digunakan oleh pengguna jaringan atau sebuah sistem operasi yang mendukung banyak pengguna (*multiuser*) untuk memverifikasi identitas dirinya kepada sistem keamanan yang dimiliki oleh jaringan atau sistem tersebut. Saat *user* akan melakukan programming atau proses pembuatan sebuah aplikasi sering sekali *user* menjumpai atau mengharuskan *user* untuk membuat form login untuk keperluan tertentu. Mengingat banyaknya masalah yang timbul apabila *password* tidak *terenkripsi* dengan baik maka tidak menutup kemungkinan apabila akun terkena *hack* dari orang-orang yang nakal.

Karenanya *user* memerlukan pengamanan tambahan guna melakukan mengantisipasi hal-hal yang tidak diinginkan tersebut. Salah satunya adalah dengan menggunakan biometrik sebagai keamananya. Biometrik merupakan pengenalan identitas seseorang berdasarkan bentuk fisiknya contohnya wajah, sidik jari, garis telapak tangan, retina mata, ataupun suara/wicara manusia itu sendiri.

Pengenalan biometrik pada wajah merupakan salah satu bidang penelitian yang penting, dan dimanfaatkan dibidang komersial maupun bidang hukum. Teknik pengenalan wajah pada saat ini mengalami kemajuan berati. Melalui pengembangan suatu teknik seperti *eigenface* untuk ekstraksi karakteristik wajah. Variasi dalam pencahayaan dan pose seharusnya diminimalisasi untuk pengenalan wajah secara optimal, dengan demikian *preprocessing* dari database dan percobaan harus dilakukan. Dalam penelitian ini akan menggunakan karakteristrik ekstraksi dari wajah, dan menggunakan metodologi *Eigenface Face Recognition.*

1.2 Rumusan Masalah

Rumusan masalah dalam penelitian ini diantaranya adalah sebagai berikut.

1. Bagaimana analisis biometrik wajah dengan menggunakan metode *eigenface* ?
2. Bagaimana desain perancangan sistem *login* pada biometrik wajah ?
3. Bagaimana implementasi kode pada biometrik wajah menggunakan metode *eigenface* ?
4. Bagaimana hasil unjuk kerja pada biometrik wajah menggunakan metode *eigenface* ?

1.3 Tujuan

Tujuan penelitian ini adalah untuk membuat *prototype* keamanan *login* pada sistem *login* menggunakan biometrik wajah.

1.4 Manfaat

Manfaat penelitian ini secara umum adalah dapat digunakan sebagai pengamanan sistem *login* degan biometrik pada wajah.

1.5 Batasan Masalah

Adapun yang menjadi batasan masalah dalam penelitian ini adalah:

1. Data train wajah yang disimpan berupa citra *grayscale* bertipe bitmap (*.bmp).
2. Data *train* wajah yaitu lebih dari satu wajah.
3. Data wajah Menggunakan 15 wajah.
4. Sistem hanya untuk membedakan *user* yang sudah di *train* wajahnya.

BAB II
TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Penelitian yang dilakukan oleh (Hasmin, 2016) yang berjudul "Aplikasi rekam kehadiran dengan deteksi wajah menggunakan metode *eigenface* pada kejaksaan tinggi sulawesi selatan". Beberapa bagian tubuh manusia bisa menjadi identitas pribadi seseorang yang menjadikan seorang manusia berbeda dengan manusia lainnya diantaranya adalah sidik jari, DNA, retina mata dan wajah, tentu saja dibutuhkan perangkat khusus untuk dapat mengenali dan mengubah data tersebut menjadi data yang dapat dikenali manusia, seperti teknologi *finger print* dan pemindai retina.

Penelitian yang dilakukan oleh (Andarinny, Widodo, & Adi, 2017) Dalam Penelitianya dengan judul "Perancangan sistem identifikasi biometrik jari tangan menggunakan *Laplacian of Gaussian* dan ektraksi kontur". Pada penelitian ini dilakukan perancangan sistem untuk identifikasi atau mengenali seseorang menggunakan biometrik khususnya tangan menggunakan deteksi tepi *Laplacian of Gaussian* dan Ekstraksi Kontur. Sistem identifikasi dengan biometrik memiliki banyak kelebihan karena selalu melekat pada tubuh dan tidak dapat dipisahkan dengan manusia. Tahapan yang dilakukan dalam pengolahan citra berupa konversi citra RGB menjadi biner menggunakan *threshodling*, deteksi tepi menggunakan *Laplacian of Gaussian* serta ekstraksi kontur menggunakan representasi kode rantai. Hasil dari ekstraksi kontur merupakan banyaknya *pixel* yang dihitung pada setiap segmen pada jari tangan yang dibagi menjadi 8 bagian. Proses pencocokan pada sistem dilakukan dengan menggunakan jarak *euclidean*, yaitu menghitung jarak terkecil kemiripan antara citra uji dengan basis data.

Penelitian yang dilakukan oleh (Apriadi, Michrandi, & Azmi, 2016) Dalam penelitianya mengembangkan "Perancangan otentikasi sidik jari pada *biometric payment design of authentication fingerprint for biometric payment*". Saat ini sudah banyak digunakan kunci berupa biometrik anggota tubuh manusia. Anggota tubuh manusia yang sering digunakan adalah pola retina, garis telapak

tangan, dan sidik jari. Sidik jari mempunyai sifat yang unik untuk setiap individu dan mempunyai sifat yang konsisten. Sistem pembayaran menggunakan otentikasi sidik jari jauh aman dan mudah digunakan, tanpa menggunakan *password* atau *PIN* untuk mengingat jika dibandingkan dengan sistem pembayaran kartu kredit.

Penelitian yang dilakukan oleh (Agustina, Fauziah, & Gunaryati, 2016) Dalam penelitianya yaitu "Biometrik pola suara dengan jaringan saraf tiruan" Metode yang digunakan dalam penelitian ini adalah biometrik pengenalan identitas berdasarkan karakteristik fisik. Pengenalan pola suara biometrik yang memiliki biaya rendah. Penelitian ini merancang pola penggunaan suara biometrik. sinyal input suara dalam bentuk suara yang direkam dengan durasi sekitar dua (2) detik pada sistem dan sinyal dekomposisi menggunakan transformasi *wavelet diskrit*. sistem verifikasi menggunakan ekstraksi karakteristik Koefisien *Sub Band* berdasarkan parameter *cepstral (SBC)* dengan menggunakan jumlah koefisien 12.

Penelitian yang dilakukan oleh (Auliannisa, Rizkia Dwi; Suratman, Fiky Yosef; Rizal, Achmad, 2017) Dalam penelitianya yaitu "Deteksi Katarak Menggunakan Metode *Transformasi Hough* Berbasis Android" Merancang sebuah aplikasi android yang dapat mendeteksi penyakit katarak dengan Transformasi *Hough* sebagai metode ekstrasi ciri citra pada bagian yag dikehendaki atau menggunakan metode *Region of Interest (ROI)* yang biasa digunakan pada pengolahan citra medis. Karena Transformasi Hough merupakan algoritma pada *Computer Vision* yang digunakan untuk mencari bentuk geometrik seperti garis dan lingkaran. Operator ini dapat menentukan lingkaran dalam dan luar pada lensa mata. Kemudian dilakukan pengklasifikasian citra menggunakan metode *K-Nearest Neighbor (KNN)* dengan parameter penempuan titik usat lingkaran dan perbandingan jumal *pixel* citra keabuan pada lingkaran yang didapat.

Berdasarkan beberapa penelitian yang sudah ada, dalam penelitian ini yaitu menunjukan perancangan keamanan sistem login menggunakan biometrik wajah, yang nantinya akan digunakan sebagai verifikasi pada saat login setelah *user* mengisi *username* dan *password* pada form login aplikasi.

2.2 Landasan Teori

2.2.1 Biometrik

Biometrik adalah metode otomatis untuk mengenali seseorang berdasarkan karakteristik *fisiologis* atau perilaku. Teknologi biometrik menjadi dasar dari berbagai pilihan yang sangat aman solusi identifikasi dan verifikasi pribadi. Sebagai syarat penting untuk menekan tingkat penipuan dan pelanggaran keamanan, teknologi biometrik menjadi persyaratan pribadi sebagai identifikasi dan verifikasi pribadi yang aman menjadi jelas. Transaksi berbasis biometrik dapat menjadi solusi untuk transaksi keuangan rahasia dan privasi data pribadi. (Kalyani, 2017).

Biometriks *Authentification* dalam *security* adalah hal yang sangat penting untuk menjaga keamanan data, namun sudah banyak teknologi yang diterapkan untuk mejaga keautentikan tersebut, akan tetapi hal itu banyak kendala dalam penerapanya dan masih kurang memberikan perlindungan yang aman. Teknologi biometrik menawarkan *autentikasi* secara *biologis* memungkinkan sistem dapat mengenali penggunanya lebih tepat.

Terdapat beberapa metode diantaranya: *fingerprint scanning*, retina *scanning,* dan DNA *scanning*. Dua metode terakhir masih dalam taraf penelitian, sedangkan *fingerprint scanning* saat ini telah digunakan secara luas dan digunakan bersama-sama dengan *smartcard* dalam proses *autentikasi*.

Biometrik secara teoritis dapat lebih efektif untuk mengindentifikasi pribadi seseorang karena biomatrik mengukur karakteristik masing-masing pribadi untuk membedakan setiap orang. Tidak seperti dengan metode *identifikasi konvensional* yang menggunakan sesuatu yang pengguna punyai, misalnya kartu indentitas untuk akses masuk ke suatu bangunan, atau suatu yang anda ketahui, seperti *password* untuk *logon* ke *system* komputer dan lain-lain. Ketika digunakan untuk indentifikasi pribadi, teknologi biometrik mengukur dan menganalisa karakteristik tingkah laku dan fisiologis manusia. Mengindentifikasi karakteristik fisiologis seseorang yang didasarkan pada pengukuran langsung bagian dari *body–fingertips, hand geometry, facial geometry* dan *eye retinas* serta *irises*.

Ditengah kemajuan jaringan komunikasi dan mobilitas alat, dibutuhkan sebuah metode yang andal untuk mengidentifikasi seseorang. Dalam perkembangan sistem identifikasi sampai dengan awal tahun 2000, terdapat 2 metode yang banyak dipakai, yaitu :

1. **Sistem identifikasi berdasarkan kepemilikan (*possession based* atau "*what you have*").**

   Proses keamanan didasarkan pada sesuatu benda ("*token*") yang dimiliki seseorang, misalkan *credit card*. Kelemahan sistem ini adalah apabila alat identifikasi tersebut hilang, maka orang lain yang menemukanya dapat menyalahgunakan fungsi alat untuk kepentingan pribadi.

2. **Sistem identifikasi berdasarkan pengetahuan (*knowledge based* atau *"what you know"*).**

   Biasanya, sistem otentifikasi jenis ini menggunakan *password* sebagai media utamanya. Kelemahan sistem ini adalah apabila *password* terlalu pendek, hal ini beresiko untuk dapat ditebak oleh orang lain, dan bila terlalu panjang akan mudah terlupa. Menurut para ahli keamanan, saat ini sudah ada *cracking tool* yang mampu memindai kata maupun menebak *password* berupa kombinasi huruf dan angka. Sebagai contoh, *"LoftCrack"*, salah satu program penjebol sandi. Dengan program ini, hanya dibutuhkan waktu sekitar 48 jam untuk dapat menemukan seluruh arsip *password* dalam suatu perusahaan. Untuk mengatasi masalah tersebut, para ahli mencari cara lain yang lebih aman. Cara yang kini telah dikembangkan adalah dengan menggunakan biometrika, yaitu suatu keadaan fisik tertentu ataupun suatu perilaku tertentu unik yang ada pada seseorang. Teknologi biometrika merupakan sebuah teknologi baru yang memiliki fungsi utama untuk mengenali manusia melalui sidik jari, mata wajah, atau bagian tubuh yang lain. Biometrika berasal dari kata bios, yang berarti kehidupan dan *metron*, yang berarti ukuran. Biometrika merupakan teknologi untuk mengenali seseorang secara unik. Dengan didukung oleh faktor biaya penggunaan teknologi yang sudah semakin terjangkau dan *fleksibilitas* teknologi ini, teknologi ini dirasa akan dapat menggusur penggunaan kata sandi (*Password*) ataupun kartu (misal *credit card*) sebagai alat otentikasi maupun identifikasi. Teknologi identifikasi ini juga disebut

dengan teknologi "*what you are*". Keunggulan sistem biometrika adalah sebagai berikut:

1. Biometrika tidak dapat hilang (fisik) atau terlupa (perilaku), kecuali karena faktor trauma.
2. Biometrika sulit untuk di-kopi/ ditiru ataupun dipindah tangankan ke pihak lain.
3. Biometrika mengharuskan orang yang bersangkutan untuk ada ditempat identifikasi dilakukan.

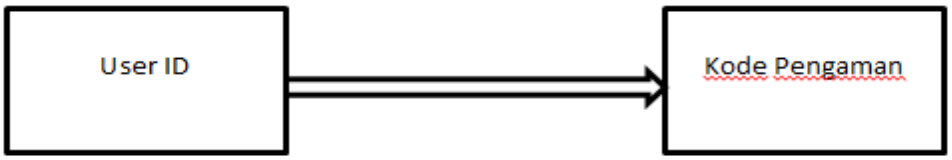
Penggunaan biometrika yang dikombinasi dengan *password* atau kartu identitas akan semakin meningkatkan keandalan kemanan identifikasi. (Nugroho E. , 2009).

2.2.2 Pengamanan dengan Biometrik

Di dalam suatu teknologi akses, selalu terjadi dua buah proses otomatis untuk menjawab dua pertanyaan yang penting untuk pengenalan identitas seseorang, yaitu "Siapakah Anda?", dan " Apakah Anda benar seperti yang Anda katakan ?" pertanyaan pertama mewakili fungsi identifikasi, dan pertanyaan kedua mewakili fungsi *otentikasi* atau *verifikasi* (pembuktian). Sebagai pendekatan umum untuk memperoleh suatu akses dapat digunakan sebagai metode pengenalan satu factor (*single-factor security*). Sebagai contoh kunci rumah atau *password.* Dua factor (two-*factor security*), yang dianggap lebih andal. Contoh paling umum adalah kartu Anjungan Tunai Mandiri (ATM). Dalam teknologi ini digunakan kartu atm dan nomor Pin sebagai dua alat yang saling tak terpisahkan untuk dapat mengakses sesuatu, misalnya dalam contoh ATM adalah mengakses nomor rekening tabungan. Dalam sistem pengamanan ini, kedua tanda yang harus berada pada peminta akses. Jika hanya ada kartu saja atau pin saja, sistem tidak akan bekerja. Kartu ini berfungsi sebagai "*user id*" (*user identification*). Data-data pemilik kartu (missal kartu ATM) sudah dicatat oleh penerbit kartu (Misalnya bank), baik nama, alamat, tanggal lahir, dan berada dalam *database* penerbit kartu. Pada dasarnya terdapat dua proses pengamanan dalam penggunaan sistem pengamanan dua faktor, antara lain:

1. Verifikasi untuk keperluan Otentikasi.

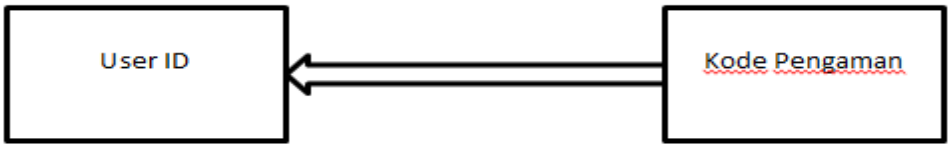
Proses ini bertujuan untuk menjawab "Betulkah orang ini sesuai dengan proses identitas yang dia katakana?". Pada proses ini, orang (subyek) memasukan atau menunjukan "*user-id*"nya (missal kartu ATM), lalu diikuti dengan memasukan kode pengamanya (missal pin, atau *password*, atau sidik jarinya). Apabila terdapat kesesuaian antara keduanya, maka akses dapat diberikan. Pada jenis ini, Proses dilakukan secara 1 terhadap 1 (*one to one*). Lihat Gambar 2.1. Proses ini disebut otentikasi atau verifikasi.



Gambar 2. 1 Verifikasi / Otentikasi. **(Nugroho E. , 2009)**

2. Rekognisi untuk keperluan Identifikasi.

Proses ini dilakukan untuk menjawab "Siapakah dia?". Dalam proses ini, orang (*subjek*) memasukan atau menunjukan kode pengamananya (misal pin, atau *password*, atau sidik jarinya). Setelah itu, alat (misal komputer) melacak *database* yang ada, dan mencari objek yang sesuai. Bila kesesuaian tersebut diketemukan, identitas pemilik kode pengamanan tersebut akan ditampilkan. Pada jenis ini, proses dilakukan terhadap n (banyak), atau istilah lainya "*one to many*". Lihat Gambar 2.2. Proses ini disebut identifikasi. Pada dasarnya. Proses "*screening*" (pencocokan seseorang dengan daftar pencarian yang ada) merupakan proses identifikasi. (Nugroho E. , 2009).



Gambar 2. 2 Rekognisi/ Identifikasi, **(Nugroho E. , 2009)**

2.2.3 Jenis Biometrik

Para ahli telah sepakat bahwa pada dasarnya, setiap organ tubuh seseorang bersifat unik. Tidak ada dua orangpun yang mempunyai bagian tubuh yang sama. Hal inilah yang melandasi perkembangan sistem biometrika. Beberapa hal yang sudah dikembangkan menjadi alat biometrika antara lain :

1. Biometrika Fisiologis atau Biometrika Statis

Biometrika fisiologis merupakan jenis sistem biometrika yang dikembangkan Berdasarkan keberadaan fisik atau fisiologis seseorang, yang meliputi:

a. Sidik jari (*fingerprint*). Alur tonjolan (*Ridge*) dan lembah (*Valley*) pada permukaan kulit digunakan sebagai alat identifikasi.
b. Ukuran jari (*Finger Geometry*). Ukuran 3 dimensi jari tangan sebagai alat identifikasi.
c. Ukuran tangan (*hand geometry*). Ukuran 3 dimensi tangan digunakan sebagai alat identifikasi, antara lain ukuran panjang jari, dan lebar telapak tangan.
d. Wajah (*Face recognition*). Pola fitur wajah digunakan sebagai alat identifikasi.
e. Iris mata. Iris adalah bagian hitam (kalau dinegara barat berwarna biru) yang melingkar pada pola mata.
f. Retina mata. Retina adalah bagian mata yang berfungsi untuk menangkap cahaya.
g. Telinga. Ukuran telinga dipergunakan sebagai alat identifikasi.
h. Vena tangan (*Hand vein*). Pola pembuluh darah orang juga dapat dipergunakan untuk identifikasi.
i. Bau badan. Bau badan seseorang ternyata unik dan dapat dipergunakan untuk alat identifikasi.
j. DNA (*Desoxi ribose nuclead acid*). DNA seseorang digunakan sebagai alat identifikasi.
k. Panas wajah.
l. Sidik telapak tangan.

2. Biometrika Perilaku atau Biometrika Dinamis

a. Suara (*Voice recognition*). Identifikasi menggunakan analisis spectrum suara.
b. Tanda tangan (*signature recognition*). Pola, bentuk dan tekanan tanda tangan seseorang dipergunakan sebagai alat identifikasi.
c. Cara mengetik (*typing recognition*). Cara mengetik seseorang juga dapat menjadi alat identifikasi.
d. Gaya berjalan (*gait*). Cara berjalan seseorang juga dapat digunakan untuk alat identifikasi. (Nugroho E. , 2009).

Sistem identifikasi biometrika juga dikelompokan menjadi:

a. Kelompok "*vision bases*", yaitu sistem identifikasi berbasiskan pada penglihatan. Organ tubuh yang termasuk kelompok ini antara lain adalah sidik jari, ukuran jari, ukuran tangan, wajah, iris mata, retina mata, telinga, vena, DNA, panas wajah, dan sidik telapak tangan.
b. Kelompok "*non vision based*", yaitu sistem identifikasi yang berbasiskan bukan pada penglihatan. Organ yang termasuk dalam kelompok ini antara lain bau badan, suara, tanda tangan, cara mengetik, dan gaya berjalan.

Beberapa sidat yang harus dipertimbangkan dalam memilih alat biometrika adalah:

a. *Universalitas*. Hal ini berarti seberapa jauh sifat biometrika dimiliki semua orang. Sebagai contoh, sidik jari dimiliki oleh setiap orang yang normal.
b. Keunikan. Hal ini berarti seberapa jauh sifat biometrika mampu membedakan seseorang dengan orang lain. Sebagai contoh, sidik jari seseorang dengan orang yang lain.

c. Permanen. Hal ini berarti seberapa jauh sifat biometrika dipengaruhi oleh usia seseorang. Sebagai contoh, sidik jari seseorang tidak berubah karena usia. Perubahan terjadi apabila terjadi kerusakan pada sidik jari.
d. *Kolektabilitas*. Hal ini berarti seberapa jauh sifat biometrika mudah diperoleh atau diukur dari seseorang. Sebagai contoh, sidik jari seseorang mudah di scan.
e. Kinerja. Hal ini berarti seberapa teliti dan cepat alat biometrika bekerja. Sebagai contoh, peralatan sidik jari bekerja dengan cepat dan andal.
f. *Akseptabilitas*. Hal ini berarti seberapa jauh alat biometrika diterima oleh masyarakat. Sebagai contoh, masyarakat tidak keberatan untuk di scan terhadap sidik jarinya.
g. *Circumvention*. Hal ini berarti seberapa sulit alat biometrika untuk dikecoh/ ditipu. Sebagai contoh, peralatan sidik jari relatif tidak mudah dikecoh oleh orang bermaksud tidak baik.

2.2.4 Klasifikasi Biometrika

Klasifikasi biometrika menurut Dr. Ir. Eko Nugroho, M.Si. dalam bukunya yaitu biometrika mengenal sistem identifikasi masa depan akan dijelaskan pada Tabel 2.1.

Tabel 2. 1 Klasifikasi biometrika **(Nugroho E. , 2009)**

Biometrika	**Universitas**	**Keunikan**	**Permanen**	**Kolektabilitas**	**Kinerja**	**Akseptabilitas**	**Circumvention**
Wajah	Baik	Buruk	Sedang	Baik	Buruk	Baik	Baik
Sidik jari	Sedang	Baik	Baik	Sedang	Baik	Sedang	Baik
Geometri tangan	Sedang	Sedang	Sedang	Baik	Sedang	Sedang	Sedang
Cara mengetik	Buruk	Buruk	Buruk	Sedang	Buruk	Sedang	Sedang
Vena tangan	Sedang	Sedang	Sedang	Sedang	Sedang	Sedang	Baik
Iris	Baik	Baik	Baik	Sedang	Baik	Buruk	Baik
Retina	Baik	Baik	Sedang	Buruk	Baik	Buruk	Baik

Tabel 2. 2 Klasifikasi biometrika **(Nugroho E. , 2009)** (Lanjutan)

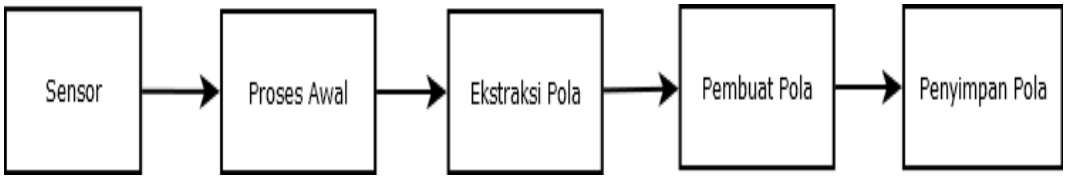
Biometrika	**Universitas**	**Keunikan**	**Permanen**	**Kolektabilitas**	**Kinerja**	**Aksept abilitas**	**Circumvention**
Tanda tangan	Buruk	Buruk	Buruk	Baik	Buruk	Baik	Buruk
Suara	-	Buruk	Buruk	Sedang	Buruk	Baik	Buruk
Panas wajah	Baik	Baik	Buruk	Baik	Sedang	Baik	Baik
Bau badan	Baik	Baik	Baik	Buruk	Buruk	Sedang	Buruk
DNA	Baik	Baik	Baik	Buruk	Baik	Buruk	Buruk
Cara berjalan	Sedang	Buruk	Buruk	Baik	Buruk	Baik	Sedang
Telinga	Sedang	Sedang	Baik	Sedang	Sedang	Baik	Sedang

2.2.5 Cara Kerja Sistem Biometrika

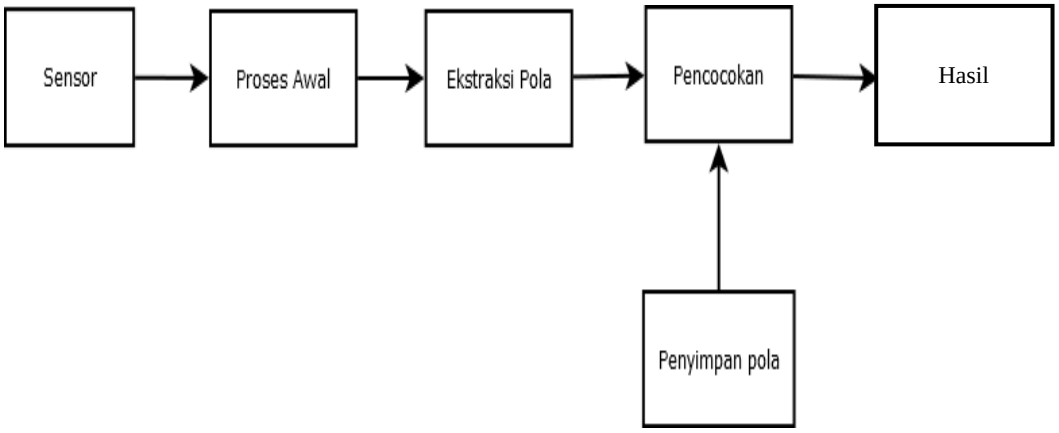
Proses biometrika terdiri dari:

1. Proses Pendaftaran (*enrolment*). Dalam proses ini, data objek, misalnya rekaman sidik jari, disimpan pada *database*.
2. Proses pemindaian (*Scanning*). Dalam proses ini, data hasil pemindaian dicocokan dengan *database* yang ada.

Cara kerja *enrollment* maupun *scanning* dilihat pada Gambar 2.3. dan Gambar 2.4.



Gambar 2. 3 Pendaftaran **(Nugroho E. , 2009)**



Gambar 2. 4 Pemindaian **(Nugroho E. , 2009)**

Keterangan:

Sensor : Alat pembaca dari objek.

Pemroses awal : Proses persiapan sebelum data diolah, misalnya proses pemindahan dari citra ke bentuk digital.

Penggali fitur : Proses untuk mengidentifikasi fitur-fiur tertentu yang akan diolah dari objek hasil pembacaan.

Pembangkit pola : Proses untuk menyusun pola-pola sesuai dengan karakteristik metode biometrika yang dipakai.

Pnyimpanan pola *database* tempat penyimpanan pola-pola hasil pendaftaran (*enrollment*)

Pencocokan : Proses pencocokan antara hasil pembacaan dengan database yang ada.

2.2.6 Pengukuran Kinerja Alat Biometrika

Kinerja suatu alat biometrika dapat diukur dengan beberapa parameter berikut:

1. *False accept Rate* (FAR) atau *False Match Rate* (FMR), yaitu peluang alat identifikasi untuk menyatakan sah terhadap data masuk yang sebenarnya tidak sah. Jadi, sistem menyatakan adanya kesesuaian antara data masuk dan data dalam *database* yang sebenarnya tidak sesuai. Jadi dalam kasus ini,

orang yang tidak sah diterima masuk sebagai orang yang sah. FAR dalam bahasa ilmu statistik termasuk dalam kelompok kesalahan identifikasi *Type* II, yaitu kesalahan menganggap benar sesuatu yang sebenarnya salah.

2. *False Eject Rate* (FER) atau *False Nonmatch Rate* (FNMR), yaitu peluang alat tersebut menyatakan tidak sah terhadap data masuk yang sebenarnya sah. Jadi, sistem menyatakan tidak adanya kesesuaian antara data masuk dan data dalam *database* yang sebenarnya sesuai. Dalam kasus ini, orang yang sah ditolak dan dianggap sebagai orang yang tidak sah. FRR dalam bahasa ilmu statistik termasuk dalam kelompok kesalahan *Type* I, yaitu kesalahan menganggap salah atas sesuatu yang sesungguhnya benar. Kesalahan *Type* I dinilai lebih ringan dari kesalahan *Type* II.
3. *Receiver* (atau *relative*) *operating characteristics*. (ROC), merupakan tingkat kepekaan alat, yang tentu saja mempengaruhi oleh FAR dan FRR. Dalam praktiknya, tingkat kepekaan ini dapat diatur. Pengaturan ini akan menyebabkan *trade off* antara FAR dan FRR. Diagram *trade off* antara FAR dan FRR akibat kepekaan alat (misalnya pengaturan tingkat ambang "threshold" alat) disebut sebagai diagram ROC.
4. *Equal Error Rate* (EER), merupakan tingkat persentasi yang menunjukan bahwa bilangan FAR dan FRR sama. Untuk membandingkan 2 buah alat biometrika secara cepat, maka alat yang bilangan EERnya rendah dianggap lebih akurat.
5. *Failure to Enroll Rare* (FTE atau FER), merupakan tingkat persentase orang gagal mendaftar (*enroll*) ke sistem.
6. *Failure to Capture Rate* (FTC), merupakan tingkat persentase orang gagal terdeteksi oleh sistem.
7. *Template Capacity*, merupakan jumlah maksimum orang yang dapat tersimpan pada alat sebagai data. (Nugroho E. , 2009)

2.2.7 FAR, FRR dan EER

Parameter terpenting dari sistem biometrika adalah FAR, FRR dan EER. Dalam sistem biometrika, digunakan skor untuk menyatakan kemiripan antara

contoh dalam database dengan objek yang dipindai. Semakin tinggi nilai skor maka tingkat kemiripanya juga semakin tinggi. Tingkat skor yang dinyatakan dalam batas minimal kemiripan disebut ambang atau *threshold*. Apabila tingkat kemiripan antara objek yang dipindai dengan contoh dalam *database* melebihi ambang yang ditetapkan, maka dinyatakan objek yang dipindai adalah sama dengan contoh yang tersimpan dalam database. Secara teori, skor yang akan dicapai oleh objek asli akan selalu lebih tinggi dari skor yang dicapai oleh pemalsu. Meskipun demikian, dapat ditetapkan suatu ambang skor yang akan dapat memisahkan antara objek yang asli dengan objek pemalsu.

Meskipun demikian, karena alasan tertentu karena keterbatasan alat, tidak jarang skor yang dicapai pemalsu justru lebih tinggi dari skor asli, sehingga berapapun ambang yang ditetapkan, terjadinya *error* pada sistem masih sering terjadi. Sebagai contoh, misal ditetapkan suatu skor ambang yang sangat tinggi. Hal ini berakibat tidak seorang pemalsupun dapat melampaui batas tersebut. Tidak sebuah polapun yang dihasilkan oleh pemalsu yang akan lolos dan diterima sistem. Akan tetapi, pada keadaan ini, pola hasil pemindaian dari objek yang aslipun juga akan bisa ditolak ketika hasil skornya berada dibawah ambang batas yang ditetapkan. Kondisi ini disebut sebagai kesalahan menolak (FRR=*False Rejection Rate*). Dilain pihak bisa juga menetapkan suatu ambang batas yang sedemikian rendah sehingga semua pola objek asli pasti akan diterima. Akan tetapi. Dalam keadaan seperti ini akan sangat mungkin pola hasil pemindaian milik pemalsupun akan lolos diterima sistem. Kondisi seperti ini disebut kondisi penerimaan yang salah (FAR=*False Acceptance Rate*).

Kemungkinan lain yang bisa terjadi, pada suatu titik tertentu, kedua kesalahan FAR dan FRR bisa terjadi. Bila nilai ambang terlalu tinggi, maka potensi terjadinya FRR (kesalahan menolak yang benar) akan sangat tinggi, dan semakin rendah nilai ambang, maka FRR akan semakin rendah juga. Nilai FRR=1 artinya bahwa semua objek asli akan ditolak, dan bila nilai FRR=0, artinya semua objek asli akan diterima. FRR dihitung dari pembagian antara jumlah objek asli ditolak dibagi dengan total jumlah objek asli. Pada suatu titik ambang tertentu,

nilai FRR akan sama dengan nilai FAR, titik ini disebut titik EER (*Equal Error Rate*). (Nugroho E. , 2009).

2.2.8 Biometrik Wajah

Setiap individu di dunia memiliki wajah yang unik dan bahkan untuk dua anak kembar yang mata manusia hampir sulit untuk membedakanya. Mungkin sesuatu yang kecil seperti penempatan alis yang sedikit unik, lebarnya mata, atau luasnya hidung. Ada penanda yang pasti untuk dapat mengaktifkan pengenalan biometrik ini dalam sepersekian detik untuk dapat mengenali keunikan setiap individu untuk memeriksa keunikan wajah dari setiap elemen. Dengan cara ini dapat memaksimalkan penggunaan *gadget* untuk menjamin itu untuk dapat memperoleh hak akses penggunaan. Pengakuan orang-orang dalam pemanfaatan kualitas wajah tidak dapat disangkal menyadari pentingnya sistem pengenalan wajah . (Kalyani, 2017).

Seiring kemajuan teknologi, sistem pengenalan wajah semakin luas dipergunakan. Salah satu teknologi terapan yang dikembangkan adalah bahwa komputer dapat mengenali wajah pemiliknya, sehingga bila sebuah komputer dibuka, komputer tersebut dapat segera mengenali pengguna komputer tersebut. Tentu saja jika yang membuka bukan pemiliknya, maka komputer tersebut tidak akan membuka akses. Ada banyak aspek dari wajah yang dapat dipergunakan sebagai alat pembeda seseorang dengan yang lainya, antara lain:

1. **Bentuk geometri wajah (*Face morphometrics*)**

   Dengan metode identifikasi melalui geometri wajah, maka ditetapkan fitur-fitur wajah yang akan dipergunakan sebagai patokan pengukuran, seperti mata, hidung, mulut dan lain-lain. Dengan mengukur jarak antar fitur tersebut, akan dapat ditetapkan karakteristik yang unik dari ukuran geometric seseorang.

2. **Pola kulit wajah (*Face skin surface*)**

   Dengan metode identifikasi melalui kulit wajah, kontur kulit seseorang dipergunakan sebagai alat identifikasi seseorang.

3. **Temperature pada wajah (*Face thermogram*)**

Karena wajah setiap orang mempunyai suhu tertentu yang diakibatkan karena panas tubuh, maka pola *temperature* panas pada wajah dapat juga dipergunakan sebagai alat identifikasi seseorang.

Masing-masing aspek dapat secara terpisah dipergunakan sebagai alat identifikasi. Menggabungkan aspek-aspek tersebut untuk meningkatkan akurasi identifikasi seringkali juga dilakukan. (Nugroho E. , 2009).

2.2.9 Teknik Identifikasi Wajah

Dalam tahap identifikasi biometrik dapat mengidentifikasi individu-individu berdasarkan perbedaan lingkup karakteristik *behaviour/*psikologi (*biometric identifier*). Hal ini dimungkinkan bahwa karakteristik psikologi/*behaviour* setiap manusia berbeda-beda. Selain itu *identifier* biometrik dianggap lebih *reliable* dibandingkan berdasarkan pemasukan *token* dan pengenalan *knowledge*.

Sistem biometrik belumlah sempurna, karena suatu saat masih dapat melakukan kesalahan dengan menerima *impostor* sebagai invidu yang juga *valid* (terjadi kesalahan pencocokan), sebaliknya terjadi penolakan terhadap individu yang *valid* (terjadi kesalahan ketidakcocokan). Untuk menjamin terhindarnya kesalahan seperti itu, sesuai referensi memadukan ciri biometrik wajah dengan ucapan, serta dari referensi memadukan biometrik wajah dengan ciri tanda-tangan. Selain itu dalam penerapannya ukuran database template sangatlah besar, bahkan dalam *database* perbankan pusat pernah terjadi *bottleneck* saat proses identifikasi.

Mekanisme sistem biometrik dapat digambarkan dengan beberapa fase :

1. **Fase Penggolongan (enrollment).**

Pada fase ini masukan akan di pindai (scan) oleh sensor biometrik, yang merupakan representasi karakteristik digital.

2. **Fase Pencocokan.**

Dalam fase ini inputan database akan dicocokkan dengan identifikasi data. Dapat dimungkinkan adanya reduksi, sehingga dihasilkan representasi digital. Hasil ini akan diproses dengan ekstraktor ciri untuk menghasilkan suatu representasi yang ekspresif dalam bentuk *template*. Bergantung aplikasinya *template* dapat disimpan dalam *database* di sistem biometrik atau dapat direkam pada kartu magnetik (atau *smartcard*).

3. Fase Pengenalan.

Karakteristik individu dibaca oleh pembaca biometrik (*reader*). Selanjutnya dikonversi dengan format digital, untuk diproses sebagai *ekstraktor* ciri (*template*). Hasil template ini selanjutnya dicocokkan dengan identifikasi individu.

Pembacaan wajah umumnya dilakukan secara optis (kamera), dan sensor panas menggunakan analisis *temperature*, teknik identifikasi wajah yang populer dilakukan adalah dengan menggunakan aspek geometri wajah. Dalam metode ini, dilakukan analisis sistem yang memisahkan citra wajah kedalam komponen wajah, antara lain mata, hidung, mulut, dan batas wajah pada citra diam tunggal yang diambil dari posisi tampak depan. Jarak antar tiap komponen diukur, kemudian dikombinasikan dengan fitur lainya untuk membentuk *semantic* wajah. Mata mempunyai jarak yang proporsional dengan fitur-fitur wajah yang lain. Jika satu fitur terdeteksi, maka posisi fitur lainya akan diketahui, dan fitur tersebut dapat diekstraksi.

Tahapan pemrosesan citra wajah yang dilakukan dalam penelitian Agushinta dan kawan-kawan adalah sebagai berikut:

1. Citra wajah seseorang diambil
2. Warna citra tersebut dikonversikan ke dalam warna dasar YCbCr. Hal ini disebabkan karena gambar yang diperoleh dari kamera memiliki struktur warna menggunakan sistem *Red-Green-Blue* (RGB). Transformasi ke sistem warna YCrCb dilakukan untuk memisahkan intensitas Y dengan chromaticity saturasi dari cahaya yang tertangkap kamera, harga Cr dan

Cb sangat stabil, sehingga nilai Cr dan Cb merupakan informasi handal yang dapat digunakan untuk proses klasifikasi warna.

3. Citra hasil pada proses konversi diolah dengan *tapis Gauss* (filter matematik).
4. Kemudian diolah dengan tingkat ambang yang ditetapkan, kemudian diubah ke citra biner dan selanjutnya ditetapkan ukuran-ukuran yang akan digunakan dalam analisis citra tersebut. (Nugroho E. , 2009).

2.2.10 Pengenalan Pola

Pola adalah suatu bentuk dimana masing-masing pola memiliki ciri-cirinya. Ciri-ciri tersebut digunakan utuk membedakan suatu pola dengan pola yang lainnya. Ciri yang baik adalah ciri yang memiliki daya pembeda yang tinggi, se-hingga pengelompokkan pola berdasarkan ciri yang dimiliki dapat dilakukan dengan keakuratan yang tinggi.

Pengenalan pola merupakan langkah perantaraan bagi proses menghilangkan dan menormalkan gambar dalam satu cara pemrosesan gambar (*image processing*), teks dll.), pengiraan ciri-ciri, pengkelasan dan akhirnya post-pemrosesan berdasarkan kelas pengenalan dan aras keyakinan. Pengenalan pola berkaitan dengan langkah pengklasifikasian. Dalam kasus tertentu, sebagaimana dalam jaringan syaraf (*neural networks*), pemilihan ciri-ciri dan pengambilan juga boleh dilaksanakan secara semi otomatis atau otomatis sepenuhnya. (Masrani, Ilhamsyah, & Ruslianto, 2018).

2.2.11 Citra RGB

Citra RGB mempunyai array berukuran m x n x 3 yang mendefinisikan warna merah, warna hijau, dan warna biru untuk setiap pikselnya. Warna pada setiap piksel ditentukan dari kombinasi merah, hijau, dan biru. RGB merupakan citra 24 bit dengan komponen merah, hijau, dan biru disimpan masing-masing berukuran 8 bit yang berarti memiliki tingkat kecerahan warna sampai 256 level. (Masrani, Ilhamsyah, & Ruslianto, 2018).

2.2.12 Algoritma Eigenface

Pengenalan *Eigenface* berasal dari prefiks bahasa Jerman "*eigen*", yang berarti "sendiri/*individual*". Metode *eigenface* dianggap sebagai teknologi pengenalan wajah otomatis pertama yang pernah diciptakan. Teori ini dikembangkan oleh Turk dan Pentland. Teori ini dikembangkan dengan membagi sebuah citra wajah menjadi data set fitur karakteristik yang disebut *eigenface.*

Eigenface adalah suatu metode pengenalan wajah yang berdasarkan pada algoritma *Principal Component Analysis* (PCA). Secara singkat prosesnya adalah citra direpresentasikan dalam sebuah gabungan *vector* yang dijadikan satu matriks tunggal. Dari matriks tunggal ini akan diekstraksi suatu ciri utama yang akan membedakan antara citra wajah satu dengan citra wajah lainnya. Citra yang digunakan adalah citra digital dengan format *grayscale* untuk mempermudah komputasinya. Dengan membandingkan antara citra uji dengan citra referensi menggunakan konsep jarak *euclidean*, maka akan didapat kesimpulan apakah suatu citra wajah dikenali atau tidak dikenali. (Kustian, 2017).

Rumus Perhitungan untuk menggunakan algoritma *eigenface* adalah dengan cara :

1. Pencarian *Matrix* N x N dapat dihitung dengan :

   $$A \,.\, x \;=\; \lambda \,.\, x \quad \text{..... (2.1)}$$

   Keterangan :

   λ adalah *Eigenvalue* dari matrix A

   x adalah *eigenvector* yang sama dengan scalar (λ)
2. Pencarian determinan dari *matrix* A dengan rumus :

   $$|A - 1\lambda| = 0 \quad \text{..... (2.2)}$$
3. Setelah nilai *Eigenvalue* (λ) ditemukan langkah selanjutnya adalah mencari *Eigenvektor* dengan menggunakan rumus :

   $$A - \lambda I \,.\, x = 0 \quad \text{..... (2.3)}$$

2.2.12.1 Segmentasi Citra

Segmentasi adalah proses pembagian sebuah citra kedalam sejumlah bagian atau obyek. Segmentasi merupakan suatu bagian yang sangat penting

dalam analisis citra secara otomatis, sebab pada prosedur ini obyek yang diinginkan akan disadap untuk proses selanjutnya. Algoritma segmentasi didasarkan pada 2 buah karakteristik nilai derajat kecerahan pada citra yaitu: *discontinuity* dan *similarity*. Pada item pertama, citra dipisahkan/dibagi atas dasar perubahan yang mencolok dari derajad kecerahannya. Aplikasi yang umum adalah untuk deteksi titik, garis, area, dan sisi citra. Pada kategori kedua, didasarkan atas *thresholding*, *region growing*, dan *region spiltting and merging*. Tiap piksel dalam suatu wilayah mempunyai kesamaan karakteristik atau propeti yang dapat dihitung (*computed property*), seperti: warna (*color*), intensitas (*intensity*),dan tekstur (*texture*). Dalam kasus ini digunakan segmentasi citra berdasarkan area tekstur wajah, karena pada pengolahan awal citra hanya ingin mengambil fokus di bagian komposisi wajah inti mulai dari alis hingga dagu. (Kustian, 2017)

2.2.12.2 Perhitungan Principal Component Analysis (PCA)

PCA adalah teknik statistik yang sudah digunakan secara luas baik dalam hal pengolahan data, pembelajaran mesin, maupun pengolahan citra atau pemrosesan signal. Metode *Principal Component Analysis* (PCA) dibuat pertama kali oleh para ahli statistik dan ditemukan oleh Karl Pearson pada tahun 1901 yang memakainya pada bidang biologi. Pada tahun 1947 teori ini ditemukan kembali oleh Karhunen, dan kemudian dikembangkan oleh Loeve pada tahun l963, sehingga teori ini juga dinamakan *Karhunen-Loeve transform* pada bidang ilmu telekomunikasi.

PCA digunakan digunakan untuk menyederhanakan suatu data, dengan cara mentransformasi data secara linier sehingga terbentuk sistem koordinat baru dengan *varians* maksimum. Analisis komponen utama dapat digunakan untuk mereduksi dimensi suatu data tanpa mengurangi karakteristik data tersebut secara signifikan. Analisis komponen utama juga sering digunakan untuk menghindari masalah *multikolinearitas* antar peubah bebas dalam model regresi berganda. Analisis komponen utama merupakan analisis antara dari suatu proses penelitian yang besar atau suatu awalan dari analisis berikutnya, bukan merupakan suatu analisis yang langsung berakhir. Misalnya komponen utama bisa merupakan

masukan untuk regresi berganda atau analisis faktor atau analisis gerombol. Analisis komponen utama juga merupakan salah satu teknik statistika multivariat yang dapat menemukan karakteristik data yang tersembunyi.Dalam penerapannya, Analisis komponen utama, justru dibatasi oleh asumsi-asumsinya,yaitu asumsi kelinearan model regresi, asumsi keorthogonalan komponen utama, dan asumsi *varians* yang besar memiliki struktur yang penting.

Principal Component Analysis (PCA) merupakan salah satu hasil berharga dari aljabar linear terapan. Prosedur PCA pada dasarnya adalah bertujuan untuk menyederhanakan variabel yang diamati dengan cara menyusutkan (mereduksi) dimensinya. Hal ini dilakukan dengan cara menghilangkan korelasi diantara variabel bebas melalui transformasi variabel bebas asal ke variabel baru yang tidak berkorelasi sama sekali tanpa menghilangkan informasi penting yang ada di dalamnya atau yang biasa disebut dengan *principal component*. Dengan reduksi ini maka waktu komputasi dapat dikurangi dan *kompleksititas* dari citra wajah yang tidak perlu dapat dihilangkan. *Principal Component Analysis* menggunakan vektor-vektor yang disebut dengan *eigenvector* dan nilai-nilai yang disebut dengan *eigenvalue* untuk mendapatkan fitur yang paling signifikan pada dataset. (Kustian, 2017).

Proses Perhitungan *Principal Component Analysis* (PCA) dapat dilakukan dengan langkah-langkah sebagai berikut :

1. Mengambil satu set citra wajah (M). misalkan M berjumlah 10 buah citra wajah.
2. Inisialisasi *r* untuk tiap citra wajah dari *set training r* adalah sebuah *vector* n^2 x 1 berdasarkan *matrix* dari citra wajah yang berukuran NxN.
3. Menghitung rata-rata *vector* citra wajah

   $$Y = \frac{1}{M}\sum_{i-1}^{M} r_i \quad (2.4)$$

4. Melakukan normalisasi ukuran citra dengan melakukan pengurangan *vector* citra wajah dengan nilai rata-rata tersebut.

   $$Q = r_i \quad (2.5)$$

5. Jika sudah maka langkah selanjutnya adalah menghitung *matrix* kovarian

$$C = \frac{1}{M} \sum_{i-1}^{M} Q_n Q_n^T = AA^T (matriks\ N^2 x N^2) \quad \text{..... (2.6)}$$

Dimana :

$$A = [Q_1 Q_2 Q_3 \dots Q_M] (matriks\ N^2 x M \quad \text{..... (2.7)}$$

6. Jika ukuran *matrix* terlalu besar maka pencarian *matrix* kovarian menjadi :

$$C = A^T A \quad \text{..... (2.8)}$$

7. Menghitung *eigenvalue* (λ) dan *eigenvector* (x) dari *matrix* kovarian

$$C = A^T A \quad \text{..... (2.9)}$$

8. Menghitung *eigenvector* sebanyak M dari *matrix* kovarian

$$C = AA^T \quad \text{..... (2.10)}$$

Dengan persamaan :

$$U_i = A.x_1 \quad \text{..... (2.11)}$$

9. Melakukan normalisasi terhadap u.
10. Mengumpulkan *eigenvector* sebanyak K.

Contoh Perhitungan Manual:

Perhitungan Manual Eigenface

1. Terdapat 3 buah image dengan matriks 2x2

Image 1 $\begin{bmatrix} 1 & 4 \\ 2 & 5 \end{bmatrix} = \begin{bmatrix} 1 \\ 4 \\ 2 \\ 5 \end{bmatrix}$

Image 2 $\begin{bmatrix} 3 & 5 \\ 2 & 6 \end{bmatrix} = \begin{bmatrix} 3 \\ 2 \\ 4 \\ 6 \end{bmatrix}$

Image 3 $\begin{bmatrix} 7 & 3 \\ 8 & 4 \end{bmatrix} = \begin{bmatrix} 7 \\ 3 \\ 8 \\ 4 \end{bmatrix}$

2. Lalu ditransformasikan ke matriks n^2 x image

$$S = \begin{bmatrix} 1 & 3 & 7 \\ 4 & 2 & 3 \\ 2 & 4 & 8 \\ 5 & 6 & 4 \end{bmatrix}$$

3. Lalu mencari rata-rata vector

Y1 $\frac{1+3+7}{3}$ = 3,66

Y2 $\frac{4+2+3}{3}$ = 3

Y3 $\frac{2+4+8}{3}$ = 4,66

Y4 $\frac{5+6+4}{3}$ = 5

Sehingga didapatkan:

$$Y=\begin{bmatrix}3,66\\3\\4,66\\5\end{bmatrix}$$

4. Mengurangi vector citra wajah dengan nilai Y

$$Q1=\begin{bmatrix}1\\4\\2\\5\end{bmatrix}-\begin{bmatrix}3,66\\3\\4,66\\5\end{bmatrix}=\begin{bmatrix}-2,66\\1\\-2,66\\0\end{bmatrix}$$

$$Q2=\begin{bmatrix}3\\2\\4\\6\end{bmatrix}-\begin{bmatrix}3,66\\3\\4,66\\5\end{bmatrix}=\begin{bmatrix}-0,66\\-1\\-0,66\\1\end{bmatrix}$$

$$Q3=\begin{bmatrix}7\\3\\8\\4\end{bmatrix}-\begin{bmatrix}3,66\\3\\4,66\\5\end{bmatrix}=\begin{bmatrix}4,66\\0\\4,66\\-1\end{bmatrix}$$

5. Selanjutnya Q1,Q2,Q3 digabungkan menjadi satu matriks A

$$A=\begin{bmatrix}-2,66 & -0,66 & 4,66\\1 & -1 & 0\\-2,66 & -0,66 & 4,66\\0 & 1 & -1\end{bmatrix}$$

6. Menghitung nilai kovarian menggunakan rumus $A^T.A$

$$C=\begin{bmatrix}-2,66 & 1 & -2,66 & 0\\-0,66 & -1 & -0,66 & 1\\4,66 & 0 & 4,66 & -1\end{bmatrix}x\begin{bmatrix}-2,66 & -0,66 & 4,66\\1 & -1 & 0\\-2,66 & -0,66 & 4,66\\0 & 1 & -1\end{bmatrix}$$

$$C=\begin{bmatrix} 11{,}64 & 2{,}5112 & -24{,}7912 \\ 2{,}5112 & 2{,}8712 & -7{,}5112 \\ -24{,}7912 & -5{,}8316 & 44{,}4312 \end{bmatrix}$$

7. Menghitung Eigenvalue dan eigenvector dari kovarian matriks C. (Untuk mempermudah perhitungan kita rubah angkanya jadi lebih kecil)

$$C=\begin{bmatrix} 2 & 0 & 1 \\ -1 & 4 & -1 \\ -1 & 2 & 0 \end{bmatrix}$$

$$\text{Determinan}=\begin{bmatrix} (2-\lambda) & 0 & 1 \\ -1 & (4-\lambda) & -1 \\ -1 & 2 & -\lambda \end{bmatrix}=0$$

$=((2-\lambda)*\{(4-\lambda)(-\lambda)-(2)(-1)\}) - (0*\{(-1)(-\lambda) - (-1)(-1)\}) + (1*\{(-1)(2) - (4-\lambda)(-1)\})=0$

$=(2-\lambda)\{-4\lambda+\lambda^2+2\}-0+(-\lambda+2)=0$

$=(2-\lambda)\{(\lambda^2 - 4\lambda+2\}+1=0$

$=(2-\lambda)\{(\lambda^2 - 4\lambda+3\}=0$

$=(2-\lambda)\ (\lambda - 1)\ (\lambda - 3)$

$=\lambda_1=2\ ,\ \lambda_2=1\ ,\ \lambda_3=3$

8. Eigenvalue λ_1 , λ_2 , λ_3 yang didapat dari matriks C digunakan untuk menghitung Eigenvektor

$$\rightarrow\lambda_1=2\begin{bmatrix} (2-2) & 0 & 1 \\ -1 & (4-2) & -1 \\ -1 & 2 & -2 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}=0$$

$$\begin{bmatrix} 0 & 0 & 1 \\ -1 & 2 & -1 \\ -1 & 2 & -2 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}=0$$

Maka Persamaanya adalah:

$-x_1 + 2\ x_2 = 0$

$-x_1 + 2\ x_2 = 0$

$-x_1 + 2\ x_2 = 0$

Langkah selanjutnya dilakukan proses eliminasi sehingga diperoleh,

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2x_2 \\ x_2 \\ 0 \end{bmatrix}$$

Eigenvektornya Adalah

$$x_2 = \begin{bmatrix} 2 \\ 1 \\ 0 \end{bmatrix}$$

$$\rightarrow \lambda_1 = 1 \begin{bmatrix} (2-1) & 0 & 1 \\ -1 & (4-1) & -1 \\ -1 & 2 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = 0$$

$$\begin{bmatrix} 1 & 0 & 1 \\ -1 & 3 & -1 \\ -1 & 2 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = 0$$

Maka Persamaanya adalah:

$x_1 + x_3 = 0$

$-x_1 + 3\,x_2 - x_3 = 0$

$-x_1 + 2\,x_2 - x_3 = 0$

Langkah selanjutnya dilakukan proses eliminasi sehingga diperoleh,

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -x_3 \\ 0 \\ x_3 \end{bmatrix}$$

Eigenvektornya Adalah

$$x_3 = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$

$$\rightarrow \lambda_1 = 3 \begin{bmatrix} (2-3) & 0 & 1 \\ -1 & (4-3) & -1 \\ -1 & 2 & -3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = 0$$

$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 1 & -1 \\ -1 & 2 & -3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = 0$$

Maka Persamaanya adalah:

$-x_1 = x_3$

$--x_1 + x_2 - x_1 = 0$

$-x_1 + 2\,x_2 - 3x_3 = 0$

Langkah selanjutnya dilakukan proses eliminasi sehingga diperoleh,

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} x_1 \\ 2 \\ x_1 \end{bmatrix}$$

Eigenvektornya Adalah

$$x_3 = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$$

9. Menghitung eigenvector sebanyak M dari matriks A

$$u_1 = \begin{bmatrix} -2{,}66 & -0{,}66 & 4{,}66 \\ 1 & -1 & 0 \\ -2{,}66 & -0{,}66 & 4{,}66 \\ 0 & 1 & -1 \end{bmatrix} \text{x} \begin{bmatrix} 2 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 4{,}66 \\ 3 \\ 4{,}66 \\ 2 \end{bmatrix}$$

$$u_2 = \begin{bmatrix} -2{,}66 & -0{,}66 & 4{,}66 \\ 1 & -1 & 0 \\ -2{,}66 & -0{,}66 & 4{,}66 \\ 0 & 1 & -1 \end{bmatrix} \text{x} \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 7{,}32 \\ -1 \\ 7{,}32 \\ -1 \end{bmatrix}$$

$$u_3 = \begin{bmatrix} -2{,}66 & -0{,}66 & 4{,}66 \\ 1 & -1 & 0 \\ -2{,}66 & -0{,}66 & 4{,}66 \\ 0 & 1 & -1 \end{bmatrix} \text{x} \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 3.32 \\ 0 \\ 3.32 \\ 1 \end{bmatrix}$$

10. Selanjutnya Eigenvalue diurutkan dari yang terkecil ke terbesar

$$\lambda_3 = \begin{bmatrix} 3.32 \\ 0 \\ 3.32 \\ 1 \end{bmatrix}$$

$$\lambda_2 = \begin{bmatrix} 7{,}32 \\ -1 \\ 7{,}32 \\ -1 \end{bmatrix}$$

$\lambda_1 = \begin{bmatrix} 4{,}66 \\ 3 \\ 4{,}66 \\ 2 \end{bmatrix}$

11. Memproyeksikan Q_i kedalam Eigenspace

W_1=

$[3.32 \quad 0 \quad 3.32 \quad 1]$ x $\begin{bmatrix} -2{,}66 \\ 1 \\ -2{,}66 \\ 0 \end{bmatrix}$=17,6624

$[3.32 \quad 0 \quad 3.32 \quad 1]$ x $\begin{bmatrix} -0{,}66 \\ -1 \\ -0{,}66 \\ 1 \end{bmatrix}$=-3,3824

$[3.32 \quad 0 \quad 3.32 \quad 1]$ x $\begin{bmatrix} 4{,}66 \\ 0 \\ 4{,}66 \\ -1 \end{bmatrix}$=29,9424

W_2=

$[7.32 \quad -1 \quad 7.32 \quad -1]$ x $\begin{bmatrix} -2{,}66 \\ 1 \\ -2{,}66 \\ 0 \end{bmatrix}$=-39,9424

$[7.32 \quad -1 \quad 7.32 \quad -1]$ x $\begin{bmatrix} -0{,}66 \\ -1 \\ -0{,}66 \\ 1 \end{bmatrix}$=-9,6624

$[7.32 \quad -1 \quad 7.32 \quad -1]$ x $\begin{bmatrix} 4{,}66 \\ 0 \\ 4{,}66 \\ -1 \end{bmatrix}$=69,2224

W_3=

$[4{,}66 \quad 3 \quad 4{,}66 \quad 2]$ x $\begin{bmatrix} -2{,}66 \\ 1 \\ -2{,}66 \\ 0 \end{bmatrix}$=-21,7912

$$[4{,}66 \quad 3 \quad 4{,}66 \quad 2] \text{ x } \begin{bmatrix} -0{,}66 \\ -1 \\ -0{,}66 \\ 1 \end{bmatrix} = -7{,}1512$$

$$[4{,}66 \quad 3 \quad 4{,}66 \quad 2] \text{ x } \begin{bmatrix} 4{,}66 \\ 0 \\ 4{,}66 \\ -1 \end{bmatrix} = 41{,}4312$$

12. Masukan nilai W_i kedalam sebuah vector

$$\Omega = \begin{bmatrix} -17{,}6624 & -39{,}9424 & -21{,}7912 \\ -3{,}3824 & -9{,}6624 & -7{,}1512 \\ 29{,}9424 & 69{,}2224 & 41{,}4312 \end{bmatrix}$$

13. Selanjutnya dari matriks omega ini akan didapatkan mean (μ) dan standar deviasi (σ).

BAB III
METODOLOGI

3.1 Bahan Penelitian

Bahan penelitian adalah bahan–bahan yang digunakan dalam penelitian. Bahan penelitian yang digunakan pada proses penelitian ini bersumber dari jurnal, artikel ilmiah, dan buku-buku pendukung yang terkait dengan biometrik wajah. Dalam penelitian ini akan digunakan bahan berupa *database* MySql dan wajah berupa *file image* atau gambar.

3.2 Alat Penelitian

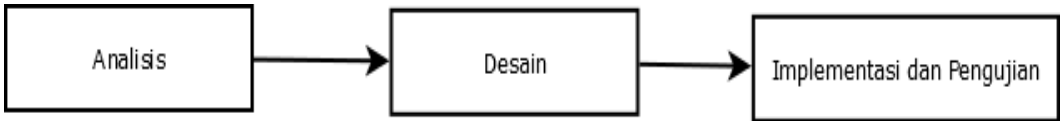
Dalam penelitian ini, alat penelitian yang digunakan berupa piranti-piranti yaitu alat pendukung berupa penggaris, busur derajat dan *smartphone* yang sudah terinstall *lux light meter*. Beserta perangkat lunak dan perangkat keras. Perangkat lunak yang digunakan yakni Xampp V3.2.2, Sharpdevelop dan Adobe Dreamweaver CS3, sedangkan perangkat keras yang digunakan tertera dalam Tabel 3.1 berikut ini beserta spesifikasinya.

Tabel 3. 1 Alat Penelitian

Merek	**Samsung**
Processor	Intel Celeron
RAM	4 GB
Harddisk	320 GB
Layar	14 inchi

3.3 Jalan Penelitian

Dalam penelitian ini, tahap-tahap metodologi yang digunakan sesuai dengan blok diagram Gambar 3.1.



Gambar 3. 1 Blok Diagram Perancangan Sistem *Login*

3.3.1 Tahap Analisis

Tahap analisis yang dilakukan pada penelitian ini adalah analisis data yang diperlukan untuk penelitian berupa data wajah *user* yang diperoleh pada saat *user* mendaftar.

3.3.2 Tahap Desain

Desain adalah proses penerjemahan sistem sesuai algoritma yang digunakan. Hal ini bertujuan agar program yang dibuat sesuai dengan analisis kebutuhan. Hasil perancangan *system* adalah bagan proses dan perancangan antar muka.

3.3.3 Tahap Implementasi dan Pengujian

Implementasi dan pengujian merupakan metode terakhir yang digunakan setelah analisis dan perancangan sistem selesai dilakukan. Metode ini akan menjelaskan tentang penerapan jalannya rancang bangun yang telah dianalisa dan dirancang. Aplikasi yang telah di rancang dan di analisa selenjutnya diimplementasikan dan dilakukan pengujian untuk mengetahui tingkat keberhasilan aplikasi yang telah ada.

3.3.4 Tahap Kesimpulan dan Saran

Tahapan kesimpulan dan saran merupakan akhir dari penelitian tugas akhir ini. Tahapan ini berisi tentang kesimpulan dari hasil-hasil penelitian dan pengujian yang telah dilakukan pada penelitian tugas akhir ini, yaitu perancangan sistem *login* dengan biometrik wajah dan berisi saran-saran membangun yang dapat dijadikan bahan penelitian ulang untuk meneliti dan merancang aplikasi biometrik wajah yang lebih baik.

3.4 Perancangan Sistem

3.4.1 Tabel

Adapaun rancangan *database* berisi tabel yang digunakan adalah sebagai berikut :

a. Tabel Biodata

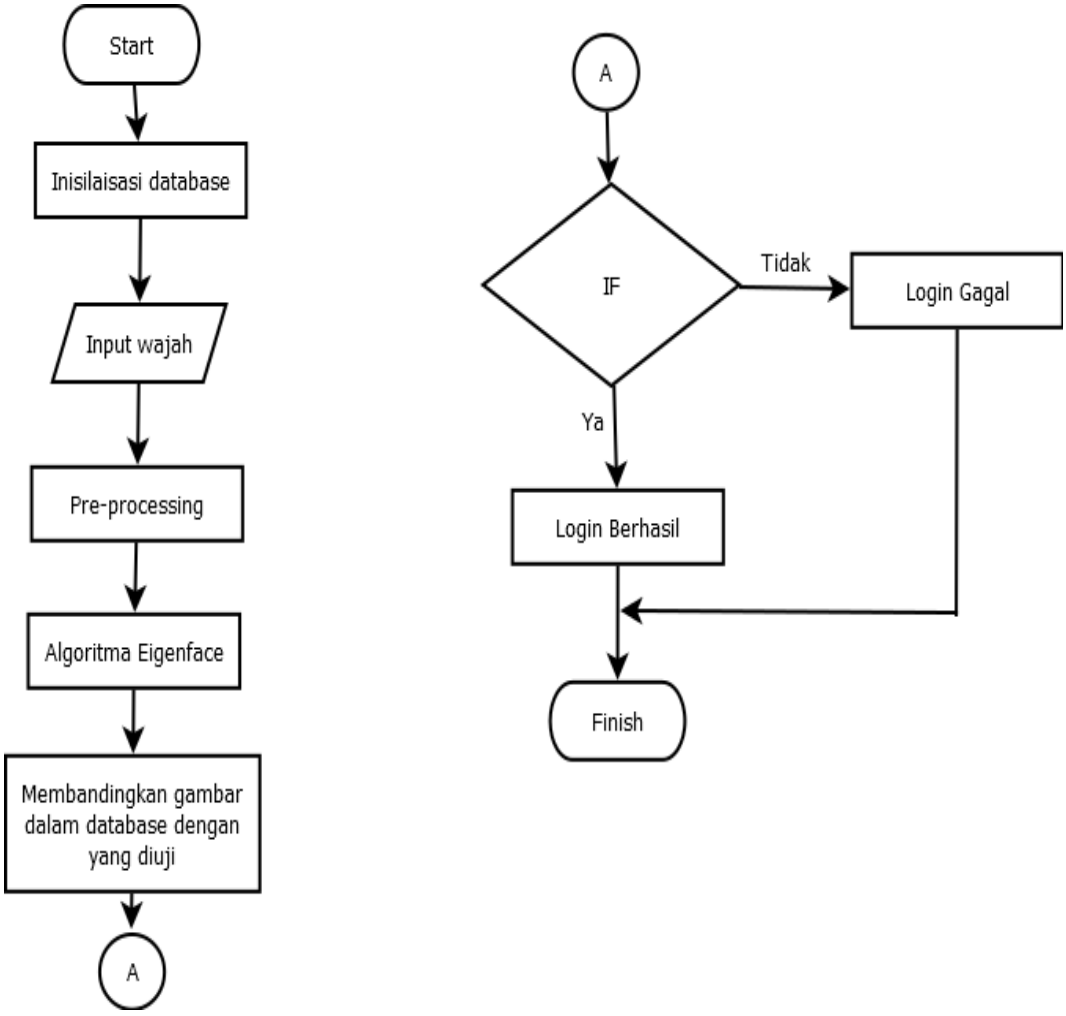
Tabel Biodata digunakan untuk proses mendaftar anggota baru dan juga berfungsi untuk data *login*. Struktur tabel biodata dapat dilihat pada Tabel 3.2.

Tabel 3. 2 Tabel Biodata

Nama Field	**Type**	**Length**	**Keterangan**
ID	Int	10	Primary key
Nama	Varchar	30	
Tanggal	Date	-	
JamMasuk	Time	-	
Keterangan	Varchar	20	

3.4.2 Desain Sistem

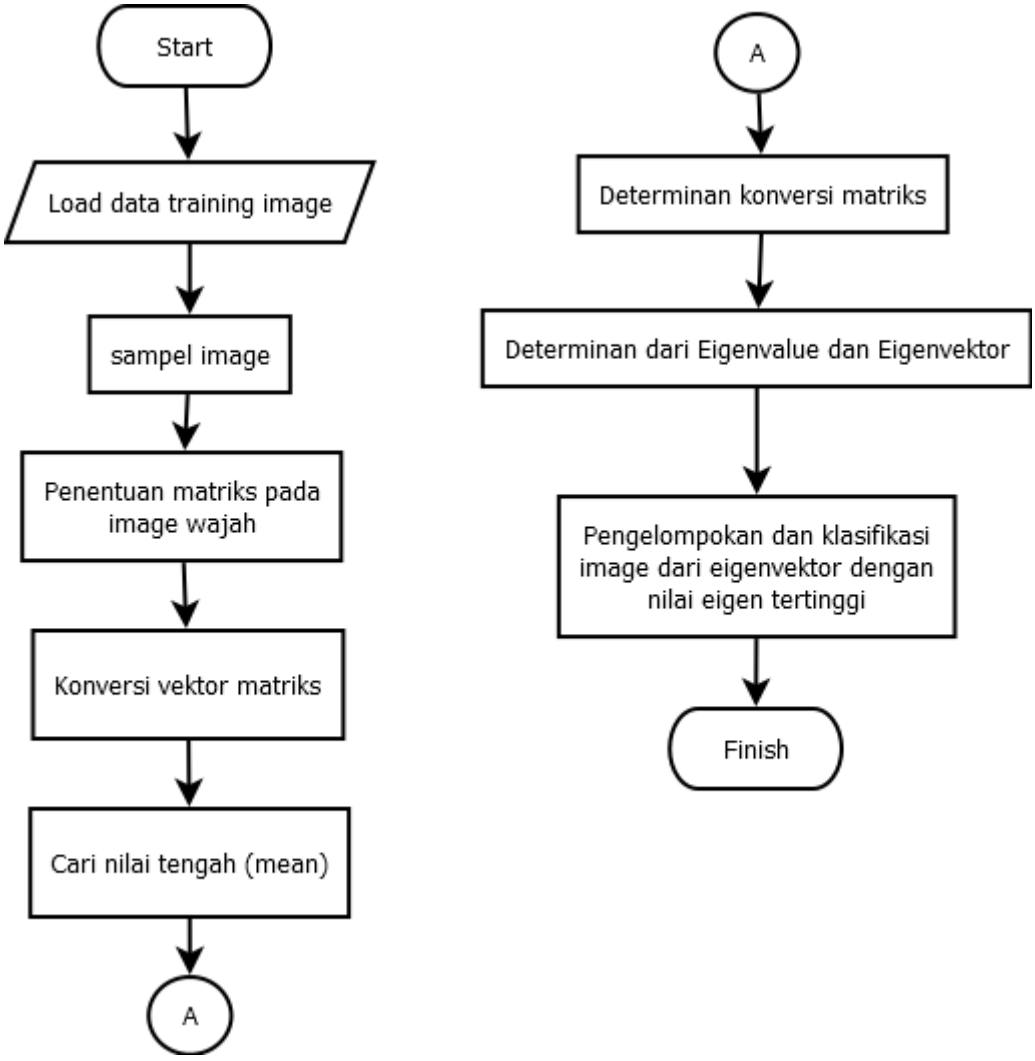
Perangkat lunak akan melakukan *pre-processing*, kemudian mengekstraksi karakteristik wajah menggunakan *eigenface* lalu menyimpan nilai *matrix* untuk proses identifikasi. Pencarian karakteristik gambar akan melalui *pre-processing* kemudian gambar akan diproses menggunakan *eigenface* gambar akan dicari nilai *matrix* yang lebih kecil dan disimpan sebagai karakteristik. Alur sistem keseluruhan dijelaskan pada Gambar 3.2.



Gambar 3. 2 Flowchart Sistem Keseluruhan

3.4.3 Desain Algoritma Identifkasi Image

Desain Algoritma Identifikasi *Image* dengan *Eigenface* dapat dilihat pada Gambar 3.3.



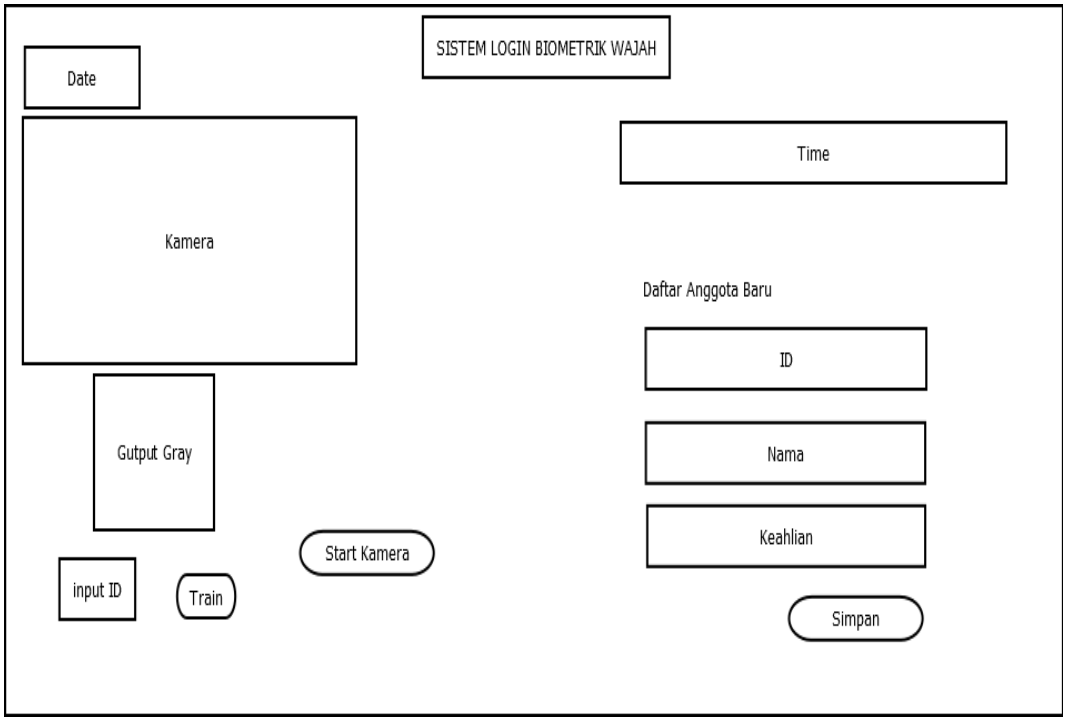
Gambar 3. 3 Desain Algoritma Identifikasi *Image* dengan *Eigenface*

3.4.4 Desain Antar Muka

Perancangan antarmuka diperlukan untuk mempermudah pengguna menggunakan sistem biometrik wajah ini. Sistem ini memiliki satu tampilan *front end* yang terdiri dari beberapa menu diantaranya adalah menu daftar anggota, menu train dan menu login.

3.4.4.1 Rancangan halaman utama (Front End)

Halaman utama *(Front End)* dapat dilihat pada gambar 3.4.



Gambar 3. 4 Halaman utama (*front end*)

BAB IV
ANALISIS DAN PEMBAHASAN

Analisis dan pembahasan akan menjelaskan mengenai hasil dari penelitian *prototype* desain keamanan login menggunakan biometrik wajah yang telah diproses dari hasil pengujian.

4.1 Hasil Penelitian

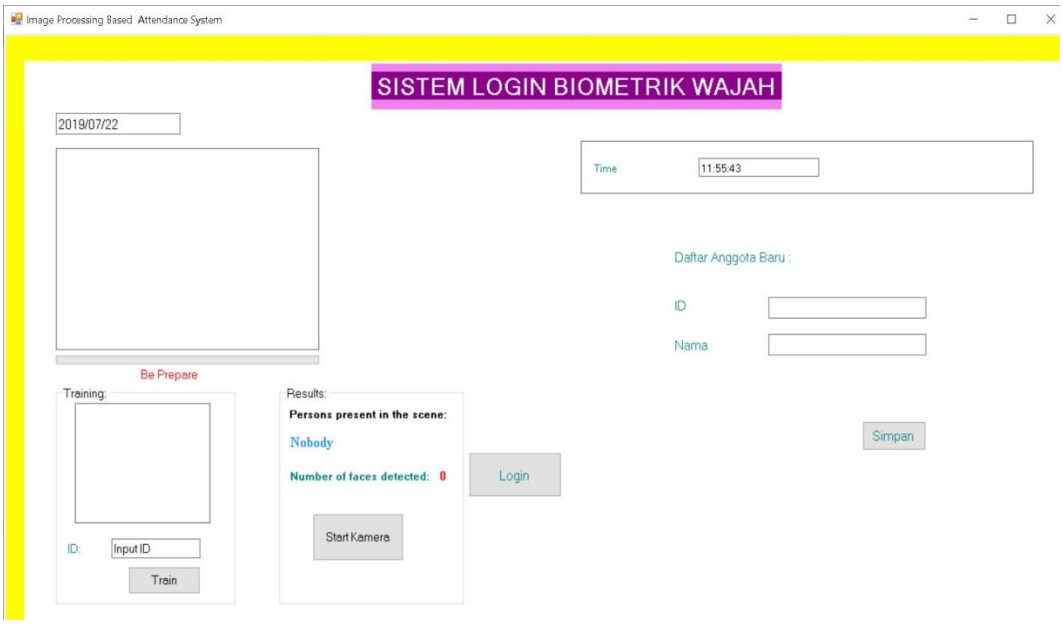
Hasil Pengujian yang dilakukan pada sistem keamanan login biometrik wajah dengan metode *eigenface* menunjukan unjuk kerja sistem yang sesuai dengan data yang didapat.

4.1.1 Hasil Implementasi Desain Program

Implementasi desain sistem adalah pengkodean sistem agar sesuai dengan desain sistem yang telah didesain sebelumnya, implementasi desain program dapat dilihat pad Sub-bab 4.1.1.1.

4.1.1.1 Tampilan Halaman Awal (Front End)

Berikut ini adalah halaman awal ketika sistem dijalankan, dapat dilihat pada Gambar 4.1.



Gambar 4. 1 Halaman Utama (Front End)

4.2 Analisis dan Pembahasan

Analisis sistem dilakukan dengan melakukan proses pendaftaran *user*, kemudian melakukan proses *train* atau pengenalan wajah, semakin banyak wajah di *train* maka akan semakin baik sistem mengenali, sebelum *image* disimpan kedalam *database*, *image* akan melalui proses *grayscaling* yaitu mengubah *image* RGB menjadi image *grayscale* untuk mendapatkan vector ciri dan mendapatkan jarak *distance* yang nantinya akan digunakan pada proses pengenalan wajah.

Setelah didapatkan nilai *grayscale* kemudian image akan melalui tahap *thresholding*, yaitu untuk mengubah citra berderajat keabuan menjadi citra biner atau hitam putih. Intensitas *pixel* citra hasil *grayscale* dibandingkan dengan nilai ambang (*threshold*). Jika nilai *pixel* lebih besar dari nilai ambang, maka *pixel* akan direpresentasikan dengan warna putih. Dan sebaliknya jika nilai *pixel* lebih kecil maka akan direpresentasikan dengan warna hitam. Setelah proses *grayscale* dan *threshold* terpenuhi maka image akan disimpan kedalam database.

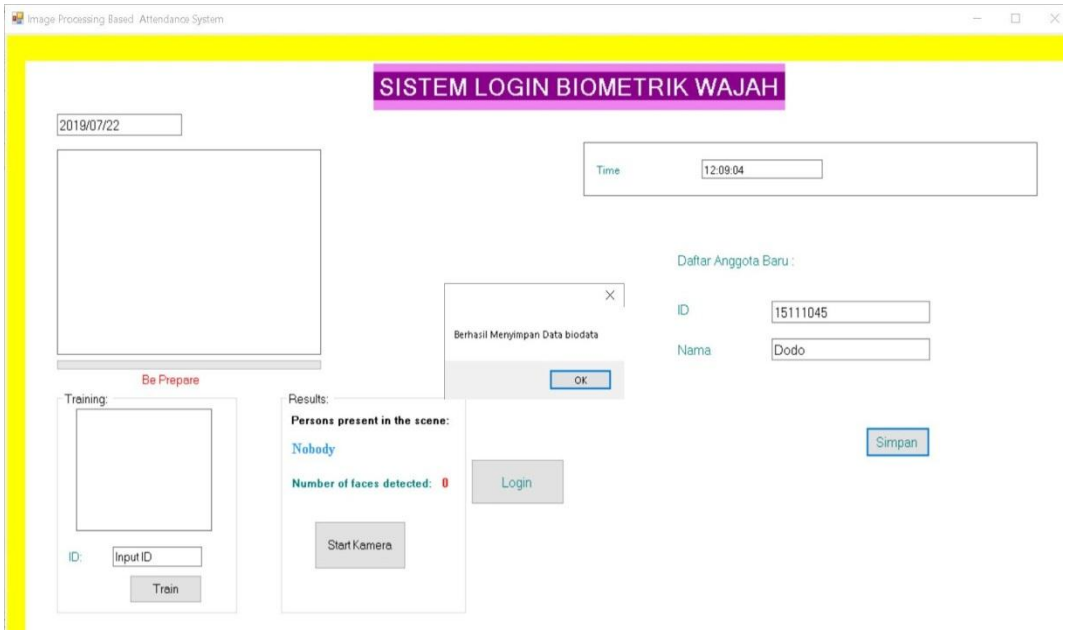
Dilanjutkan dengan tes *login* pada aplikasi. Proses *login* akan membandingkan citra yang ada dalam database dengan citra tes yang akan dibandingkan dengan menggunakan algoritma *eigenface* untuk mencari nilai citra yang terdekat pada database dengan nilai citra tes.

4.2.1 Uji Fungsionalitas

Pada uji fungsionalitas sistem akan dilakukan proses *train* (pengenalan wajah) dan proses tes *login* yaitu proses pencocokan wajah yang ada dalam *database* dengan wajah baru saat proses *login*.

4.2.1.1 Proses Tambah Anggota

Proses tambah anggota berfungsi untuk mendaftarkan ID yang digunakan untuk melakukan proses *train* wajah. Proses tambah anggota dapat dilihat pada Gambar 4.2.



Gambar 4. 2 Proses tambah anggota

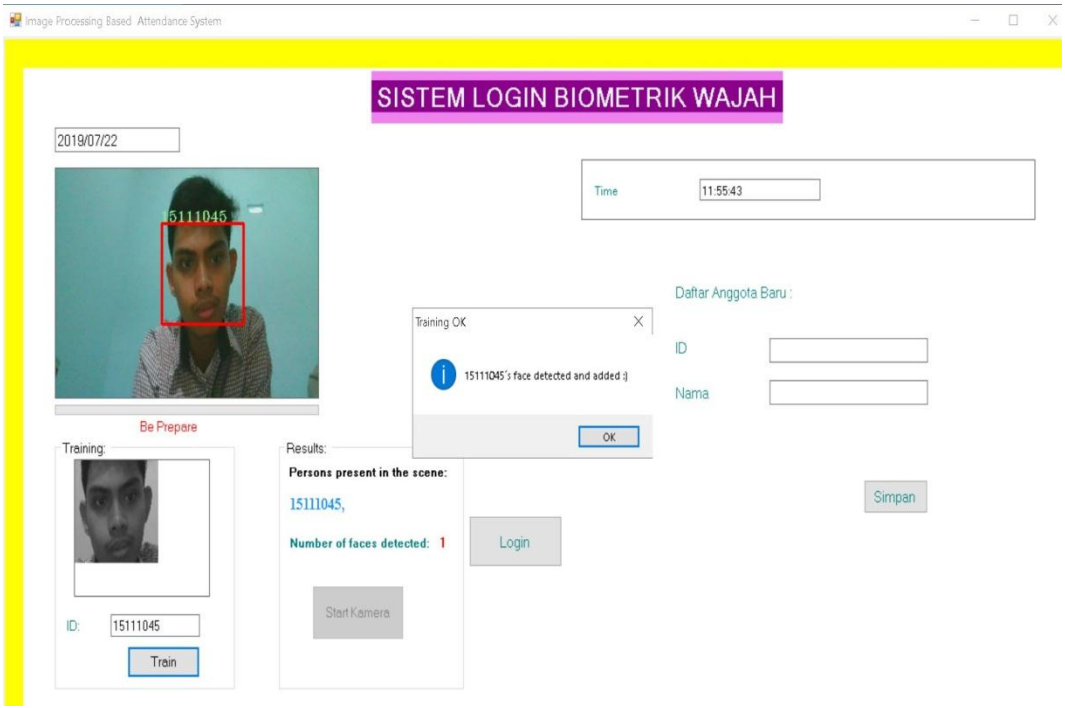
Proses tambah anggota akan disimpan pada *database* MySQL. Untuk lebih jelasnya lihat Gambar 4.3.

	ID	Nama	Tanggal	jamMasuk	keterangan
Ubah Salin Hapus	1506110	Beatrice K. Ingulima	2019-08-07	09:20:33	Berhasil
Ubah Salin Hapus	1900001	Laode hartono	2019-08-06	00:00:00	
Ubah Salin Hapus	15011059	Teguh Supriono	2019-08-06	12:04:15	Berhasil
Ubah Salin Hapus	15021064	M. hasbi	0000-00-00	00:00:00	
Ubah Salin Hapus	15021069	M. Anjas	2019-08-06	12:09:31	Berhasil
Ubah Salin Hapus	15021087	Syofiyan	2019-08-06	12:30:15	Berhasil
Ubah Salin Hapus	15021151	Nur kholik	2019-08-06	11:24:15	Berhasil
Ubah Salin Hapus	15031027	Dedek satria wijaya	2019-08-06	13:03:38	Berhasil
Ubah Salin Hapus	15071055	Indah Rahmayanti	0000-00-00	00:00:00	
Ubah Salin Hapus	15071060	Nur Chasanah	0000-00-00	00:00:00	
Ubah Salin Hapus	15081125	M. Noval K	2019-08-06	19:03:52	Berhasil
Ubah Salin Hapus	15111006	David Saputra	0000-00-00	00:00:00	

Gambar 4. 3 Data anggota disimpan pada *database*

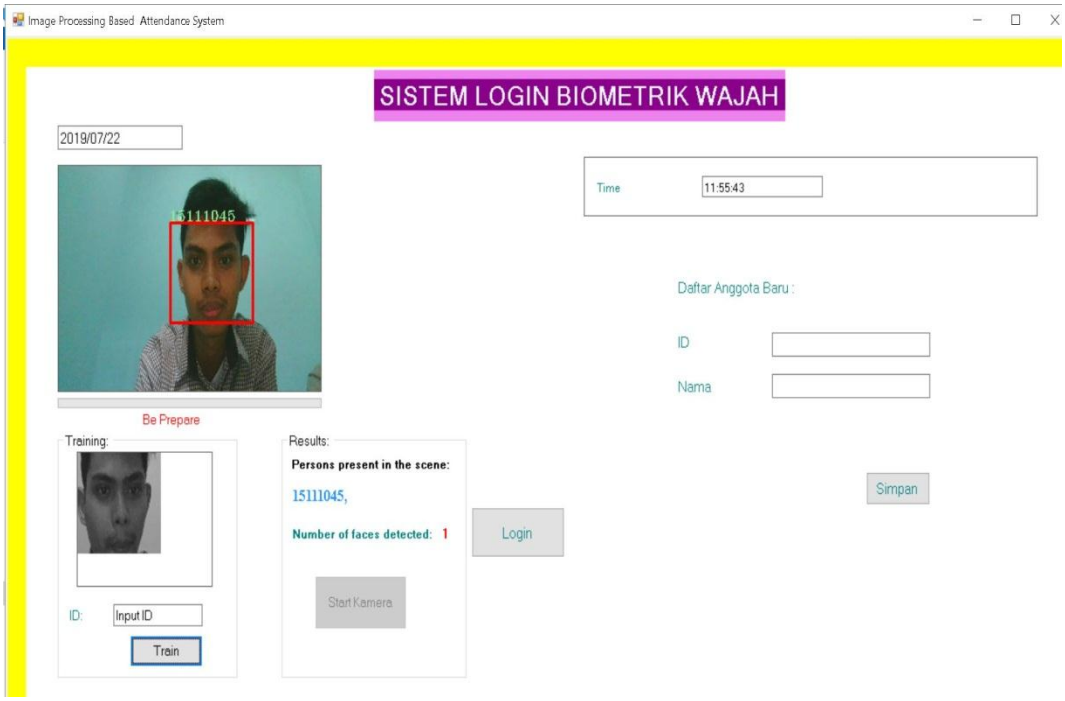
4.2.1.2 Proses Train Wajah

Proses *train* wajah dilakukan dengan melakukan lima kali *train* yaitu saat posisi wajah menghadap ke depan, kanan, kiri, atas dan bawah. guna agar sistem lebih mengenal detail wajah. Pada proses *train* wajah dilakukan uji coba dengan *lux light meter* nilai 19 *lux*. Proses *train* wajah dapat dilihat pada Gambar 4.4.



Gambar 4. 4 Proses *Train* Wajah

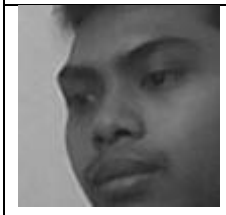
Setelah wajah di training maka setiap kali orang tersebut berada di depan kamera maka akan terlihat ID orang tersebut. Untuk lebih jelasnya lihat Gambar 4.5.



Gambar 4. 5 ID pengguna terlihat dikamera

Tabel pengujian proses *train* wajah. Pada tabel ini akan ditampilkan hasil uji dari proses train wajah dengan berbagai pose wajah. Tabel pengujian proses *train* wajah dapat dilihat pada Tabel 4.1

Tabel 4. 1 Pengujian Proses *Train* Wajah

No	ID	Pose Train Wajah	Keterangan
1	15111045	Hadap Depan	Berhasil
		 Hadap Kanan	Berhasil

Tabel 4. 2 Pengujian Proses *Train* Wajah (Lanjutan)

No	ID	Pose Train Wajah	Keterangan
1	15111045	Hadap Kiri	Berhasil
		Hadap atas	Berhasil
		Hadap Bawah	Berhasil
2	Data 1	-	Berhasil
3	Data 2	-	Berhasil
4	Data 3	-	Berhasil
5	Data 4	-	Berhasil
6	Data 5	-	Berhasil
7	Data 6	-	Berhasil
8	Data 7	-	Berhasil
9	Data 8	-	Berhasil
10	Data 9	-	Berhasil
11	Data 10	-	Berhasil
12	Data 11	-	Berhasil
13	Data 12	-	Berhasil
14	Data 13	-	Berhasil
15	Data 14	-	Berhasil

Keterangan:

Pada pengujian proses *train* wajah yaitu dengan pose wajah menghadap lurus kedepan, kanan, kiri, atas, dan bawah. Data berhasil di *input* semua kedalam *database*.

*Note: data lengkap ada di lampiran.

4.2.1.3 Proses Tes Login

Pada proses tes *login* akan di lakukan proses pencocokan secara *real time* antara citra wajah dengan data citra wajah yang ada pada *database*. Proses pengujian tes wajah akan dilakukan dengan berbagai pose, diantaranya yaitu pose saat menghadap ke depan, kanan, kiri, atas, dan bawah. Dan akan dilakukan dengan berbagai tingkat pencahayaan dengan parameter yang digunakan yaitu *lux light meter* (satuan *lux*). Proses tes *login* dapat lihat pada tabel-tabel dibawah ini:

Pengukuran jarak standar yang dapat digunakan untuk proses login. Pengukuran jarak standar dapat dilihat pada Tabel 4.3.

Tabel 4. 3 Jarak Standar

No	**Jarak (*Centi Meter*)**	**Keterangan**
1	20 cm	Tidak teridentifikasi
2	30 cm	Tidak teridentifikasi
3	40 cm	Tidak teridentifikasi
4	50 cm	Teridentifikasi
5	60 cm	Teridentifikasi
6	70 cm	Tidak teridentifikasi
7	80 cm	Tidak teridentifikasi
8	90 cm	Tidak teridentifikasi
9	100 cm	Tidak teridentifikasi

Keterangan:

Dari percobaan jarak wajah dan webcam yang didapat maka jarak standarnya adalah 50-60 cm.

Tingkat kemiringan (sudut) wajah, mengukur sudut wajah dilakukan dengan menggunakan busur derajat sebagai alat ukur. Tingkat sudut wajah saat menengok akan sangat berpengaruh dalam penentuan *matching* sistem saat *login*. Tingkat sudut wajah yang telah diukur dapat dilihat pada Tabel 4.4.

Tabel 4. 4 Tes Sudut Kemiringan Wajah

No	**Posisi Wajah**	**Tingkat Kemiringan**	**Keterangan**
1	Lurus depan	0°	Berhasil
2	Menghadap kanan	10°	Berhasil
		20°	Berhasil
		30°	Berhasil
		40°	Berhasil
		50°	Gagal
3	Menghadap kiri	10°	Berhasil
		20°	Berhasil
		30°	Berhasil
		40°	Berhasil
		50°	Gagal
4	Menghadap atas	10°	Berhasil
		20°	Berhasil
		30°	Berhasil
		40°	Berhasil
		50°	Gagal
5	Menghadap bawah	10°	Berhasil
		20°	Berhasil
		30°	Berhasil
		40°	Berhasil
		50°	Gagal

Keterangan:
Dari tes sudut kemiringan maka dapat disimpulkan yaitu sudut kemiringan yang masih dapat mengenali wajah adalah 40°.

Tes *gesture* wajah dan *accessoris*, gesture wajah *user* yang berubah-ubah akan di tes *login* dalam sistem beserta *accessoris* yang digunakan sehingga nantinya akan didapatkan data hasil. Tabel Tes *gesture* wajah dan *accessoris* dapat dilihat pada Tabel 4.5.

Tabel 4. 5 Tes *Gesture* Wajah dan *Accerssoris*

No	***Gesture* dan *Accesoris***	**Keterangan**
1	Senyum	Berhasil
2	Tertawa	Berhasil
3	Memejamkan mata	Berhasil
4	Menjulurkan lidah	Berhasil
5	Mengerutkan kening	Berhasil
6	Mata melotot	Berhasil
7	Mulut manyun	Berhasil
8	Pipi menggembung	Berhasil
9	Memakai kacamata	Gagal
10	Memakai peci	Berhasil
11	Memakai masker	Gagal

Keterangan:
Total dalat adalah 11 data, diantaranya 9 data berhasil dan 2 data gagal. Maka hasil dari percobaan login pada *gesture* dan *accessoris* wajah, gagal pada kondisi saat wajah memakai kacamata dan memakai masker.

Pada pengujian untuk pencahayaan dengan menggunakan *lux light meter* akan dijelaskan pada gambar berikut ini. Pengujian *login* dengan *lux light meter* dapat dilihat pada Tabel 4.6.

Tabel 4. 6 Tes Tingkat Pencahayaan

No	Tingkat Pencahayaan	Keterangan
1	1 *lux*	Gagal
2	3 *lux*	Gagal
3	5 *lux*	Berhasil
4	10 *lux*	Berhasil
5	15 *lux*	Berhasil
6	19 *lux*	Berhasil
7	25 *lux*	Berhasil
8	46 *lux*	Berhasil
9	70 *lux*	Berhasil
10	152 *lux*	Berhasil
11	212 *lux*	Berhasil
12	241 *lux*	Berhasil

Keterangan:

Dari tes tingkat pencahayaan maka dapat disimpulkan apabila tingkat pencahayaan dibawah 5 *lux* maka wajah tidak terdefinisi.

Percobaan tes *login*, pada percobaan ini akan dilakukan guna untuk mengetahui seberapa akurat aplikasi dapat mengenali wajah *user*. Percobaan tes *login* dapat dilihat pada Tabel 4.7.

Tabel 4. 7 Tes Login

No	Nim (ID)	Status
1	15111045	Berhasil
2	15031027	Berhasil
3	15021069	Berhasil
4	15021154	Berhasil
5	15021087	Berhasil
6	15011059	Berhasil

Tabel 4. 8 Tes Login (Lanjutan)

No	Nim (ID)	Status
7	15021064	Gagal
8	18021018	Berhasil
9	17081172	Berhasil
10	17021089	Berhasil
11	15111058	Berhasil
12	18031008	Gagal
13	15111027	Berhasil
14	15111059	Berhasil
15	15111018	Berhasil

Keterangan:

Pada 15 kali percobaan, didapat data 13 kali berhasil dan 2 kali gagal. Maka diperoleh 86%.

4.2.1.4 Pengujian Aplikasi Halaman Utama (Front End)

Pengujian aplikasi pada halaman utama (*Front End*) dilakukan untuk menguji fungsi menu yang ada pada halaman utama. Hasil pengujian aplikasi pada halaman utama (*Front End*) dapat dilihat pada Tabel 4.9.

Tabel 4. 9 Hasil Pengujian Aplikasi Pada Halaman Utama (Front End)

No	Menu Yang Diuji	Persyaratan	Hasil Yang Diterapkan	Hasil Uji Coba
1	Start Kamera	Membuka menu utama	Dapat membuka kamera	OK
2	Time	Membuka menu utama	Dapat menyesuaikan waktu yang sama	OK
3	Date	Membuka menu utama	Dapat menyamakan tanggal	OK

Tabel 4. 10 Hasil Pengujian Aplikasi Pada Halaman Utama (Front End) (Lanjutan)

No	**Menu Yang Diuji**	**Persyaratan**	**Hasil Yang Diterapkan**	**Hasil Uji Coba**
4	Daftar Anggota	Membuka menu utama	Dapat menginputkan data biodata kedalam database	OK
5	Train	Membuka menu utama	Dapat melakukan pengenalan wajah	OK
6	Login	Membuka menu utama	Dapat melakukan login	OK

BAB V
PENUTUP

5.1 Kesimpulan

Dari penelitian yang dilakukan, kesimpulan yang dapat diambil adalah :

1. Jarak wajah dengan webcam standarnya adalah 50-60 cm, sedangkan tingkat pencahayaan minimal wajah dapat dikenali adalah 5 *lux*, dan sudut kemiringan wajah yang masih dapat dikenali sistem adalah 40°.
2. Pada pengujian *gesture* dan *accessories* wajah didapatkan total data 11 data, 9 data berhasil dan 2 data gagal, yaitu pada kondisi saat wajah memakai kacamata dan memakai masker.
3. Pada pengujian tes *login* ada 15 total data, 13 data berhasil dan 2 data gagal, maka persentase keberhasilanya adalah 86%.

5.2 Saran

Berdasarkan penelitian mengenai pembuatan *prototype* desain keamanan login pada menggunakan biometrik wajah dengan metode *eigenface* yang telah dilakukan, untuk penelitian lebih lanjut sangat diperlukan adanya pengembangan terhadap *prototype* ini, saran-saran yang dapat penulis berikan adalah :

1. Adanya penambahan metode lanjutan pada program agar meningkatkan unjuk kerja sistem yang lebih akurat.
2. Menentukan batas ambang (*thresholding*) yang sesuai untuk sistem *prototype* ini dapat membedakan ciri wajah setiap orang.
3. Sistem dikembangkan lebih ramah pengguna (*user friendly*).

DAFTAR PUSTAKA

Agustina, I., Fauziah, & Gunaryati, A. 2016. Biometrik pola suara dengan jaringan saraf tiruan. *JURNAL TEKNIK INFORMATIKA VOL 9 NO. 2*, Hal: 140-147, ISSN: 1979-9160.

Andarinny, A. A., Widodo, C. E., & Adi, K. 2017. Perancangan sistem identifikasi biometrik jari tangan menggunakan Laplacian of Gaussian dan ektraksi kontur. *Youngster Physics Journal*, Hal: 304-314, ISSN: 2302 - 7371.

Apriadi, A., Michrandi, S., & Azmi, F. 2016. Perancangan otentikasi sidik jari pada biometrik payment design of authentication fingerprint for biometric payment. *e-Proceeding of Engineering*, Hal: 824-830, ISSN : 2355-9365.

Auliannisa, Rizkia Dwi; Suratman, Fiky Yosef; Rizal, Achmad. 2017. Deteksi Katarak Menggunakan Metode Transformasi Hough Berbasis Android. *e-Proceeding of Engineering : Vol.4, No.3*, Hal: 3310-3319, ISSN: 2355-9365.

Barri, M. W., Lumenta, A. S., & Wowor, A. 2015. Perancangan Aplikasi SMS GATEWAY Untuk Pembuatan Kartu Perpustakaan di Fakultas Teknik Unsrat. *E-journal Teknik Elektro dan Komputer*, Hal : 23-28, ISSN : 2301-8402.

Firman, A., Wowor, H., & Najoan, X. 2016. Sistem Informasi Perpustakaan Online Berbasis Web. *E-journal Teknik Elektro dan Komputer*, Hal : 29-36, ISSN : 2301-8402.

Hasmin, E. 2016. Aplikasi rekam kehadiran dengan deteksi wajah menggunakan metode eignface pada kejaksaan tinggi sulawesi selatan. *Seminar Nasional Teknologi Informasi dan Multimedia 2016*, Hal: 411-416, ISSN: 2302-3805.

Kalyani. 2017. Various Biometric Authentication Techniques: A Review. *Journal of Biometrics & Biostatistics*, Hal: 1-5, ISSN: 2155-6180.

Kustian, N. 2017. Analisis komponen utama menggunakan metode eigenface terhadap pengenalan citra wajah. *jurnal teknologi*, Hal : 44-48, ISSN : 2085 – 1669.

Masrani, H., Ilhamsyah, & Ruslianto, I. 2018. Aplikasi pengenalan pola pada huruf tulisan tangan menggunakan jaringan saraf tiruan dengan metode ekstraksi fitur geometri. *Jurnal Coding, Sistem Komputer Untan*, Hal: 69-78, ISSN: 2338-493X.

Nugroho, E. 2009. *Biometrika Mengenal Sistem Identifikasi Masa Depan.* Yogyakarta: C.V ANDI OFFSET.

Nugroho, H. 2017. Image Enhancement Pada Screen Capture CCTV Dengan Menggunakan Metode Histogram Ekualisasi. *KINETIK*, hal: 99-106, ISSN: 2503-2259.

Pamungkas, P. D., & Hariri, F. R. 2016. Pengenalan Citra Tanda Tangan Menggunakan Metode. *Citec Journal*, Hal: 269-279, ISSN: 2460-4259.

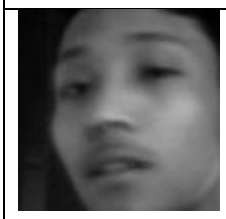
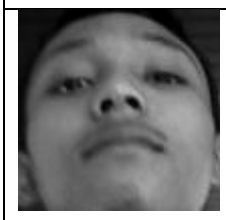
Yahya, & Nur, M. A. 2018. Pengaruh Aplikasi C# dalam Proses Perhitungan Numerik. *Jurnal Informatika dan Teknologi*, hal: 79 – 87, e-ISSN 2614-8773.

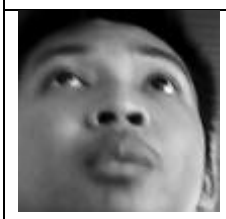
LAMPIRAN

Lampiran A. Biodata

Nama Lengkap	:	Sugeng Widodo
NIM	:	15111045
Alamat Asal	:	Jln. Jaga Desa, Pahonjean, Majenang.
Alamat di Yogyakarta	:	Jln. Jembatan Merah 2, Kampung Prayan Kulon, RW 37, Soropadan, Condongcatur, Depok, Sleman, Yogyakarta.
No. Hp	:	08882744178
Alamat E-Mail	:	kalibaru09@gmail.com
Website	:	-
Fakultas	:	Teknologi Informasi
Jurusan	:	Informatika
Universitas	:	Universitas Mercu Buana Yogyakarta

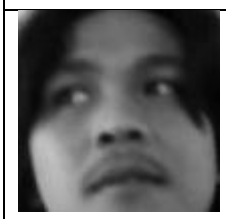
Lampiran B. Data Train Wajah

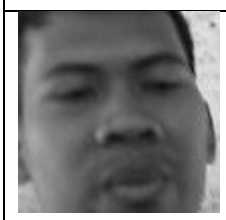
No	ID	Pose Train Wajah	Keterangan
1	15031027	Hadap Depan	Berhasil
		 Hadap Kanan	Berhasil
		Hadap Kiri	Berhasil
		 Hadap Atas	Berhasil
		Hadap Bawah	Berhasil

2	15021064	 Hadap Depan	Berhasil
		 Hadap Kanan	Berhasil
		 Hadap Kiri	Berhasil
		 Hadap Atas	Berhasil
		 Hadap Bawah	Berhasil
3	15021154	 Hadap Depan	Berhasil

		 Hadap Kanan	Berhasil
		 Hadap Kiri	Berhasil
		 Hadap Atas	Berhasil
		 Hadap Bawah	Berhasil
4	15021087	 Hadap Depan	Berhasil
		 Hadap Kanan	Berhasil

		 Hadap Kiri	Berhasil
		 Hadap Atas	Berhasil
		 Hadap Bawah	Berhasil
5	15011059	 Hadap Depan	Berhasil
		 Hadap Kanan	Berhasil
		 Hadap Kiri	Berhasil

		 Hadap Atas	Berhasil
		 Hadap Bawah	Berhasil
6	15021064	 Hadap Depan	Berhasil
		 Hadap Kanan	Berhasil
		 Hadap Kiri	Berhasil
		 Hadap Atas	Berhasil

		 Hadap Bawah	Berhasil
7	18021018	 Hadap Depan	Berhasil
		 Hadap Kanan	Berhasil
		 Hadap Kiri	Berhasil
		 Hadap Atas	Berhasil
		 Hadap Bawah	Berhasil

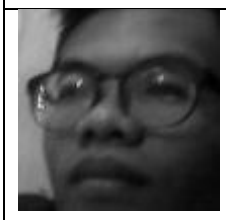
8	17081172	 Hadap Depan	Berhasil
		 Hadap Kanan	Berhasil
		 Hadap Kiri	Berhasil
		 Hadap Atas	Berhasil
		 Hadap Bawah	Berhasil
9	17021089	 Hadap Depan	Berhasil

		 Hadap Kanan	Berhasil
		 Hadap Kiri	Berhasil
		 Hadap Atas	Berhasil
		 Hadap Bawah	Berhasil
10	15111027	 Hadap Depan	Berhasil
		 Hadap Kanan	Berhasil

		 Hadap Kiri	Berhasil
		 Hadap Atas	Berhasil
		 Hadap Bawah	Berhasil
11	18031008	 Hadap Depan	Berhasil
		 Hadap Kanan	Berhasil
		 Hadap Kiri	Berhasil

		 Hadap Atas	Berhasil
		 Hadap Bawah	Berhasil
12	15111027	 Hadap Depan	Berhasil
		 Hadap Kanan	Berhasil
		 Hadap Kiri	Berhasil
		 Hadap Atas	Berhasil

		 Hadap Bawah	Berhasil
13	15111059	 Hadap Depan	Berhasil
		 Hadap Kanan	Berhasil
		 Hadap Kiri	Berhasil
		 Hadap Atas	Berhasil
		 Hadap Bawah	Berhasil

14	15111018	 Hadap Depan	Berhasil
		 Hadap Kanan	Berhasil
		 Hadap Kiri	Berhasil
		 Hadap Atas	Berhasil
		 Hadap Bawah	Berhasil

Lampiran C. Source Code MainForm.cs

```
using System;
using System.Collections.Generic;
using System.Drawing;
using System.Windows.Forms;
using Emgu.CV;
using Emgu.CV.Structure;
using Emgu.CV.CvEnum;
using System.IO;
using System.Diagnostics;
using System.Data.Sql;
using System.Linq;
using System.Data.SqlClient;
using System.Data;
using MySql.Data.MySqlClient;

namespace MultiFaceRec
{

    public partial class FrmPrincipal : Form
    {
        // variables, vector, dan haarcascade
        //-----------------------------------------------

        Image<Bgr, Byte> currentFrame;
        Capture grabber;
        HaarCascade face;
        MCvFont font = new MCvFont(FONT.CV_FONT_HERSHEY_TRIPLEX, 0.5d, 0.5d);
        Image<Gray, byte> result, TrainedFace = null;
        Image<Gray, byte> gray = null;
        List<Image<Gray, byte>> trainingImages = new List<Image<Gray, byte>>();
        List<string> labels = new List<string>();
        List<string> NamePersons = new List<string>();
        int ContTrain, NumLabels, t;
        public static string name ="";
        public static string names = "";


        //mulai
        //------------
        public FrmPrincipal()
        {
            InitializeComponent();

            face = new HaarCascade("haarcascade_frontalface_default.xml");
            try
            {
                string Labelsinfo = File.ReadAllText(Application.StartupPath +
"/TrainedFaces/TrainedLabels.txt");
                string[] Labels = Labelsinfo.Split('%');
                NumLabels = Convert.ToInt16(Labels[0]);
                ContTrain = NumLabels;
                string LoadFaces;

                for (int tf = 1; tf < NumLabels + 1; tf++)
                {
                    LoadFaces = "face" + tf + ".bmp";
                    trainingImages.Add(new Image<Gray,
```

```
byte>(Application.StartupPath + "/TrainedFaces/" + LoadFaces));
                    labels.Add(Labels[tf]);
                }

            }
            catch (Exception e)
            {
                //MessageBox.Show(e.ToString());
                MessageBox.Show("Data wajah dalam database masih kosong).",
"Trained faces load", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
            }

        }

        //start kemera process
        //------------------------
        public void button1_Click(object sender, EventArgs e)
        {

            grabber = new Capture();
            grabber.QueryFrame();

            Application.Idle += new EventHandler(FrameGrabber);
            button1.Enabled = false; //cemra start button


        }
        //start train img process
        //---------------------------
        public void button2_Click(object sender, System.EventArgs e)
        {
            try
            {

                ContTrain = ContTrain + 1;

                //ambil gambar dari kamera 320x240 dan buat jadi 20x20 for
TrainedFace
                gray = grabber.QueryGrayFrame().Resize(320, 240,
Emgu.CV.CvEnum.INTER.CV_INTER_CUBIC);

                //TestImageBox.Image = gray;
                MCvAvgComp[][] facesDetected = gray.DetectHaarCascade(
                face,
                1.2,
                7,
                Emgu.CV.CvEnum.HAAR_DETECTION_TYPE.DO_CANNY_PRUNING,
                new Size(20, 20)); // new size of pic only face

                foreach (MCvAvgComp f in facesDetected[0])
                {
                    TrainedFace = currentFrame.Copy(f.rect).Convert<Gray,
byte>(); //converting the pic to gray and save it to TranedFace
                    break;
                }

                //Trained image is been save to "Trainedface" variable
                TrainedFace = result.Resize(100, 100,
Emgu.CV.CvEnum.INTER.CV_INTER_CUBIC);
                trainingImages.Add(TrainedFace);
```

```
                //adding text box train name to label
                labels.Add(textBox1.Text);

                imageBox1.Image = TrainedFace;

                File.WriteAllText(Application.StartupPath +
"/TrainedFaces/TrainedLabels.txt", trainingImages.ToArray().Length.ToString() +
"%");


                for (int i = 1; i < trainingImages.ToArray().Length + 1; i++)
                {
                    trainingImages.ToArray()[i - 1].Save(Application.StartupPath
+ "/TrainedFaces/face" + i + ".bmp");
                    File.AppendAllText(Application.StartupPath +
"/TrainedFaces/TrainedLabels.txt", labels.ToArray()[i - 1] + "%");
                }

                MessageBox.Show(textBox1.Text + "´s face detected and added :)",
"Training OK", MessageBoxButtons.OK, MessageBoxIcon.Information);
            }
            catch
            {
                MessageBox.Show("Enable the face detection first", "Training
Fail", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
            }
        }

        //Matching caotured image with the TrainedFace
        //-----------------------------------------
        public void FrameGrabber(object sender, EventArgs e)
        {
            label3.Text = "0";

            NamePersons.Add("");


            // capture a frame form  device both face and all things on the
image
            currentFrame = grabber.QueryFrame().Resize(320, 240,
Emgu.CV.CvEnum.INTER.CV_INTER_CUBIC);


            gray = currentFrame.Convert<Gray, Byte>();

//            (TestImageBox.Image = currentFrame);

//            Result of haarCascade will be on the "MCvAvgComp"-facedetected (is
it face or not )
            MCvAvgComp[][] facesDetected = gray.DetectHaarCascade(
          face,
          1.2,
          7,
          Emgu.CV.CvEnum.HAAR_DETECTION_TYPE.DO_CANNY_PRUNING,
          new Size(20, 20));


            foreach (MCvAvgComp f in facesDetected[0])
            {
                t = t + 1;
```

```
                result = currentFrame.Copy(f.rect).Convert<Gray,
byte>().Resize(100, 100, Emgu.CV.CvEnum.INTER.CV_INTER_CUBIC);

                currentFrame.Draw(f.rect, new Bgr(Color.Red), 2); //Frame detect
colour is 'read'


                if (trainingImages.ToArray().Length != 0)
                {
                    MCvTermCriteria termCrit = new MCvTermCriteria(ContTrain,
0.001);


                    EigenObjectRecognizer recognizer = new
EigenObjectRecognizer(
                       trainingImages.ToArray(),
                       labels.ToArray(),
                       3000,
                       ref termCrit);

                    name = recognizer.Recognize(result); // detected name of the
face is been saved  to the 'name'-variable

                    //the colour of  the face label name
                    currentFrame.Draw(name, ref font, new Point(f.rect.X - 2,
f.rect.Y - 2), new Bgr(Color.LightGreen));

                }

                NamePersons[t - 1] = name;
                NamePersons.Add("");

                label3.Text = facesDetected[0].Length.ToString();

            }
            t = 0;


            for (int nnn = 0; nnn < facesDetected[0].Length; nnn++)
            {
                names = names + NamePersons[nnn] + ", ";
            }

            imageBoxFrameGrabber.Image = currentFrame;
            label4.Text = names;

            names = "";

            NamePersons.Clear();

        }
        //============================================================
        //data base saving process ids done here
        //============================================================

        ConnectionString cs = new ConnectionString();
        CommonClasses cc = new CommonClasses();
        clsFunc cf = new clsFunc();
        string st1;
        string st2;
```

```
        public void FrmPrincipal_Load(object sender, EventArgs e)
        {
            textBox2.Text = DateTime.Now.ToLongTimeString();
            textBox3.Text = DateTime.Now.ToString("yyyy/MM/dd");
        }


        //Timer Method
        //------------
        public void timer1_Tick(object sender, EventArgs e)
        {


                //this.Hide();


        }


         private void releaseObject(object obj)

        {

            try

            {

                System.Runtime.InteropServices.Marshal.ReleaseComObject(obj);

                obj = null;

            }

            catch (Exception ex)

            {

                obj = null;

                MessageBox.Show("Exception Occured while releasing object " +
ex.ToString());

            }

            finally

            {

                GC.Collect();

            }

        }

         private void biodata_Click(object sender, EventArgs e)
         {
             string connectionString =
@"server=localhost;userid=root;database=skripsi_login";
```

```
            MySqlConnection connection = null;
            if (name == "")
            {
              MessageBox.Show("Data Tidak Ditemukan");
            }
            else
            {
            try
            {
                connection = new MySqlConnection(connectionString);
                connection.Open();
                MySqlCommand cmd = new MySqlCommand();
                cmd.Connection = connection;
                string sql = "UPDATE  biodata set Tanggal ='" + textBox3.Text +
"',jamMasuk = '"+textBox2.Text+"',keterangan='Berhasil' where ID = '"+name+"'";
                cmd.CommandText = sql;
                cmd.Prepare();
                cmd.ExecuteNonQuery();
                MessageBox.Show("Login Berhasil");
            }
            finally
            {
                if (connection != null)
                    connection.Close();
            }

         }
         }


         private void textBox3_TextChanged(object sender, EventArgs e)
         {

         }
        void BtnPresentClick(object sender, EventArgs e)
        {

         }

        void Label7Click(object sender, EventArgs e)
        {

         }

        void Panel3Paint(object sender, PaintEventArgs e)
        {

         }

        void Panel1Paint(object sender, PaintEventArgs e)
        {

         }

        void Label3Click(object sender, EventArgs e)
        {

         }

        void Button3Click(object sender, EventArgs e)
```

```
        {
               string connectionString =
@"server=localhost;userid=root;database=skripsi_login";

            MySqlConnection connection = null;
            try
            {
                connection = new MySqlConnection(connectionString);
                connection.Open();
                MySqlCommand cmd = new MySqlCommand();
                cmd.Connection = connection;
                string id = txtID.Text;
                string nama = txtNama2.Text;
                string jurusan = txtJurusan2.Text;
                string sql = "Insert into biodata (ID,Nama,Keahlian) values ('"
+ id + "','"+nama+"','"+jurusan+"')";
                cmd.CommandText = sql;
                cmd.Prepare();
                cmd.ExecuteNonQuery();
                MessageBox.Show("Berhasil Menyimpan Data biodata");
            }
            finally
            {
                if (connection != null)
                    connection.Close();
            }
        }

        void TextBox2TextChanged(object sender, EventArgs e)
        {

         }

        void TextBox4TextChanged(object sender, EventArgs e)
        {

         }

        void PresentClick(object sender, EventArgs e)
        {

         }
    }


    }
```

Lampiran D. Source Code Recognizer.cs

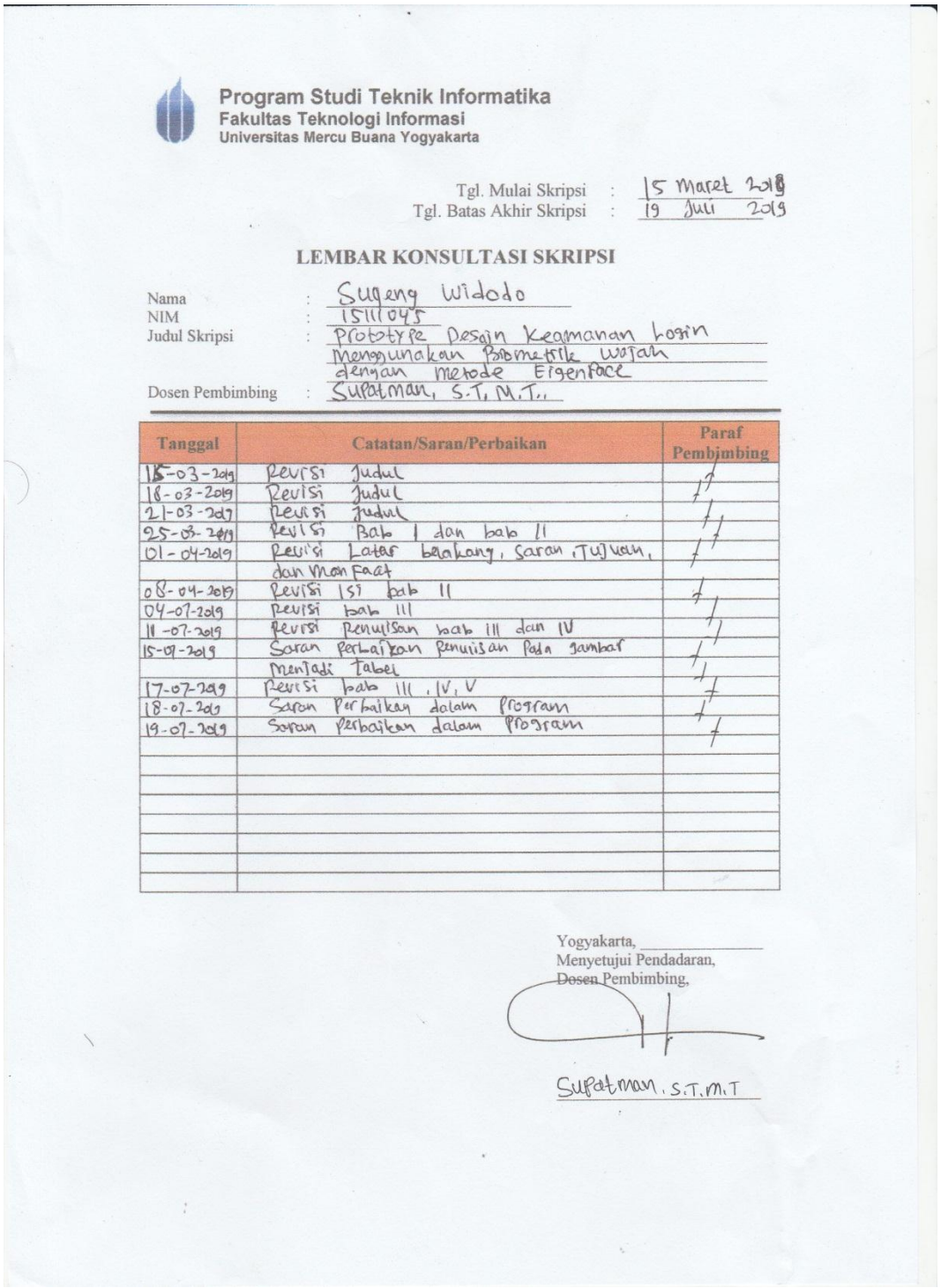
```
using System;
using System.Diagnostics;
using Emgu.CV.Structure;
namespace Emgu.CV
{

//=======================================================================================
========================================

   [Serializable]
   public class EigenObjectRecognizer
   {
      private Image<Gray, Single>[] _eigenImages;
      private Image<Gray, Single> _avgImage;
      private Matrix<float>[] _eigenValues;
      private string[] _labels;
      private double _eigenDistanceThreshold;
      public Image<Gray, Single>[] EigenImages
      {
         get { return _eigenImages; }
         set { _eigenImages = value; }
      }

      public String[] Labels
      {
         get { return _labels; }
         set { _labels = value; }
      }


      public double EigenDistanceThreshold
      {
         get { return _eigenDistanceThreshold; }
         set { _eigenDistanceThreshold = value; }
      }


      public Image<Gray, Single> AverageImage
      {
         get { return _avgImage; }
         set { _avgImage = value; }
      }


      public Matrix<float>[] EigenValues
      {
         get { return _eigenValues; }
         set { _eigenValues = value; }
      }

      private EigenObjectRecognizer()
      {
      }


      public EigenObjectRecognizer(Image<Gray, Byte>[] images, ref
MCvTermCriteria termCrit)
         : this(images, GenerateLabels(images.Length), ref termCrit)
```

```
      {
      }

      private static String[] GenerateLabels(int size)
      {
         String[] labels = new string[size];
         for (int i = 0; i < size; i++)
            labels[i] = i.ToString();
         return labels;
      }


      public EigenObjectRecognizer(Image<Gray, Byte>[] images, String[] labels,
ref MCvTermCriteria termCrit)
         : this(images, labels, 0, ref termCrit)
      {
      }

      public EigenObjectRecognizer(Image<Gray, Byte>[] images, String[] labels,
double eigenDistanceThreshold, ref MCvTermCriteria termCrit)
      {
         Debug.Assert(images.Length == labels.Length, "The number of images
should equals the number of labels");
         Debug.Assert(eigenDistanceThreshold >= 0.0, "Eigen-distance threshold
should always >= 0.0");

         CalcEigenObjects(images, ref termCrit, out _eigenImages, out
_avgImage);


         _eigenValues = Array.ConvertAll<Image<Gray, Byte>,
Matrix<float>>(images,
             delegate(Image<Gray, Byte> img)
             {
                return new Matrix<float>(EigenDecomposite(img, _eigenImages,
_avgImage));
             });

         _labels = labels;

         _eigenDistanceThreshold = eigenDistanceThreshold;
      }

      #region static methods

      public static void CalcEigenObjects(Image<Gray, Byte>[] trainingImages,
ref MCvTermCriteria termCrit, out Image<Gray, Single>[] eigenImages, out
Image<Gray, Single> avg)
      {
         int width = trainingImages[0].Width;
         int height = trainingImages[0].Height;

         IntPtr[] inObjs = Array.ConvertAll<Image<Gray, Byte>,
IntPtr>(trainingImages, delegate(Image<Gray, Byte> img) { return img.Ptr; });

         if (termCrit.max_iter <= 0 || termCrit.max_iter >
trainingImages.Length)
            termCrit.max_iter = trainingImages.Length;

         int maxEigenObjs = termCrit.max_iter;
```

```
         #region initialize eigen images
         eigenImages = new Image<Gray, float>[maxEigenObjs];
         for (int i = 0; i < eigenImages.Length; i++)
            eigenImages[i] = new Image<Gray, float>(width, height);
         IntPtr[] eigObjs = Array.ConvertAll<Image<Gray, Single>,
IntPtr>(eigenImages, delegate(Image<Gray, Single> img) { return img.Ptr; });
         #endregion

         avg = new Image<Gray, Single>(width, height);

         CvInvoke.cvCalcEigenObjects(
             inObjs,
             ref termCrit,
             eigObjs,
             null,
             avg.Ptr);
      }

      public static float[] EigenDecomposite(Image<Gray, Byte> src, Image<Gray,
Single>[] eigenImages, Image<Gray, Single> avg)
      {
         return CvInvoke.cvEigenDecomposite(
             src.Ptr,
             Array.ConvertAll<Image<Gray, Single>, IntPtr>(eigenImages,
delegate(Image<Gray, Single> img) { return img.Ptr; }),
             avg.Ptr);
      }
      #endregion


      public Image<Gray, Byte> EigenProjection(float[] eigenValue)
      {
         Image<Gray, Byte> res = new Image<Gray, byte>(_avgImage.Width,
_avgImage.Height);
         CvInvoke.cvEigenProjection(
             Array.ConvertAll<Image<Gray, Single>, IntPtr>(_eigenImages,
delegate(Image<Gray, Single> img) { return img.Ptr; }),
             eigenValue,
             _avgImage.Ptr,
             res.Ptr);
         return res;
      }

      public float[] GetEigenDistances(Image<Gray, Byte> image)
      {
         using (Matrix<float> eigenValue = new
Matrix<float>(EigenDecomposite(image, _eigenImages, _avgImage)))
            return Array.ConvertAll<Matrix<float>, float>(_eigenValues,
                delegate(Matrix<float> eigenValueI)
                {
                   return (float)CvInvoke.cvNorm(eigenValue.Ptr,
eigenValueI.Ptr, Emgu.CV.CvEnum.NORM_TYPE.CV_L2, IntPtr.Zero);
                });
      }

      public void FindMostSimilarObject(Image<Gray, Byte> image, out int index,
out float eigenDistance, out String label)
      {
         float[] dist = GetEigenDistances(image);
```

```
         index = 0;
         eigenDistance = dist[0];
         for (int i = 1; i < dist.Length; i++)
         {
            if (dist[i] < eigenDistance)
            {
               index = i;
               eigenDistance = dist[i];
            }
         }
         label = Labels[index];
      }

      public String Recognize(Image<Gray, Byte> image)
      {
         int index;
         float eigenDistance;
         String label;
         FindMostSimilarObject(image, out index, out eigenDistance, out label);

         return (_eigenDistanceThreshold <= 0 || eigenDistance <
_eigenDistanceThreshold )  ? _labels[index] : String.Empty;
      }
   }
}
```

Lampiran E. Scan Form Bimibingan Skripsi



Program Studi Teknik Informatika
Fakultas Teknologi Informasi
Universitas Mercu Buana Yogyakarta

Tgl. Mulai Skripsi : 15 Maret 2019
Tgl. Batas Akhir Skripsi : 19 Juli 2019

LEMBAR KONSULTASI SKRIPSI

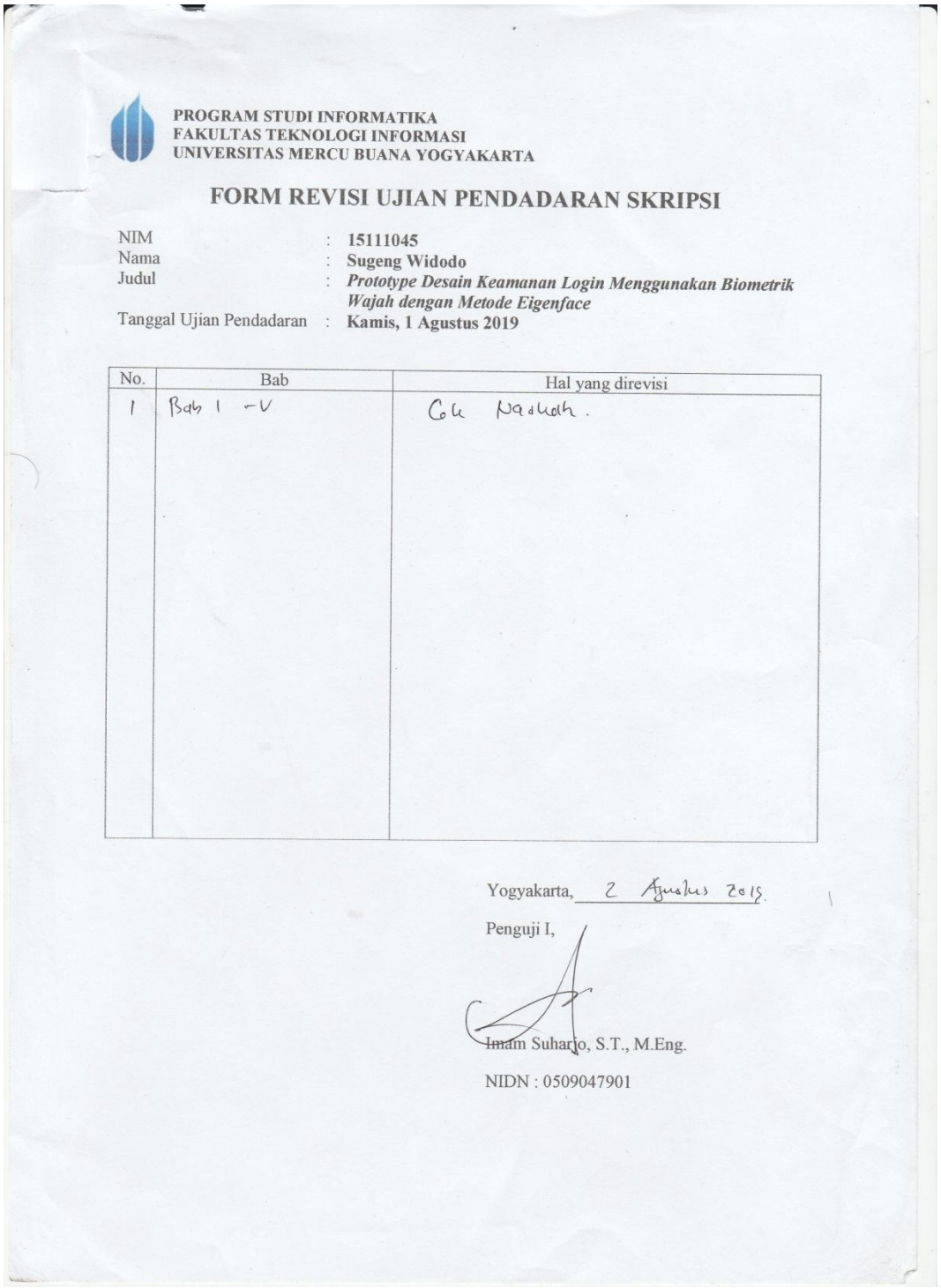
Nama : Sugeng Widodo
NIM : 15111045
Judul Skripsi : Prototype Desain Keamanan Login Menggunakan Biometrik Wajah dengan metode Eigenface
Dosen Pembimbing : Supatman, S.T, M.T.,

Tanggal	Catatan/Saran/Perbaikan	Paraf Pembimbing
15-03-2019	Revisi Judul	
18-03-2019	Revisi Judul	
21-03-2019	Revisi Judul	
25-03-2019	Revisi Bab I dan bab II	
01-04-2019	Revisi Latar belakang, Saran, Tujuan, dan manfaat	
08-04-2019	Revisi isi bab II	
04-07-2019	Revisi bab III	
11-07-2019	Revisi Penulisan bab III dan IV	
15-07-2019	Saran Perbaikan Penulisan Pada gambar menjadi Tabel	
17-07-2019	Revisi bab III, IV, V	
18-07-2019	Saran Perbaikan dalam Program	
19-07-2019	Saran Perbaikan dalam Program	

Yogyakarta, _____
Menyetujui Pendadaran,
Dosen Pembimbing,

Supatman, S.T, M.T

Lampiran F. Scan Form Revisi



PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS MERCU BUANA YOGYAKARTA

FORM REVISI UJIAN PENDADARAN SKRIPSI

NIM : **15111045**
Nama : **Sugeng Widodo**
Judul : ***Prototype Desain Keamanan Login Menggunakan Biometrik Wajah dengan Metode Eigenface***
Tanggal Ujian Pendadaran : **Kamis, 1 Agustus 2019**

No.	Bab	Hal yang direvisi
1	Bab 1 - V	Cek Naskah.

Yogyakarta, 2 Agustus 2019.

Penguji I,

Imam Suharjo, S.T., M.Eng.

NIDN : 0509047901

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS MERCU BUANA YOGYAKARTA

FORM REVISI UJIAN PENDADARAN SKRIPSI

NIM	:	**15111045**
Nama	:	**Sugeng Widodo**
Judul	:	***Prototype Desain Keamanan Login Menggunakan Biometrik Wajah dengan Metode Eigenface***
Tanggal Uian Pendadaran	:	**Kamis, 1 Agustus 2019**

No.	Bab	Hal yang direvisi
1.	1-5	1. Buku
2.	4	2. Uji dengan data 30, Tarik validitas

Yogyakarta, 8/8/2019

Penguji III,

Arita Witanti, S.T., M.T.

NIDN : 0422018102

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS MERCU BUANA YOGYAKARTA

FORM REVISI UJIAN PENDADARAN SKRIPSI

NIM : **15111045**
Nama : **Sugeng Widodo**
Judul : ***Prototype Desain Keamanan Login Menggunakan Biometrik Wajah dengan Metode Eigenface***
Tanggal Ujian Pendadaran : **Kamis, 1 Agustus 2019**

No.	Bab	Hal yang direvisi
(1)		+ TOLERANSI BATAS KEDEKATAN JARAK - OVER FITTING,

Yogyakarta, 01-08-19

Penguji III,

Mutaqin Akbar, S.Kom., M.T.

NIDN : 0528078902