

## BAB III PERANCANGAN DAN IMPLEMENTASI

### 1.1 Desain Mekanik *Quadcopter*

Berikut rancangan *quadcopter* yang dibuat dengan menggunakan *frame* S500 dan terdapat dua *hardware* yaitu *pixhawk px4* sebagai *flight controller* dan *odroid-xu4* sebagai perangkat pengolah citra (*image processing*).



Gambar 3.1. *Packaging quadcopter* tampak depan dan tampak atas

Quadcopter yang dirancang memiliki dimensi : 39 x 39 x 25 cm, yang tersusun didalam sebuah rangka *frame* S500. Dalam *packaging quadcopter* terdapat beberapa modul dan perangkat utama serta pendukung sehingga menjadi quadcopter yang siap diterbangkan untuk menjalankan misi.

### 1.2 Desain Elektronik *Quadcopter*

Dalam penelitian ini produk yang dibangun dari beberapa *hardware* penting dalam membangun sistem *autonomous quadcopter*. Berikut *hardware* yang digunakan.

#### a) Odroid-xu4

Odroid-xu4 adalah sebuah komputer *single board* yang menggunakan prosesor *octa-core* Exynos 5422 dan berjalan di sistem operasi seperti *Ubuntu*, *Android*, *Fedora*, *ARCHLinux*, *Debian*, dan *OpenELEC* dengan ribuan paket terbuka perangkat lunak yang gratis.

Sistem ini merupakan sistem yang membantu *quadcopter* dapat membuat keputusan cerdas seperti *autonomous flight control realtime*, *detection object*

dan lainnya. Namun dalam penelitian ini sistem komputer pendamping digunakan untuk sistem navigasi *quadcopter* seperti *auto takeoff*, *hover time delay*, menuju *waypoint* dan *auto landing*. Komputer pendamping yang digunakan kedalam penelitian ini adalah mikro komputer dengan pemrosesan yang cepat yaitu odroid-xu4.



Gambar 3.2. Board odroid-xu4

b) Pixhawk

Sistem yang berfungsi sebagai pengendali utama penerbangan. Kami merencanakan penggunaan pixhawk sebagai *flight controller*. Pixhawk merupakan perangkat keras dan perangkat lunak yang didesain pada kendaraan tanpa awak untuk mampu terbang secara otomatis dengan menggunakan pengolahan *on-board* visi komputer. Pixhawk ini di sajikan kedalam bentuk sebuah *platform* penelitian *open-source* yang memungkinkan pengolahan penuh pada kendaraan tanpa awak.



Gambar 3.3. Pixhawk

c) FTDI USB

FTDI USB adalah sebuah modul antarmuka *USB to Serial UART* yang menggunakan chip FT322R dan dapat digunakan sebagai *USB to Serial/TTL Converter*. Dengan menggunakan USB ini komunikasi antar perangkat dapat berjalan secara serial.



Gambar 3.4. FDTI USB

d) *Buzzer*

*Buzzer* adalah komponen elektronika yang dapat memunculkan suara berdasarkan pulsa frekuensi yang diberikan baik itu digital maupun analog. Pada sistem *autonomous quadcopter* yang dibuat, buzzer digunakan sebagai indikator kepada pengguna bila objek terdeteksi.



Gambar 3.5. *Buzzer*

### 1.3 *Ground Control Station (GCS)*

Parameter-parameter yang terukur dikirimkan secara *wireless* menuju komputer yang berada pada GCS dengan menggunakan komunikasi serial frekuensi 433 Mhz. Data tersebut langsung dibaca pada *software* mission planner pada GCS. Untuk *quadcopter* dengan komputer pendamping GCS ditambahkan dengan *software* tambahan seperti PuTTY untuk mengakses (*Secure Shell*) SSH dan VNC Viewer untuk menampilkan layar *desktop* dari komputer pendamping yang tertanam pada *quadcopter*.



Gambar 3.6. Telemetri dan GCS

a) *Router Tenda*

Router tenda merupakan router yang memiliki standar dan protocol IEEE802.11n digunakan untuk sistem jaringan *wireless* yang memberikan kecepatan *wireless* hingga 300 Mbps. *Router* ini digunakan sebagai perantara sambungan dari komputer GCS dan odroid-xu4.



Gambar 3.7. *Router Tenda*

b) Modul WiFi Tp-Link TL-722N

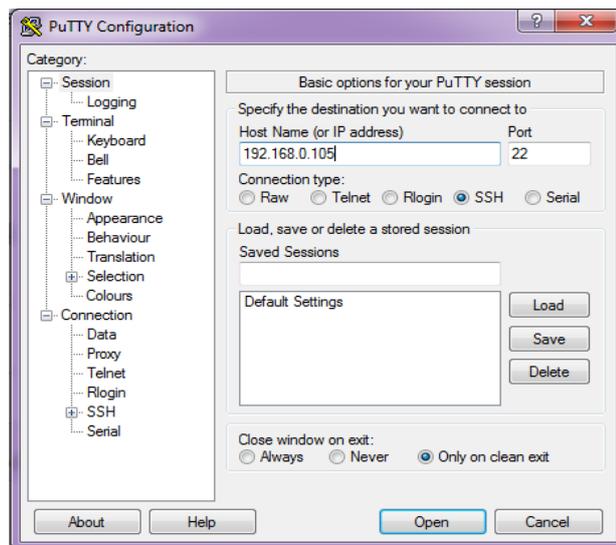
Modul wifi ini merupakan adapter USB *high gain* nirkabel untuk menyambungkan perangkat komputer atau *notebook* ke jaringan nirkabel yang memiliki kecepatan transmisi hingga 150 Mbps. Adapter ini digunakan pada perangkat odroid-xu4 sebagai adapter wifi sehingga memperkuat sambungan jaringan nirkabel dengan internet. Modul ini digunakan sebagai penangkap sinyal wifi yang dipancarkan oleh *router tenda*.



Gambar 3.8. Modul WiFi TP-Link

### c) PuTTY

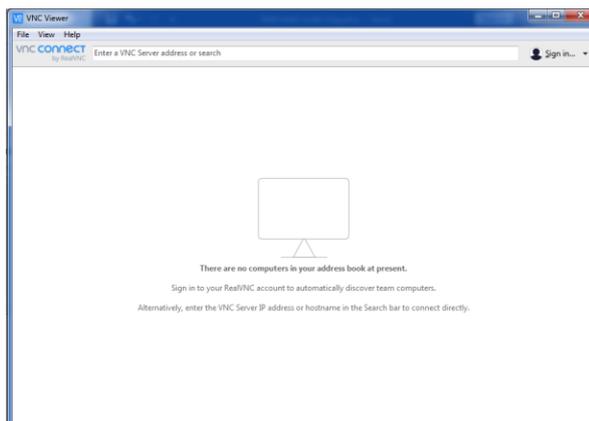
PuTTY adalah *software open-source* yang digunakan untuk melakukan *remote acces*, seperti Rlogin, (*Secure Shell*) SSH, dan Telnet. Dengan menggunakan PuTTY, pengguna dapat mengendalikan system dari jarak jauh untuk mengakses *server*. Untuk dapat menggunakan PuTTY, pengguna memanfaatkan protocol jaringan berupa IP (*Internet Protocol*) *server* agar pengguna melakukan remote pada perangkat yang ingin dikendalikan. PuTTY hanya akan menampilkan *command* teks untuk menjalankan perintah tertentu berdasarkan kendali dari pengguna. Layaknya aplikasi SSH lain, tentunya PuTTY memiliki fitur untuk melakukan koneksi SSH dan menjadikannya klien SSH. Pada sistem ini PuTTY digunakan untuk me-*remote* odroid-xu4 dari jarak jauh.



Gambar 3.9. Tampilan PuTTY

### d) VNC Viewer

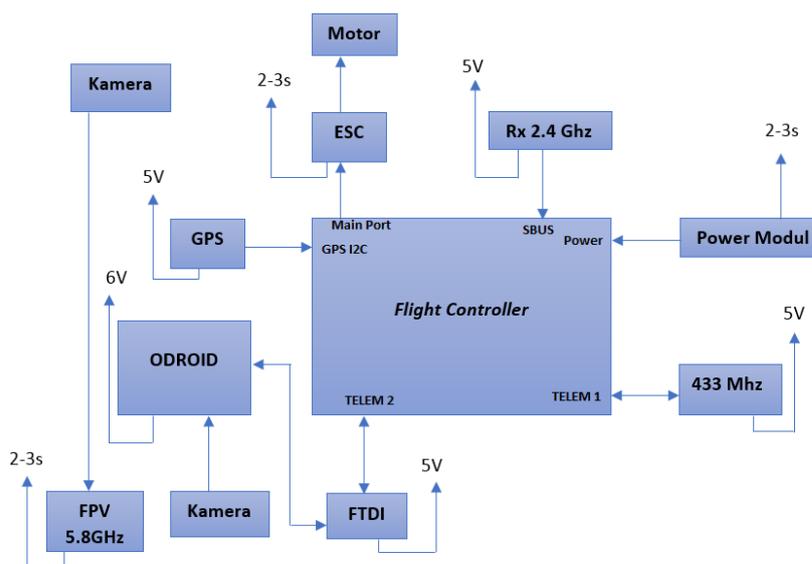
VNC *Viewer* adalah *software* yang digunakan untuk me-*remote desktop* komputer dari jarak jauh menggunakan jaringan nirkabel. *Software* ini digunakan untuk melihat tampilan *virtual environment* dari OS Ubuntu dan sekaligus me-*remote* odroid-xu4 dari jarak jauh.



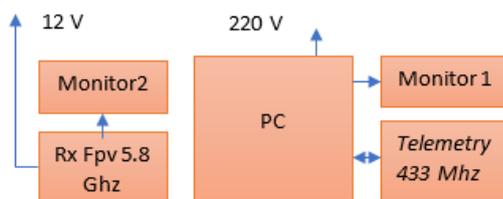
Gambar 3.10. Tampilan VNC Viewer

### 1.4 Diagram Blok Sistem dan Flowchart

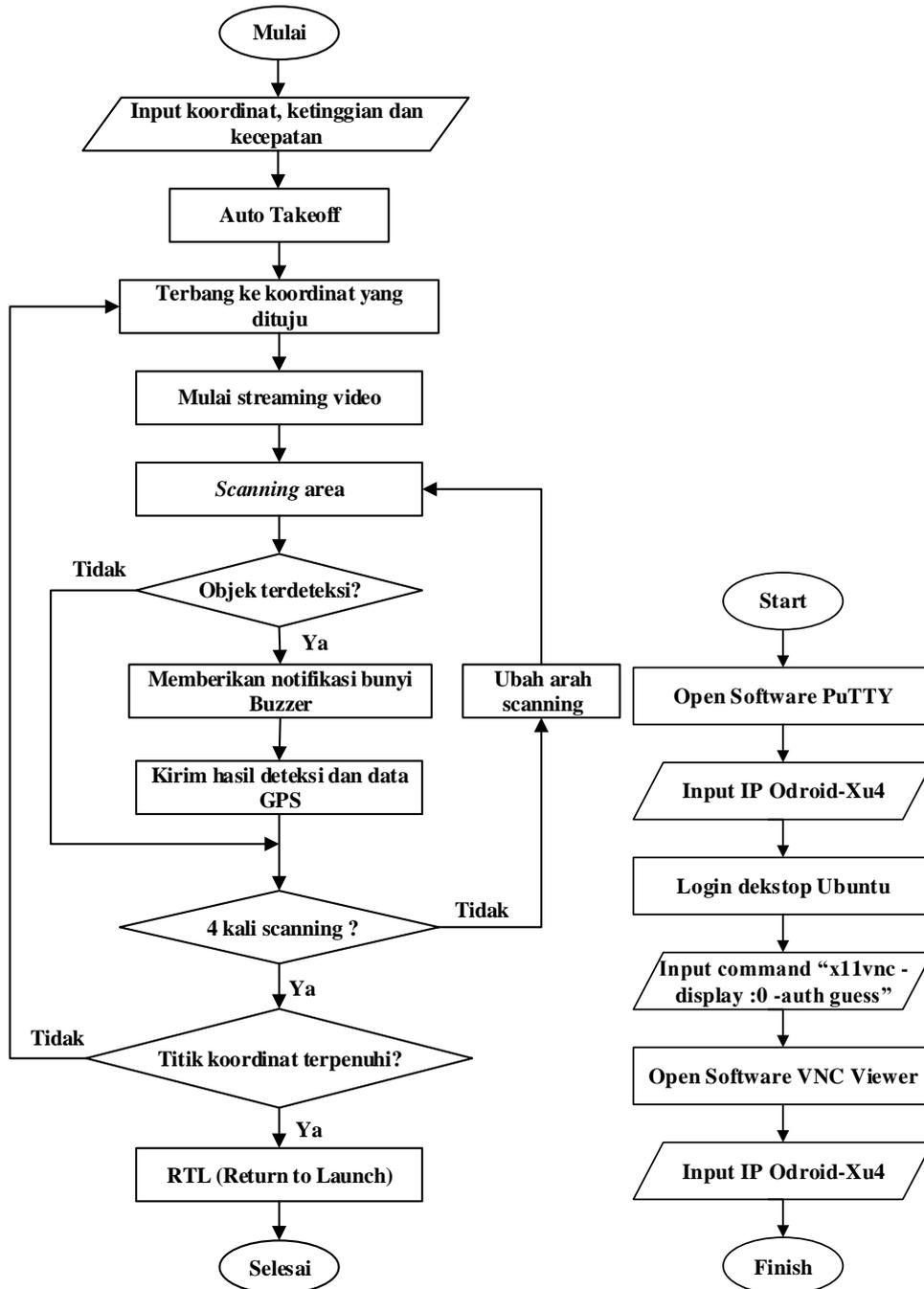
Pada bab ini berisi tentang diagram blok modul udara dan GCS dalam membangun sistem *autonomous quadcopter* beserta *flowchart* kerja sistem.



Gambar 3.11. Diagram blok modul udara



Gambar 3.12. Diagram blok modul GCS



Gambar 3.13. Flowchart sistem kerja *quadcopter* dan antarmuka

## 1.5 Perancangan Pendeteksian Objek Metode HOG dan SVM

Pada bab ini dibuat sebuah perancangan sistem pendeteksian objek dengan metode matematis metode HOG dan SVM. Langkah pertama pada proses pendeteksian objek adalah menormalisasikan gamma dan warna pada citra yang di-*input*-kan untuk meningkatkan kontras dari citra dengan melakukan tranformasi tingkat keabuaan citra. Berikut fungsi transformasinya (3.1) [6].

$$y(n_1, n_2) = x^\gamma(n_1, n_2). \quad (3.1)$$

Dimana,  $y(n_1, n_2)$  adalah citra *input*,  $x(n_1, n_2)$  adalah citra *output*, dan  $\gamma$  adalah parameter gamma [6]. Selanjutnya, mengitung gradien dari citra yaitu penyaringan dengan filter 2 dimensi (filter vertical dan filter horizontal). Dihitung dengan rumus persamaan (3.2) [7].

$$|G| = \sqrt{I_x^2 + I_y^2} \quad (3.2)$$

Dimana  $I$  adalah *grayscale level*,  $I_x$  adalah matrik terhadap sumbu- $x$ ,  $I_y$  adalah matrik terhadap sumbu- $y$  yang dihitung dengan metode 1-D *Centered* pada persamaan (3.3) - (3.4) [8][9].

$$I_x = I * D_x, I_y = I * D_y, \quad (3.3)$$

$$D_x = I(x + 1, y) - I(x - 1, y), \quad (3.4)$$

$$D_y = I(x, y + 1) - I(x, y - 1). \quad (3.5)$$

$D_x$  adalah nilai gradien citra sumbu  $x$  dengan mask  $[-1, 0, 1]$ , sedangkan  $D_y$  adalah nilai gradien citra sumbu  $y$  dengan *mask*  $\begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$  [7].

Dari nilai gradien tersebut, dihitung *magnitude*-nya dan nilai gradien ditransformasikan ke dalam sumbu dengan sudut  $0^\circ$  dan  $180^\circ$  yang disebut orientasi gradien [8]. Berikut perhitungan orientasi gradien pada persamaan (3.6) dan (3.7).

$$m(x, y) = \sqrt{D_x^2 + D_y^2}, \quad (3.6)$$

$$\theta(x,y) = \arctan \left( \frac{Dx}{Dy} \right). \quad (3.7)$$

Tahap selanjutnya adalah perhitungan histogram dari orientasi gradien pada tiap *cell* dengan melakukan perhitungan *bin*-nya dengan persamaan (3.8) dan (3.9) [8].

$$hist(xA) = hist(xA) + mag(x, y) \times (B - \theta(x,y)), \quad (3.8)$$

$$hist(xB) = hist(xB) + mag(x, y) \times (\theta(x, y) - A). \quad (3.9)$$

Dimana, nilai  $x_A$  = nilai *bin* ke-A,  $x_B$  = nilai *bin* ke-B,  $A$  = nilai sudut *bin* ke-A dan  $B$  = nilai sudut *bin* ke-B[8]. Setelah dilakukan perhitungan histogram pada tiap *cell* yang selanjutnya dilakukan penggabungan seluruh *cell* dalam satu blok [7][8]. Hasil tersebut akan dinormasillasikan dengan persamaan (3.10) dan (3.11).

$$L_1 - norm \ v' = \frac{v}{\sqrt{\|v\|^2 + \varepsilon}}, \quad (3.10)$$

$$L_2 - norm \ v'' = \frac{v}{\sqrt{\|v\|_2^2 + \varepsilon}}. \quad (3.11)$$

Dimana, nilai  $v$  = fitur blok yang mengandung histogram,  $\varepsilon = 0,1$  (*konstanta*), dan  $L - norm = normalisasi$  [8]. Pada bagian ini citra hasil ekstraksi fitur HOG dilinierkan dengan menggunakan *hyperplane* terbaik dari dua buah class pada input space yang berbeda untuk menentukan citra *positive* dan *negative* hasil ekstraksi fitur HOG. Formula untuk *hyperplane* disajikan pada persamaan (3.12) – (3.14)[7].

$$w^T x + b = 0, \quad (3.12)$$

Untuk citra *positive*, jika  $w^T x + b > 0$ ,

$$w^T x + b = 1, \quad (3.13)$$

Untuk citra *negative*, jika  $w^T x + b < 0$ ,

$$w^T x + b = -1. \quad (3.14)$$

Untuk mencari *margin* sebagai *space* dari *hyperplane*, digunakan persamaan (3.15).

$$M = (x^+ - x^-) \cdot n = (x^+ - x^-) \cdot \frac{w}{\|w\|} = \frac{2}{\|w\|}. \quad (3.15)$$

Variable  $n$  merupakan panjang vektor pada *hyperplane*.