

Sistem Drone Cerdas Yang Dilengkapi Face Detection dan Face Recognition Untuk Pembuatan Sinematik Video

Yunan Kholilul Fatah^a, Yosi Kristian^b, Devi Dwi Purwanto^{c*}

^aDepartemen Teknologi Informasi, Institut Sains dan Teknologi Terpadu Surabaya

^bDepartemen Teknologi Informasi, Institut Sains dan Teknologi Terpadu Surabaya

^cDepartemen Sistem Informasi, Institut Sains dan Teknologi Terpadu Surabaya
E-mail: yunanelfath@gmail.com, yosi@stts.edu, devi@stts.edu

Abstrak— Pembuatan konten video melalui drone pada umumnya membutuhkan seorang profesional untuk mengendalikan drone tersebut agar mampu menghasilkan pergerakan drone dengan hasil video yang diinginkan. Video sinematik adalah video yang mempunyai alur cerita atau dapat menyampaikan sebuah cerita pada film ataupun video pendek. Pengarahan alur cerita tersebut membutuhkan juga seorang film director yang mengarahkan adegan yang akan ditampilkan didalam video. Penelitian ini mengusulkan untuk membuat autonomous drone agar mampu bergerak dan menangkap video sesuai dengan arahan film director. Penelitian ini menggunakan face detection dan face recognition dengan algoritma *Local Binary Pattern Histogram* (LBPH) dan memanfaatkan Rule base system sebagai system cerdas yang terdapat pada system agar drone mampu mengikuti wajah yang dikenali sesuai pergerakan subject yang telah di rencanakan oleh film director. Setiap pergerakan drone memiliki catatan terbang yang terdapat pada system drone berupa *Inertial Measurement Unit* (IMU) sehingga system mampu memberikan grafik 3 dimensi setelah drone sudah tidak berada di udara. Skenario yang diusulkan dalam penelitian ini membuktikan bahwa drone mampu bergerak sesuai ekspektasi penulis dan film director. Selain itu survey berupa kuesioner untuk responden umum juga membuktikan bahwa drone sudah mengikuti salah satu konsep cinematography seperti jarak sudut pandang, object yang menarik (*shape saliency*) dan area pergerakan kamera.

Kata Kunci— autonomous drone, video sinematik, algoritma LBPH, face detection, face recognition

I. PENDAHULUAN

Unarmed Aerial Vehicle (UAV) atau drone merupakan alat yang bisa dikendalikan oleh manusia dalam beberapa hal seperti *live broadcast*, *agriculture*, *delivery*, *sport* atau hobi [2]. Pada beberapa jenis drone terdapat kamera yang berfungsi sebagai visualisasi untuk mengambil

gambar atau merekam video dengan sudut pandang orang pertama atau *first person video* yang biasanya terletak di posisi depan *drone* itu sendiri. Dengan visual tersebut orang yang mengendalikan *drone* seolah-olah sedang berada di dalam *drone*.

Salah satu fungsi *drone* yang digunakan dalam penelitian ini adalah pembuatan konten video sinematik. Menurut Blain Brown dalam bukunya Teknik sinematik merupakan rangkaian metode dan keahlian yang digunakan untuk menambahkan arti dan cerita pada sebuah konten atau film seperti actor, latar belakang, dialog dan adegan [3]. Sehingga video sinematik merupakan hasil karya dari metode dan keahlian tersebut untuk membuat sebuah video dengan arti dan cerita yang terkandung secara tersirat dalam video tersebut.

Penelitian terdahulu yang dilakukan Chong Huang dkk melakukan penelitian yang disebut *an Autonomous Cinematography System* "ACT" dengan membuat *autonomous drone* untuk menangkap video sinematik pada sebuah adegan aksi berdasarkan ekstraksi titik 3D anggota tubuh yang melalui kamera *stereo drone* dan melakukan perencanaan kamera secara dinamis dan *realtime* yang sanggup memenuhi nilai aesthetics film secara objektif dan memahami keterbatasan yang ada pada *drone* [4].

Penelitian ini memanfaatkan kamera *drone* yang dilengkapi *Face detection* dan *Face Recognition* sehingga mampu mengenali subject yang berada di depan *drone*. Selain itu *drone* juga mampu bergerak secara otomatis mengikuti wajah yang dikenal agar tetap berada pada posisi tengah kamera. Algoritma yang digunakan menggunakan *Local Binary Pattern Histogram* (LBPH) dengan memanfaatkan *library python* openCV dan haar cascade classifier. Perencanaan adegan dan skenario pada penelitian ini melibatkan *film director* [1] yang mengarahkan *subject* untuk berada pada lokasi yang telah ditentukan sehingga *drone* akan mengikuti pergerakan wajah *subject* tersebut dan menghasilkan video sinematik yang sesuai dengan skenario yang telah diberikan *film director*.

Pergerakan *drone* secara autonomous mengikuti wajah subject yang memanfaatkan *Fuzzy Rule Based* [5] dengan menentukan formula berdasarkan koordinat x dan y yang

Naskah Masuk : 25 Januari 2022

Naskah Direvisi : 03 April 2022

Naskah Diterima : 04 April 2022

*Corresponding Author : devi@stts.edu



This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.

ditampilkan pada layar pada setiap wajah yang dikenali. Kontribusi penelitian ini adalah dengan cara membuat perencanaan kamera dengan bantuan *film director* serta *face detection* dan *face recognition* agar *drone* mampu mengenali *subject* dan mengikuti wajah pergerakan *subject* dengan memanfaatkan *rule based system* untuk menghitung koordinat wajah *subject* yang berada di depan layar kamera.

II. PENELITIAN TERDAHULU

Pada tahun 2018 Chong huang dkk melakukan penelitian mengenai *autonomous cinematography system* yang mengemukakan bahwa kemunculan *drone* telah meningkatkan kebutuhan para pembuat film untuk meningkatkan kualitas sinematik video dan cerita yang dapat disampaikan oleh para pembuat film [3]. Proses syuting melalui udara dengan *drone* pada adegan aksi sulit dilakukan karena membutuhkan orang yang mampu memahami skenario secara dinamis dan mengendalikan *drone* secara bersamaan. Chong huang dkk melakukan dua usulan yang menurut mereka mampu mengatasi latar belakang tersebut, menurut mereka hal ini merupakan system *drone* pertama yang mampu menangkap sinematik video dari film aksi baik di dalam ruangan maupun di luar ruangan. Pertama, mereka mengusulkan untuk melakukan ekstraksi titik 3D anggota tubuh yang melalui kamera stereo *drone* secara efisien. Kedua, mereka melakukan perancangan kamera *planning* secara dinamis dan *realtime* yang sanggup memenuhi nilai *aesthetics* film secara objektif dan memahami keterbatasan dari perangkat *drone*. Hasil uji coba yang mereka dapatkan dari penelitian ini secara virtual simulasi ataupun fakta dilapangan menyimpulkan bahwa *system* mereka mampu menangkap pergerakan seseorang dalam film aksi dengan ekspresif.

A. 3D Skeleton Detection

Penelitian terdahulu yang dicantumkan Chong huang dkk mengenai perkembangan 3D *skeleton detection* yang sudah ada, metode yang dipakai bergantung pada kedalaman sensor infra merah. *Kinect sensor* merupakan sensor yang digunakan untuk mendeteksi kedalaman kerangka. *Kinect* sensor mampu mendeteksi *subject* tanpa mengharuskan pengguna untuk memakai sensor tambahan. Ukurannya yang kecil menjadikannya kelebihan karena dapat dipasangkan pada robot untuk melihat lingkungan yang tidak dikenal. Tapi bagaimanapun juga *kinect sensor* tidak bisa bekerja di luar ruangan karena menggunakan *projector* laser infra merah.

Chong huang dkk mengenalkan deteksi 3D *skeleton* dengan menggunakan stereo kamera. Untuk menghitung ketajaman gambar, mereka juga menambahkan constraint untuk menyaring hasil yang di hasilkan jika terdapat ketajaman yang tidak sesuai pada area dengan gerakan yang kabur. Penelitian ini menggunakan *library opensource* OpenPose untuk mendeteksi ketajaman *skeleton* secara 2D, jika semua anggota tubuh terdeteksi maka OpenPose mampu mendeteksi 13 titik termasuk kepala, hidung, panggul, bahu kiri dan kanan, siku, tangan dan kaki seperti terlihat pada gambar 1.

Berdasarkan ketajaman yang berhasil di tangkap dari kamera sebelah kiri, mereka bisa *mengconvert* c_y pada masing-masing titik anggota tubuh dari 2D (x,y) menjadi titik anggota tubuh secara 3D (x,y,z) sebagai suatu persamaan

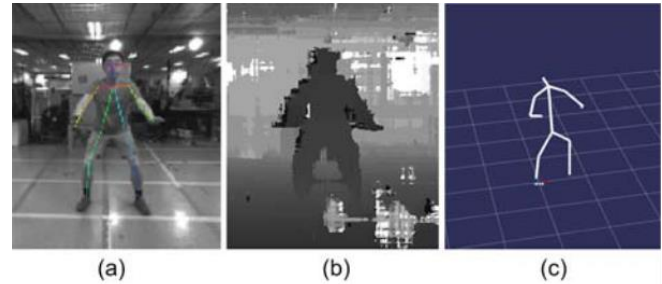
sebagai berikut:

$$Z = \text{depth}(x, y)$$

$$X = (x - c_x) \cdot Z / f_x,$$

$$Y = (y - c_y) \cdot Z / f_y,$$

Dimana c_x , c_y merupakan titik tengah gambar dan f_x , f_y merupakan focal length kamera dari kedua sumbu x dan y.



Gambar. 1.

(a) 2D skeleton, (b) depth map, (c) 3D Skeleton

B. Camera Planning Based on Next Best View

Pada usulan kedua ini Chong huang dkk menggunakan perencanaan tangkapan kamera yang bertujuan untuk mendesign *trajectory* yang memenuhi *aesthetic* dari object dan memahami kekurangan yang ada pada *drone*. Langkah pertama yang mereka lakukan adalah memprediksi gaya dari *subject* untuk *frame* berikutnya dengan menggunakan predictor *raw depth acquisition*. Kemudian memperhitungkan sudut pandang terbaik dari aksi *subject*, lalu generate secara fisik *trajectory* yang mengarah pada sudut pandang tersebut. Mereka mengestimasi sudut pandang terbaik dengan memperhatikan area pandangan yang merupakan titik tengah *subject* pada masing-masing gaya seseorang. *Radius* atau jarak antara kamera terhadap *subject* merepresentasikan sebagai ukuran *subject* pada gambar. *Radius* dapat di estimasi dengan formula sebagai berikut:

$$r = r_0(1 + k_v)$$

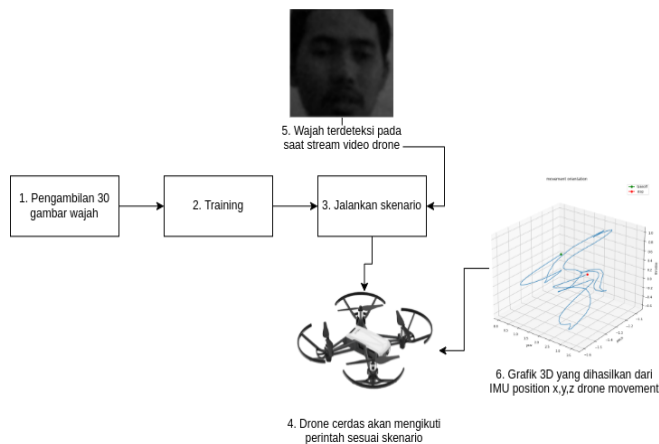
Dimana r_0 adalah jarak minimum kamera dan k adalah konstanta parameter untuk mengetahui jarak kamera. v adalah kecepatan saat ini pada *subject* yang menentukan rata-rata kecepatan pada leher dan panggul. Untuk menjamin penglihatan dari keseluruhan tubuh *subject*, mereka memberikan nilai jarak minimal pada r_0 dengan nilai 3 meter. Lalu memberikan nilai k dengan 0.4 untuk menjaga kehalusan dan kestabilan pergerakan kamera.

Hasil penelitian yang mereka lakukan dengan mengevaluasi system pada CMU dataset pergerakan yang ditangkap dan adegan aksi secara *realtime*. Mereka membandingkan *system* mereka dengan *state-of-the-art* teknik pembuatan film secara otomatis yang disebut "Active Track". "Active Track" merupakan *mode* terbang cerdas pada DJI Mavic Pro dimana kamera bisa menjaga jarak dengan target secara otomatis dan menyesuaikan kamera agar berada pada titik tengah *subject* pada layar kamera. percobaan yang mereka lakukan adalah dengan membandingkan algoritma deteksi anggota tubuh

menggunakan *Kinect sensor* pada ruangan tertutup. Mereka melakukan uji coba ini dengan gerakan lambat pada adegan TaiChi dan gerakan cepata pada adegan tarian “*Gangnam Style*”. Mereka mengklaim bahwa system *drone* mereka mampu menangkap rekaman dari sudut pandang yang bagus dengan frekuensi yang lebih tinggi. Dibandingkan dengan *Active Track*, system yang mereka ajukan menunjukkan bahwa jalur kamera dapat terpenuhi dengan lebih banyak sudut pandang dan lebih banyak rekaman kreatif yang ditangkap.

III. PENELITIAN YANG DIAJUKAN

Metode yang di lakukan dalam penelitian ini terdiri dari dua tahapan, yaitu perencanaan pembuatan skenario yang diberikan oleh seorang film director serta *drone* yang mampu mendeteksi wajah dan mengenali wajah dan mengikuti wajah *subject* berdasarkan pergerakan yang telah diberikan sesuai skenario sebelumnya. *Input* dari video diperoleh dari *stream drone* yang menangkap satu gambar wajah sebanyak 30 gambar yang dikonversi menjadi *grayscale* sekaligus melabeli gambar tersebut. Kemudian mengimplementasikan *LBP Operation* dengan membuat gambar perantara yang menjelaskan gambar asli dengan lebih baik dan menyoroti karakteristik wajah [7]. Arsitektur diagram penelitian ini dapat dilihat pada gambar 2.



Gambar. 2. Arsitektur Diagram

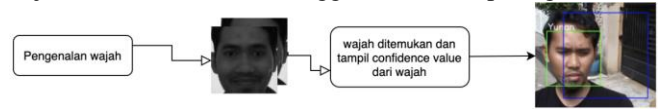
Setelah pengambilan gambar wajah selesai dilakukan pada setiap *input* wajah orang, langkah berikutnya adalah melakukan *training* data wajah tersebut dengan menggunakan *classifier* yang sama yaitu *haarcascade frontalface default* namun pada proses ini dilakukan juga *training* proses dengan algoritma *LBPH Recognizer* untuk mengenali wajah tersebut. Pada proses *training output* yang dihasilkan berupa *.yml file yang akan menyimpan data berupa data *matrix histogram*. Dari file *.yml tersebut langkah terakhir pada system ini *drone* akan terbang dan menjalankan scenario dengan mendeteksi wajah yang berada di depannya sehingga *output* yang dihasilkan berupa grafik 3 dimensi pada saat *drone* terbang.

Grafik 3 dimensi yang berupa *path* atau jalur terbang *drone* berikutnya dievaluasi oleh seorang *film director* agar mendapatkan hasil video *sinematography* yang sesuai skenario yang telah diberikan. Selain itu hasil video tersebut di upload ke *platform public* agar orang awam lain dapat memberikan responden yang telah diberikan pada saat penelitian ini dilakukan.

A. Face detection dan face recognition

Pada penelitian ini *library opencv* dan *classifier haarcascade frontal face default* digunakan dengan memakai parameter *minNeighbors* sebagai *sliding window* sebesar 5 dan *scale* sebagai *radius* sebesar 1,3. Langkah pada *Face Recognition* akan dilanjutkan dengan melakukan ekstraksi *histogram* dengan menggunakan gambar pada tahap terakhir seperti terlihat pada gambar 2. Sebagai parameter untuk membagi gambar menjadi *multiple grids* bisa dilakukan dengan menggunakan Grid X dan Grid Y. Kemudian untuk menemukan gambar yang cocok dengan *input* gambar adalah dengan membandingkan dua *histograms* dan mengembalikannya dengan gambar yang terdekat dengan *histogram*, beberapa pendekatan bisa dilakukan untuk mendapatkan nilai tersebut dalam penelitian ini akan menggunakan *Euclidean Distance* Sehingga hasil dari *Face Recognition* akan menampilkan *realtime* gambar wajah yang sudah dilabeli.

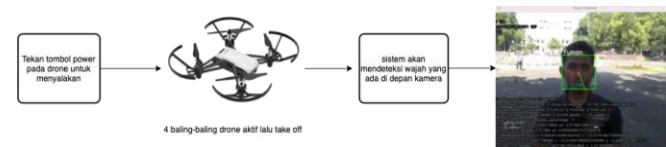
Tahap akhir dalam penelitian ini adalah *Face Detection Tracking* menggunakan Algoritma *haar cascade classifier* dan memanfaatkan *library* OpenCV agar *drone* mampu mengikuti wajah yang dikenali sehingga menghasilkan video *sinematografi* cerdas yang menarik dan mudah dioperasikan. *Haar cascade* merupakan pendekatan berdasarkan *machine learning* yang membutuhkan beberapa gambar *positive* dan *negative* untuk melakukan *train* pada *classifier* [6], *Fitur Haar* ini akan melakukan *scanning* agar bisa mendeteksi wajah mulai dari kiri atas hingga titik bawah pada gambar 3.



Gambar. 3. Wajah ditemukan

B. Drone controlling & Movement

Pada penelitian ini fungsi untuk mengontrol dan mengirim perintah ke *drone* dilakukan dengan program *python* dan memanfaatkan fitur yang tersedia di *official documentation* DJI Tello Ryze berupa SDK v1.0 yang bisa terhubung melalui jaringan *wireless* pada drone DJI Tello. Dalam dokumentasi tersebut sudah terdapat fungsi utama untuk melakukan kontrol seperti *take off*, *landing*, *go left*, *go right* dan seterusnya. Berdasarkan SDK tersebut penelitian ini secara spesifik menggunakan *library python* hanyazou/TelloPy yang telah mempermudah untuk mengirim perintah dalam bahasa pemrograman *python*.



Gambar. 4 pergerakan drone

Pada gambar 4 menjelaskan bagaimana proses awal *drone* setelah memiliki data wajah yang dikenali sehingga *system* mampu memberikan label pada wajah yang didepan kamera. Untuk wajah yang sudah dikenali akan diberikan label sesuai data yang diinput waktu proses *input* dan pengambilan wajah, sedangkan untuk wajah yang belum dikenal akan tampil berupa kotak hijau dan juga label “*anonymous*”. Setelah *system* mendeteksi wajah yang dikenali, *system* akan

menghitung jarak antara titik tengah dari wajah dan titik tengah dari kamera *drone* lalu akan menampilkan arah panah dari titik tengah kamera ke titik tengah pada wajah. Hal ini bertujuan untuk melihat seberapa besar nilai yang akan dikirim *system* untuk bergerak mendekati titik tengah yang diharapkan sesuai skenario.

Pergerakan *maneuver* pada *drone* memanfaatkan tiga tipe *maneuver* untuk mengirimkan perintah dari program yaitu *yaw*, *pitch* dan *throttle*. Dari tipe tersebut penelitian menentukan *variable* yang berfungsi untuk menghitung kecepatan dalam melakukan *maneuvering* seperti pergerakan maju-mundur (*pitch*), rotate kiri-kanan (*yaw*) dan up-down (*throttle*). Penelitian ini menggunakan *library* OpenCV dan TelloPy untuk menampilkan *stream* video dari kamera *drone*, dari *stream* video tersebut dapat diketahui bahwa program hanya menerima sebuah *frame* dengan tampilan 2 dimensi atau koordinat berupa sumbu x untuk bergerak ke kiri-kanan dan sumbu y untuk pergerakan naik-turun. Adapun *drone* membutuhkan 3 koordinat untuk melakukan pergerakan maju-mundur sehingga pergerakan ini menggunakan *variable* lebar box wajah yang terdeteksi pada *system*. Pergerakan maju-mundur disebut yang disebut sebagai z merupakan lebar box wajah yang dideteksi secara *realtime*. Dengan demikian dapat diketahui *variable* sebagai berikut:

- Titik x : pergerakan kiri-kanan
- Titik y: pergerakan naik-turun
- Titik z: pergerakan maju-mundur

Titik tengah wajah akan dibandingkan dengan titik tengah dari *frame* kamera *drone*, sehingga dapat ditentukan nilai kecepatan dan *threshold* untuk melakukan pergerakan *drone*. Setelah menentukan *variable* yang diketahui penelitian ini dilanjutkan dengan menentukan formula untuk melakukan pergerakan-pergerakan agar bisa mengikuti pergerakan sesuai skenario yang diinginkan. Formula tersebut kemudian diterangkan pada *library python* hanyazou/TelloPy yang bertugas mengirim perintah dan melakukan *maneuvering* ke DJI Tello.

Library hanyazou/TelloPy merupakan merupakan *library python open source* yang dapat diakses secara *public*. Penelitian ini secara garis besar hanya menggunakan 3 fungsi didalam *library* ini yaitu *set_yaw*, *set_pitch*, *set_throttle*. Dimana *set_yaw* merupakan perintah yang dikirimkan *system* untuk melakukan rotasi *maneuver* kearah kiri dan kanan dengan nilai jika positif maka *drone* akan mengarah ke kanan, jika *negative* maka *drone* akan mengarah ke kiri. Kemudian *set_pitch* merupakan perintah yang dikirimkan *system* untuk melakukan *maneuver* ke arah maju dan mundur dengan nilai *variable* jika positif akan mengarah ke depan dan jika *negative* akan mengarah ke kiri. Lalu yang terakhir adalah *set_throttle* merupakan perintah yang dikirimkan *system* untuk melakukan *maneuver* ke atas dan bawah dengan nilai yang dikirimkan sebagai *variable* jika positif maka akan mengarah ke arah atas dan jika ke *negative* akan mengarah ke bawah. Dari ketiga fungsi tersebut nilai yang dikirimkan merupakan nilai satuan kecepatan (jarak dibagi waktu). Selain tiga perintah dasar tersebut penulis juga membutuhkan dua perintah utama yaitu untuk melakukan *take off* dan *landing*. Berikut ini adalah contoh *logic* sederhana untuk melakukan perintah dasar tersebut

```

1: try:
2:     drone.connect()
3:     drone.takeoff()
4:     sleep(3)
5:     drone.set_yaw(1)
6:     sleep(3)
7:     drone.set_throttle(1)
8:     sleep(3)
9:     drone.set_pitch(1)
10:    sleep(3)
11:    drone.land()
12: finally:
13:    drone.land()

```

Logika sederhana dari sekumpulan *flow* segmen program 1 dijelaskan bahwa hal pertama yang dilakukan *system* adalah melakukan koneksi lebih dulu terhadap *drone*, lalu setelah koneksi terhubung *drone* akan terbang dengan perintah *takeoff()*, baris selanjutnya *system* akan memberikan jeda yang dikirimkan ke *drone* selama 3 detik untuk terbang di tempat hal ini akan dilakukan pengulangan pada setiap perubahan perintah yang akan dikirimkan kepada *drone*. Setelah *takeoff()* *drone* akan melakukan rotasi ke kanan dengan kecepatan 1cm/s, lalu dengan kecepatan yang sama *drone* selanjutnya akan bergerak ke atas dengan fungsi *set_throttle*, kemudian *drone* akan bergerak ke depan dengan fungsi *set_pitch* dan yang terakhir *drone* akan mendarat dengan menggunakan fungsi *land*.

Untuk mendapatkan formula yang sesuai, pada penelitian mengambil ide dari formula dari sebuah *source code public* dari *github* mengenai pendeteksi dan pengendalian *tello drone* berdasarkan deteksi wajah[8]. Dari *source code* tersebut disebutkan bahwa *left-right movement* dan *updown movement* jika lebih dari *maximum speed threshold* maka nilai kecepatan yang dikirimkan yaitu *negative*, jika kurang dari maka positif. Dari *constraint* tersebut penulis masih menemukan ketidakstabilan pada *drone* saat melakukan *maneuvering* berdasarkan wajah yang terdeteksi. Selain karena *constraint* tersebut masih tidak stabil, *source code* tersebut juga menggunakan *library* yang berbeda dengan penulis. sehingga penulis melakukan modifikasi pada *constraint* tersebut dengan beberapa *input* yang relevan dengan *library* yang dipakai penulis. Formula untuk menghitung kecepatan pergerakan *drone* secara *realtime* pada program adalah sebagai berikut:

$$\text{Step_power} = \text{max_speed} / \text{step_length}$$

$$\text{Movement_step} = (\text{object} / \text{distance_treshold}) - \text{step_length}$$

$$\text{Movement_speed} = \text{movement_step} * \text{step_power}$$

Dimana: *max_speed* = konstanta *variable* dari kecepatan maksimal yang dapat ditempuh *drone*.

Step_length = jarak dari langkah *drone* untuk bergerak dari satu titik ke titik lainnya.

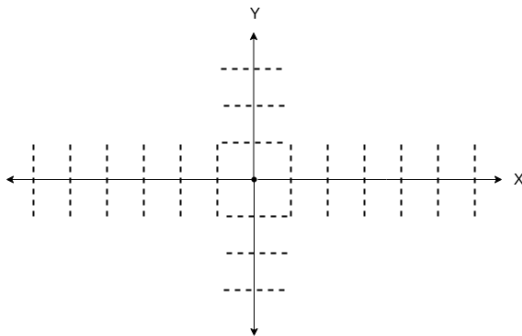
Distance_treshold = jarak *min-max* titik koordinat *object*

Movement_speed = nilai kecepatan yang akan dikirim ke *drone*

Pada saat *stream* video dari *drone*, *library* openCv akan menampilkan video tampilan 2 dimensi yang memiliki koordinat axis X dan Y. Berdasarkan koordinat tersebut diketahui *center point* yang akan digunakan *system* untuk mengirim perintah agar *drone* diam karena sudah berada di

center point pada sudut pandang kamera *drone*. Pada sumbu X diketahui bahwa titik awal sampai akhir dari tampilan 2 dimensi ini merupakan resolusi lebar *window* yang ditampilkan dari openCV sedangkan pada sumbu Y diketahui bahwa titik awal sampai akhir dari tampilan 2 dimensi merupakan resolusi tinggi *window* yang ditampilkan pada openCV. Adapun resolusi lebar dan tinggi yang digunakan pada penelitian ini adalah 960*720.

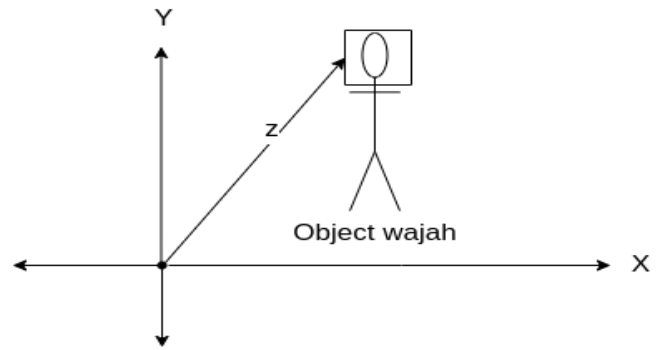
Dari pergerakan yang dilakukan *drone* diketahui bahwa setiap pergerakan memiliki jarak dari titik awal sampai titik dimana *drone* itu berhenti sebelum batas lebar atau tinggi dari *window*. Sehingga dari pergerakan tersebut dapat disimpulkan setiap pergerakan dapat dihitung berapa langkah yang dapat di tempuh *drone* dari satu titik ke titik lainnya. Dengan demikian titik-titik tersebut dapat dijadikan sebuah *variable* yang berguna untuk menghitung berapa total langkah sebuah *drone* dari titik awal sampai titik akhir. Setelah mendapatkan total tersebut akhirnya titik tersebut juga memiliki jarak antara satu titik dengan titik lainnya sehingga dapat dirumuskan sebagai *step_length* dalam penelitian ini. Seperti terlihat pada gambar 3.5 terdapat garis titik-titik yang merupakan representasi dari langkah yang ditempuh *drone* dari satu titik ke titik lainnya baik itu untuk pergerakan rotasi *drone* ke kanan-kiri (x) ataupun pergerakan ke atas-bawah (y).



Gambar 5. Koordinat 2 dimensi dari OpenCV x & y

Setelah mendapatkan berapa jarak yang dapat ditempuh pada setiap langkah pergerakan *drone* seperti terlihat pada gambar 5 dengan tanda garis titik-titik, selanjutnya dibutuhkan juga konstanta *variable* yang digunakan *drone* untuk bergerak secara maksimal. Dalam hal ini angka maksimal yang digunakan merupakan maksimal kecepatan yang dapat ditempuh *drone* dalam melakukan *maneuvering* atau pergerakan

Setelah mengetahui nilai pergerakan atas-bawah dan kanan-kiri *drone* mampu menghasilkan 2 dimensi berdasarkan layar *window* yang didapatkan dari *library* openCV. Sehingga untuk menghasilkan pergerakan secara 3 dimensi diperlukan satu *variable* lagi untuk menentukan nilai maju dan mundur dari sebuah *object*. Maka penelitian ini menggunakan nilai lebar box wajah yang dikenali dari sebuah *object* dengan *face recognition* untuk melakukan pergerakan maju dan mundur yang disebut sebagai koordinat z.



Gambar 6. Koordinat z Untuk 3 Dimensi

Pada gambar 6 terlihat bahwa koordinat z di sesuaikan dengan berapa besar *bounding box* yang mendeteksi wajah *subject*. Adapun koordinat z memiliki nilai tengah sebesar 250 *pixel* dari lebar box wajah yang terdeteksi pada layar *window* openCV. Koordinat awal z merupakan nilai awal sebesar 75 yang merupakan jarak terjauh antara *object* dengan kamera *drone* ketika *object* wajah tersebut dikenali dan titik akhir dari lebar box wajah memiliki nilai sebesar 425 *pixel* yang merupakan jarak terdekat antara *object* dengan kamera *drone* ketika *object* wajah dikenali. Setelah diketahui nilai lebar minimal dan maksimal dari *object* maka dapat diketahui juga nilai titik tengah dari garis z dimana pada posisi ini nilai yang dikirimkan ke *drone* adalah 0. Nilai 0 ini merupakan perintah *drone* untuk berhenti melakukan pergerakan maju-mundur sehingga *drone* akan berada di depan *object* dan berhenti di titik tengah *object* wajah yang dikenali.

C. Inertial Measurements Unit (IMU)

Untuk mendapatkan *log data* penerbangan seperti navigasi dan *positioning* pada *drone* penelitian ini memanfaatkan komponen IMU yang menjadi salah satu nilai untuk memastikan data penerbangan pada *drone*. IMU biasanya terdiri dari tiga *tri-axial sensors* seperti *accelerometer*, *gyroscope* dan *magnetometer* [9]. Pengukuran *gyroscope* & *accelerometer* memberikan nilai penting pada pengukuran kecepatan (*velocity*), ketinggian (*altitude*) dan akselerasi (*acceleration*) yang terdapat pada *drone*, IMU data merupakan bagian integral dari INS (*Inertial Navigation System*) [10].

Hasil dari IMU akan disimpan kedalam *log file* yang akan ditampilkan setelah *drone* mendarat, IMU pada *library python* hanyazou/Tellopy terdapat pada *flight_data log* yang bisa diakses secara *realtime*. Namun untuk meningkatkan performa saat *drone* berada di udara *log* akan ditampilkan pada saat *drone* sudah sudah berada di permukaan. Penelitian ini memproses *log* tersebut menjadi sebuah grafik 4 Dimensi menggunakan *library python* matplotlib.pyplot, grafik akan menunjukkan *movement orientation* dari *drone* sehingga terlihat secara *visual track* atau *path* saat *drone take off* sampai akhirnya *landing*.

D. Sinematik Video Measurement

Sebelum melakukan proses perekaman video menggunakan *drone* DJI Tello secara otomatis, terdapat beberapa parameter yang perlu di perhatikan agar bisa menghasilkan video sinematik. Selain parameter konstant yang berupa *variable* program yang dijalankan pada system, parameter penting lainnya perlu ditambahkan untuk melakukan penilaian

terhadap video. Diantara parameter yang dilakukan pada penelitian ini adalah melibatkan *film director* untuk memberikan saran skenario apa saja yang dapat dinilai sebagai sinematik video, selain itu penilaian berupa kuesioner yang berisi video hasil rekaman dari *system drone* cerdas yang diberikan kepada orang umum.

IV. HASIL DAN UJI COBA

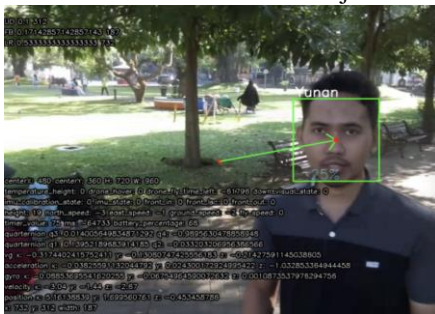
Penelitian ini menjadikan *drone* sebagai media untuk melakukan perekaman yang akan di proses oleh *system* sehingga menjadi *training set*. *Data training set* ini digunakan ketika *drone* terbang dan juga ketika *system* berjalan sehingga *input* gambar diterima dari *stream* video yang akhirnya mampu mengidentifikasi gambar wajah tersebut.

Algoritma pada proses pengenalan wajah menggunakan *haarcascade frontalface default cascade classifier* pada *library python* openCV melalui kamera *drone*, dengan kamera dan *library* tersebut *system* mampu mendeteksi wajah yang didepannya dan menyimpan gambar kedalam *harddisk computer*. Adapun hasil gambar disimpan memiliki spesifikasi *file* dengan ukuran 16KB, type JPEG image, dimensi 287x287 dengan warna abu-abu.

Penelitian ini juga menggukan *library* LBPH pada openCV *python* yaitu *face.LBPHFaceRecognizer_create()* dan menggunakan *detector* *haarcascade_frontend_default* untuk mendeteksi wajah dengan nilai parameter sebagai berikut:

- threshold*: 1.7976931348623157e+308
- radius*: 1
- neighbors*: 8
- grid_x*: 8
- grid_y*: 8

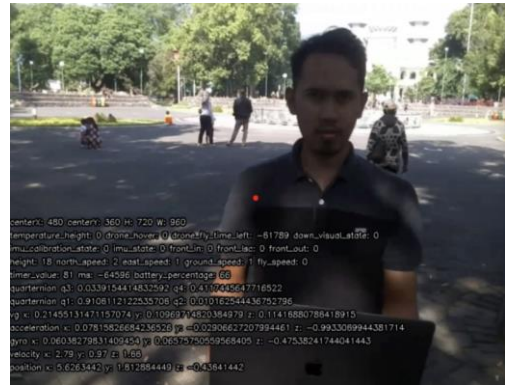
Proses uji coba yang pertama kali dilakukan adalah mengenali wajah yang berada di depan kamera *drone* dengan menggunakan *data training input data*. Dalam penelitian ini dilakukan pengenalan data wajah dan tersimpan di dalam *harddisk* komputer sehingga program dapat memberikan label pada kamera sesuai wajah tersebut. Setelah melakukan uji coba menerbangkan *drone* sekaligus mengaktifkan salah satu skenario berikut juga *face recognition* program dapat menampilkan wajah yang dikenali dan mampu memberikan label sekaligus *confidence value* pada setiap wajah yang dikenali. Adapun *confidence value* tersebut tergantung dengan jarak antara kamera *drone* dan wajah



Gambar 7 Hasil Face Recognition

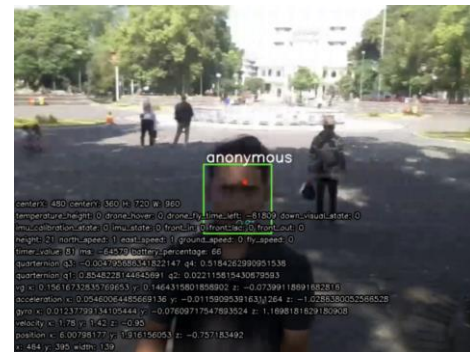
Pada saat uji coba terdapat juga kondisi yang mempengaruhi program tidak dapat menampilkan label wajah yang sudah dikenali yaitu *factor backlight* yang berada di sekitar *subject*, sehingga wajah yang berada di depan

kamera tidak terdeteksi sama sekali. Dapat dilihat pada gambar 8.



Gambar 8 face detection tidak dapat mendeteksi wajah

Selain itu pada saat uji coba ketika *drone* berada di udara juga ditemukan wajah yang sudah dikenali namun program mendeteksi sebagai "*anonymous*".Seperti terlihat pada gambar 9.

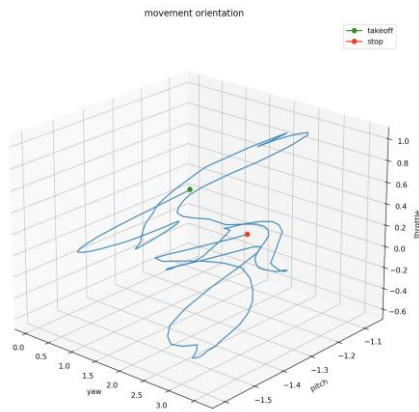


Gambar 9 face recognition yang tidak dapat mengenali wajah dan memberi label "anonymous"

Dari hasil uji coba tersebut penelitian ini juga melakukan uji coba dengan mendapatkan hasil riwayat terbang yang mampu di tampilkan pada grafik 3 Dimensi seperti pada gambar 10. Prosedur pembuatan media aplikasi melibatkan *film director* untuk menentukan spesifikasi awal pada rencana sinematik video yang terdiri dari urutan adegan, *type* pergerakan camera, jenis bidikan kamera dan komposisi layar [11]. Pada penelitian ini penulis telah berhasil menggambarkan *route* atau *path* pergerakan seperti pada gambar 10.

Lalu penelitian ini juga melibatkan *film director* untuk memberikan skenario agar video yang di tangkap dapat memberikan hasil sesuai dengan salah satu perencanaan *sinematography*. Adapun *type* pergerakan kamera yang telah dilakukan dalam penelitian ini hanya mampu menangkap adegan yang berada di depan kamera *drone*, karena kamera *drone* hanya terletak di sisi depan saja dan tidak bisa bergerak. Pembuatan sinematik video ini sangat bergantung pada aktor dan *film director* yang menentukan hasil dari pembuatan video sinematik seperti pergerakan *subject* dengan wajah yang dikenali agar sesuai dengan skenario atau adegan yang telah diberikan oleh *film director*. Pengenalan wajah pada penelitian ini berhasil mendeteksi wajah *subject* yang dikenali dengan baik. Sehingga dengan kombinasi *Rule based* yang telah diformulasikan pada *drone*, *drone* dapat

mengikuti wajah *subject* dengan baik.



Gambar 10 posisi koordinat yang menampilkan x,y,z untuk skenario drone mengikuti wajah

V. KESIMPULAN DAN SARAN

Pada hasil uji coba yang telah dilakukan penelitian ini telah berhasil mengenali wajah dengan baik menggunakan algoritma LBPH dan juga *haar cascade classifier* yang memanfaatkan *library* openCV melalui kamera *drone*. Selain itu pengaturan *control* menggunakan *tellopy library python* dapat dengan mudah diimplementasikan jika titik yang akan dituju sudah diketahui, lalu untuk melakukan pengendalian *drone* mampu bergerak lebih stabil mengikuti *subject* daripada menggunakan *control* dari *joystick* atau *controller* yang dikendalikan manusia secara manual.

Database pengenalan wajah yang disimpan dalam penelitian ini diharapkan mampu menjadi sumber untuk mengenali wajah untuk penelitian-penelitian berikutnya. Sehingga proses *input* dan perekaman wajah dapat dikurangi pada penelitian selanjutnya.

Kecerdasan yang menggunakan rumus mampu diterapkan untuk menghitung koordinat wajah yang terekam di layar pada *system drone* dengan memanfaatkan *rule base system* terbukti mampu mengendalikan pergerakan *drone* secara stabil mengikuti pergerakan wajah *subject*.

Skenario yang diterapkan dalam penelitian ini juga mampu membuktikan bahwa *drone* mampu bergerak sesuai ekspektasi peneliti sehingga menghasilkan video sinematik yang stabil dan telah dilakukan *survey questioner* untuk membuktikan bahwa video telah mengikuti salah satu konsep *cinematography* seperti jarak sudut pandang, *object* yang menarik dan area pergerakan kamera.

Selain hasil uji coba yang dapat disimpulkan penelitian ini juga memberikan beberapa saran yang dapat membantu penelitian selanjut pada bidang yang sesuai, diantaranya seperti memodifikasi *drone* dengan merancang kamera agar terpasang di bawah *drone* sehingga mampu menghasilkan sudut pandang yang lebih luas disbanding dengan penelitian saat ini yang hanya mendapatkan sudut pandang dari depan saja selain itu melakukan implementasi algoritma lainnya untuk mendeteksi dan mengenali wajah untuk membuktikan kestabilan *drone* dengan mencari wajah yang dipilih agar lebih baik.

DAFTAR PUSTAKA

- [1] Adelia Sabirova, Roman Fedorenko, "Drone Cinematography System Design and New Guideline Model for Scene Objects Interaction.", (2020), University Of Prince Edward Island, IEEE Xplore.
- [2] Osman Terik Centikaya, Semint Sandal, Erkan Bostanci, Mehmet Serdar Guzel "A Fuzzy Rule Based Visual Human System for Drones", (2019), 4rd International Conference on Computer Science And Engineering.
- [3] Brown, Blain, , "Cinematography; Theory and practice for cinematographers and directors", (2022), Routledge, 2 Park Square, Milton Park, Abingdon, Oxon OX14 4RN.
- [4] Chong Huang, Fei Gao, Jie Pan, Zhenyu Yang, Weihao Qiu, Peng Chen, Xin Yang, Shaojie Shen, and Kwang-Ting (Tim) Cheng, "ACT: An Autonomous Drone Cinematography System for Action Scenes", (2018), 2018 IEEE International Conference on Robotics and Automation (ICRA), p. 7039.
- [5] Osman Terik Centikaya, Semint Sandal, Erkan Bostanci, Mehmet Serdar Guzel "A Fuzzy Rule Based Visual Human System for Drones", (2019), 4rd International Conference on Computer Science And Engineering, p. 275.
- [6] Osman Terik Centikaya, Semint Sandal, Erkan Bostanci, Mehmet Serdar Guzel "A Fuzzy Rule Based Visual Human System for Drones", (2019), 4rd International Conference on Computer Science And Engineering, p. 276.
- [7] Kevin Salton do Prado, "Face Recognition: Understanding LBPH Algorithm", (2017), <<https://towardsdatascience.com/face-recognition-how-lbph-works-90ec258c3d6b>>
- [8] <https://github.com/youngsoul/tello-sandbox>, diakses pada 16 Januari 2022.
- [9] Sitank Bhatia*, Hai Yang, Rui Zhang, Fabian Hoflinger, Leonhard Reindl, "Development of an Analytical Method for IMU Calibration", (2016), 2016 13th International Multi-Conference on Systems, Signals & Devices.
- [10] Akhila Madhu, Harshith Kumar M B, Prajeesha, "Positioning Optimization of Drones using IMU and Securing UAV Communication by implementing Hybrid Cryptosystem", (2021), Proceedings of the Fifth International Conference on Trends in Electronics and Informatics (ICOEI)
- [11] Iason Karakotas, Ioannis Mademlis, Nikos Nikolaidis & Ioannis Pita, "Shot Type Feasibility in Autonomous UAV Cinematography", (2019), ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)