

# Metode Pemantau Posisi dan Arah Gerak Helikopter Tanpa Awak dengan Google Maps API

Akbari Indra Basuki  
P2 Informatika-LIPI  
akbari@informatika.lipi.go.id

Oka Mahendra  
P2 Informatika-LIPI  
oka@informatika.lipi.go.id

## Abstract

*This paper discuss direction and position monitoring method of unmanned helicopter use Google Maps API service and GPS sensor. The signal generated by GPS sensor is a coordinate point of latitude and longitude. The direction of helicopter is obtained from GPS module by calculating data position history. GPS signal sent by radio modem in helicopter and received by radio modem in computer. The data received by computer then processed by Visual Basic 6 application. The processed data was used by Javascript application to display the helicopter position and direction on the map which retrieved by Google Maps API service. As the result, the user can observe helicopter direction and position using web browsers like Mozilla Firefox, Google Chrome or other web browser which already support HTML5. The update speed of data position in local server can reach up to 20 Hz. This ability is sufficient in the fact that the best commercial GPS today, only capable generate data in 5-10 Hz. By using this method the GPS data can be processed without any doubt about missing of the data.*

*Keywords: Google Maps, position, direction, GPS, unmanned helicopter.*

## Abstrak

*Makalah ini membahas metode pemantau posisi dan arah gerak helikopter tanpa awak dengan bantuan layanan Google Maps API dan sensor GPS. Sinyal yang dihasilkan oleh GPS berupa data posisi dalam koordinat lintang dan bujur. Arah gerak helikopter diperoleh dari modul GPS dengan mengkalkulasi data riwayat posisi. Sinyal dari GPS selanjutnya dipancarkan oleh modem radio di helikopter dan diterima oleh modem radio di komputer. Data kemudian diolah oleh program aplikasi yang dibuat dengan Visual Basic 6. Data hasil olahan digunakan oleh program javascript untuk menampilkan posisi dan arah gerak helikopter ke dalam peta yang diperoleh dari layanan Google Maps API. Sebagai hasil akhir, pengguna dapat melihat posisi dan arah gerak helikopter pada browser seperti Mozilla Firefox, Google Chrome maupun browser lain yang mendukung HTML5. Kecepatan pembaharuan data posisi pada jaringan lokal dapat mencapai dua puluh data perdetik atau 20 Hz. Kemampuan ini sangat lebih dari cukup mengingat kemampuan akuisisi sensor GPS komersial yang paling mutakhir hanya berkisar antara 5-10 Hz. Dengan menggunakan metode ini data GPS dapat diproses tanpa ada data yang terlewat.*

*Kata kunci: Google Maps, posisi, arah, GPS, helikopter tanpa awak.*

## 1. Pendahuluan

Unmanned aerial Vehicle atau UAV merupakan suatu pesawat udara tanpa awak yang dapat bekerja secara otomatis. Bentuk fisik UAV dapat berupa pesawat terbang mini, helikopter mini maupun pesawat balon-helium [1]. UAV banyak digunakan untuk penelitian [1][2], pemetaan lahan [3] maupun pengawasan sistem irigasi [4].

Tidak seperti pesawat pada umumnya yang memiliki pilot dan kru pesawat untuk mengontrol dan mengawasi secara langsung kondisi pesawat, pada pesawat tanpa awak kondisi pesawat tidak dapat dikontrol secara langsung karena memang tidak memiliki kru pesawat. Proses kontrol pesawat sepenuhnya dilakukan oleh sistem autopilot dengan mengacu pada parameter-parameter yang telah ditentukan oleh pengguna sebelum terbang. Hal ini memiliki konsekuensi akan timbulnya masalah apabila UAV mengalami kerusakan atau tidak dapat melaksanakan fungsi sesuai perintah. Oleh karena

itu dibutuhkan suatu sistem pemantau yang memadai untuk memantau kinerja UAV.

Penelitian mengenai sistem pemantau UAV untuk misi penelitian telah dilakukan oleh Jean[1] dan Roglia[2]. Pada penelitian yang dilakukan oleh Jean[1], sistem pemantau digunakan untuk mengetahui posisi dan ketinggian pesawat balon-helium sekaligus untuk memvisualisasikan hasil pembacaan sensor-sensor penelitian. Pada penelitian tersebut komunikasi data antara pesawat balon-helium dan stasiun penerima dilakukan melalui modem radio FSK dan modem GSM. Modem GSM digunakan untuk mentransfer data pada fase naik sedangkan modem radio FSK digunakan pada fase pesawat balon-helium turun. Sistem pemantau dibuat berbasis layanan web, sehingga dapat diakses oleh masyarakat umum. Selain itu, layanan Google Maps API digunakan untuk memetakan posisi dari pesawat balon-helium. Pada penelitian tersebut berhasil dilakukan pemantauan posisi dan data sensor secara realtime dengan menempuh jarak sejauh 150 Km dan waktu 3 jam.

Pada penelitian yang dilakukan oleh Zongjean[3] dan Chao[4], sistem pemantauan UAV digunakan untuk mengamati pergerakan UAV dalam melakukan pengambilan data gambar. Pada penelitian yang dilakukan oleh Zongjean[3], UAV memiliki 4 buah kamera untuk mengambil gambar udara. Pemrosesan pemetaan dilakukan dengan metode triangulasi dengan koreksi posisi GPS. Dengan demikian didapatkan hasil rekonstruksi topologi antar gambar yang lebih akurat.

Dari pengamatan beberapa penelitian tersebut dapat disimpulkan bahwa sistem pemantauan memegang peranan penting dalam sistem UAV. Baik untuk pengamatan pergerakan balon-helium dalam mengambil data sensor maupun pesawat mini dalam mengambil data gambar. Dengan adanya sistem pemantau dapat memungkinkan untuk dilakukan pengkoreksian data. Selain itu juga memperkuat validitas data yang dihasilkan oleh UAV karena adanya data riwayat saat pengamatan dilakukan. Waktu pembaharuan data dari UAV ke stasiun pemantau juga sangat penting agar tercapai kondisi realtime. Dengan demikian data UAV yang diterima oleh stasiun pemantau memang merupakan data terkini dari UAV.

Makalah ini membahas tentang metode pemantauan UAV yang memiliki waktu tanggap cepat, sehingga dapat memantau UAV dalam kondisi realtime dan lebih akurat. Metode yang dikemukakan pada makalah ini terbatas untuk pemantauan data posisi dan arah hadap UAV. UAV yang digunakan dalam penelitian ini merupakan jenis helikopter mini tanpa awak. Secara umum metode yang dikemukakan dalam makalah ini bekerja dengan memanfaatkan media penyimpanan file data yang tersimpan di server. Sedangkan data posisi dan arah dari helikopter tanpa awak dikirim oleh helikopter melalui modem radio dan diterima oleh komputer. Untuk menampilkan data hasil pemantauan digunakan media yang sama dengan penelitian yang dilakukan oleh Jean[1], yaitu menggunakan layanan peta dari Google Maps API, dengan tambahan fasilitas yang lebih interaktif yaitu kemampuan mengubah arah ikon secara realtime.

Tujuan utama pembahasan makalah ini adalah untuk mengaplikasikan Google Maps API dalam menampilkan data hasil pemantauan posisi dan arah helikopter. Masalah yang akan diselesaikan dalam makalah ini meliputi bagaimana data diambil dari sensor GPS receiver, ditransmisikan, dan ditampilkan pada browser secara realtime dan interaktif dengan kemampuan merubah arah ikon. Analisis kemampuan difokuskan pada besaran simpangan rata-rata data GPS receiver, serta tanggap realtime dari sistem dalam mengolah data dan menampilkan posisi dan arah gerak helikopter mini tanpa awak pada browser.

## 2. Perancangan

### 2.1 Akuisisi Data dari GPS Receiver dan Sistem Komunikasi Data dengan Ground Station

Penelitian ini menggunakan modul GPS Receiver tipe D2523T[5]. Modul ini menggunakan interface 3.3v UART dengan standar data *National Marine Electronics*

*Association* atau NMEA. Agar dapat terhubung ke modem radio dengan *interfacing* UART RS232, diperlukan konverter 3.3v-5v *bidirectional*. Dengan antenna *helic* (terlihat pada Gambar 1), modul GPS receiver ini sangat peka dan mampu menerima sinyal dari maksimal 12 satelit sekaligus.

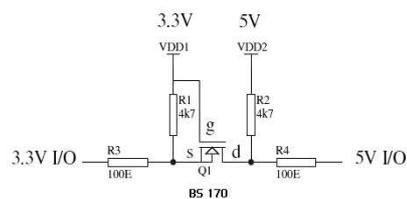


Gambar 1 Modul GPS Receiver D2523T

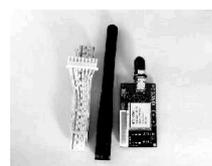
Data 3.3v dikonversi menjadi 5v oleh rangkaian MOSFET BS-170 sebagai *bidirectional logic converter* seperti terlihat pada gambar 2. Karena konverter ini bersifat dua arah, maka digunakan rangkaian yang sama untuk jalur TX dan RX. Setelah diperoleh level tegangan 5v, digunakan MAX 232 untuk mengubah level tegangan TTL 5v menjadi RS 232.

Output dari MAX 232 yang merupakan data yang berasal dari GPS selanjutnya diberikan kepada modem radio tipe YS-1100U[6] (terlihat pada Gambar 3).

Di sisi komputer pengendali dan pemantau, digunakan modem yang sama, yaitu YS-1100U. Pada helikopter digunakan modem radio dengan *interface* serial dan pada komputer pengendali dan pemantau digunakan modem radio dengan *interface* serial atau USB.



Gambar 2 Bidirectional logic converter



(a)



(b)

Gambar 3 Modem Radio YS-1100U Serial (b) USB

Selain data posisi (derajat lintang dan bujur) dan arah gerak dari helikopter, dibuat pula sistem pemantau data lain yang diperoleh dari sensor GPS, yaitu ketinggian di atas permukaan air laut, kecepatan gerak dan jumlah satelit yang diterima oleh sensor. Selain mamantau kondisi helikopter, pengguna juga dapat memberikan perintah gerak, misalnya maju, mundur, ke kiri, atau ke kanan kepada helikopter. Secara umum, sistem

komunikasi helikopter dengan komputer pengendali dan pemantau dapat terlihat pada gambar 4.



**Gambar 4 Sistem komunikasi dua arah antara pengendali/monitoring dan helikopter**

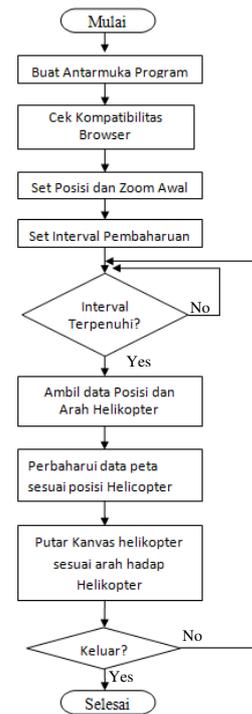
## 2.2 Google Maps API sebagai Penampil Helikopter Tanpa Awak

Dalam menampilkan data hasil pemantauan UAV, terdapat beberapa jenis metode. Pada penelitian yang dilakukan oleh Jean[1] layanan Google Maps API digunakan untuk pengambilan data peta. Pada penelitian yang dilakukan oleh Roglia[2] suatu layanan berbasis web yang disebut *Service Oriented Architecture* atau SOA digunakan untuk memantau kondisi UAV. Google Maps API merupakan layanan gratis yang disediakan oleh Google untuk memperoleh data peta, sedangkan SOA merupakan arsitektur berbasis layanan yang memungkinkan suatu sistem dapat saling berinteraksi dengan yang sistem yang lain. SOA cocok untuk diaplikasikan pada sistem yang melibatkan beberapa UAV sekaligus seperti penelitian Roglia[2].

Pada penelitian ini, digunakan metode yang hampir sama dengan Jean[1]. Hal yang membedakan dari penelitian Jean[1] adalah untuk menampilkan posisi dan arah helikopter diperlukan tampilan yang lebih interaktif daripada tampilan untuk memantau pesawat balon-helium. Aplikasi yang dibuat pada penelitian ini tidak dapat hanya menampilkan tanda balon sebagaimana yang dilakukan oleh Jean[1]. Untuk menampilkan gambar helikopter diperlukan gambar ikon helikopter dan bantuan *Canvas* dari HTML5.

*Canvas* merupakan salah satu jenis elemen pada HTML5 yang digunakan untuk menggambar grafik atau objek gambar. *Canvas* dapat digunakan untuk memutar gambar yang diletakkan di atasnya sehingga dapat digunakan untuk menggambarkan arah helikopter pada sistem monitoring. Selain itu untuk menempatkan *canvas* diatas Google Map digunakan elemen ekstensi dari Google Map API yang disebut *Elabel*. Prosedur dalam penampilan posisi dan arah helikopter ditunjukkan oleh Gambar 5.

Data hasil olahan program di komputer digunakan untuk memperbaharui data posisi dan arah helikopter yang tersimpan di *server*. File yang tersimpan di *server* berisi tiga komponen data, yaitu posisi bujur dan lintang dari helikopter serta arah gerak helikopter. Data bujur bernilai  $-180^{\circ}$  sampai  $180^{\circ}$ , sedangkan data lintang bernilai  $-90^{\circ}$  sampai  $90^{\circ}$ . Arah gerak helikopter bernilai  $0^{\circ}$  -  $359,99^{\circ}$ , yang mana nilai  $0^{\circ}$  menyatakan arah utara.



**Gambar 5 Diagram alir sistem penampilan posisi dan arah hadap helikopter tanpa awak**

Langkah pertama pemetaan posisi oleh sistem pemantau dengan google map adalah membuat antarmuka untuk pengendalian peta. Antarmuka ini terdiri dari dua jenis, yaitu antarmuka standar dari Google API dan antarmuka buatan sendiri untuk menampilkan posisi terakhir dari helikopter. Antarmuka standar dari Google API dalah antarmuka Zoom dan Navigation serta menu untuk pemilihan jenis peta. Langkah kedua adalah mengecek kompatibilitas browser pemakai apakah sudah mendukung fitur kanvas dari HTML5. Jika belum maka akan dipakai marker standar dari Google API, namun apabila sudah mendukung maka nantinya canvas dapat digunakan untuk menampilkan dan memutar gambar helikopter sesuai arah hadap helikopter.

Tahap berikutnya adalah penentuan posisi awal helikopter dan pengaturan interval waktu pembaharuan. Posisi awal yang dipilih adalah posisi bujur dan lintang dari tempat Puslit Informatika LIPI karena pengujian helikopter berada di tempat yang hampir sama. Penentuan ini bertujuan agar pengamatan mudah dilakukan tanpa harus mencari-cari posisi awal dari helikopter. Interval waktu yang dipilih merupakan variabel pengamatan pada pengujian yang dilakukan pada bagian pengujian. Bagian utama program adalah iterasi untuk pembacaan data posisi dan arah gerak dari file data serta pengambilan data peta yang sesuai dari layanan Google Map.

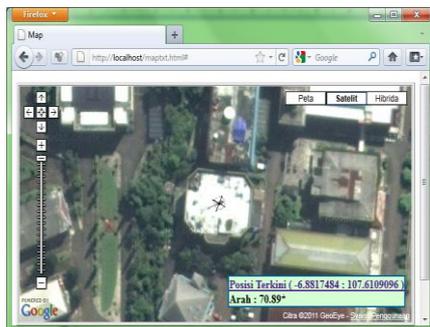
Pembacaan file data dilakukan secara tidak langsung melalui program readtxt.php. Dengan memakai program php, isi file text terlebih dahulu dibaca baru kemudian ditampilkan menjadi suatu halaman website. Isi dari halaman website inilah yang dibaca oleh program javascript. Cara ini ditempuh untuk menghindari adanya chache dari browser terhadap suatu file yang telah dibacanya, apabila pembacaan dilakukan secara langsung. Dengan adanya chacing, maka isi file yang bersangkutan akan tetap sama walaupun file sumber telah berubah isinya karena browser menyimpan file tersebut kedalam buffer.

Setelah data posisi dan arah hadap diperoleh, langkah selanjutnya adalah memindahkan kanvas helikopter sesuai koordinat bujur dan lintang serta memutar kanvas sesuai arah hadap dari Helikopter. Proses ini berulang-ulang selama helikopter tanpa awak beroperasi. Hasil akhir sistem pemantau ditunjukkan oleh Gambar 6.

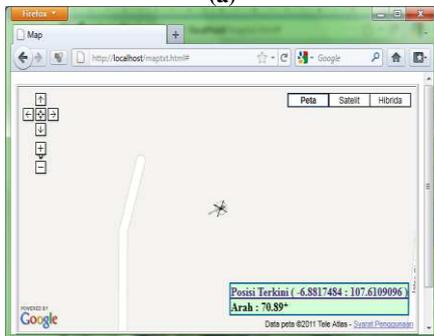
### 3. Pengujian

#### 3.1 Uji coba simpangan data GPS

Meskipun modul GPS receiver ditempatkan pada posisi yang tetap, data yang diperoleh akan selalu berubah. Hal ini disebabkan oleh perbedaan perhitungan oleh engine GPS receiver, jumlah serta lokasi satelit yang digunakan dalam perhitungan posisi. Dalam penelitian ini digunakan lima buah lokasi percobaan seperti yang terlihat pada Tabel 1.



(a)



(b)

**Gambar 6** Tampilan sistem penampil posisi dan arah helikopter tanpa awak dalam mode pilihan (a) satelit dan (b) peta

**Tabel 1** Nilai rata-rata data pada jumlah satelit yang diterima mencapai maksimal di masing-masing lokasi

Lokasi	Lintang	Bujur	Tinggi	Jumlah satelit GPS
1	-6.881525	107.610990	850.92	8
2	-6.881514	107.610885	842.39	12
3	-6.881537	107.610936	852.31	12
4	-6.881511	107.610923	850.39	12
5	-6.881307	107.611024	842.31	12

Pada kondisi dengan jumlah satelit yang lebih sedikit, data GPS menyimpang dengan lebih besar. Sebaliknya pada kondisi dengan jumlah satelit yang lebih banyak (maksimal 12 satelit untuk modul GPS D2523T), data GPS menyimpang lebih kecil. Karena data GPS semakin akurat dengan semakin banyaknya jumlah satelit yang diterima, maka simpangan rata-rata data menggambarkan tingkat akurasi GPS receiver tersebut. Simpangan rata-rata dihitung berdasarkan selisih data dengan nilai rata-rata pada kondisi satelit terbanyak pada lokasi tersebut. Nilai rata-rata ini (terlihat pada Tabel 1) diasumsikan sebagai nilai yang paling akurat yang merepresentasikan data GPS pada lokasi tersebut.

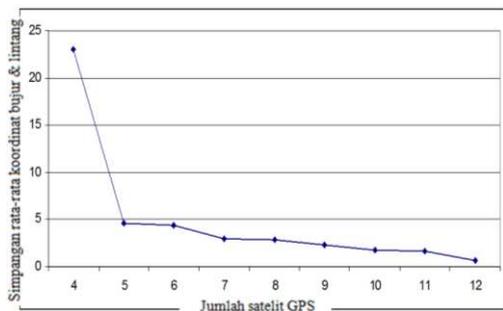
Dengan patokan data seperti pada Tabel 1, selanjutnya dihitung nilai simpangan rata-rata (harga mutlak) dari masing-masing lokasi dengan kondisi jumlah satelit tertentu seperti terlihat pada tabel 2. Data yang kosong menunjukkan bahwa selama percobaan, di lokasi tersebut tidak pernah terdapat jumlah satelit tertentu. Pengambilan data dilakukan minimal 20 kali untuk masing-masing jumlah satelit. Dengan konversi 1 derajat adalah sebesar 111.000 km, maka diperoleh tabel simpangan rata-rata data GPS (lintang dan bujur) seperti terlihat pada Tabel 2.

**Tabel 2** Simpangan rata-rata data lintang dan bujur dengan nilai rata-rata data pada saat GPS menerima satelit terbanyak

Lokasi	Simpangan rata-rata berdasarkan jumlah Satelit (x10 <sup>-5</sup> ) dalam derajat								
	4	5	6	7	8	9	10	11	12
1	-	3,95	4,57	2,43	1,34	-	-	-	-
2	-	-	-	2,20	-	-	1,55	1,21	0,847
3	20,7	4,29	3,40	3,45	3,85	3,71	1,50	1,65	0,75
4	-	-	-	-	-	0,534	0,987	0,888	0,273
5	-	-	-	-	-	-	2,21	2,08	0,306
Rata-rata (derajat)	20,7	4,12	3,99	2,69	2,60	2,12	1,56	1,46	0,544
Rata-rata (meter)	22,97	4,57	4,42	2,99	2,89	2,35	1,73	1,62	0,60

Nilai rata-rata simpangan dalam meter dapat dilihat pada Grafik 1. Dari data ini diperoleh bahwa simpangan rata-rata terkecil untuk data lintang dan bujur diperoleh pada saat GPS memperoleh 12 satelit (jumlah satelit terbanyak yang dapat diterima oleh modul GPS ini), yaitu sebesar 0,60 meter.

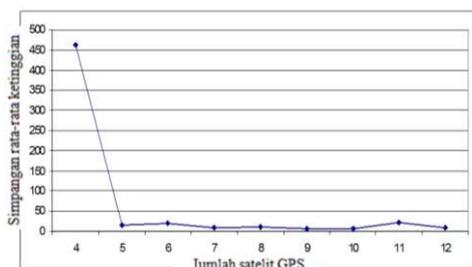
Selain menghasilkan data derajat lintang dan bujur, GPS juga menghasilkan data tinggi lokasi yang diukur di atas permukaan air laut (DPL). Simpangan rata-rata data tinggi terhadap nilai rata-rata pada saat GPS menerima satelit terbanyak dapat dilihat pada Tabel 3. Pengambilan data minimal 20 kali untuk masing-masing jumlah satelit. Dari percobaan ini diperoleh data pada saat modul GPS memperoleh jumlah satelit terbanyak, simpangan data tinggi DPL adalah sebesar 7,92 meter.



Gambar 7 Simpangan rata-rata data lintang dan bujur GPS berdasarkan jumlah satelit yang diterima.

Tabel 3 Simpangan rata-rata data tinggi dengan nilai rata-rata data pada saat GPS menerima satelit terbanyak

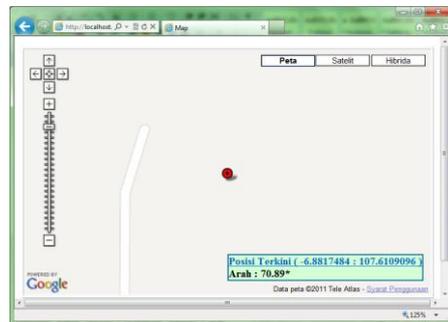
Lokasi	Simpangan rata-rata berdasarkan jumlah Satelit dalam meter								
	4	5	6	7	8	9	10	11	12
1	-	4,84	1,29	4,19	1,72	-	-	-	-
2	-	-	-	1,54	-	-	6,52	5,20	4,76
3	462,7	26,09	18,16	18,77	18,29	13,67	1,64	1,56	0,62
4	-	-	-	-	-	0,97	1,14	1,66	1,00
5							13,33	13,12	1,54
Rata-rata (meter)	462,7	15,47	19,45	8,17	10,00	7,32	5,66	21,54	7,92



Gambar 8 Simpangan rata-rata data tinggi DPL GPS berdasarkan jumlah satelit yang diterima

### 3.2 Uji coba Penampilan data GPS dengan Google Maps API

Sistem pemantau posisi dan arah helikopter tanpa awak bekerja dengan memanfaatkan fitur elemen *canvas* pada HTML5. Oleh karena itu, sistem ini hanya mampu menampilkan tampilan helikopter tanpa awak pada *browser* yang mendukung penuh fitur elemen *canvas* pada HTML5. Apabila *browser* tidak mendukung sepenuhnya fitur elemen *canvas* HTML5 seperti *browser* Internet Explorer 9, maka akan ditampilkan lingkaran merah kecil yang menandakan posisi helikopter sebagaimana ditunjukkan oleh gambar 7. *Browser* yang mendukung fitur elemen *canvas* pada HTML5 antara lain *browser* versi terkini dari Google Chrome, Mozilla Firefox, Opera dan Safari, sedangkan Internet Explorer versi 9 tidak mendukung penuh fitur elemen *canvas*.



Gambar 9 Tampilan *browser* IE 9 yang tidak mendukung sepenuhnya elemen *canvas* HTML5

Salah satu kemampuan dari sistem pemantau UAV yang paling penting adalah kemampuan untuk menampilkan kondisi UAV secara *realtime*. Oleh karena itu perlu dilakukan pengujian terhadap sistem pemantau posisi dan arah helikopter tanpa awak ini sehingga mampu mendukung transmisi data secara *realtime*.

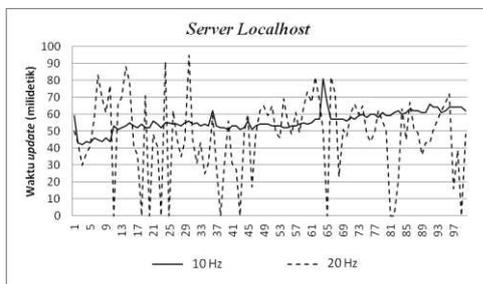
Pengujian pertama adalah menghitung waktu tanggap atau *time respon* dari *server* untuk melakukan pembaharuan data GPS. Pengujian dilakukan di empat *server* yang berbeda, yaitu *server* Localhost, *server* LAN, dan dua buah *server* eksternal. *Server* eksternal yang dipilih yaitu 000webhost.com dan freeweostingarea.com. Pengujian dilakukan dengan menghitung selisih waktu antara pembaharuan data dengan waktu penampilan data pada *browser* pengguna. Hasil yang diperoleh dari pengujian ini terlihat pada tabel 4.

Berdasarkan hasil tersebut waktu tanggap pembaharuan yang diterima oleh *server* Localhost dan *server* LAN sangat cepat, yaitu kurang dari 1 detik. Sedangkan untuk *server* eksternal waktu tanggap pembaharuan berkisar antara 2 sampai 3 detik. Hal tersebut dikarenakan letak *server* yang berada di luar negeri, sehingga membutuhkan waktu pengiriman yang lama. Dengan demikian, *server* eksternal tidak dapat digunakan sebagai penampil posisi dan arah gerak helikopter tanpa awak karena tidak memenuhi minimal waktu pembaharuan data GPS sebesar 1 data per detik.

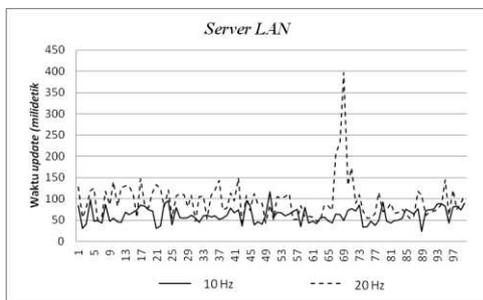
**Tabel 4** Tabel hasil pengujian waktu tanggap pembaharuan data

Server	Waktu Tanggap
Localhost	< 1 detik
LAN	< 1 detik
Eksternal 1 (000webhost.com)	> 3 detik
Eksternal 2 (freewebhostingarea.com)	> 2 detik

Pengujian selanjutnya adalah menghitung laju waktu *update* yang dapat dilakukan oleh aplikasi ini. Pengujian kali ini bertujuan untuk mengetest kemampuan sistem penampil dalam menangani data dalam frekuensi tinggi sampai 20 data per detik. Karena kemampuan sensor GPS hanya mampu menghasilkan data dalam frekuensi 1 Hz maka pengujian dilakukan dengan membuat program aplikasi yang mensimulasikan data GPS dalam frekuensi diatas 1 Hz. Dikarenakan kemampuan *server* eksternal tidak memadai untuk menampilkan data maka pada pengujian kali ini tidak ikut disertakan. Pengujian dilakukan untuk data sebanyak 100 kali. Masing-masing *server* baik *server* Localhost maupun *server* LAN diuji coba untuk kecepatan 20 Hz atau *interval* 50 milidetik dan 10 Hz atau *interval* 100 milidetik. Hasil pengujian dapat ditampilkan dalam bentuk grafik. Grafik 3 merupakan hasil pengujian untuk *server* Localhost sedangkan grafik 4 merupakan hasil pengujian untuk *server* LAN. Hasil rata-rata data pengujian ditampilkan pada tabel 5.



**Gambar 10** Hasil pengujian untuk *server* Localhost



**Gambar 11** Hasil pengujian untuk *server* LAN

Sesuai hasil yang terdapat pada tabel 5, pada *server* Localhost diperoleh waktu *update* sebesar 53,27 milidetik untuk kecepatan 20 Hz dan 55,66 milidetik untuk kecepatan *update* 10 Hz.

Sebagai hasilnya tampilan pada *browser* dapat berjalan hampir secara kontinyu karena waktu transmisi data dapat dilakukan dengan cepat. Hal tersebut disebabkan *server* berada pada satu komputer dengan komputer penghasil data GPS.

Kelemahan pada *server* Localhost adalah terjadinya data yang hilang karena keterbatasan komputasi di komputer, yang menyebabkan terjadinya konflik saat pembacaan *file* antara program penerima data GPS dengan *server* Localhost. Rasio keberhasilan pada saat kecepatan 20 Hz adalah 89 %. Hal ini menyebabkan gambar helikopter pada tampilan berkedip dalam sepersekian detik.

Walau demikian hal ini hampir tidak terasa karena penampilan dilakukan dalam kecepatan 20 Hz atau dengan kata lain rata-rata lama waktu tidak tampilnya helikopter pada menu tampilan per detik adalah :

$$= \% \text{ kegagalan} \times \text{jumlah data} \times \text{periode}$$

$$= 11 \% \times 20 \times 50 \text{ milidetik}$$

$$= 110 \text{ milidetik}$$

**Tabel 5** Tabel perbandingan waktu pembaharuan

Server	Waktu <i>update</i> (milidetik)	Rasio keberhasilan penampilan	Hasil tampilan
Localhost 20 Hz	53,27	89 %	Sedikit patah-patah
Localhost 10 Hz	55,66	100 %	Kontinyu
Server LAN 20 Hz	95,89	100 %	Patah-patah
Server LAN 10 Hz	62,18	100 %	Hampir kontinyu

Sedangkan pada kecepatan 10 Hz program penampil dapat berjalan dengan baik. Waktu *update* yang berkisar 55,66 milidetik menyebabkan tampilan dapat berjalan secara kontinyu karena waktu untuk penampilan sendiri di cuplik dalam *interval* 100 milidetik. Dengan *interval* penampilan sebesar 100 milidetik mengakibatkan beban komputasi komputer menjadi lebih ringan sehingga kejadian hilangnya data menjadi turun drastis menjadi 0 %.

Pada pengujian pada *server* LAN, tidak terjadi adanya data yang hilang karena spesifikasi *server* yang tinggi. Pada pengujian dengan kecepatan 20 Hz, program penampil mengalami patah-patah atau *lag*. *Lag* disebabkan oleh proses waktu *update* data yang jauh lebih besar daripada *interval* penampilan data. Kejadian tersebut terjadi karena adanya *delay* transmisi data dari dan ke *server*. Sebagai akibatnya gerakan helikopter seperti meloncat-loncat dari satu titik ke titik lain. Untuk pengujian pada kecepatan 10 Hz, tampilan dapat berjalan agak lancar. Pada kecepatan ini masih terjadi sedikit lag pada tampilan yang disebabkan oleh *delay* transmisi data dari *server* LAN ke komputer penampil.

#### 4. Kesimpulan

Dari penelitian ini dapat disimpulkan :

1. Google Maps API sesuai untuk menampilkan posisi UAV karena kemampuan kecepatan *update* yang tinggi (maksimal 20 data per detik) dan keakuratan dalam menampilkan posisi UAV (resolusi  $10^{-7}$  derajat bujur dan lintang).
2. Kecepatan update sistem dibatasi oleh kemampuan komputer yang dipergunakan sebagai *server*, terutama waktu baca dan tulis ke media penyimpanan, dan kualitas jaringan internet.
3. Penelitian dapat dikembangkan dengan arsitektur database sebagai tempat data yang akan ditulis dan ditampilkan.
4. Kecepatan sensor GPS yang digunakan dalam penelitian ini sebesar 1 Hz perlu ditingkatkan hingga 10 Hz agar *update* data dapat lebih cepat.

#### 5. Daftar Pustaka

- [1] Sebastien Jean, Kiev Gamma, Didier Donsez, Andre Lagreze, *Toward a Monitoring System for High Altitude Objects*. Proceedings of the 6th International Conference on Mobile Technology, Application & Systems, Nice France, 2009
- [2] E. Roglia and R. Meo, *A soa-based system for territory monitoring*. Geospatial Web Services: Advances in Information Interoperability, <http://www.di.unito.it/~meo/Pubblicazioni/A-SOA-Based-System-For-Territory-Monitoring.zip>, diakses tanggal 24 Oktober 2011.
- [3] LIN Zongjian, *UAV For Mapping – Low Altitude Photogrammetric Survey*. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Netherlands, 2008.
- [4] Haiyang Chao, Marc Baumann, Austin Jensen, YangQuan Chen, Yongcan Cao, Wei Ren, and Mac Mc- Kee, *Band-reconfigurable multi-UAV-based cooperative remote sensing for real-time water management and distributed irrigation control*, Proc. of the IFAC World Congress, Seoul, Korea, 2008.
- [5] Sparkfun.com, *50 Channel D2523T Helical GPS Receiver*, [www.sparkfun.com/products/9566](http://www.sparkfun.com/products/9566), diakses tanggal 24 Oktober 2011.
- [6] ShenZhen YiShi Electronic Technology Development Co. Ltd, *YS-1100U RF DataTransceiver*, [http://www.compel.ru/images/catalog/1188/YS-1100U\\_manual.pdf](http://www.compel.ru/images/catalog/1188/YS-1100U_manual.pdf), diakses tanggal 24 Oktober 2011.