

Aplikasi Android Untuk Remote Control Quadcopter

Ryan Nathanael Soenjoto Widodo, Petrus Santoso, Handry Khoswanto

Jurusan Teknik Elektro, Universitas Kristen Petra

Jl. Siwalankerto 121-131, Surabaya 60236, Indonesia

E-Mail: m23409003@john.petra.ac.id ; petrus@peter.petra.ac.id ; handry@peter.petra.ac.id

Abstrak—Android merupakan sistem operasi yang populer saat ini untuk perangkat *mobile*, dan perangkat Android ini dapat dikembangkan lebih lanjut dengan memanfaatkan sensor layar sentuh dan *accelerometer* yang ada dan perangkat WiFi untuk mengendalikan *quadcopter*. Berdasarkan fakta tersebut dibuat aplikasi Android untuk mengontrol *quadcopter* buatan Parrot yang memiliki empat motor penopang, yaitu AR.Drone 2.0. Berdasarkan pengujian aplikasi dapat mengontrol AR.Drone 2.0 dengan memanfaatkan layar sentuh dan sensor *accelerometer* dengan baik. Data navigasi dan data video berhasil diterima, dan diolah dengan baik. Namun aplikasi yang dibuat memiliki kekurangan, yaitu video dan fitur penerimaan data navigasi yang tidak lancar. Selain itu Java Android yang digunakan sebagai bahasa pemrograman utama tidak dapat langsung memproses paket UDP dengan delay kurang dari 140 ms. Untuk perintah AT*PCMD yang memuat empat arah (*pitch*, *roll*, *gaz*, dan *yaw*) dalam satu perintah AT untuk kontrol, jika digunakan untuk lebih dari satu arah untuk satu perintah AT*PCMD maka pergerakan AR.Drone 2.0 akan menjadi tidak teratur. Perintah itu diabaikan oleh AR.Drone 2.0 selama AR.Drone 2.0 tidak terbang. Selain itu AR.Drone 2.0 hanya dapat menerima perintah dari satu perangkat saja.

Kata kunci— AR.Drone 2.0, Android, remote control, data navigasi, video

I. PENDAHULUAN

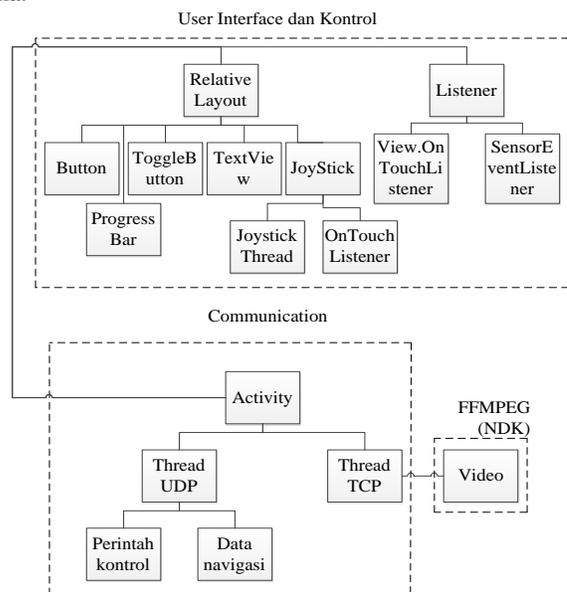
PEMANFAATAN *Unmanned Aerial Vehicle* (UAV) untuk pengawasan, pengintaian, mengambil data di tempat yang sulit dijangkau oleh manusia, dan lain-lainnya sangatlah membantu manusia dalam melakukan tugasnya. UAV dapat dikontrol secara nirkabel ataupun *autonomous*. Salah satu bentuk dari UAV adalah *quadcopter*. *Quadcopter* memiliki empat motor yang berfungsi sebagai penopang.

Salah satu perangkat yang populer saat ini adalah perangkat yang berbasis sistem operasi Android Hampir semua perangkat Android memiliki berbagai sensor untuk mengendalikan sistem operasi Android, sebagai contoh yaitu sensor layar sentuh dan *accelerometer*. Dengan adanya sensor-sensor tersebut dan perangkat WiFi, perangkat Android cocok untuk digunakan untuk mengendalikan *quadcopter* buatan Parrot, yaitu AR.Drone 2.0.

II. DESAIN DAN IMPLEMENTASI SISTEM

Aplikasi yang dibuat mempunyai berbagai fungsi untuk mengontrol AR.Drone 2.0, antara lain mengontrol pergerakan

AR.Drone 2.0, mengambil gambar dari kamera utama, mengambil data navigasi dari AR.Drone 2.0 yang berisi tentang status AR.Drone 2.0 dan data-data mengenai persentase baterai, kemiringan, ketinggian, dan lain-lainnya, dan mengolah data video yang diterima dari AR.Drone 2.0. Gambar 1 menunjukkan blok diagram dari aplikasi. Aplikasi memiliki dua thread yang digunakan untuk berkomunikasi dengan AR.Drone 2.0, yaitu thread UDP dan thread TCP. Sedangkan untuk tampilan diatur oleh thread UI atau thread utama.



Gambar. 1. Blok diagram sistem.

A. Desain Aplikasi

Thread utama merupakan activity utama yang dapat beradaptasi untuk tiap orientasi dengan *layout* yang berbeda.

Untuk kontrol digunakan thread UI dari Activity. Pada tampilan antar muka digunakan Button, ToggleButton, TextView, ProgressBar, dan JoyStick. JoyStick adalah View yang dibuat khusus untuk menampilkan gambar hasil video dan joystick yang digunakan untuk kontrol. Semua View tersebut dapat diakses melalui Activity.

Pada thread utama tidak ada koneksi socket, hal ini dikarenakan mulai dengan *update* Honeycomb, koneksi socket tidak boleh dilakukan di thread utama, maka dari itu setiap koneksi socket dilakukan di thread yang berbeda.

Pada awal aplikasi dijalankan pengguna harus memposisikan perangkat androidnya ke posisi portrait supaya

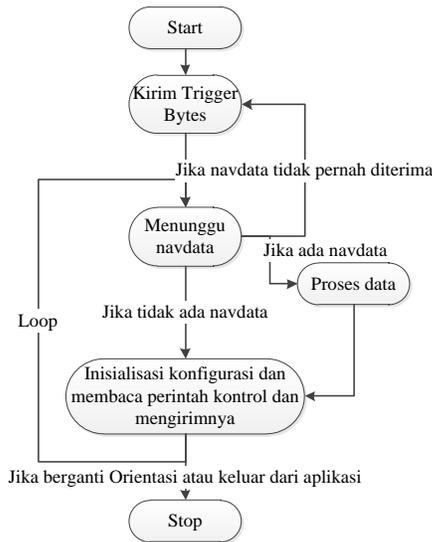
masuk mode konfigurasi, di mode konfigurasi pengguna dapat melakukan konfigurasi mengenai ketinggian AR.Drone 2.0, kecepatan, dan menyalakan atau mematikan video. Jika ToggleButton On/Off ditekan dan sudah terhubung ke WiFi maka thread untuk mengatur komunikasi UDP untuk kontrol dan menerima data navigasi diaktifkan.

Jika perangkat android berorientasi portrait maka *layout* akan berganti menjadi mode kontrol. Jika pada mode konfigurasi video dinyalakan, maka ketika berganti *layout* ke *layout* mode kontrol, thread yang digunakan untuk mengatur komunikasi TCP untuk video dinyalakan. Pada mode kontrol pengguna dapat melakukan kontrol AR.Drone 2.0.

B. Desain Thread UDP

Thread UDP akan mengatur pengiriman perintah kontrol dan penerimaan data navigasi. Thread ini akan mengaktifkan pengiriman data navigasi pada AR.Drone 2.0 dengan mengirimkan *trigger byte* terlebih dahulu. Kemudian dimulai pengiriman perintah kontrol dan konfigurasi, dan penerimaan data navigasi. Untuk mendapatkan delay yang akan digunakan untuk delay pengiriman kontrol digunakan timeout dari penerimaan data navigasi sehingga lebih efisien. Delay didapatkan jika tidak didapat data navigasi, atau data navigasi datang terlambat.

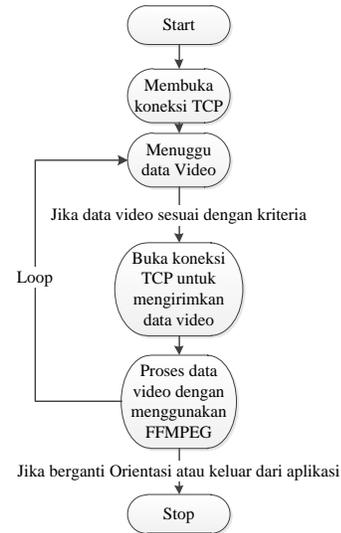
Jika tidak ada perintah AT atau konfigurasi yang harus dikirim maka perintah kontrol untuk diam di tempat akan dikirim.



Gambar. 2. State diagram thread UDP.

C. Desain Thread TCP

Thread TCP digunakan untuk mengendalikan komunikasi TCP antara aplikasi dan AR.Drone 2.0. TCP digunakan untuk menerima data video yang dikirim AR.Drone 2.0. Thread ini akan membuat thread baru yang berfungsi untuk berkomunikasi dengan FFMPEG setelah mendapatkan data video yang sesuai kriteria. Setelah selesai mengirim data video, socket ditutup, dan akan dibuka lagi ketika ada data video yang akan diproses.

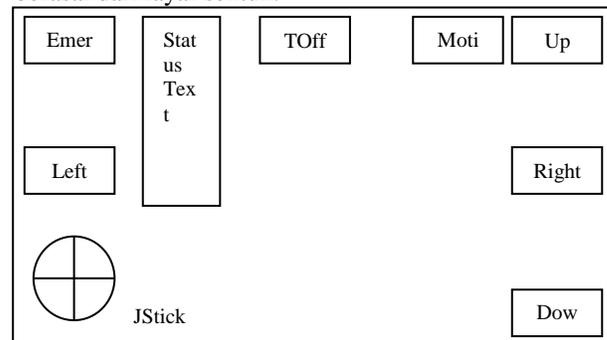


Gambar. 3. State diagram thread TCP.

Aplikasi ini memiliki satu Activity dan dua layout, layout untuk posisi *landscape* dan layout untuk posisi *portrait*. Layout dengan posisi *landscape* digunakan untuk kontrol, dan layout dengan posisi *portrait* digunakan untuk konfigurasi. Untuk mengetahui orientasi, maka pada View Button tRighr diberi kondisi *visibility="gone"* pada *landscape portrait*.

Pada posisi *landscape* akan ditampilkan layout yang didesain untuk kontrol. Pada layout ini terdapat View Buton, ToggleButton, TextView, dan JoyStick. Berikut penjelasan View-View yang digunakan:

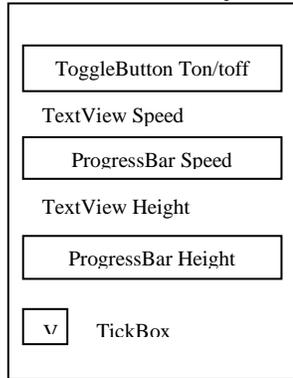
- TextView digunakan untuk menampilkan data yang didapat dari data navigasi. Button digunakan untuk kontrol.
- Button Left dan Right digunakan untuk memutar AR.Drone 2.0 ke kiri dan ke kanan.
- Button Up dan Down digunakan untuk menambah dan mengurangi ketinggian AR.Drone 2.0.
- Button Emergency digunakan untuk pendaratan darurat.
- Button Takeoff digunakan untuk lepas landas, dan mendarat.
- ToggleButton Motion digunakan untuk mengaktifkan kontrol dengan menggunakan sensor *accelerometer* dari perangkat Android.
- JoyStick merupakan View yang dibuat khusus untuk menampilkan gambar video AR.Drone 2.0, menampilkan gambar joystick dan membaca masukan pengguna yang berasal dari layar sentuh.



Gambar. 4. Layout landscape.

Layout portrait didesain untuk konfigurasi.View yang digunakan pada layout ini adalah ToggleButton, ProgressBar, TickBox, dan TextView. Berikut penjelasan dari kegunaan masing-masing View:

- ToggleButton untuk memulai dan menutup koneksi antara perangkat Android dan AR.Drone 2.0.
- ProgressBar untuk mengatur tingkat kecepatan dan ketinggian maksimum.
- TickBox digunakan untuk mengaktifkan dan menonaktifkan fitur video.
- TextView digunakan untuk menunjukkan nilai konfigurasi.



Gambar. 5. *Layout portrait*.

D. Implementasi Sistem

Implementasi system dapat dirangkum dalam beberapa proses berikut:

- 1) *Proses pengiriman perintah kontrol (AT)*: Perintah kontrol selalu dikirimkan langsung ke AR.Drone 2.0, dan AR.Drone 2.0 meresponnya dengan gerakan. Perintah kontrol dikirim sebanyak 30 kali tiap detik dengan cara memanfaatkan fitur `OnTouchListener`, dan *timeout* dari socket. *Delay* didapatkan dengan cara memberi *timeout* pada socket. *Timeout* yang diberikan adalah 25 ms dengan kompensasi *lag* pada aplikasi. `OnTouchListener` digunakan pada `JoyStick` dan `Button` agar dapat diketahui interaksi yang dilakukan pengguna dengan `Button` atau `JoyStick`. Jika disentuh maka akan memberitahu thread UDP untuk mengirim perintah kontrol. Untuk data *accelerometer* dan koordinat layar sentuh diubah menjadi arah berdasarkan batasan-batasan tertentu AR.Drone 2.0. Data arah kemudian diberikan ke thread UDP untuk dikirim ke AR.Drone 2.0. Perintah AT yang digunakan untuk mengontrol AR.Drone 2.0 adalah `AT*PCMD`.
- 2) *Proses pengiriman perintah konfigurasi (AT)*: Untuk perintah konfigurasi, sebelum mengirimkan perintah konfigurasi, perintah autentikasi untuk konfigurasi harus dikirimkan (`AT*CONFIG_IDS`) setelah perintah konfigurasi dikirimkan direkomendasikan untuk menunggu hingga control command ack di data navigasi meminta untuk di reset dengan cara mengirimkan `AT*CTRL` ke AR.Drone.
- 3) *Proses penerimaan dan pengolahan data navigasi*: Untuk mengaktifkan pengiriman data navigasi perlu dikirimkan *trigger* bit ke UDP port 5554. *Trigger* bit adalah byte

array dengan isi {0x01,0x00,0x00,0x00}. *Trigger* bit diperlukan untuk memberitahu kemana Data navigasi harus dikirimkan. Setelah *trigger* bit dikirimkan diperlukan konfigurasi agar keluar dari mode *bootstrap*. Data navigasi yang diterima memiliki format *little endian*. Sebagai contoh data 4 byte yaitu 0x55667788 namun pada saat dikirimkan yang dikirimkan yaitu 0x88776655.

Header		Drone state		Sequence number		Vision flag	
0x55667788		32-bit int.		32-bit int.		32-bit int.	
Option 1				Checksum block			
id	size	data	...	cks id	size	cks data	...
16-bit int.	16-bit int.	16-bit int.	16-bit int.	32-bit int.	...

Gambar. 6. Format data navigasi.

- 4) *Penerimaan data video*: Karena AR.Drone 2.0 tidak memberi tahu terlebih dahulu ukuran data yang akan dikirim maka diperlukan cara untuk menerima data keseluruhan. Data video yang dikirimkan memiliki pola ukuran maksimum 1460 byte. Sehingga jika data yang diterima lebih kecil dari 1460 berarti data sudah yang diterima sudah lengkap.
- 5) *Pengolahan data video*: Pada gambar 8 ditunjukkan bahwa Setelah data diterima data dicek apakah memiliki *header* PaVE, Jika memiliki *header* PaVE dan ukurannya lebih besar dari 7000 byte maka data akan diproses oleh FFMPEG sehingga dihasilkan gambar dalam format BMP. Gambar dengan format BMP akan ditampilkan melalui `View JoyStick`.

III. PENGUJIAN SISTEM

Berikut merupakan hasil pengujian dari aplikasi yang dibuat:

A. Pengujian Terbang dan Kontrol

Pada awal aplikasi dibuka pengguna akan disambut dengan *activity* utama dengan *layout* sesuai dengan orientasi. Pengguna harus merubah orientasi perangkat Android menjadi *portrait* agar dapat melakukan konfigurasi dan mulai berkomunikasi dengan AR.Drone 2.0. Sebelum mengaktifkan koneksi (komunikasi) maka pengguna harus sudah terhubung dengan jaringan WiFi dari AR.Drone 2.0, apabila belum terhubung dan `ToggleButton on/off` sudah ditekan maka pengguna akan mendapatkan pemberitahuan melalui `Toast` bahwa belum terkoneksi dengan jaringan WiFi AR.Drone 2.0, hal ini dapat dilihat pada gambar 7.

Setelah terhubung orientasi perangkat harus dirubah menjadi *landscape* agar masuk ke *layout* kontrol. Semua tombol, `JoyStick`, dan sensor *accelerometer* telah dicoba dan berhasil. AR.Drone 2.0 bergerak sesuai dengan perintah yang diberikan. `JoyStick` berjalan dengan lancar ketika video dimatikan. Namun ketika video diaktifkan, terlihat adanya penurunan *frame rate* sehingga kelancaran joystick agak berkurang. Namun hal serupa tidak tampak pada sensor *accelerometer*, karena data gerakan yang ditampilkan hanyalah *text* pada `TextView`. `JoyStick` dan sensor



Gambar. 10. Konfigurasi uji coba koneksi dari dua perangkat.

IV. KESIMPULAN

Kesimpulan yang didapat dari pengerjaan tugas akhir ini adalah sebagai berikut:

- Aplikasi yang dibuat dapat berjalan dengan baik dengan memanfaatkan sensor-sensor yang ada dan koneksi WiFi.
- Berdasarkan pengujian, fitur-fitur yang terdapat pada aplikasi, antara lain, fitur kontrol dengan layar sentuh dan sensor *accelerometer*, fitur penerimaan dan pengolahan data navigasi dan data video dapat berfungsi sesuai dengan fungsinya.
- Berdasarkan perbandingan dengan aplikasi pembanding, dapat dilihat bahwa terdapat beberapa kekurangan dalam hal fitur. Aplikasi yang dibuat dapat berjalan dengan baik dengan memanfaatkan sensor-sensor yang ada dan koneksi WiFi.
- Berdasarkan pengujian, fitur-fitur yang terdapat pada aplikasi, antara lain, fitur kontrol dengan layar sentuh dan sensor *accelerometer*, fitur penerimaan dan pengolahan data navigasi dan data video dapat berfungsi sesuai dengan fungsinya.
- AT*PCMD yang memuat empat arah (*pitch*, *roll*, *gaz*, dan *yaw*) dalam satu perintah AT untuk kontrol, jika digunakan untuk lebih dari satu arah untuk satu perintah AT, maka pergerakannya akan tidak teratur.
- Perintah AT*PCMD tidak diproses oleh AR.Drone 2.0 selama AR.Drone 2.0 tidak terbang.
- Java Android tidak dapat langsung memproses paket UDP dengan *delay* kurang dari 140 ms.
- AR.Drone 2.0 hanya dapat menerima perintah dari satu perangkat saja.

REFERENSI

- [1] Piskorski, Stephane, et al. AR.Drone Developer Guide. Parrot S.A., 2012
- [2] Gargenta, Marko. Learning Android. Sebastopol: O'Reilly Media, Inc, 2011.
- [3] Guihot, Hervé. Pro Android Apps Performance Optimization. New York: Apress Media, L.L.C, 2012.
- [4] "About Ffmpeg." Ffmpeg. 12 October 2012. <<http://ffmpeg.org/about.html>>
- [5] "Activities." Android Developers. 16 November 2012. Google, Inc. 21 November 2012. <<http://developer.android.com/guide/components/activities.html>>
- [6] "Android (operating system)." Wikipedia the free encyclopedia. 29 November 2012. Wikimedia Foundation, Inc. 29 November 2012. <<http://www.tech2crack.com/history-android/>>
- [7] Vogel, Lars. "Android Development Tutorial." Vogella. 11 September 2012. Vogella. 17 September 2012. <<http://www.vogella.com/articles/Android/article.html>>
- [8] "Application Fundamentals." Android Developers. 16 November 2012. Google, Inc. 21 November 2012. <<http://developer.android.com/guide/components/fundamentals.html>>
- [9] "AR.Drone Developer Guide." ARDRONE open API platform. 21 May 2012. Parrot. 17 September 2012. <https://projects.ardrone.org/attachments/download/434/ARDrone_SD_K_2_0.tar.gz>
- [10] "AR.FreeFlight 2.0." Parrot SA. July 2012. Parrot. 17 September 2012. <<https://play.google.com/store/apps/details?id=com.parrot.freeflight>>
- [11] "Layouts." Android Developers. 16 November 2012. Google, Inc. 21 November 2012. <<http://developer.android.com/guide/topics/ui/declaring-layout.html#CommonLayouts>>
- [12] "NetworkOnMainThreadException." Android Developers. 16 November 2012. Google, Inc. 10 October 2012. <<http://developer.android.com/reference/android/os/NetworkOnMainThreadException.html>>
- [13] "Platform Versions." Android Developers. 16 November 2012. Google, Inc. 21 November 2012. <<http://developer.android.com/about/dashboards/index.html>>
- [14] "Recreating an Activity." Android Developers. 16 November 2012. Google, Inc. 21 November 2012. <<http://developer.android.com/training/basics/activity-lifecycle/recreating.html>>
- [15] "Technical Specifications." Parrot AR.Drone 2.0. Parrot. 21 November 2012. <<http://ardrone2.parrot.com/ardrone-2/specifications/>>
- [16] "The UAV – Unmanned Aerial Vehicle." The UAV. The UAV. 16 July 2012. <<http://www.theuav.com/>>
- [17] Aubrey, Chris. "Understand the navdata..." Online posting 24 August 2012, 12 October 2012 <<https://projects.ardrone.org/boards/1/topics/show/4780>>
- [18] Lockwood, Alex. "Why Ice Cream Sandwich Crashes Your App." ANDROID DESIGN PATTERNS. 2012. Alex Lockwood. 10 October 2012. <<http://www.androiddesignpatterns.com/2012/06/app-force-close-honeycomb-ics.html>>
- [19] Yadav, Manish. "History of Android." Tech2Crack. 2011. 16 December 2012 <<http://www.tech2crack.com/history-android/>>