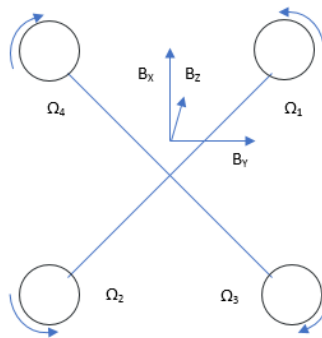


BAB II

TINJAUAN PUSTAKA

1.1 Konsep Dasar *Quadcopter*

Quadcopter merupakan kendaraan tanpa awak memiliki empat baling-baling yang terhubung melalui lengan yang saling bersilangan membentuk huruf “X”. Baling-baling ini memiliki *pitch* dan diameter tertentu sehingga saat digerakkan oleh motor elektrik menghasilkan gaya angkat yang lebih besar dari gaya hambatnya.



Gambar 2.1. Konsep dasar *quadcopter* dalam posisi *hover*

Arah depan pada *quadcopter* mewakili sumbu X pada badan *quadcopter*. Perputaran pada keempat baling-baling tidak semuanya sama yaitu pada lengan diagonal baling-baling 1 dan 2 akan berputar berlawanan arah jarum jam. Sedangkan pada baling-baling 3 dan 4 akan berputar searah jarum jam. Keuntungan dari arah perputaran baling-baling ini adalah untuk meniadakan efek momen inersia dan *quadcopter* dapat melayang stabil diudara. Dari kombinasi pergerakan keempat motor tersebut, *quadcopter* memiliki empat pergerakan sebagai berikut:

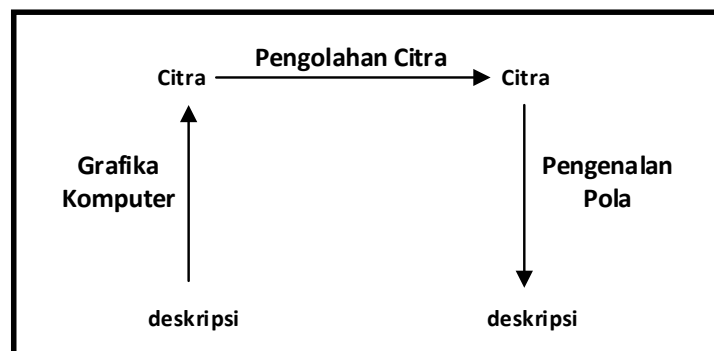
- Thrust* adalah gaya yang menimbulkan *quadcopter* bergerak searah sumbu Z vertikal dari bumi. Untuk menimbulkan gaya *thrust*, kecepatan keempat motor harus sama dan secara bersamaan motor tersebut ditambah kecepatannya hal ini akan menimbulkan gaya angkat lebih besar dan *quadcopter* akan bergerak naik keatas begitu juga sebaliknya.
- Roll* adalah torsi yang mengakibatkan *quadcopter* berputar disepanjang sumbu X dari bodinya. Hal ini terjadi ketika dua motor berputar lebih cepat dari dua

motor lainnya. Misalnya jika *quadcopter* akan melakukan *roll* kearah kanan maka yang terjadi adalah dua motor sebelah kiri akan berputar lebih cepat dari dua motor sebelah kanan begitu juga sebaliknya.

- c) *Pitch* adalah torsi yang mengakibatkan *quadcopter* berputar disepanjang sumbu Y dari bodinya. Hal ini terjadi ketika dua motor berputar lebih cepat dari dua motor lainnya. Misalnya jika *quadcopter* akan melakukan *pitch* kedepan maka yang terjadi adalah dua motor bagian belakang akan berputar lebih cepat dari dua motor bagian depan begitu juga sebaliknya.
- d) *Yaw* adalah torsi yang mengakibatkan *quadcopter* berputar disepanjang sumbu Z dari *body*-nya. Pergerakan ini dipengaruhi oleh perubahan kecepatan dari keempat motor. Jika kecepatan motor depan dan belakang diperlambat sedangkan kecepatan motor kanan dan kiri dipercepat maka *quadcopter* akan bergerak menyimpang kekiri begitu pula sebaliknya.

1.2 Pengolahan Citra

Pengolahan citra adalah pemrosesan citra yang dilakukan pada sebuah komputer yang bertujuan meningkatkan kualitas dari citra. Didalam bidang komputer, terdapat tiga bidang yang berkaitan dengan citra yang disajikan dalam hubungan gambar berikut ini.



Gambar 2.2. Hubungan antar bidang terkait citra

- a) Grafika komputer (*computer graphics*) bertujuan untuk menghasilkan citra (grafik atau *picture*) dengan primitive-primitif geometric geometri seperti garis, lingkaran, dan sebagainya. Primitif-primitif geometri tersebut memerlukan data deskriptif untuk melukis elemen-elemen gambar. Contoh data deskriptif adalah koordinat titik, panjang garis, jari-jari lingkaran, tebal garis, warna, dan

sebagainya. Grafika komputer memainkan peranan penting dalam visualisasi dan *virtual reality*.

- b) Pengolahan citra (*image processing*) bertujuan memperbaiki kualitas citra agar mudah diinterpretasi oleh manusia atau mesin (dalam hal ini komputer). Teknik-teknik pengolahan citra mentransformasikan citra menjadi citra lain. Jadi, masukannya adalah citra dan keluarannya juga citra, namun citra keluaran mempunyai kualitas lebih baik daripada citra masukan.
- c) Pengenalan pola (Pengenalan Pola (*pattern recognition/image interpretation*) adalah pengelompokan data numerik dan simbolik (termasuk citra) secara otomatis oleh mesin (dalam hal ini komputer). Tujuan pengelompokan adalah untuk mengenali suatu objek di dalam citra. Manusia bisa mengenali objek yang dilihatnya karena otak manusia telah belajar mengklasifikasi objek-objek di alam sehingga mampu membedakan suatu objek dengan objek lainnya. Kemampuan sistem visual manusia inilah yang dicoba ditiru oleh mesin. Komputer menerima masukan berupa citra objek yang akan diidentifikasi, memproses citra tersebut, dan memberikan keluaran berupa deskripsi objek di dalam citra.

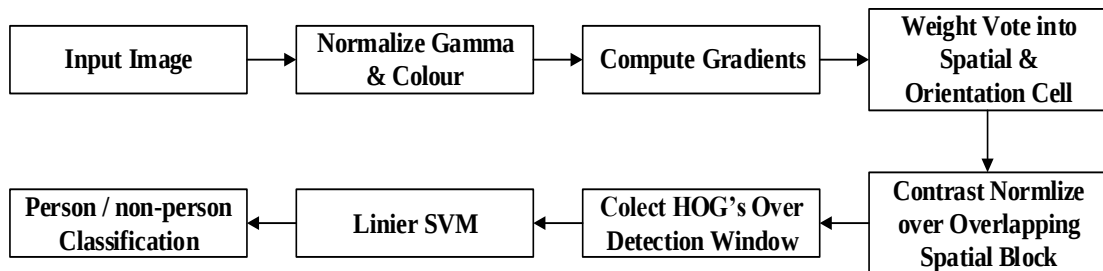
1.3 Library OpenCV

OpenCV adalah sebuah *library open source computer vision* yang ditulis dalam bahasa C dan C++ dan telah dikembangkan untuk menyederhanakan programing terkait pengolahan citra dinamis secara *realtime*. Selain dalam Bahasa C dan C++, OpenCV sudah dikembangkan ke python, java, dan matlab serta dapat dijalankan diberbagai *operating system* yaitu Linux, Windows, dan sekarang dapat dijalankan di Mac OS X dan Android [1]. OpenCV dirancang untuk efektivitas dalam pengolahan citra dan difokuskan dalam implementasi untuk pembuatan aplikasi vision, contohnya algoritma untuk teknik kalibrasi (*Camera Calibration*), deteksi fitur (*Feature*) dan pelacakan (*Optical Flow*), analisis bentuk (*Geometry, Contour Processing*), analisis gerak (*Motion Templates, Estimators*), rekonstruksi 3D (*View Morphing*), objek segmentasi dan pengenalan (*Histogram, Embedded Hidden Markov Models, Eigen Objects*). OpenCV menyediakan infrasturkutr computer

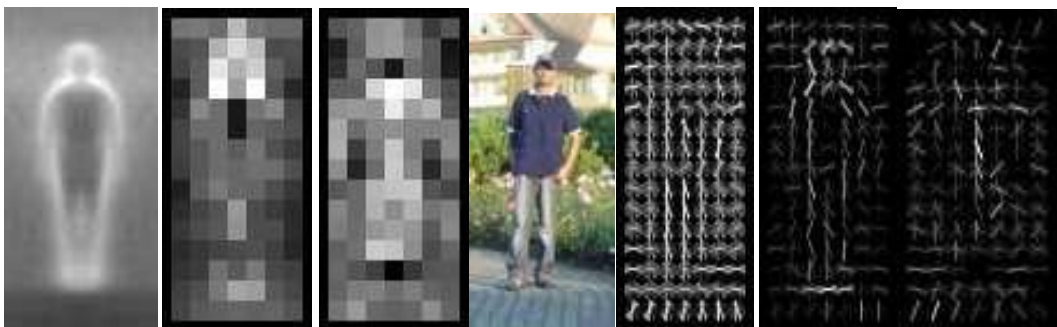
vision yang mudah pengaplikasiannya sehingga dapat digunakan dengan cepat untuk membangun sebuah aplikasi vision [5]

1.4 Histogram of Oriented Gradients (HOG)

Histogram of Oriented Gradients merupakan metode pengolahan citra yang digunakan untuk mendeteksi objek (manusia). Metode ini digunakan untuk memutuskan hasil pendeteksian apakah manusia atau bukan manusia. Teknik yang digunakan pada metode ini dengan cara menghitung gradien dalam daerah tertentu dalam sebuah citra. Proses pendeteksian objek (manusia) pada metode ini dilakukan per-*frame* dari citra yang ditangkap sebagai hasil pencacahan video. *Frame* hasil pencacahan video memiliki karakteristik yang berbeda yang ditunjukkan melalui distribusi gradien. Karakteristik ini diimplementasi dengan membagi citra ke dalam jendela-jendela kecil (*cell*). Tiap *cell* tersebut akan gabungan kembali ke sebuah histogram dari sebuah gradien. Gabungan dari histogram ini akan membentuk representasi dari citra yang mewakili objek [2][4]. Berikut ini proses ekstraksi dari metode HOG dilakukan.



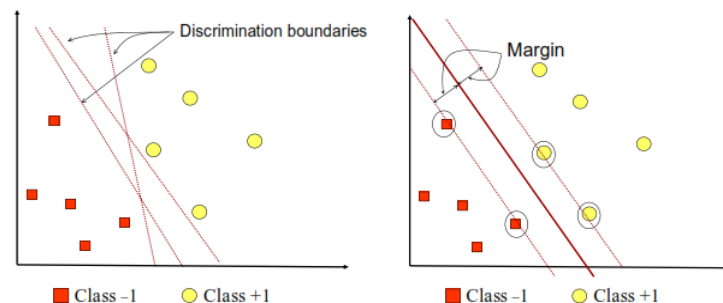
Gambar 2.3. Sistem proses ekstraksi HOG [2]



Gambar 2.4. Proses ekstraksi citra dengan metode HOG [2]

1.5 Support Vector Machine (SVM)

Support Vector Machine (SVM) merupakan metode yang digunakan untuk mengidentifikasi dan membedakan objek yang terdeteksi manusia atau bukan manusia serta pencocokan objek (manusia) yang dideteksi [4]. Proses dari metode ini bekerja saat framework HOG pada OpenCV hasil pencacahan video dari penangkapan video oleh *webcam*. Dalam proses ini SVM berfungsi sebagai pemisah dua buah *class* pada *input space* antara : -1 dan +1. *Class* -1 dilambangkan kotak warna merah dan *class* +1 dilambangkan lingkaran warna kuning. Prinsip dasar dari SVM sendiri adalah sebagai *linier classifier* yang bisa dilihat pada gambar dibawah ini.



Gambar 2.5. Proses SVM menemukan hyperlane terbaik untuk memisahkan kedua class-1 dan +1 [2]

1.6 Sistem Komputer Pendamping

Komputer pendamping merupakan *hardware* tambahan yang berfungsi untuk membuat keputusan cerdas. *Flight controller* saat ini mungkin dapat membuat keputusan namun terbatas hanya meliputi *autonomous* yang standar. Sebelum komputer pendamping, pengembang UAV tidak mempunyai pilihan lebih selain terpaku pada kode pengendali penerbangan yang terbatas [10]. Sekarang dengan hadirnya komputer pendamping, pengembang *drone* dapat membangun aplikasi *drone* canggih dengan intelijen *onboard*, konektivitas tanpa batas ke *cloud* melalui 4G/LTE, mengintegrasikan muatan *custom* dan membangun & menghubungkan Antarmuka *Web / Mobile custom*. Jalur komunikasi yang digunakan melalui jaringan kabel dengan modul *driver* FTDI yang terlihat pada gambar (2.6) [11]. *Driver* ini digunakan untuk mengonveri saluran komunikasi serial ke UART pada pengendali penerbangan.



Gambar 2.6. Sambungan komunikasi odroid ke pixhawk

Dalam pengoperasiannya sistem operasi yang terpasang pada komputer pendamping yaitu linux ubuntu dan telah terinstal *mavproxy* dan *library dronekit*. *Mavproxy* atau MAVLink adalah protokol pengiriman pesan yang sangat ringan untuk berkomunikasi dengan *drone* dan antara komponen *drone* ke *onboard computer* sedangkan *dronekit* merupakan *platform open source* yang menangani algoritma penerbangan berbasis bahasa python. Berikut merupakan perintah dasar pada pemrograman python pada *library dronekit*:

```
"# mavproxy.py --master=/dev/ttyUSB0 --baudrate 1500000 --aircraft MyCopter"
```

Dimana, *ttyUSB0* merupakan *port* modul *driver* FDTI dan nilai 1500000 merupakan nilai *baudrate* yang digunakan untuk komunikasi antara odroid-xu4 dan pixhawk.