

# PENERAPAN LOW PASS FILTER DAN FINITE IMPULSE RESPONSE DALAM SISTEM PEMANTAUAN POSISI PEJALAN KAKI MENGUNAKAN GPS

B Devina Puspasari<sup>\*)</sup>, Iwan Setiawan, and Darjat

Jurusan Teknik Elektro, Fakultas Teknik, Universitas Diponegoro,  
Jln. Prof. Sudharto, Tembalang, Semarang, Indonesia

*E-mail : puspasari.devina@yahoo.com*

## Abstrak

Sistem pemantauan posisi merupakan satu hal yang penting digunakan dalam sejumlah bidang aplikasi seperti militer, keamanan dan layanan darurat. Salah satu contoh penggunaannya adalah memberikan informasi posisi pejalan kaki yang tersesat. Sampai saat ini, salah satu solusi untuk memenuhi kebutuhan positioning tersebut umumnya dengan memanfaatkan teknologi GNSS (Global Navigation Satelity Systems). Hal ini seperti GPS-Global Positioning Systems (Amerika), Glonass (Rusia) dan Galileo (Eropa). Namun, pemakaian GPS rawan tak stabil karena adanya noise yang disebabkan banyaknya faktor yang mempengaruhi seperti multipath. Tugas akhir ini dilakukan dengan perancangan sistem pemantauan posisi pejalan kaki dengan menggunakan GPS. Untuk mengolah data, digunakan dua jenis filter yang akan diperbandingkan kinerjanya. Metode dua jenis filter tersebut adalah low pass filter dan finite impluse response. Untuk Central Processing Unitnya menggunakan mikrokontroller ATMEGA164 dan menggunakan bahasa C. GUI dimanfaatkan untuk memonitoring trajektori dari pejalan kaki.

*Kata Kunci : Navigasi, Pejalan kaki, GPS, LPS, FIR, GUI monitoring, trajektori*

## Abstract

The position monitoring is one important thing that is used in a number of areas such as military applications, security and emergency services. One example of its use is to provide the information of pedestrian's position that in lost. Until now, one of the solutions to meet the needs of the positioning generally by utilizing GNSS (Global Navigation Satelity Systems). This is like GPS-Global Positioning Systems (USA), Glonass (Russia) and Galileo (Europe). However, the use of the GPS receiver is prone to be unstable because of noise that caused by various things like multipath. The final task was to design a pedestrian navigation system using GPS. To process the data, use two types of filters that will be compared to performance. The two types of filter method, the low pass filter and moving average that be implemented with finite impulse response. For central processing unit was using ATMEGA164 microcontroller and using C language GUI used to monitor the trajectory of the pedestrian.

*Keywords: Navigation, Pedestrians, GPS, LPS, FIR, GUI monitoring, trajectory*

## 1. Pendahuluan

Fungsi navigasi antara lain menyediakan pemantauan posisi dan arah. Sistem pemantauan posisi ini merupakan satu hal yang penting digunakan dalam sejumlah bidang aplikasi seperti militer, keamanan dan layanan darurat. Salah satu contoh penggunaannya adalah memberikan informasi posisi pejalan kaki yang tersesat. Sampai saat ini, salah satu solusi untuk memenuhi kebutuhan positioning tersebut umumnya dengan memanfaatkan teknologi GNSS (Global Navigation Satelity Systems) yaitu seperti GPS-Global Positioning Systems (Amerika), Glonass (Rusia) dan Galileo (Eropa).

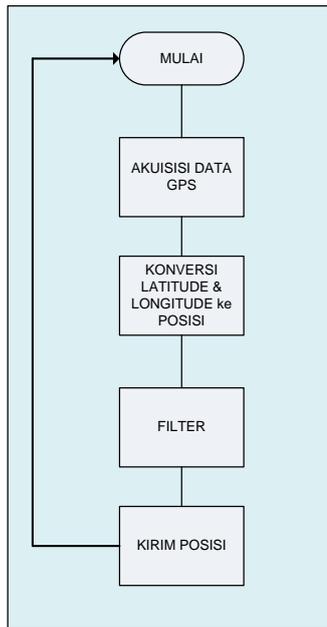
Kendala perancangan sistem pemantauan posisi menggunakan GPS ini adalah *noise* yang menyebabkan

pembacaan trajektori pada antar muka komputer menjadi bias. Hal ini bisa dijelaskan saat seorang diam, yang nampak pada grafik seharusnya satu titik namun karena ada gangguan pada penerimaan yang nampak pada grafik adalah beberapa titik sehingga penerima tidak dapat menentukan posisi navigator dengan tepat. Untuk meminimalkan gangguan / *noise* tersebut digunakan *low pass filter* dan juga metode *finite impulse response*. Kedua metode tersebut diperbandingkan kinerjanya untuk mengetahui mana yang lebih baik.

## 2. Metode

Perancangan sistem pemantauan posisi pejalan kaki menggunakan sensor GPS ini pada dasarnya terbagi





Gambar 5. Diagram alir perancangan perangkat lunak.

Sesuai dengan gambar 6 dapat dilihat bahwa pada diagram alir *firmware* terbagi menjadi beberapa blok proses utama, yaitu blok akuisisi data GPS, konversi *longitude* dan *latitude*, penerapan filter, dan mengirim posisi ke program antar muka.

### 2.2.1 Perancangan Perangkat Lunak pada CodeVision AVR

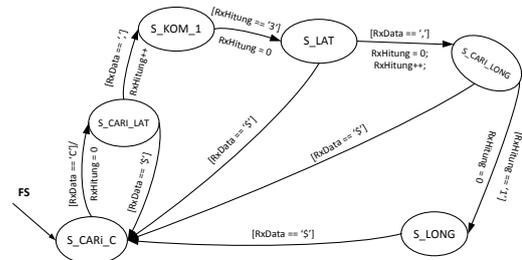
Mikrokontroler ATmega 164 dapat diprogram menggunakan CodeVision AVR. Dalam pembuatan program menggunakan CodeVision AVR, digunakan CodeWizard.

Pertama, dipilih *chip* ATmega164 karena akan digunakan mikrokontroler ATmega164 dalam pemrograman dengan nilai *clock* sebesar 11,59 MHz. Selanjutnya pengaturan untuk inisialisasi USART1 yang digunakan untuk mengakses data yang dikirim oleh GPS dimana *baudrate* yang digunakan sebesar 4800 bps. Selanjutnya pengaturan untuk inisialisasi USART0 yang digunakan untuk berkomunikasi dengan komputer dimana *baudrate* yang digunakan sebesar 9600. Inisialisasi LCD, di mana menggunakan PORT B untuk terhubung pada mikrokontroler. Dengan adanya CodeWizard, maka pemrograman akan lebih mudah karena tidak perlu mengatur secara manual register yang digunakan. CodeVision AVR telah siap digunakan untuk membuat program.

#### A. Akuisisi Data GPS

Dalam menentukan posisi koordinat lintang dan bujur maka harus terlebih dahulu ekstraksi terhadap data yang

diterima oleh GPS Receiver. Data yang diterima merupakan kumpulan karakter dengan format NMEA 0183 yang didalamnya mengandung kalimat GPRMC yang berisi data posisi lintang dan bujur. Gambar 6 menjelaskan diagram state pengambilan data posisi lintang dan bujur dari kalimat NMEA 0183.



Gambar 6. Diagram State Pembacaan Data GPS

Pada program akses data GPS digunakan fasilitas interrupt dalam menerima data serial yaitu interrupt [USART1\_RXC] void usart1\_rx\_isr(void) sehingga apabila ada karakter yang diterima oleh mikrokontroler, maka mikrokontroler akan langsung mengeksekusi program yang berkaitan dengan karakter yang diterima. Karakter C berfungsi sebagai penanda bahwa kalimat GPRMC telah diterima oleh komputer sehingga perhitungan karakter selanjutnya akan dimulai. Karakter , (koma) berfungsi penanda setiap parameter yang terdapat pada kalimat GPRMC antara lain kecepatan, garis lintang, dan garis bujur. Karakter \$ digunakan sebagai penanda bahwa eksekusi program dimulai dari awal lagi dengan menunggu datangnya karakter C.

#### B. Konversi Latitude dan Longitude ke Posisi

Dalam perancangan perangkat lunak ini, untuk mengkonversi Latitude dan Longitude yang diterima sensor GPS menjadi posisi adalah melalui listing program sebagai berikut :

```

switch (state_phase)
{
    case INIT:
        if ((timer_init >= 215)
            && (statusGPS == 'A'))
        {
            hilangkan_derajat_lintang();
            hilangkan_derajat_bujur();
            LongitudeOrigin
            = ((atof(LongitudeStrSementara) -
              atof(LongitudeStrBelakang)) / 100) +
              (atof(LongitudeStrBelakang) / 60);
            LatitudeOrigin
            = ((atof(LatitudeStrSementara) -

```

```

atof(LatitudeStrBelakang))/100) +
(atof(LatitudeStrBelakang)/60);
    if
((LongitudeOrigin>=batas_Longitude[0])
&&(LongitudeOrigin<=batas_Longitude[1]
)&&(LatitudeOrigin>=batas_Latitude[0])
&&(LatitudeOrigin<=batas_Latitude[1]))
    {
        flag_update_GPS=0;
        timer_GPS=0;
        state_phase=UPDATE_GPS;
    }
    timer_init++;
break;

```

### C. Penerapan Filter

Seperti yang sudah dibahas dalam laporan bab sebelumnya, bahwa dalam tugas akhir ini menerapkan dua jenis filter yang akan dibandingkan performanya yakni filter FIR (*Finite Impulse Response*) dan LPF (*Low Pass Filter*). Penerapan kedua filter tersebut secara digital sehingga ditambahkan melalui listing pada CodeVision AVR. Adapun listing program adalah sebagai berikut :

```

void filter(unsigned char jenis)
{
    float Tf=5,T=1,alpha=Tf/(Tf+T);
//dalam detik
    if (jenis==FIR)
    {
        filtered_posisiX[0]=filtered_posisiX[1];
        filtered_posisiX[1]=filtered_posisiX[2];
        filtered_posisiX[2]=filtered_posisiX[3];
        filtered_posisiX[3]=filtered_posisiX[4];
        filtered_posisiX[4]=J_GPS_X;
        posisi_x=(filtered_posisiX[0]+filtered_posisiX[1]+filtered_posisiX[2]+filtered_posisiX[3]+filtered_posisiX[4])/5;
        filtered_posisiY[0]=filtered_posisiY[1];
        filtered_posisiY[1]=filtered_posisiY[2];
        filtered_posisiY[2]=filtered_posisiY[3];
        filtered_posisiY[3]=filtered_posisiY[4];
        filtered_posisiY[4]=J_GPS_Y;
    }
}

```

```

    posisi_y=(filtered_posisiY[0]+filtered_posisiY[1]+filtered_posisiY[2]+filtered_posisiY[3]+filtered_posisiY[4])/5;
    }
    else if (jenis==LOW_PASS)
    {
        filtered_posisiX[1]=alpha*filtered_posisiX[0]+(1-alpha)*J_GPS_X;
        filtered_posisiY[1]=alpha*filtered_posisiY[0]+(1-alpha)*J_GPS_Y;
        posisi_x=filtered_posisiX[1];
        posisi_y=filtered_posisiY[1];
    }
}

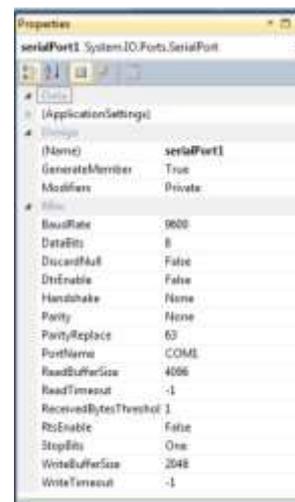
```

### 2.2.2 Perancangan Perangkat Lunak pada Visual C# 2010

Perangkat lunak Visual C# 2010 menyediakan fasilitas yang digunakan untuk perancangan sistem ini: komunikasi serial antara mikrokontroler sebagai pengirim dengan komputer sebagai penerima data serta menampilkan trajektori langkah dari pengguna. Data yang dikirimkan berupa posisi berdasarkan sumbu X dan sumbu Y.

#### A. Program Komunikasi antara Komputer dan Mikrokontroler

Agar komputer dengan mikrokontroler bisa saling berkomunikasi maka perlu dilakukan pengaturan terlebih dahulu dengan mengatur *properties* dari *serial port* seperti yang ditunjukkan pada Gambar 9.



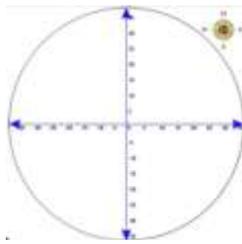
Gambar 7. Pengaturan *port* serial program C#.

Variabel yang digunakan untuk menyimpan data yang dikirim dari mikrokontroler berupa koordinat pada sumbu X dan Y adalah *string\_posisiX*, *string\_posisiY*. Karakter 'a', 'b' dan '~' merupakan penanda masing-

masing data yang diterima. Implementasi pada program adalah sebagai berikut.

### B. Program untuk Menampilkan Trajektori Langkah

Trajektori langkah pada perangkat lunak visual studio ditampilkan pada objek panel yang ditambahkan pada GUI visual studio. Objek panel ini yang nantinya akan digunakan sebagai kanvas menggambar trajektori langkah. Gambar 10 menunjukkan panel pada GUI visual studio yang akan digunakan:



Gambar 8. Panel pada GUI Visual Studio

Setelah menambahkan objek panel pada GUI kemudian perlu menambahkan komponen *brush* dan *path* agar dapat menggambar pada panel. Senarai program berikut digunakan untuk mendefinisikan komponen *brush* dan *path* :

```
System.Drawing.Point newPoint, oldPoint, startPoint;
GraphicsPath path= new GraphicsPath();
GraphicsPath pathReset = new GraphicsPath();
bool paint = false;
SolidBrush color;
```

Pada GUI, komponen *path* berfungsi untuk menyimpan nilai-nilai titik koordinat dari GPS dan komponen *brush* berfungsi untuk menampilkan trajektori dari langkah pengguna. Sehingga dengan memanfaatkan komponen *timer* pada visual studio dan event *timer\_tick* maka dapat dilakukan proses update gambar tiap detik. Senarai program berikut diletakkan dalam *event timer\_tick* dan digunakan untuk mengupdate gambar trajektori langkah:

```
dataA = Convert.ToDouble(string_posisiX) * 0.4 /
skala; //konversi ke pixel
dataB = Convert.ToDouble(string_posisiY) * 0.4 /
skala; //konversi ke pixel

lineX = (dataA) + (this.panel1.Width / 2);
//menyamakan frame trajektori dan GUI
lineY = (dataB*-1) + (this.panel1.Height / 2);
```

Data posisi X dan Y yang dikirim dari mikro harus diubah terlebih dahulu ke dalam bentuk pixel karena dalam GUI visual studio semua data koordinat hanya bisa dalam satuan pixel. Pengkonversian koordinat dari bentuk centimeter ke pixel dilakukan dengan mengkalikan nilai posisi (dalam centimeter) dengan konstanta 0,4. Nilai konstanta 0,4 (40 pixel/100 cm) didapat dikarenakan jarak antar tiap titik keterangan posisi adalah 40 pixel.

Sehingga diinginkan untuk skala 1 maka tiap 100 cm data posisi diubah menjadi 40 pixel.

Setelah dilakukan penyesuaian dengan titik origin panel maka koordinat posisi tersebut disimpan ke dalam komponen *path* dan kemudian digambar pada panel. Tahapan tersebut ditunjukkan dalam senarai program berikut:

```
path.AddEllipse((float)lineX, (float)lineY, 3, 3);
draw_path();
```

## 3. Hasil dan Analisa

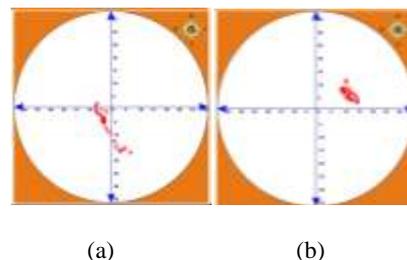
Pengujian secara garis besar meliputi pengujian data GPS tanpa filter, dengan filter FIR, dan dengan *Low Pass Filter*. Aksi yang dilakukan terhadap *handheld* sensor GPS antara lain: diam, lintasan lurus, lintasan bentuk kotak. Pengujian dilakukan di balkon lantai 3 gedung B, lapangan parkir jurusan Teknik Elektro Universitas Diponegoro. Setelah melakukan pengujian maka dapat diketahui apakah performa sistem telah sesuai dengan yang diharapkan sekaligus dapat diketahui filter mana yang lebih baik.

### 3.1 Pengujian Data GPS Tanpa Filter

Pengujian data GPS tanpa filter bertujuan untuk lebih mengetahui apakah sistem setidaknya dapat berjalan sesuai dengan syarat minimal performansi yang diharapkan. Syarat minimal tersebut adalah munculnya tanda / bintik pada antarmuka di mana sensor GPS berada terutama saat sensor diam.

#### 3.1.1 Posisi Diam

Seperti yang dijelaskan seperti di atas, data GPS tanpa filter pada posisi diam dapat menunjukkan apakah sistem alat telah dapat berfungsi sesuai dengan syarat minimal yakni munculnya tanda/bintik saat sensor pada posisi diam dengan demikian komunikasi antara sistem sensor dengan komputer berhasil berlangsung. Gambar 9 a dan b di bawah menunjukkan data GPS tanpa filter pada posisi diam.



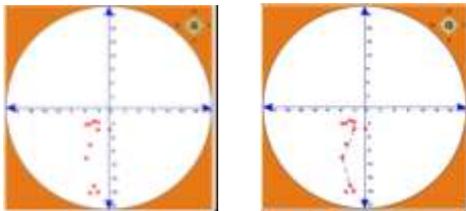
Gambar 9. Data GPS Tanpa Filter pada Posisi Diam

Dari gambar 9, nampak adanya sejumlah tanda/bintik yang merupakan representasi keberadaan sensor GPS.

Noise/gangguan masih terlihat berupa bintang/tanda masih banyak dan tersebar terutama pada gambar 9 (a). Pada gambar 9 (b) sudah menunjukkan perbaikan karena diletakkan menjauhi atap menuju ruang terbuka. Data ini tetap belum memenuhi kondisi ideal, di mana pada waktu sensor GPS pada posisi diam seharusnya hanya nampak 1 (satu) bintang/tanda saja.

### 3.1.2 Lintasan Lurus Sepanjang 15 meter

Selain posisi diam, pengujian dilakukan dengan membawa sensor GPS berjalan lurus sepanjang 15 meter. Hasil pengujian dapat dilihat pada gambar 10.

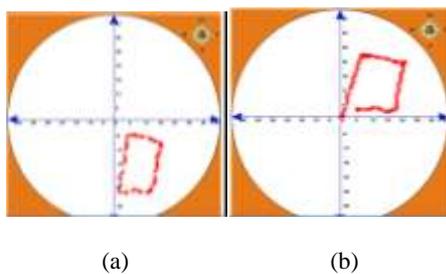


Gambar 10. Data GPS Tanpa Filter pada Posisi Lintasan Lurus 15 meter.

Pada gambar 10 (a), terlihat bahwa bintang/tanda terlihat kurang teratur. Pada gambar 10 (b), meski belum terlihat lurus, tren yang terlihat, bintang/tanda tersebut mengarah ke suatu bentuk lintasan vertikal meski belum lurus.

### 3.1.3 Lintasan Kotak

Pengujian ini merupakan aksi ketiga dalam pengujian data sensor GPS tanpa filter, yakni berjalan menurut lintasan kotak. Hasil pengujian dapat dilihat pada gambar 11.



Gambar 11. Data GPS Tanpa Filter pada Lintasan Kotak

Pada gambar 11 (a), pengujian dilakukan di lapangan voli. Terlihat bahwa tanda/bintang sudah terlihat membentuk lintasan kotak namun belum sempurna. Pada gambar 12(b), pengujian dilakukan di lapangan parkir 21,6 m x 12,6 m (y,x). Tanda/bintang sudah membentuk lintasan kotak, meski belum sempurna pula.

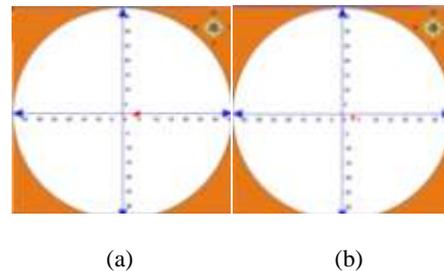
## 3.2 Pengujian dengan Filter FIR

Pengujian selanjutnya adalah pengujian data sensor GPS yang menerapkan Filter FIR (*Finite Impulse Response*).

Hal pertama yang ingin diketahui apakah penerapan FIR ini berhasil membuat performa sistem sensor berjalan lebih baik daripada saat sebelum penerapan filter apapun. Setelah berjalan semua, baru dapat diperbandingkan dengan data sensor GPS yang menerapkan *Low Pass Filter*. Pengujian dilakukan di lapangan parkir Jurusan Elektro Universitas Diponegoro yang memiliki panjang dan lebar : 21,6 m x 12,6 m (y,x).

### 3.2.1 Posisi Diam

Pengujian sensor GPS dengan filter FIR pada posisi diam dapat dilihat pada gambar 12.

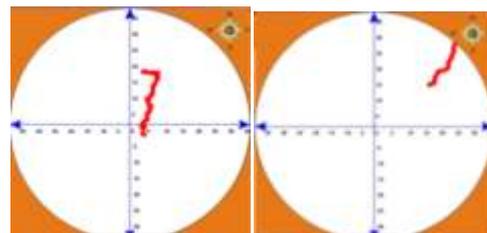


Gambar 12 Data GPS dengan Filter FIR Posisi Diam.

Dari gambar 12 (a) dan (b) dapat diketahui, bahwa tanda bintang/titik medekati bentuk sejaranya, yakni berupa satu bintang / titik saja. Hal ini suatu kemajuan apabila dibandingkan dengan sebelum memakai filter apapun.

### 3.2.2 Lintasan Lurus

Pengujian sensor GPS dengan filter FIR saat membentuk lintasan lurus dapat dilihat pada gambar 13.

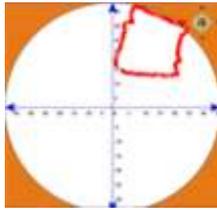


Gambar 13. Data GPS Filter FIR pada Lintasan Lurus

Dari gambar 13 dapat diketahui bahwa tanda titik/bintang membentuk tren lintasan lurus. Namun, lintasan tersebut belum sempurna, terlihat masih berkelok-kelok. Jika dibandingkan dengan pendataan hasil uji jalan sensor GPS dengan tanpa menggunakan filter apapun, data uji jalan menggunakan filter FIR menunjukkan data mengalami kemajuan.

### 3.2.3 Lintasan Bentuk Kotak

Pengujian sensor GPS dengan filter FIR saat membentuk lintasan kotak dapat diketahui pada gambar 14.



Gambar 14. Data GPS Filter FIR pada Lintasan Kotak

Dari gambar 14, nampak bahwa data hasil uji jalan sensor GPS dengan sensor FIR juga telah menunjukkan tren bentuk lintasan kotak meski belum sempurna. Tanda bintang/titik menunjukkan kontinuitas yang lebih rapat dibandingkan data hasil uji jalan sensor GPS tanpa menggunakan filter apapun.

### 3.3 Pengujian dengan Low Pass Filter

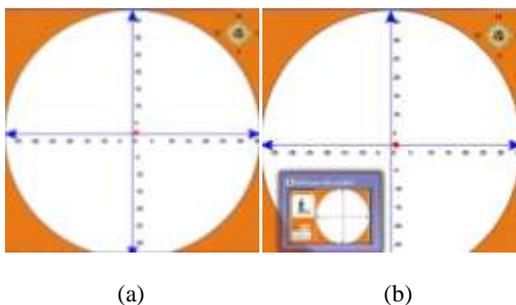
Pengujian selanjutnya adalah pengujian data sensor GPS yang menerapkan Filter LPF (*Low Pass Filter*). Hal yang ingin diketahui apakah penerapan LPF ini berhasil membuat performa sistem berjalan lebih baik daripada saat sebelum penerapan filter apapun maupun saat penerapan sensor FIR. Dalam satu aksi, diberikan 3 nilai periode tapis antara lain :  $T=9$ ,  $T=5$ ,  $T=1$ .

#### 3.3.1 Posisi Diam

Pengujian sensor GPS dengan filter LPF saat membentuk posisi diam terdiri dari tiga uji jalan yang memerlukan tiga perubahan nilai periode tapis.

##### A. Variabel Koefisien Filter $T=9$

Hasil pengujian data sensor GPS menggunakan *Low Pass Filter* dengan  $T=9$  dapat diketahui pada gambar 15.

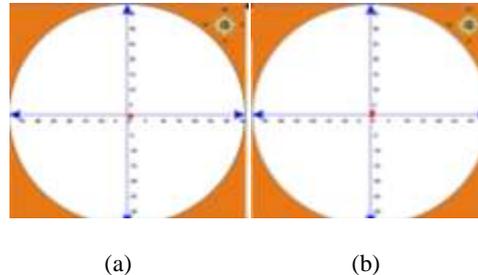


Gambar 15. Data GPS Filter LPF,  $T=9$  pada Posisi Diam

Dari gambar 15, terutama pada gambar 15(a) dapat diketahui bahwa tanda titik/bintik telah berhasil menunjukkan posisi diam di mana hanya ada satu tanda titik/bintik saja yang tampil pada GUI. Hal ini lebih baik dibandingkan dengan sensor GPS tanpa menggunakan filter apapun.

##### B. Variabel Koefisien Filter $T=5$

Hasil pengujian data sensor GPS menggunakan *Low Pass Filter* dengan  $T=5$  dapat diketahui pada gambar 16.

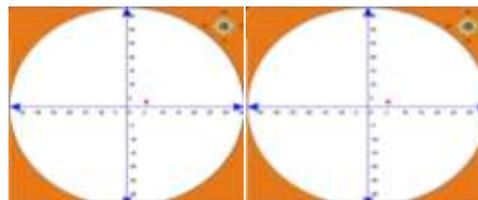


Gambar 16. Data GPS Filter LPF  $T=5$  pada Posisi Diam

Dari gambar 16(a), dapat diketahui bahwa data uji posisi diam telah menunjukkan konsistensi data yang sebenarnya pada uji pada posisi diam yakni menunjukkan hanya ada satu tanda titik/bintik saja yang tampil pada GUI. Meski pada gambar 16(b), titik terlihat kurang konsisten untuk diam meski posisi sensor GPS tidak bergerak. Hal ini disebabkan karena periode tapis yang digunakan semakin kecil sehingga frekuensi semakin besar (periode berbanding terbalik dengan frekuensi) sehingga jika frekuensi semakin besar mengakibatkan respon sensor semakin besar namun juga semakin tidak stabil (ada riak).

##### C. Variabel Koefisien Filter $T=1$

Hasil pengujian data sensor GPS menggunakan *Low Pass Filter* dengan  $T=1$  dapat diketahui pada gambar 17.



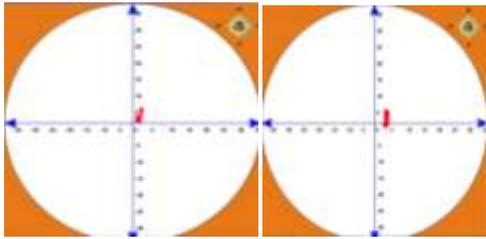
Gambar 17. Data GPS Filter LPF  $T=1$  pada Posisi Diam

Dari gambar 17, dapat diketahui bahwa data uji posisi diam telah menunjukkan konsistensi data yang sebenarnya pada uji pada posisi diam yakni menunjukkan hanya ada satu tanda titik/bintik saja yang tampil pada GUI. Meski periode semakin kecil sehingga secara teori respon sensor semakin tidak stabil namun data menunjukkan bahwa penerapan filter LPF dengan  $T=1$  pada posisi diam tetap memperlihatkan data yang bagus.

### 3.3.2 Lintasan Lurus Sepanjang 10 meter

#### A. Variabel Koefisien Filter $T=9$

Hasil pengujian data sensor GPS menggunakan *Low Pass Filter* dengan  $T=9$  dapat diketahui pada gambar 18.

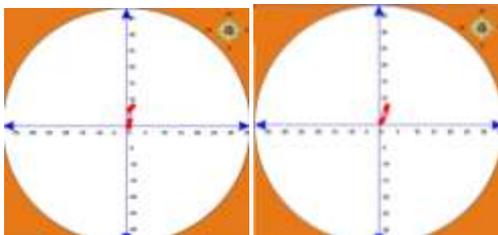


**Gambar 18. Data GPS Filter LPF T=9 Lintasan Lurus.**

Dari gambar 18, dapat diketahui bahwa hasil uji jalan lintasan lurus sensor GPS menggunakan filter LPF dengan  $T=9$  telah menunjukkan tren garis lurus yang tampil pada GUI. Namun garis tersebut berupa garis pendek padahal panjang lintasan sebenarnya yang dilalui sama dengan panjang lintasan yang digunakan untuk uji jalan pada jenis filter lain dan tanpa filter. Hal ini disebabkan karena periode tapis,  $T=9$  di mana frekuensi kecil ( $T$  berbanding terbalik dengan  $f$  di mana semakin besar  $T$ , frekuensi semakin kecil), sistem semakin kurang responsif terhadap perubahan namun mengakibatkan sistem lebih stabil.

#### B. Variabel Koefisien Filter T=5

Hasil pengujian data sensor GPS menggunakan *Low Pass Filter* dengan  $T=5$  dapat diketahui pada gambar 19.

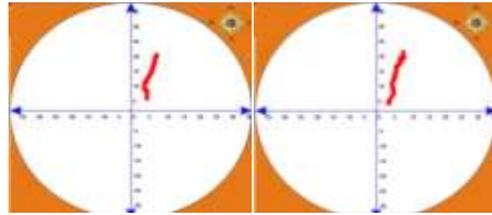


**Gambar 19. Data GPS Filter LPF T=5 Lintasan Lurus.**

Dari gambar 19, dapat diketahui bahwa hasil uji jalan lintasan lurus sensor GPS menggunakan filter LPF dengan  $T=5$  telah menunjukkan tren garis lurus yang tampil pada GUI. Namun garis tersebut berupa garis pendek meski lebih panjang saat uji jalan sensor GPS menggunakan LPF dengan  $T=9$ . Hal ini ikut disebabkan karena periode tapis lebih kecil dari pengujian sebelumnya ( $T=9$ ) sehingga telah sesuai teori di mana, semakin besar frekuensi, sistem akan semakin responsif (cepat merespon) terhadap perubahan.

#### C. Variabel Koefisien Filter T=1

Hasil pengujian data sensor GPS menggunakan *Low Pass Filter* dengan  $T=1$  dapat diketahui pada gambar 20.



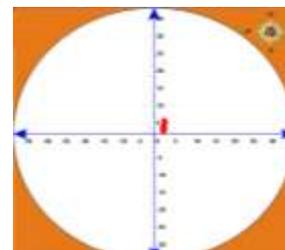
**Gambar 20. Data GPS Filter LPF T=1 Lintasan Lurus.**

Dari gambar 20, dapat diketahui bahwa hasil uji jalan lintasan lurus sensor GPS menggunakan filter LPF dengan  $T=1$  telah menunjukkan tren garis lurus yang tampil pada GUI. Tampilan lintasan tersebut berupa garis yang lebih panjang dari saat uji jalan menggunakan sensor LPF dengan  $T=9$ , maupun  $T=5$ . Hal ini telah sesuai teori di mana semakin besar frekuensi (periode semakin kecil), sistem semakin responsif terhadap perubahan masukan, namun akibatnya sistem mengalami lebih banyak riak (kurang stabil) seperti yang tampak pada gambar di mana saat berjalan dalam lintasan lurus, garis pada GUI tidak menampilkan garis lurus sempurna meski tren sudah mengarah ke garis lurus.

### 3.3.3 Lintasan Bentuk Kotak 10x10 meter.

#### A. Variabel Koefisien Filter T=9

Hasil pengujian data sensor GPS menggunakan *Low Pass Filter* dengan  $T=9$  dapat diketahui pada gambar 21.

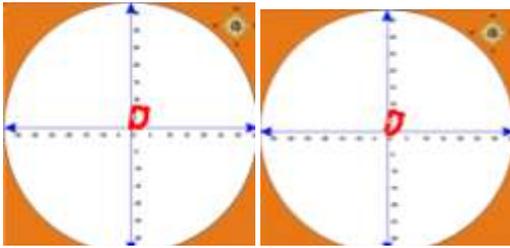


**Gambar 21 Data GPS Filter LPF T=9 pada Lintasan Kotak.**

Dari gambar 21, dapat diketahui bahwa hasil uji jalan lintasan kotak sensor GPS menggunakan filter LPF dengan  $T=9$  tidak berhasil menunjukkan tampilan lintasan kotak seperti yang diinginkan meski telah berkali-kali diulang dalam pengujian padahal digunakan berjalan seturut lintasan kotak di lapangan parkir Jurusan Teknik Elektro Undip. Hal ini ikut disebabkan karena periode yang besar berarti frekuensi semakin kecil, yang mengakibatkan sistem sensor kurang responsif terhadap perubahan masukan sehingga meski berjalan lambat sesuai lintasan kotak pada lapangan parkir, tampilan tetap menunjukkan hanya garis lurus pendek.

#### B. Variabel Koefisien Filter T=5

Hasil pengujian data sensor GPS menggunakan *Low Pass Filter* dengan  $T=5$  dapat diketahui pada gambar 22.

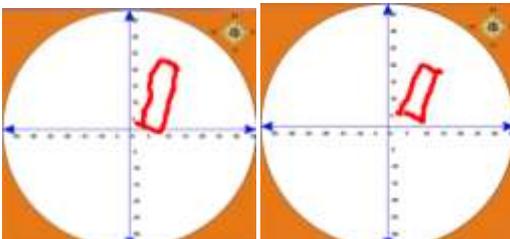


**Gambar 22** Data GPS Filter LPF T=5 pada Lintasan Kotak.

Dari gambar 22, dapat diketahui bahwa hasil uji jalan lintasan kotak sensor GPS menggunakan filter LPF dengan  $T=5$  telah berhasil menunjukkan tren tampilan lintasan kotak meski belum sempurna. Hal ini lebih baik dibandingkan saat pengujian sensor dengan filter LPF dengan  $T=9$ . Pada pengujian ini, tampilan berupa kotak yang kecil meski telah dilakukan uji jalan seturut lintasan kotak lebih dari sekali. Frekuensi yang semakin besar ( $T$  / periode tapis lebih kecil) saat  $T=5$ , membuat respon sensor semakin baik dari saat  $T=9$  meski hasil uji jalan belum sepenuhnya sesuai dengan harapan.

### C. Variabel Koefisien Filter $T=1$

Hasil pengujian data sensor GPS menggunakan *Low Pass Filter* dengan  $T=1$  dapat diketahui pada gambar 23.



**Gambar 23.** Data GPS Filter LPF  $T=9$  pada Lintasan Kotak.

Dari gambar 23, dapat diketahui bahwa hasil uji jalan lintasan kotak sensor GPS menggunakan filter LPF dengan  $T=1$  telah berhasil menunjukkan tren tampilan lintasan kotak yang lebih baik dari pada dengan  $T=5$  maupun dengan  $T=9$ . Tampilan berupa kotak yang lebih besar dengan kontinuitas titik/titik yang rapat dan tidak ada perenggangan. Hal ini ikut disebabkan karena sistem semakin responsif terhadap perubahan masukan diakibatkan frekuensi yang semakin besar sehingga respon semakin mendekati harapan yaitu membentuk lintasan kotak yang semakin baik untuk diamati meski belum membentuk lintasan kotak sempurna.

## 4. Kesimpulan

Performansi sistem pemantauan posisi dengan sensor GPS sangat ditentukan oleh letaknya. Filter yang diterapkan juga terbukti mempengaruhi kinerja dan hasil uji jalan. Dari kedua filter yang diperbandingkan, filter LPF

menunjukkan hasil yang lebih memuaskan. Periode tapis pada filter LPF berpengaruh pada hasil uji. Secara keseluruhan, filter LPF dengan  $T=1$  merupakan penerapan yang paling menunjukkan hasil yang mendekati harapan.

## Referensi

- [1]. El-Rabbany, Ahmed, Introduction to GPS: The Global Positioning System, Artech House, Inc, 2002.
- [2]. Gopi, Satheesh, Global Positioning System : Principle and Applications, Tata McGraw-Hill Publishing Company Limited, 2005.
- [3]. Basuki, Akbari Indra dan Oka Mahendra, Metode Pemantau Posisi dan Arah Gerak Helikopter Tanpa Awak dengan Google Maps API, Penelitian P2 Informatika-LIPI.
- [4]. Anshori, Muhammad Ikhsan, Desain Kontrol Autopilot pada UGV (Unmanned Ground Vehicle) Berbasis GPS (Global Positioning System), Tugas Akhir Teknik Elektro Universitas Diponegoro, Semarang 2011.
- [5]. Halidah, Hafsa, dkk, Vehicle Security And Locator (Very-Lo) sebagai Aplikasi Sms Center dan Global Positioning System (Gps) untuk Alternatif Alat Pengaman Kendaraan Yang Murah dan Efektif dengan Handphone Bekas, Program Kreativitas Mahasiswa, Institut Teknologi Bandung, 2010.
- [6]. Chand, Mahesh, Graphics Programming with GDI+, Pearson Education, Inc, 2004.
- [7]. Mossakowski, Krzysztof, GDI- Graphics Device Interface, Power Point Lecture 3 Faculty of Mathematics and Information Science.
- [8]. -----, ATmega 164 Datasheet, [www.atmel.com](http://www.atmel.com).
- [9]. Summerville, Douglas, Embedded System Interfacing for Engineers Using the Freescale HCS08 Microcontroller, Morgan and Claypool Publishers.
- [10]. Sedra, Adel, Smith, Kenneth C. *Microelectronic Circuits, 3 ed.* Saunders College Publishing. 1991.
- [11]. K. V. Cartwright, P. Russell and E. J. Kaminsky, "Finding the maximum magnitude response (gain) of second-order filters without calculus," Lat. Am. J. Phys. Educ. Vol. 6, No. 4, pp. 559-565, 2012.
- [12]. Sumardi, Syahid, Simulasi Penekanan Derau dengan Metode Finite Impulse Response (FIR) secara Adaptif Menggunakan Algoritma Least Mean Square, Teknik Elektro Universitas Diponegoro.
- [13]. -----, EM-411 Product Guide, [http://fl.chf.free.fr/PDF/acrianGHZ/airwave%20modules/EM-411Product\\_Guide1.1.pdf](http://fl.chf.free.fr/PDF/acrianGHZ/airwave%20modules/EM-411Product_Guide1.1.pdf).
- [14]. -----, GPS Receiver Module Data Sheet, <http://www.polstargps.com>.