

Penerapan Estimasi Posisi dan *Tracking* Wajah Pada Sistem Presensi Mahasiswa

<http://dx.doi.org/10.28932/jutisi.v6i2.2730>

Afrillebar Putra Pratama [✉]#1, Agi Prasetiadi*², Elisa Usada^{#3}

[#]Jurusan SI Teknik Informatika, Institut Teknologi Telkom Purwokerto

Jl. DI Panjaitan No.128, Pancurawis, Purwokerto Kidul, Kec. Purwokerto Sel., Kabupaten Banyumas, Jawa Tengah

¹16102145@ittelkom-pwt.ac.id

²agi@ittelkom-pwt.ac.id

³elisa@ittelkom-pwt.ac.id

Abstract — The current presence system can be done with a computerized system, one of which is the face biometric system. This study focuses on the application of position estimation and tracking based on clustering on people's faces to determine the position in three dimensions. Position estimation can be obtained by making a kernel that is ready to be used to predict three-dimensional coordinates of faces based on two-dimensional coordinates of two images. Position estimation can be done by utilizing the Machine Learning algorithm family. In this study, Least Absolute Shrinkage and Selection Operators (LASSO) is used to perform the position estimation. Meanwhile, clustering in this study uses the K-Means algorithm. Based on the test results, the kernel error obtained in estimating the face location is 9.23 cm. The tracking accuracy of an object based on clustering is 100%.

Keywords— K-Means; LASSO; Machine Learning; Object Tracking; Position Estimation.

I. PENDAHULUAN

Sistem presensi berdasarkan cara penggunaannya ada dua, yaitu manual dan non-manual. Presensi manual dilakukan dengan tanda tangan di buku kehadiran. Sedangkan non-manual dilakukan dengan alat yang sudah terkomputerisasi [1].

Presensi manual memiliki banyak kelemahan seperti terjadinya tindak kecurangan oleh mahasiswa dengan memalsukan tanda tangan serta dapat terjadi akibat *human error* ketika *staff* kampus memasukkan data ke *database*. Saat ini, presensi manual dapat digantikan dengan sistem yang sudah terkomputerisasi yaitu dengan sistem biometrik contohnya sidik jari, suara, wajah, dan lain-lain [2].

Perkembangan sistem presensi saat ini sudah mulai merambah melalui identifikasi wajah yang dilakukan oleh kamera di dalam ruangan yang berfungsi untuk mengidentifikasi dan mengetahui posisi wajah [3]. Pemanfaatan kamera digital dilakukan untuk mengetahui estimasi posisi atau menentukan letak lokasi suatu objek

dalam suatu ruang yang terjangkau pandangan kamera sehingga dapat mengenali koordinat lokasi objek [4].

Estimasi posisi dengan pendekatan ciri visual pada umumnya dapat dilakukan dengan dua kamera (stereoskopik) atau kamera 3D yang sudah dilengkapi dengan sensor kedalaman. Teknik-teknik tersebut membutuhkan sensor yang lebih banyak sehingga harga perangkat keras untuk aplikasi menjadi lebih mahal [5]. Sebagai alternatifnya, penelitian ini menggunakan dua kamera *smartphone* dengan meniru cara kerja dari mata manusia yang memiliki dua sudut pandang berbeda. Kamera *smartphone* tersebut digunakan sebagai masukkan citra untuk menghasilkan citra digital yang kemudian diolah dengan model *Machine Learning* untuk memprediksi lokasi dalam bentuk koordinat 3D dan dapat membedakan orang satu dengan orang lainnya.

Penelitian ini mengacu pada penelitian sebelumnya yang bertujuan merancang simulasi penentuan koordinat objek di dalam ruang menggunakan kamera digital. Penelitian tersebut menerapkan pendekatan proyeksi perspektif hasil kamera yang dapat dirumuskan secara matematis menggunakan dua titik hilang dengan empat titik acuan menggunakan objek berupa pola ubin berukuran 40,25x40,25 cm. Penggunaan metode ini memperoleh tingkat presisi mencapai 2,71 cm [4].

Berdasarkan acuan penelitian tersebut, penelitian ini melakukan alternatif pemodelan lain untuk memprediksi lokasi wajah dalam bentuk koordinat 3D menggunakan algoritma *Least Absolute Shrinkage and Selection Operator (LASSO)* dan melakukan *tracking* berbasis *clustering* untuk melacak tiap-tiap wajah dengan algoritma *K-Means*. Kedua algoritma tersebut dijalankan dengan memanfaatkan *Scikit-Learn Library*. Hasil dari estimasi posisi dan *clustering* kemudian di *plotting* ke dalam *3D Scatter Plot* dengan memanfaatkan *Matplotlib Library*. Dengan menggabungkan kedua fungsi tersebut, model ini dapat digunakan sebagai penunjang sistem presensi yang dibangun menggunakan *Tkinter GUI* sehingga dapat memberikan informasi baru berupa lokasi wajah sebagai alternatif model.

II. LANDASAN TEORI

A. Presensi

Presensi merupakan tindakan pendataan kehadiran sebagai aktifitas pelaporan di sebuah institusi yang telah tersusun rapi sehingga mudah dicari dan digunakan kembali saat diperlukan. Di dunia perkuliahan, presensi menjadi salah satu aspek bagi dosen dalam menentukan nilai akhir mahasiswa [1].

B. Estimasi Posisi

Estimasi posisi adalah proses melakukan estimasi terhadap posisi objek pada koordinat nyata berdasarkan koordinat objek pada video. Estimasi posisi dilakukan agar informasi mengenai pergerakan objek dapat dimengerti oleh manusia karena menggunakan koordinat dunia nyata [6].

Salah satu sistem koordinat yang terkenal adalah sistem koordinat kartesian. Sistem koordinat kartesian pada dasarnya terdiri dari sistem koordinat kartesian 2D dan 3D. Perbedaannya pada sistem koordinat 3D terdapat sumbu selain x dan y , yaitu sumbu z yang arahnya tegak lurus terhadap sumbu x dan y [7].

C. Citra Digital

Citra merupakan representasi 2D dari bentuk fisik nyata 3D. Citra digital merupakan citra 2D yang dapat ditampilkan pada layar komputer dalam bentuk piksel. Posisi koordinat pada bidang citra di layar mengacu pada sistem koordinat kartesian dengan sumbu mendatar dinyatakan dengan sumbu x , dan sumbu tegak dinyatakan dengan sumbu y [8].

D. Pengenalan Wajah

Sistem pengenalan wajah bekerja dengan cara mengambil data wajah digital dari sebuah citra atau *frame* pada video dan membandingkannya dengan data wajah yang tersimpan di *database*. Sistem pengenalan wajah yang ada harus mampu membedakan wajah dengan latar belakang citra agar dapat mengenali wajah manusia atau bukan [9].

E. OpenCV

OpenCV (*Open Computer Vision Library*) merupakan *open source library* yang memiliki lebih dari 500 algoritma dan telah dioptimasi serta disediakan untuk menangani permasalahan mengenai *computer vision* dan *machine learning* [3].

F. Machine Learning

Machine learning biasa digunakan untuk menirukan perilaku manusia dalam menyelesaikan suatu masalah atau melakukan otomatisasi dan ciri khasnya adalah proses pelatihan dan pembelajaran [10].

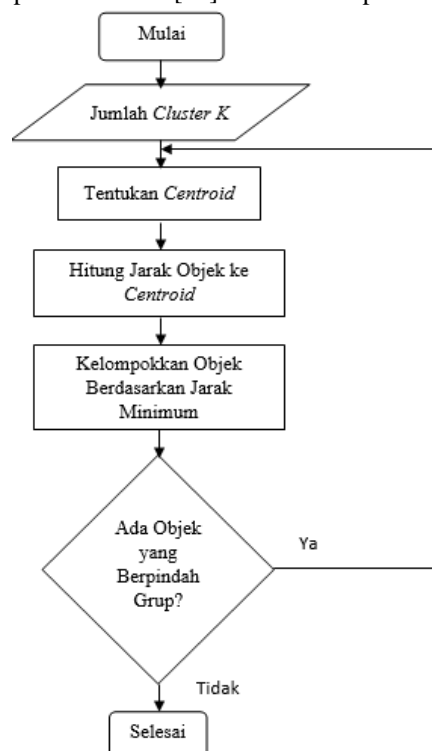
G. LASSO

LASSO merupakan salah satu metode *machine learning* yang cocok diterapkan pada data *non-linear* yang bersifat *polynomial* dan mengandung multikolinieritas karena antara

salah satu variabel atau lebih dengan variabel lainnya memiliki hubungan yang kuat. LASSO mengurangi sejumlah koefisien ke arah nol bahkan tepat nol sehingga model yang dihasilkan menjadi lebih sederhana dan terhindar dari multikolinieritas [11]. Algoritma LASSO ini digunakan untuk memprediksi koordinat 3D (x, y, z) berdasarkan masukkan koordinat 2D dari dua kamera ($x1, y1, x2, y2$).

H. K-Means

Klasterisasi yang ada pada *K-Means* dilakukan dengan cara memisahkan data ke dalam kelompok yang berbeda atau biasa disebut dengan cara *partitioning*. Cara ini mampu meminimalkan rata-rata jarak setiap data ke klasternya dan mampu melakukan komputasi lebih cepat dari pengelompokan hirarkis [12]. Berikut tahapan *K-Means*:



Gambar 1. Tahapan K-Means

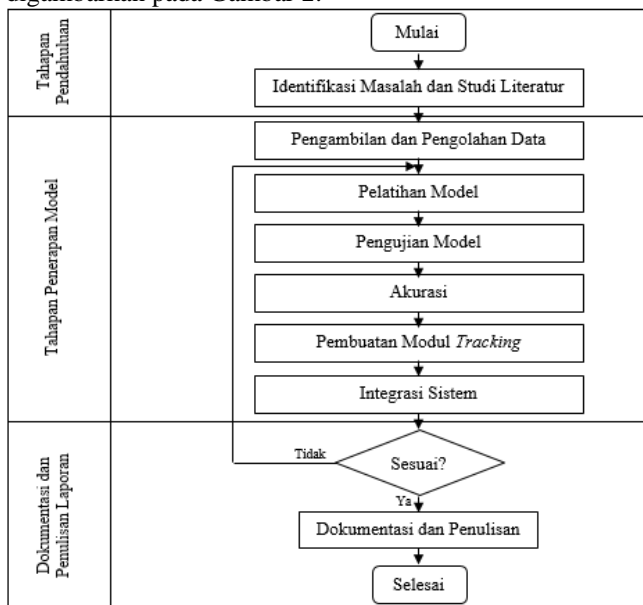
K-Means digunakan untuk mengelompokkan data piksel 2D berdasarkan jumlah orang. Jumlah dari data tersebut tergantung pada hasil tangkapan setiap *frame* pada durasi video yang telah ditentukan. Berdasarkan Gambar 1 dapat dijelaskan bahwa tahapan *K-Means* yaitu:

1. Tentukan nilai k sebagai jumlah klaster
Jumlah klaster tergantung pada jumlah orang yang ada di video demo yang digunakan sebagai data uji. Jika terdapat satu orang, maka jumlah klasternya adalah satu. Jika dua orang, maka jumlah klasternya dua dan begitu seterusnya.
2. Inisialiasasi k *centroid* awal secara acak
Setelah jumlah klaster ditentukan, selanjutnya menginisialisasi *centroid* secara acak sehingga didapatkan nilai tengah dari klaster yang akan dibentuk.

3. Hitung jarak objek ke *centroid*
Langkah ini dilakukan untuk mengelompokkan setiap data yang berada di jarak terdekat berdasarkan jumlah klaster. Penghitungan jarak yang digunakan pada penelitian ini yaitu menggunakan *Euclidean Distance*.
4. Tempatkan setiap data ke klaster terdekat
Tahap selanjutnya setelah perhitungan jarak objek ke *centroid* adalah dengan menempatkan setiap data atau objek berdasarkan jarak terdekat. Pada langkah ini data-data akan dikelompokkan menjadi beberapa kelompok sesuai dengan jumlah klaster yang telah ditentukan.
5. Pilih kembali *centroid* untuk tiap-tiap klaster
Apabila pengelompokkan telah dilakukan, selanjutnya pilih kembali *centroid* untuk tiap-tiap klaster dengan tujuan mengetahui apakah masih ada data yang berpindah kelompok atau tidak. Jika sudah tidak ada lagi yang berpindah kelompok, maka *clustering* telah selesai dilakukan. Namun, jika masih terdapat perpindahan kelompok maka kembali ke langkah dua hingga tidak ada lagi data yang berpindah kelompok.

III. METODOLOGI PENELITIAN

Pada penelitian ini terdapat tiga tahapan penelitian yang digambarkan pada Gambar 2.



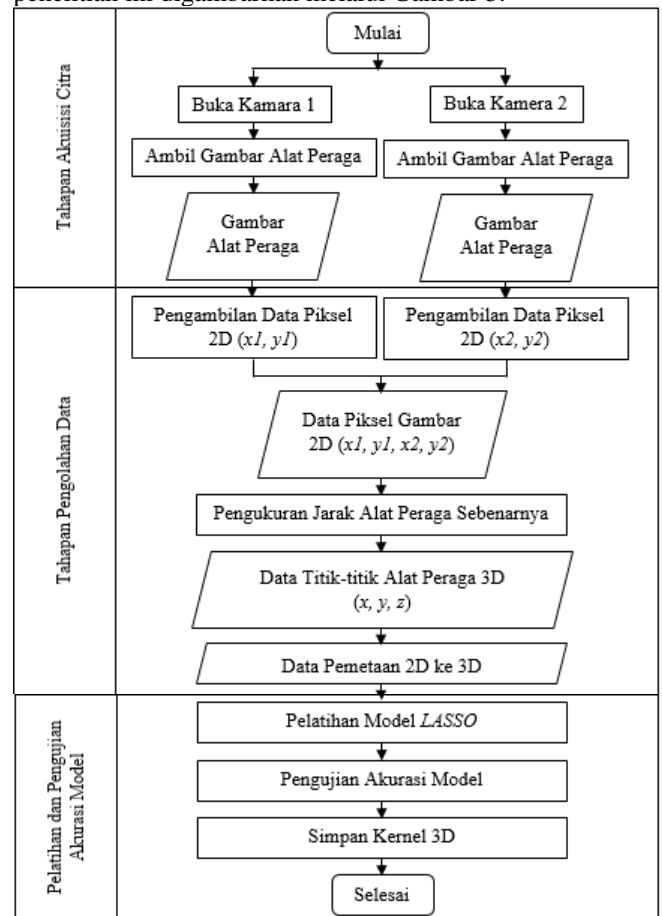
Gambar 2. Metodologi Penelitian

A. Identifikasi Masalah dan Studi Literatur

Identifikasi masalah yang dilakukan pada penelitian ini dengan cara observasi mengenai presensi yang tengah berjalan karena masih memerlukan campur tangan manusia. Permasalahan ini yang diangkat menjadi topik penelitian karena presensi bisa dilakukan secara otomatis dengan sistem biometrik wajah dan diketahui posisi untuk tiap-tiap orangnya. Sedangkan studi literatur yang dilakukan dengan membaca jurnal relevan.

B. Pengambilan dan Pengolahan Data

Tahapan pengambilan dan pengolahan data pada penelitian ini digambarkan melalui Gambar 3.



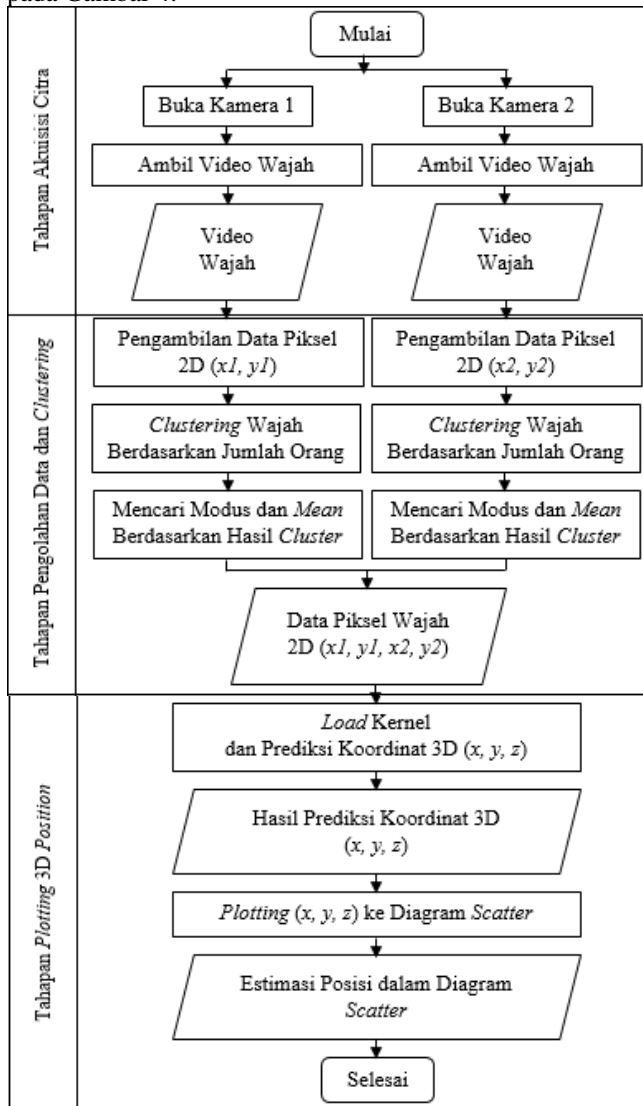
Gambar 3. Pengambilan dan Pengolahan Data

Berdasarkan Gambar 3 data diperoleh dari hasil pengambilan gambar alat peraga menggunakan dua kamera. Gambar kedua alat peraga tersebut kemudian ditentukan piksel 2D-nya lalu digabungkan sehingga menghasilkan data piksel gambar 2D (x_1, y_1, x_2, y_2). Langkah selanjutnya yaitu pengukuran jarak dari alat peraga yang sebenarnya. Pengukuran jarak ini dilakukan dengan mengukur panjang antara titik tengah alat peraga yang menyerupai koordinat kartesian 3D dengan ketiga sumbunya (x, y, z).

Apabila pengukuran jarak alat peraga sudah didapatkan, kemudian digabungkan dengan gambar 2D sehingga dihasilkan data pemetaan 2D dan 3D. Data inilah yang dijadikan sebagai data latih dan akan dieksekusi menggunakan model *LASSO*. Selanjutnya dilakukan pengujian akurasi menggunakan *Mean Absolute Error (MAE)*. Hasil dari pelatihan tersebut kemudian disimpan dengan memanfaatkan *Pickle Library* yang banyak digunakan untuk menyimpan model *Machine Learning*. Data dalam penyimpanan *Pickle* ini kemudian di *load* untuk memprediksi koordinat 3D berdasarkan masukkan piksel 2D dari data uji yang berupa titik tengah pada wajah manusia.

C. Pengujian Model

Tahapan pengujian model pada penelitian ini digambarkan pada Gambar 4.



Gambar 4. Pengujian Model

Gambar 4 merupakan tahapan pengujian model dimana data uji yang digunakan adalah data baru berupa koordinat dari citra wajah pada layar komputer. Koordinat tersebut didapatkan dari titik tengah piksel pada *bounding box* citra wajah yang telah terdeteksi. Selanjutnya dilakukan *tracking* berbasis *clustering* berdasarkan jumlah orang yang ada untuk tiap-tiap video yang diambil dari dua kamera berbeda, kemudian dicari modus untuk memastikan kebenaran setiap orang dari *clustering* yang telah dilakukan.

Langkah selanjutnya yaitu mencari *mean* berdasarkan pergerakan setiap orang yang ada pada video. Penggunaan *mean* ini bertujuan untuk mengetahui lokasi orang yang ada pada video karena pada dasarnya setiap mahasiswa tidak banyak bergerak ketika kegiatan belajar mengajar berlangsung.

Hasil *clustering* untuk tiap-tiap video dari dua kamera yang berbeda tersebut kemudian digabungkan sehingga didapatkan data piksel wajah 2D. Data piksel wajah 2D inilah yang akan diprediksi koordinat 3D-nya. Hasil prediksi tersebut kemudian di *plotting* ke dalam *3D Scatter Plot*.

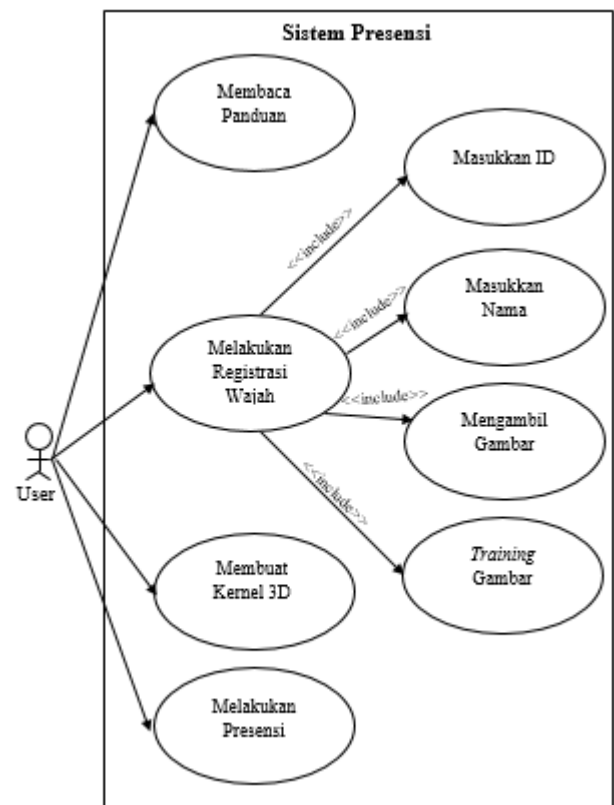
D. Pengembangan Sistem

Metode yang diterapkan dalam membangun perangkat lunak yaitu menggunakan SDLC. Berikut tahapan metode pada SDLC:

1) *Tahap Perencanaan:* Tahap awal ini menjelaskan identifikasi masalah, pengumpulan data, dan studi literatur yang bertujuan agar permasalahan yang akan diteliti dapat teridentifikasi dengan jelas.

2) *Tahap Analisis Kebutuhan:* Tahapan ini dilaksanakan untuk mengidentifikasi beberapa kebutuhan dasar dari sistem serta mempelajari teknik-teknik yang diperlukan untuk pengembangan sistem.

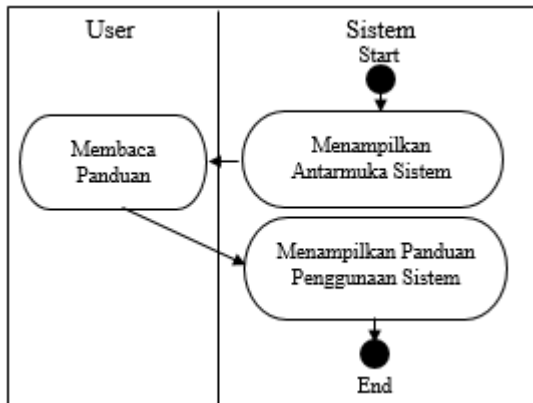
3) *Tahap Perancangan:* Tahap ini menggunakan *Unified Modeling Language (UML)* untuk membuat gambaran umum dan alur logika sistem.



Gambar 5. Use Case Diagram

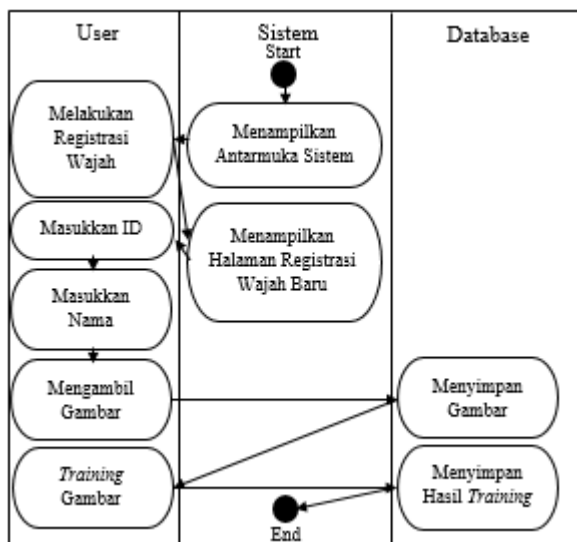
Gambar 5 merupakan aksi yang dilakukan *user* untuk dapat melakukan presensi pada sistem yang dibuat. *User* dapat membaca panduan untuk mengetahui panduan dalam menjalankan sistem, *user* juga dapat melakukan registrasi wajah baru dengan cara memasukkan ID, dan nama

kemudian diambil gambarnya untuk memperoleh *dataset* dan di *training*. Adapun *Activity Diagram* dari *Use Case* dapat dilihat pada Gambar 6, Gambar 7, Gambar 8, dan Gambar 9.



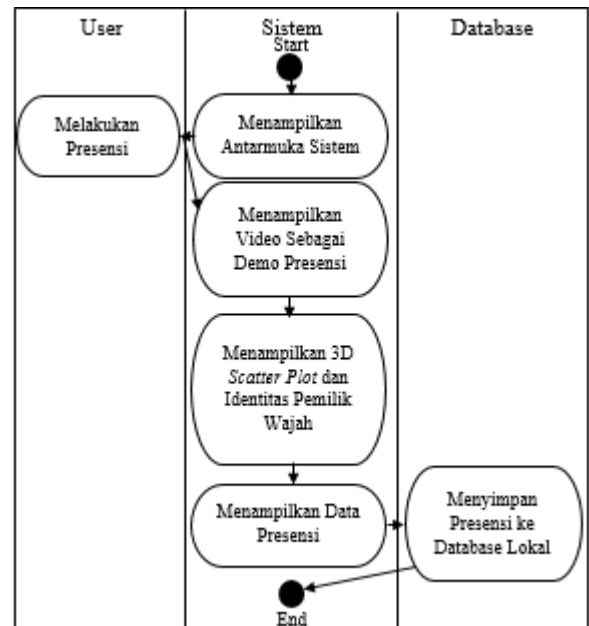
Gambar 6. Activity Diagram: Membaca Panduan

Gambar 6 menjelaskan bahwa *user* dapat mengetahui alur kerja dari sistem dengan cara membaca panduan yang ada di antarmuka sistem. Panduan tersebut berisi alur kerja bagaimana presensi dapat dilakukan dengan baik pada sistem tersebut.



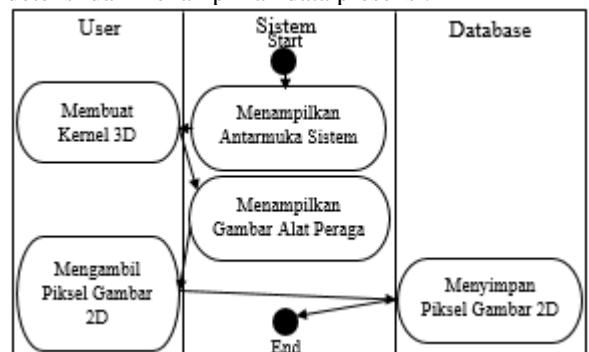
Gambar 7. Activity Diagram: Melakukan Registrasi Wajah

Gambar 7 menjelaskan mengenai tahapan yang harus dilalui *user* untuk melakukan registrasi wajah baru hingga dihasilkan data *training* agar dapat dilakukan identifikasi atau pengenalan wajah. *User* dapat melakukan registrasi wajah dengan memasukkan ID dan nama terlebih dahulu sebagai identitas dari orang tersebut, kemudian dilakukan proses pengambilan gambar untuk *dataset* wajah menggunakan *Haar Cascade Classifier* dan di *training* dengan *LBPH*.



Gambar 8. Activity Diagram: Melakukan Presensi

Gambar 8 menjelaskan bahwa presensi dapat dilakukan menggunakan video demo yang telah dibuat sebelumnya dan berisi beberapa orang sebagai simulasi presensi. Hasil akhir dari presensi ini akan menampilkan *mapping* berdasarkan estimasi posisi untuk setiap orang yang telah terdeteksi dan menampilkan data presensi.



Gambar 9. Activity Diagram: Membuat Kernel 3D

Gambar 9 menjelaskan mengenai pembuatan kernel 3D yang perlu dilakukan terlebih dahulu sebelum presensi. Hal ini diperlukan untuk mengetahui posisi dari setiap orang yang terdeteksi. Pembuatan kernel ini dilakukan menggunakan algoritma *LASSO*. Pikel 2D dari gambar alat peraga diperoleh dari program *Mouse Event Click Python*.

4) *Tahap Implementasi*: Tahap ini sistem mulai dibangun sesuai analisis kebutuhan dan perancangan yang telah dibuat. Penelitian ini menggunakan perangkat keras Laptop Asus A455L untuk mengimplementasikan sistem. Sedangkan untuk proses pengambilan citra digital menggunakan kamera *smartphone* Samsung Galaxy A50, dan Vivo v91. Sedangkan untuk membuat program *Python* menggunakan *Visual Studio Code* sebagai perangkat lunak.

5) *Tahap Pengujian*: Di tahap ini sistem dilakukan pengujian terhadap hasil dari implementasi yang telah dibuat. Ada lima pengujian yang ada pada penelitian ini yaitu pengujian pada kernel yang digunakan untuk memprediksi lokasi wajah, pengujian estimasi posisi untuk mengukur keakuratan posisi asli dan prediksi, pengujian pengenalan wajah, pengujian *clustering* untuk mengukur ketepatan hasil *cluster* terhadap beberapa wajah, dan pengujian sistem untuk mengetahui keseluruhan sistem berjalan dengan baik atau tidak.

IV. HASIL PENGUJIAN DAN ANALISIS

A. Pengujian Kernel LASSO



Gambar 10. Gambar Alat Peraga

Gambar 10 merupakan gambar alat peraga yang digunakan dalam pembuatan kernel karena menyerupai sistem koordinat kartesius 3D. Data yang diambil berupa piksel 2D dengan program *Mouse Event Click* pada *OpenCV Library*. Alat peraga tersebut diletakkan di lokasi tempat pengujian bersamaan dengan pengambilan video demo.

Data piksel 2D diambil dengan melakukan klik pada titik tengah, ujung kiri, ujung kanan, dan ujung atas secara bergantian dari dua gambar alat peraga berdasarkan dua kamera berbeda yang digunakan sebagai empat titik acuan. Tabel I dan Tabel II merupakan data dari piksel 2D.

TABEL I
PIKSEL GAMBAR KAMERA 1

x1	y1
294,0	264,0
130,0	260,0
434,0	259,0
296,0	102,0

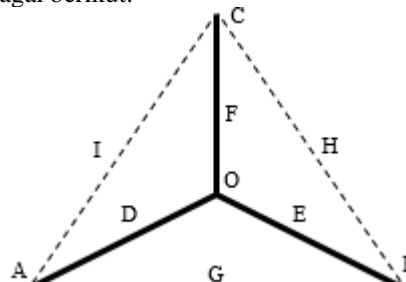
Tabel I merupakan data piksel 2D yang diperoleh dari gambar alat peraga kamera 1.

TABEL II
PIKSEL GAMBAR KAMERA 2

x2	y2
308,0	262,0
151,0	258,0
450,0	263,0
310,0	104,0

Tabel II merupakan data piksel 2D yang diperoleh dari gambar alat peraga kamera 2. Data pada Tabel I, dan Tabel II kemudian diolah dengan menambahkan titik-titik antara

titik tengah dengan tiap-tiap ujung dan antara ujung satu dengan ujung lainnya sehingga mendapatkan 10 titik gambar pada objek. Apabila digambarkan dalam bangun 3D adalah sebagai berikut:



Gambar 11. Alat Peraga dalam Bidang 3D

Keterangan:

Titik O = 0 cm

Titik A, B, C = 30 cm

Titik D, E, F, G, H, I = 15 cm

Gambar 11 menggambarkan pengukuran jarak alat peraga yang sebenarnya. Kemudian hasil pengukuran tersebut dituangkan pada Tabel III yang merupakan pemetaan 2D dan 3D yang berasal dari gambar objek pada layar sebagai piksel 2D dan besaran objek sesungguhnya di dunia nyata.

TABEL III
PEMETAAN 2D DAN 3D

x1	y1	x2	y2	x	y	z
294,0	264,0	308,0	262,0	0,0	0,0	0,0
130,0	260,0	151,0	258,0	30,0	0,0	0,0
434,0	259,0	450,0	263,0	0,0	30,0	0,0
296,0	102,0	310,0	104,0	0,0	0,0	30,0
212,0	262,0	229,5	260,0	15,0	0,0	0,0
364,0	261,5	379,0	262,5	0,0	15,0	0,0
295,0	183,0	309,0	183,0	0,0	0,0	15,0
282,0	259,5	300,5	260,5	15,0	15,0	0,0
365,0	180,5	380,0	183,5	0,0	15,0	15,0
213,0	181,0	230,5	181,0	15,0	0,0	15,0

Tabel III merupakan data pemetaan 2D dan 3D yang digunakan untuk data *training* dalam pembuatan kernel 3D. Pembuatan kernel hanya dilakukan sekali ketika letak kamera berada di lokasi yang sama, namun jika lokasi berbeda maka perlu membuat ulang kernel sesuai lokasi yang digunakan agar posisi objek dapat diketahui dengan akurat. Berdasarkan data tersebut, kemudian dilakukan pelatihan menggunakan algoritma *LASSO*. Penerapan algoritma *LASSO* akan dijelaskan pada kode program 1 berdasarkan *dataset* yang ada pada Tabel III.

Kode Program 1 Pelatihan dengan Algoritma *LASSO*

```
# Pengolahan Dataset
data = dataset.values
X = data[:, :4] # data x1,y1,x2,y2
a = data[:, 4:5] # data x
b = data[:, 5:6] # data y
c = data[:, 6:7] # data z
# creating model dan training data
lasso_a =Lasso(alpha=0.01).fit(X, a)
```

```
lasso_b =Lasso(alpha=0.01).fit(X, b)
lasso_c =Lasso(alpha=0.01).fit(X, c)
# prediction
y_pred1 = lasso_a.predict(X)
y_pred2 = lasso_b.predict(X)
y_pred3 = lasso_c.predict(X)
# save pickle
with open ('lasso_model_pickle_a.krn1','wb') as f:
    pickle.dump(lasso_a,f)

with open ('lasso_model_pickle_b.krn1','wb') as f:
    pickle.dump(lasso_b,f)

with open ('lasso_model_pickle_c.krn1','wb') as f:
    pickle.dump(lasso_c,f)
```

Hasil perbandingan antara koordinat asli dan prediksi 3D pada koordinat x, y, dan z dapat dilihat pada Tabel IV, Tabel V, dan Tabel VI.

TABEL IV
PERBANDINGAN KOORDINAT ASLI DAN PREDIKSI KOORDINAT X

No	Asli	Prediksi	Error (cm)
1	0	4,96	4,96
2	30	27,62	2,38
3	0	-2,63	2,63
4	0	0,04	0,04
5	15	16,29	1,29
6	0	1,17	1,17
7	0	2,50	2,50
8	15	12,50	2,50
9	0	-1,29	1,29
10	15	13,83	1,17
MAE			1,99

Berdasarkan Tabel IV didapatkan *error* berdasarkan perhitungan *MAE* koordinat x sebesar 1,99 cm, di mana *error* terbesar yaitu 4,96 cm dan terkecil sebesar 0,04 cm.

TABEL V
PERBANDINGAN KOORDINAT ASLI DAN PREDIKSI KOORDINAT Y

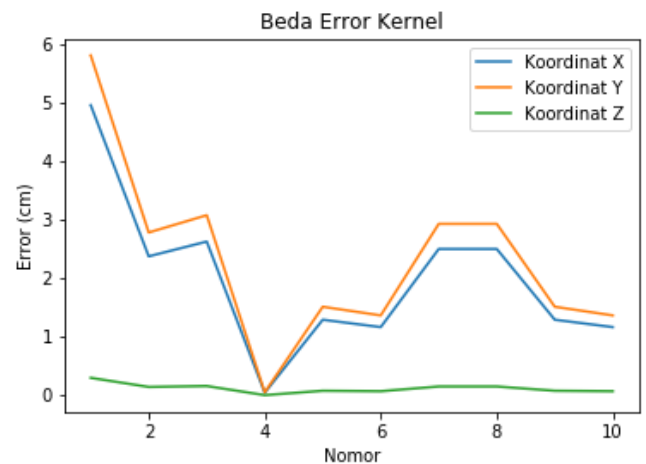
No	Asli	Prediksi	Error (cm)
1	0	5,81	5,81
2	0	-2,78	2,78
3	30	26,92	3,08
4	0	0,05	0,05
5	0	1,51	1,51
6	15	16,37	1,37
7	0	2,93	2,93
8	15	12,07	2,93
9	15	13,49	1,51
10	0	-1,37	1,37
MAE			2,33

Berdasarkan Tabel V didapatkan *error* berdasarkan perhitungan *MAE* untuk koordinat y sebesar 2,33 cm, di mana *error* terbesar yaitu 5,81 cm dan terkecil 0,05 cm.

TABEL VI
PERBANDINGAN KOORDINAT ASLI DAN PREDIKSI KOORDINAT Z

No	Asli	Prediksi	Error (cm)
1	0	-0,30	0,30
2	0	0,14	0,14
3	0	0,16	0,16
4	30	30,00	0,00
5	0	-0,08	0,08
6	0	-0,07	0,07
7	15	14,85	0,15
8	0	0,15	0,15
9	15	15,08	0,08
10	15	15,07	0,07
MAE			0,12

Berdasarkan Tabel VI didapatkan *error* berdasarkan perhitungan *MAE* untuk koordinat z sebesar 0,12 cm, di mana *error* paling besar yaitu 0,30 cm dan paling kecil 0,00 cm. Adapun grafik beda *error* dari kernel dapat dilihat pada Gambar 12.



Gambar 12. Beda *Error* Koordinat x, y, z

Gambar 12 merupakan grafik beda *error* kernel di mana koordinat x ditandai dengan warna biru, koordinat y dengan warna jingga, dan koordinat z dengan warna hijau. Dari grafik tersebut dapat diketahui bahwa tingkat *error* untuk koordinat x dan y hampir sama. Hal ini disebabkan karena terdapat perbedaan sudut pandang dari kedua kamera yang dibuat menyerupai cara kerja dari mata manusia terhadap koordinat x dan y. Sedangkan *error* dari koordinat z cenderung stabil karena tinggi dari kedua kamera *smartphone* yang digunakan relatif sama. Perhitungan *error* tersebut didasarkan pada perhitungan *MAE* dengan rumus:

$$MAE = \frac{\sum_{i=0}^n |x_i|}{n} \quad (1)$$

Adapun hasil perhitungan *MAE* tertera pada Tabel VII.

TABEL VII
AKURASI MAE

Koordinat x	Koordinat y	Koordinat z
1,99	2,33	0,12

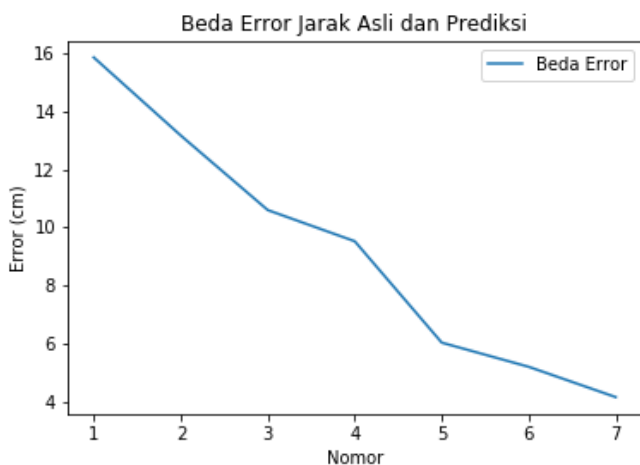
B. Pengujian Estimasi Posisi

Pengujian estimasi posisi dilakukan menggunakan data uji berdasarkan piksel 2D dari citra wajah yang berupa video. Data uji tersebut diperoleh berdasarkan koordinat tengah pada *bounding box* hasil pendeteksian wajah untuk kasus satu orang. Tabel VIII merupakan hasil pengujian estimasi posisi satu orang berdasarkan tangkapan 10 *frame*.

TABEL VIII
PERBANDINGAN JARAK ASLI DAN PREDIKSI

Jarak Asli (cm)	Jarak Prediksi (cm)	Beda Error (cm)
2	17,84	15,84
7	20,16	13,16
12	22,6	10,6
17	26,53	9,53
22	28,05	6,05
28	33,22	5,22
33	37,18	4,18
Rata-rata Error (cm)		9,23

Berdasarkan pengujian akurasi estimasi posisi yang dilakukan menghasilkan rata-rata *error* sebesar 9,23 cm. Dari hasil estimasi posisi tersebut maka dapat diketahui lokasi wajah yang terdeteksi. Pengukuran jarak tersebut dilakukan berdasarkan jarak wajah dengan alat peraga yang digunakan sebagai titik referensi. Pengukuran tersebut dilakukan dengan perhitungan *Euclidean* sederhana.



Gambar 13. Beda Error Jarak Asli dan Prediksi

Berdasarkan Gambar 13 diketahui bahwa semakin jauh jarak antara lokasi wajah dengan alat peraga, maka tingkat *error* akan semakin kecil yang ditandai dengan grafik menurun. Besar *error* dihitung menggunakan persamaan berikut,

$$R_e = \sum_{i=0}^n \frac{|r'_i - r_i|}{n} \quad (2)$$

$$r' = \sqrt{x_\mu^2 + y_\mu^2 + z_\mu^2} \quad (3)$$

Di mana R_e adalah jarak *error* rata-rata, n adalah jumlah sampel pengujian dengan jarak asli yang berbeda-beda antara wajah ke titik O pada alat peraga sebagai titik

referensi, r_i adalah jarak asli wajah ke titik O pada kasus ke- i , r'_i adalah jarak prediksi antara wajah ke titik O.

Karena disetiap kasus pengujian menggunakan video yang durasinya berbeda-beda, maka nilai r' pun merupakan rata-rata dari jarak prediksi yang didapat di setiap *frame*-nya. r' dihitung menggunakan jarak *Euclidean* sederhana di atas dengan masing-masing komponen x , y , z merupakan rata-rata dari nilai seluruh *frame*.

C. Pengujian Pengenalan Wajah

Metode pengenalan wajah yang digunakan yaitu *LBPH Face Recognizer*. Sedangkan metode yang digunakan untuk pendeteksian wajah yaitu *Haar Cascade Classifier*. Contoh akurasi pengenalan wajah dapat dilihat pada Tabel IX.

TABEL IX
CONTOH AKURASI PENGENALAN WAJAH

Nama Asli	Presentase Keberhasilan
Putra	100%
Akmal	90%
Akrom	100%
Akurasi	97%

Berdasarkan Tabel IX didapatkan akurasi pengenalan wajah sebesar 97%. Akurasi tersebut diperoleh dari hasil pengenalan wajah pada tiga video yang berbeda untuk tiap-tiap orang dalam keadaan diam dan diambil sebanyak 10 *frame* pada video. Pengenalan wajah semakin buruk apabila data latih yang digunakan memiliki banyak kesamaan antara satu dengan orang yang lain, kemudian pencahayaan yang buruk, serta sudut dan jarak yang tidak tepat dapat membuat wajah menjadi sulit untuk dikenali. Perhitungan akurasi pengenalan wajah dihitung menggunakan persamaan berikut,

$$Akurasi = \frac{\text{Jumlah Presentase Keberhasilan}}{\text{Total Percobaan}} \times 100\% \quad (4)$$

D. Pengujian Clustering

Pengujian *clustering* dilakukan dengan data beberapa video yang berisi satu orang, dua orang, dan tiga orang. Hasil pengujian *clustering* untuk satu orang dapat dilihat pada Tabel X, Tabel XI, dan Tabel XII.

TABEL X
PENGUJIAN CLUSTERING KASUS 1 ORANG VIDEO 1

No	1 Cluster	
	Kamera 1	Kamera 2
1	Putra	Putra
2	Putra	Putra
3	Putra	Putra
4	Putra	Putra
5	Putra	Putra
6	Putra	Putra
7	Putra	Putra
8	Putra	Putra
9	Putra	Putra
10	Putra	Putra
Modus	Putra	Putra
Hasil	Benar	

Dari Tabel X didapatkan nama yang paling sering keluar pada 10 *frame* untuk kamera 1 dan 2 yaitu Putra berdasarkan 10 kali program melakukan prediksi nama dengan benar. Tabel XI merupakan hasil pengujian kasus 1 orang video 2.

TABEL XI
PENGUJIAN CLUSTERING KASUS 1 ORANG VIDEO 2

No	1 Cluster	
	Kamera 1	Kamera 2
1	Akmal	Akmal
2	Akmal	Akmal
3	Akmal	Akmal
4	Akmal	Akmal
5	-	Akmal
6	-	Akmal
7	-	Akmal
8	-	Akmal
9	-	Akmal
10	-	Akmal
Modus	Akmal	Akmal
Hasil	Benar	

Dari Tabel XI didapatkan nama yang paling sering keluar untuk kamera 1 dan 2 yaitu Akmal. Pada pengujian yang dilakukan, kamera 1 hanya mengenali 4 *frame* dari total 10 *frame*. Sedangkan untuk kamera 2, program dapat mengenali Akmal berdasarkan total 10 *frame*. Tabel XII merupakan hasil pengujian kasus 1 orang video 3.

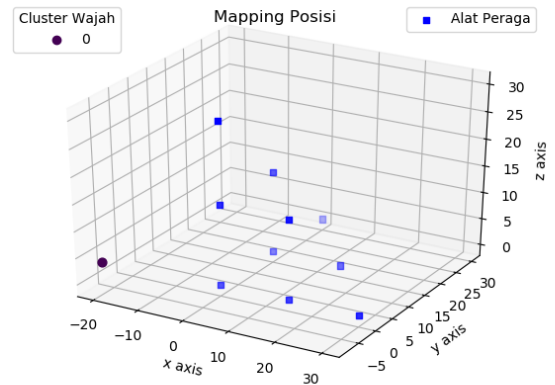
TABEL XII
PENGUJIAN CLUSTERING KASUS 1 ORANG VIDEO 3

No	1 Cluster	
	Kamera 1	Kamera 2
1	Akrom	Akrom
2	Akrom	Akrom
3	Akrom	Akrom
4	Akrom	Akrom
5	Akrom	Akrom
6	Akrom	Akrom
7	Akrom	Akrom
8	Akrom	Akrom
9	Akrom	Akrom
10	Akrom	Akrom
Modus	Akrom	Akrom
Hasil	Benar	

Dari Tabel XII didapatkan nama yang paling sering keluar untuk kamera 1 dan 2 yaitu Akrom berdasarkan 10 kali melakukan prediksi nama dengan benar pada 10 *frame*.

Pada pengujian kasus 1 orang untuk 3 video yang berbeda, didapatkan hasil pengenalan wajah yang benar sepenuhnya untuk tiap-tiap orang. Dengan hasil tersebut membuat pengujian *clustering* dapat dilakukan dengan baik dan tepat karena hanya membuat 1 *cluster*. Estimasi posisi diperoleh dari penggabungan hasil *clustering* pada kamera 1 dan 2 untuk wajah yang terdeteksi, sehingga mendapatkan data koordinat 2D pada gambar (x_1, y_1, x_2, y_2). Berikut

salah satu contoh gambar *plotting* pada 3D Scatter Plot untuk kasus 1 orang.



Gambar 14. 3D Scatter Plot Kasus 1 orang

Gambar 14 merupakan contoh *mapping* posisi berdasarkan estimasi yang telah dilakukan untuk *clustering* kasus satu orang. Isi dari *mapping* tersebut meliputi *cluster* wajah yang berjumlah satu yang ditandai dengan 'o' marker berwarna merah, sedangkan 's' marker berwarna biru menandakan 10 titik dari alat peraga yang digunakan. Lokasi orang tersebut didapatkan setelah dibuat *mean* dari pergerakan yang dilakukan sehingga menghasilkan satu titik koordinat 3D untuk orang tersebut. Pengujian *clustering* yang kedua dilakukan dengan video yang berisi 2 orang. Hasil pengujiannya dapat dilihat pada Tabel XIII, Tabel XIV, dan Tabel XV.

TABEL XIII
PENGUJIAN CLUSTERING KASUS 2 ORANG VIDEO 1

No	Kamera 1		Kamera 2	
	Cluster 0	Cluster 1	Cluster 0	Cluster 1
1	Akrom	Putra	Akrom	Putra
2	Putra	Putra	Putra	Putra
3	Akrom	Putra	Akrom	Putra
4	Akrom	Akmal	Putra	Putra
5	Akrom	Putra	Putra	Putra
6	Akrom	Akmal	Akrom	Putra
7	Akrom	Akmal	Akrom	Putra
8	Akrom	Putra	Putra	Putra
9	Akrom	Putra	Akrom	Putra
10	Akrom	Putra	Akrom	Putra
Modus	Akrom	Putra	Akrom	Putra
Hasil	Benar			

Tabel XIII merupakan hasil *clustering* dua orang pada video 1 di mana modus pada kamera 1 untuk *cluster* 0 adalah Akrom dengan 9 kali prediksi nama benar 1 salah dan *cluster* 1 adalah Putra dengan 7 kali prediksi nama benar 3 salah. Sementara itu, modus pada kamera 2 untuk *cluster* 0 adalah Akrom dengan 6 kali prediksi nama benar 4 salah dan *cluster* 1 adalah Putra dengan 10 kali prediksi

nama benar. Tabel XIV merupakan hasil *clustering* dua orang untuk video 2.

TABEL XIV
PENGUJIAN *CLUSTERING* KASUS 2 ORANG VIDEO 2

No	Kamera 1		Kamera 2	
	Cluster 0	Cluster 1	Cluster 0	Cluster 1
1	Putra	Putra	Akmal	Putra
2	Akmal	Putra	Akmal	Putra
3	Akmal	Putra	Akmal	Putra
4	Akmal	Putra	Akmal	Putra
5	Akmal	Putra	Akmal	Putra
6	Akmal	Putra	Akmal	Putra
7	Akmal	Putra	Akmal	Putra
8	Akmal	Putra	Akmal	Putra
9	Akmal	Putra	Akmal	Putra
10	Akmal	Putra	Putra	Putra
Modus	Akmal	Putra	Akmal	Putra
Hasil	Benar			

Berdasarkan Tabel XIV didapatkan modus pada kamera 1 dan kamera 2 untuk *cluster 0* adalah Akmal dengan 9 kali prediksi nama benar dari 10 *frame* dan *cluster 1* adalah Putra dengan 10 kali benar dalam melakukan prediksi nama. Tabel XV merupakan hasil *clustering* dua orang untuk video 3.

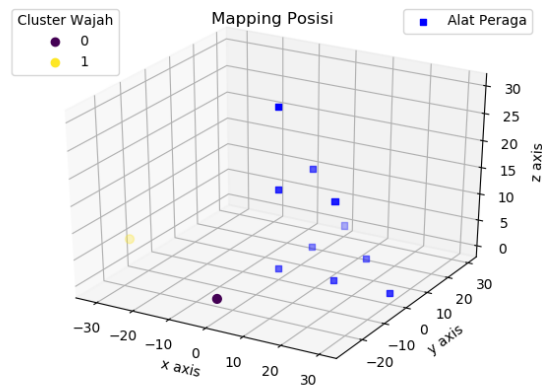
TABEL XV
PENGUJIAN *CLUSTERING* KASUS 2 ORANG VIDEO 3

No	Kamera 1		Kamera 2	
	Cluster 0	Cluster 1	Cluster 0	Cluster 1
1	Akrom	Akmal	Akrom	Akmal
2	Akrom	Akmal	Akrom	Akmal
3	Akrom	Akmal	Akrom	Akmal
4	Akrom	Akmal	Akrom	Akmal
5	Akrom	Akmal	Akrom	Akmal
6	Akrom	Akmal	Akrom	Akmal
7	Akrom	Akmal	Akrom	Akmal
8	Akrom	Akmal	Akrom	Akmal
9	Akrom	Akmal	Akrom	Akmal
10	Akrom	Akmal	Akrom	Akmal
Modus	Akrom	Akmal	Akrom	Akmal
Hasil	Benar			

Berdasarkan Tabel XV didapatkan modus pada kamera 1 dan 2 untuk *cluster 0* adalah Akrom dengan total 10 kali benar dalam melakukan prediksi nama, kemudian modus pada *cluster 1* adalah Akmal yang juga memiliki total 10 kali benar dalam memprediksi nama.

Berdasarkan hasil pengujian kasus 2 orang dengan 3 video yang berbeda didapatkan hasil *clustering* yang baik karena dapat membedakan dua wajah dengan warna yang berbeda untuk menunjukkan bahwa ada 2 *cluster* atau 2 wajah. Pada pengujian kasus 2 orang, program pengenalan wajah tidak sepenuhnya memprediksi nama dengan benar. Walaupun begitu, dengan fungsi modus ini dapat tetap memberikan hasil nama yang benar berdasarkan pada perhitungan nama yang paling banyak keluar dalam 10

frame. Berikut salah satu contoh *mapping* pada 3D Scatter Plot untuk kasus dua orang.



Gambar 15. 3D Scatter Plot Kasus 2 Orang

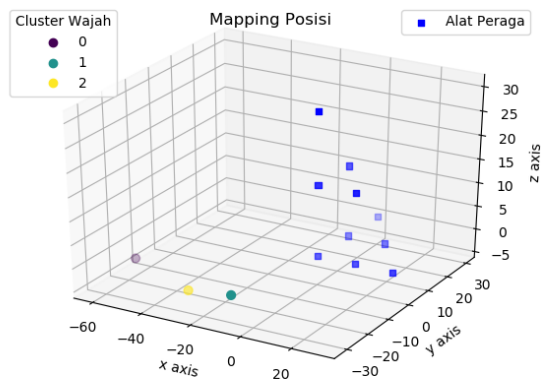
Gambar 15 merupakan contoh *mapping* posisi dalam bentuk koordinat 3D berdasarkan estimasi yang telah dilakukan untuk *clustering* kasus dua orang. Isi dari *mapping* tersebut meliputi dua *cluster* wajah yang ditandai dengan 'o' marker berwarna kuning dan merah, sedangkan 's' marker berwarna biru menandakan 10 titik dari alat peraga yang digunakan. Lokasi tersebut didapatkan setelah dibuat *mean* dari pergerakan yang dilakukan terhadap wajah sehingga menghasilkan satu titik koordinat 3D untuk tiap-tiap orang tersebut. Pengujian *clustering* yang ketiga dilakukan dengan video yang berisi tiga orang. Tabel XVI merupakan *clustering* untuk kasus 3 orang.

TABEL XVI
PENGUJIAN *CLUSTERING* KASUS 3 ORANG

No	Kamera 1			Kamera 2		
	Cluster			Cluster		
	0	1	2	0	1	2
1	Putra	Akrom	Akmal	Putra	Akrom	Akmal
2	Putra	Akrom	Akmal	Putra	Akrom	Akmal
3	Akrom	Akrom	Akmal	Putra	Akrom	Akmal
4	Putra	Akrom	-	Putra	Akrom	Akmal
5	Akrom	Akrom	-	Putra	Akrom	Akmal
6	Putra	Akrom	-	Putra	Akrom	Akmal
7	Akmal	Akrom	-	Putra	Akrom	Akmal
8	Putra	Akrom	-	Putra	Akrom	Akmal
9	Akrom	Akrom	-	Putra	Akrom	Akmal
10	Putra	Akrom	-	Putra	Akrom	Akmal
Modus	Putra	Akrom	Akmal	Putra	Akrom	Akmal
Hasil	Benar					

Dari Tabel XVI didapatkan modus pada kamera 1 untuk *cluster 0* adalah Putra dengan 6 kali prediksi nama benar dan 4 kali salah, kemudian untuk *cluster 1* adalah Akrom dengan 10 kali benar, lalu untuk *cluster 2* adalah Akmal dengan total pengenalan berjumlah 3 kali berdasarkan 10 *frame* dan benar semua. Sedangkan pada kamera 2 untuk

cluster 0 adalah Putra, cluster 1 adalah Akrom, dan cluster 2 adalah Akmal dengan 10 kali prediksi nama benar untuk tiap-tiap orang. Berikut salah satu contoh *mapping* pada 3D Scatter Plot untuk kasus tiga orang.



Gambar 16. 3D Scatter Plot Kasus 3 Orang

Gambar 16 merupakan contoh *mapping* posisi dalam bentuk koordinat 3D berdasarkan estimasi yang telah dilakukan untuk *clustering* kasus tiga orang. Isi dari *mapping* tersebut meliputi tiga *cluster* wajah yang ditandai dengan 'o' marker berwarna kuning, hijau dan merah, sedangkan 's' marker berwarna biru menandakan 10 titik dari alat peraga yang digunakan. Lokasi orang tersebut didapatkan setelah dibuat *mean* dari pergerakan wajah yang dilakukan sehingga menghasilkan satu titik koordinat 3D untuk tiap-tiap orang. Tabel XVII adalah rekapitulasi hasil pengujian *clustering*.

TABEL XVII
REKAPITULASI HASIL PENGUJIAN CLUSTERING

No	Kasus 1	Hasil	Kasus 2	Hasil	Kasus 3	Hasil
1	Putra	Benar	Akrom	Benar	Putra	Benar
2	Akmal	Benar	Putra	Benar	Akmal	Benar
3	Akrom	Benar	Akmal	Benar	Akrom	Benar
4	-	-	Putra	Benar	-	-
5	-	-	Akrom	Benar	-	-
6	-	-	Akmal	Benar	-	-
Hasil	3 Benar		6 Benar		3 Benar	
Akurasi	100 %					

Berdasarkan pengujian yang telah dilakukan, maka total akurasi *tracking* berbasis *clustering* sebesar 100% berdasarkan 12 kali benar dari total 12 percobaan.

E. Pengujian Sistem

Sistem ini menggunakan *user interface* berupa aplikasi *desktop* yang dibangun dengan Tkinter GUI karena sistem ini dibuat dengan bahasa pemrograman Python sehingga lebih mudah untuk diintegrasikan di mana proses-prosesnya dibuat sederhana agar mudah dipahami oleh *user* dan dapat

digunakan sesuai kebutuhan. Pengujian ini didasarkan pada pengujian fungsional dengan menguji beberapa parameter fungsional yang telah disusun seperti pada Tabel XVIII.

TABEL XVIII
PENGUJIAN FUNGSIONAL

Kebutuhan Fungsional	Percobaan									
	1	2	3	4	5	6	7	8	9	10
Sistem dapat menampilkan panduan	v	v	v	v	v	v	v	v	v	v
Sistem dapat mengisikan ID dan Nama untuk registrasi wajah baru	v	x	v	v	v	v	v	v	v	v
Sistem dapat membuat dataset	v	v	v	v	v	x	v	v	v	v
Sistem dapat melakukan training berdasarkan dataset	v	v	v	v	v	v	v	v	v	v
Sistem dapat melakukan presensi dan menampilkan hasil estimasi posisi serta clustering pada 3D scatter plot	v	x	v	v	v	v	v	v	v	v
Sistem dapat melakukan penyimpanan dan menampilkan data presensi	v	x	v	v	v	v	v	v	v	v

Berdasarkan pengujian yang telah dilakukan secara fungsional terhadap sistem, didapatkan tingkat akurasi sebesar 93% berdasarkan empat kali salah dari 60 kali percobaan untuk semua kebutuhan fungsional.

KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, kesimpulannya yaitu algoritma LASSO dapat digunakan untuk pembuatan kernel agar dapat memprediksi koordinat nyata berdasarkan koordinat gambar. Dari hasil kernel tersebut akan dilakukan estimasi posisi berdasarkan pengukuran jarak dari objek target dengan objek referensi. Berdasarkan pengujian estimasi posisi yang telah dilakukan, jarak antara objek target yang berupa wajah dan objek referensi yang berupa alat peraga berbentuk koordinat kartesian 3D dapat dihitung dengan perhitungan Euclidean. Berdasarkan hasil perhitungan, semakin jauh jarak antara wajah dan alat peraga maka tingkat *error* semakin kecil.

Selain itu, pengujian untuk *tracking* berbasis *clustering* agar dapat melacak posisi wajah dapat dilakukan dengan baik oleh algoritma *K-Means*. *Clustering* ini dilakukan berdasarkan hasil prediksi wajah yang didapatkan dari proses pengenalan wajah. Oleh karena itu, hasil *clustering* sangat dipengaruhi oleh hasil prediksi wajah. Apabila wajah dapat dideteksi dan dikenali dengan baik, maka proses *tracking* berbasis *clustering* dapat dilakukan dengan baik juga.

Penelitian ini berfokus pada model untuk estimasi posisi dan *tracking* berbasis *clustering* menggunakan *dataset* dari tiga orang. Model ini dapat digunakan sebagai penunjang pada sistem presensi untuk mendapatkan lokasi dari orang dalam bentuk koordinat 3D. Hal ini dimaksudkan agar sistem dapat mengenali wajah seseorang walaupun tidak secara langsung melakukan proses presensi dengan cara berdiri di depan sensor kamera. Berdasarkan hasil pengujian yang dilakukan dapat diasumsikan untuk empat atau lebih orang bisa dilakukan dengan baik.

V. SARAN

Adapun beberapa saran pada penelitian ini untuk dikembangkan lagi di penelitian selanjutnya adalah mengenai pencahayaan ketika melakukan pengenalan wajah perlu diperhatikan dan diadakan penelitian lebih lanjut mengenai batasan pencahayaan yang baik, kemudian untuk penelitian selanjutnya diharapkan menerapkan metode alternatif lain sebagai pembanding, dan sebaiknya menggunakan kamera eksternal khusus agar menghasilkan gambar yang lebih baik serta *hardware* yang mumpuni untuk implementasi secara *realtime* serta menggunakan *dataset* yang lebih banyak agar hasil pengenalan dapat prima.

DAFTAR PUSTAKA

- [1] E. B. Setiawan and B. Kurniawan, "Perancangan Sistem Absensi Kehadiran Perkuliahan dengan Menggunakan Radio Frequency Identification (RFID)," *CoreIT*, vol. 1, no. 2, pp. 44–49, 2015.
- [2] R. Rian, C. Putra, and F. P. Juniawan, "Penerapan Algoritma Fisherfaces Untuk Pengenalan Wajah Pada Sistem Kehadiran Mahasiswa Berbasis Android," *J. Telemat.*, vol. 10, no. 1, pp. 132–146, 2017.
- [3] C. Suhery and I. Ruslianto, "Identifikasi Wajah Manusia untuk Sistem Monitoring Kehadiran Perkuliahan menggunakan Ekstraksi Fitur Principal Component Analysis (PCA)," *J. Edukasi dan Penelit. Inform.*, vol. 3, no. 1, pp. 9–15, 2017.
- [4] D. Rusjdi, C. J. Kusumo, and A. Yudha, "Perancangan Simulasi Penentuan Koordinat Obyek Di Dalam Ruang Menggunakan Kamera Digital," *PETIR*, vol. 5, no. 2, pp. 76–185, 2012.
- [5] H. Afrisal, I. Soesanti, and A. I. Cahyadi, "Estimasi Posisi Dengan Menggunakan Kamera Monokular," *TRANSMISI*, vol. 21, no. 1, pp. 1–9, 2019.
- [6] M. A. Firmansyah and I. Supriana, "Multiple Object Tracking dan Estimasi Posisi untuk Menunjang Sistem Keamanan Rumah," in *Konferensi Nasional Informatika (KNIF)*, 2015, pp. 1–6.
- [7] I. Maliki, "3 Dimensi," in *Grafika Komputer*, Bandung: Universitas Komputer Indonesia, 2010, pp. 44–48.
- [8] R. Munir, "Pembentukan Citra," in *Pengolahan Citra Digital*, Bandung: Informatika, 2004, pp. 15–28.
- [9] S. Monika, A. Rakhman, and Lindawati, "Pengaman Rumah dengan Sistem Face Recognition Secara Real Time Menggunakan Metode Principal Component Analysis," *ISBN 978-602-1180-50-1*, pp. 395–401, 2017.
- [10] A. Ahmad. (2017) Mengenal Artificial Intelligence , Machine Learning, Neural Network, dan Deep Learning. [Online]. Tersedia: https://www.researchgate.net/publication/320395378_Mengenal_Artificial_Intelligence_Machine_Learning_Neural_Network_dan_Deep_Learning
- [11] F. K. H. Prabowo, Y. Wilandari, and A. Rusgiyono, "Pemodelan Pertumbuhan Ekonomi Jawa Tengah Menggunakan Pendekatan Least Absolute Shrinkage and Selection Operator (LASSO)," *GAUSSIAN*, vol. 4, no. 4, pp. 855–864, 2015.
- [12] D. P. Prabowo, K. Latifah, and R. A. Premunendar, "Pelacakan dan Segmentasi Objek Bergerak Menggunakan Metode K-Means Clustering Berbasis Variasi Jarak," *Inform. UPGRIS*, vol. 5, no. 1, pp. 111–115, 2019.