

## DESAIN DAN IMPLEMENTASI INSTRUMEN DIGITAL PENGUKUR KECEPATAN, JARAK DAN PEMBAKARAN KALORI PADA SEPEDA GUNUNG BERBASIS MIKROKONTROL ATMEGA32

<b>Lantana Dioren Rumpa</b> Teknik Informatika Universitas Kristen Indonesia Toraja Makale, Indonesia dionrumpa@ukitoraja.ac.id	<b>Enos Lolang</b> Pendidikan Matematika Universitas Kristen Indonesia Toraja Makale, Indonesia	<b>Parea R. Rangan</b> Teknik Sipil Universitas Kristen Indonesia Toraja Makale, Indonesia	<b>Alexander Pakiding</b> Pendidikan Matematika Universitas Kristen Indonesia Toraja Makale, Indonesia
--	--	--	---

**Abstract:** Bicycle is one of the most popular transportation and in Indonesia there are still many people used it. Static bike or stationary bicycle is one of many kind of bicycle and people used it in indoor room. Static bike has an electronic panel that could help people who used it to know how far the distance, what about their speed and how many calories they have burned. Many people said that static bike is more fun and more interesting than outdoor bicycle especially from mountain bike because the static bike has the electronic panel. The electronic panel from static bike will be used on mountain bike so it can be more interesting as static bike. Microcontroller will used as minicomputer to processing all data from magnetic sensor that show the rotation of bike's tires and then the microcontroller will show the result on seven segment display about the distances, the speed, and the calories burned.

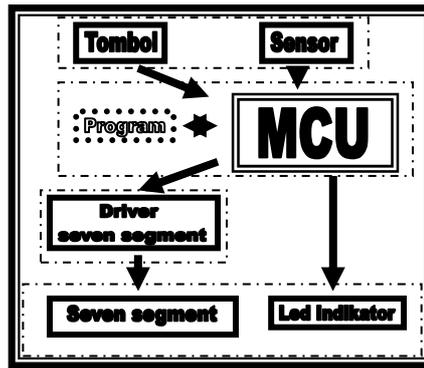
**Kata kunci:** MET, Modul *Main Panel*, Modul *User Board*, Kalori.

Instrumentasi digital yang biasanya ditemui pada panel elektronik sepeda statis merupakan salah satu alat yang dapat memberikan informasi ke pada penggunanya. Informasi tersebut biasanya berupa jarak yang sudah ditempuh dalam hal ini berapa jauh pengguna mengayuh sepedanya. Begitu juga dengan informasi kecepatan sepeda. Terakhir adalah informasi pembakaran kalori selama menggunakan sepeda tersebut. Pada sepeda *outdoor* khususnya sepeda gunung panel tersebut jarang kita temui. Walaupun ada itupun hanya untuk menghitung jarak atau kecepatan saja dan juga harganya mahal. Dengan menggunakan mikrokontrol sebagai pusat pengolah data maka panel elektronik pada sepeda statis dapat diaplikasikan pada sepeda gunung. Dengan adanya sistem tersebut pada sepeda gunung pengguna dapat menggunakan sepeda dengan tambahan informasi dari panel tersebut seperti jarak yang sudah ditempuh, kecepatan, dan pembakaran kalori sambil menikmati pemandangan luar ruangan.

Dari alasan di atas, dirancang dan dibuat sebuah instrumentasi digital sebagai panel elektronik pada sepeda gunung.

### METODE

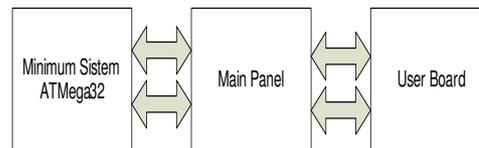
Instrumentasi ini dipasangkan pada sepeda gunung. Sistem ini sepenuhnya akan dikendalikan oleh mikrokontrol. Sebagai penghitung putaran ban digunakan sensor *hall-effect* atau yang lebih dikenal dengan nama sensor magnet. Sedangkan untuk tampilan display digunakan seven segment sebanyak 4 digit. Gambaran umum dari sistem ini dapat dilihat pada Gambar 1.



Gambar 1. Blok diagram keseluruhan sistem

### Perancangan Perangkat Keras

Perancangan Perangkat keras sistem ini dibagi menjadi tiga modul yaitu : modul minimum sistem ATmega32, modul *Main Panel*, dan modul *User Board*. Pada masing-masing modul ini akan terpasang semua komponen elektro yang dipakai dalam perancangan perangkat keras. Adapun diagram hubungan antara tiga modul tersebut dapat dilihat pada gambar 2.



Gambar 2. Blok diagram hubungan antara modul

### Modul Minimum Sistem ATmega32

Rangkaian minimum sistem ATmega32 yang digunakan sebagai pemroses input yang diperoleh dari sensor magnet maupun tombol-tombol *interfaces*, juga sebagai penyimpanan data-data yang diperoleh. Minimum sistem ini juga yang memproses seluruh perhitungan dari sistem baik itu perhitungan kecepatan, jarak, dan pembakaran kalori. Minimum sistem akan menjadi pusat pengolahan data dari modul *Main Panel* dan mengirimkan hasil olahan data ke modul *User Board* melalui modul *Main Panel*. Modul minimum sistem ATmega32 dapat dilihat pada gambar 3.

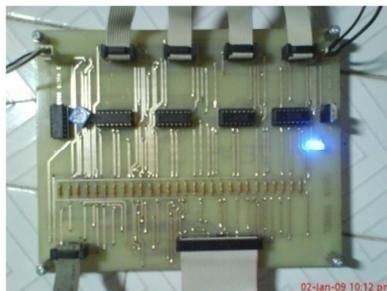


Gambar 3. Modul Minimum Sistem ATmega32

### Modul *Main Panel*

Pada *Modul Main Panel*, semua driver-driver dari seven segment yaitu IC *BCD to Decoder 7-segment 74LS47* ditempatkan disini. Modul ini juga berfungsi sebagai pembagi sumber *power* yang dialirkan ke semua modul-modul yang lainnya dan komponen lainnya. Selain itu IC 7805 yang berfungsi sebagai *regulator* juga ditempatkan

pada *modul* ini karena fungsinya sebagai perubah tegangan menjadi tegangan kerja 5V. *Modul Main Panel* seperti terlihat pada Gambar 4.



Gambar 4. Modul *Main Panel*

Pada *Modul Main Panel* juga terdapat konektor dari sensor magnet yang menghubungkan sensor dengan *modul* ini. Data yang berasal dari sensor magnet akan dibandingkan dengan  $V_{ref}$  yang berasal *single tone*. IC *Comparator LM324* akan berfungsi sebagai pembandingnya agar supaya data dari sensor dapat dibaca oleh *MCU* yang selanjutnya akan diproses sesuai program.

#### **Modul User Board**

Modul ini tersusun dari empat digit *seven segment Anoda*, empat konektor tombol, tiga led indikator, dan 2 konektor ke *main panel*. *Seven segment display* yang terdapat pada *modul* ini berfungsi untuk menampilkan hasil perhitungan dari *MCU* baik itu perhitungan kecepatan sepeda, jarak tempuh, serta pembakaran kalori. Modul ini juga disambungkan dengan empat buah tombol yang mempunyai fungsi dan warna yang berbeda. Modul User Board dapat dilihat pada gambar 5.

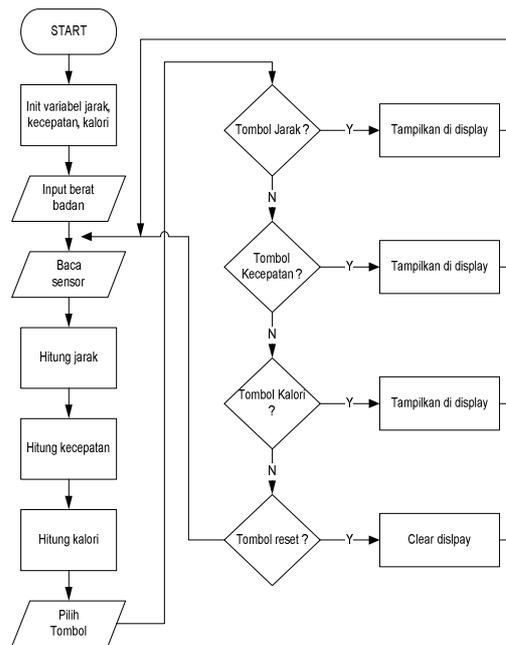


Gambar 5. Modul *User Board*

#### **Perancangan Perangkat Lunak**

Pada sistem ini perangkat lunak akan berperan untuk mengolah data dalam mikrokontroler. Adapun data yang akan diolah semuanya berasal dari sensor magnet.

Sensor magnet akan menjadi sarana untuk menghitung jumlah putaran ban. Apabila jumlah putaran ban diketahui maka jarak tempuh dapat dicari. Kecepatan dapat ditemukan apabila jarak dan waktu diketahui. Dan pembakaran kalori dapat dihitung apabila kecepatan diketahui, nilai MET, dan berat badan. Adapun flowchart dari program utama dapat dilihat pada gambar 6.



Gambar 6. Diagram alir program utama sistem

Dari gambar 6 terlihat ada tiga perhitungan yang utama yaitu perhitungan jarak, perhitungan kecepatan, dan perhitungan kalori.

Secara umum hanya ada satu program atau software yang akan di- *download* ke mikro atau *MCU*. Program inilah yang akan menjadi inti dari perhitungan data. Program tersebut akan dibuat dengan bantuan software *Code Vision AVR (CVAVR)*. Program akan dibagi ke dalam fungsi – fungsi untuk memudahkan pemrograman. Adapun beberapa fungsi (function) yang digunakan dalam pemrograman sistem ini adalah :

### Fungsi Penghitung Jarak

Untuk mencari jarak yang ditempuh selama bersepeda maka digunakan sebuah fungsi untuk menghitung jarak tempuh. Fungsi ini akan dipanggil oleh program utama. Fungsi tersebut mengacu pada persamaan (1) di bawah ini.

$$m = K \times n \dots\dots\dots (1)$$

Dimana :

- m : Jarak Tempuh (Km)
- K : Keliling ban sepeda (Km)
- n : Banyak putaran ban (n kali)

Fungsi ini menggunakan dua variabel yaitu keliling ban sepeda dan banyaknya rotasi ban sepeda. Petikan program fungsi penghitung jarak tempuh dapat dilihat di bawah ini.

```

float hitJarak (int rotasi)
{
    float k = 0.0017;
    float jarak=0;

    jarak = rotasi*k;

    return (jarak);
}
    
```

Pada fungsi ini terlihat variabel k yaitu keliling ban sepeda sebesar 0,0017. Ini hanya contoh pengukuran keliling ban sepeda. Untuk ukuran keliling ban sepeda yang

sebenarnya yaitu 0.0603508 Km. Banyaknya rotasi dapat dihitung dari berapa kali sensor magnet mendeteksi magnet yang dipasang pada ban sepeda.

### Fungsi Penghitung Kecepatan

Fungsi untuk menghitung kecepatan mengacu pada persamaan (2) di bawah ini.

$$V = m / t \dots\dots\dots (2)$$

Dimana :

- V : Kecepatan (Km/Jam)
- m : Jarak (Km)
- t : *time*/waktu (Jam)

Seperti yang dijelaskan pada persamaan (2) bahwa kecepatan dihitung dari jarak dibagi dengan waktu. Untuk mencari kecepatan maka digunakan fungsi untuk mencari kecepatan selama satu kali putaran ban. Fungsinya dapat dilihat di bawah ini.

```
float hitKcptan (float time)
{
    float k = 0.0017;
    float speed;

    speed = k/(time/3600000);

    return (speed);
}
```

Dari fungsi dapat dilihat bahwa jarak tempuh sudah ditetapkan yaitu keliling dari ban sepeda. Variabel *time* dapat dicari dengan menghitung seberapa lama putaran ban tersebut sampai mencapai satu keliling dalam satuan mili detik. Setelah itu *time* akan dibagi 3600000 untuk mendapatkan *time* dalam satuan Jam. Kecepatan akan dihitung setiap satu kali putaran ban sepeda.

### Fungsi Penghitung Pembakaran Kalori

Kalori didapatkan melalui persamaan (3) di bawah ini.

$$C = MET \times W \times t \dots\dots\dots (3)$$

Dimana :

- C : *Calories* – pembakaran kalori (Kalori)
- W : Berat badan (Kg)
- t : Waktu (Jam)
- MET : Nilai dari MET

Adapun MET adalah singkatan dari *Metabolic Equivalent* yang artinya adalah jumlah Kalori (*Cal*) yang digunakan oleh organisme atau individu per menit dalam melakukan aktivitas. 1 MET adalah kalori yang digunakan oleh individu atau organisme selama istirahat atau diam. 1 MET juga sama dengan 3,5 mililiter oksigen per menit per kilogram dari berat tubuh selama proses metabolisme tubuh. Standar perhitungan MET menggunakan 3 variabel yaitu:

- Berat badan ( Kg ).
- Waktu ( Jam ). Pada system ini waktu akan dihitung selama satu kali perputaran ban.
- Tabel standar dari nilai MET dari setiap kegiatan atau aktivitas.

Untuk mencari pembakaran kalori maka variabel yang digunakan ada tiga yaitu konstanta MET, berat badan, dan waktu. Fungsinya dapat dilihat di bawah ini.

```
float hitKal (float time, float weight, float speed)
{
    float kalori=0;
    if ( ( speed <= 19.2 ) && ( speed >= 16.1 ) )
    {
        kalori= weight*6*time;
    }
    else if ( ( speed <= 22.4 ) && ( speed >= 19.3 ) )
    {
        kalori= weight*8*time;
    }
    else if ( ( speed <= 25.6 ) && ( speed >= 22.5 ) )
    {
        kalori=weight*10*time;
    }
    else if ( ( speed <= 30.6 ) && ( speed >= 25.7 ) )
    {
        kalori=weight*12*time;
    }
    else if ( speed <= 32.2)
    {
        kalori=weight*16*time;
    }

    return (kalori);
}
```

Dari fungsi ini dapat dilihat bahwa secara tidak langsung kecepatan juga mempengaruhi pembakaran kalori. Oleh sebab itu nilai MET juga akan dipengaruhi oleh kecepatan. Kecepatan sepeda mempengaruhi nilai MET seperti pada tabel 1 di bawah ini.

Tabel 1. Nilai MET

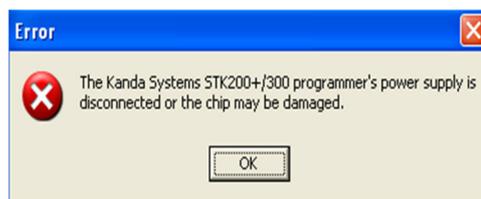
MET	Aktivitas
6	Sepeda Luar ruangan ( 16.1 - 19.2 Km/Jam)
8	Sepeda Luar ruangan ( 19.3 - 22.4 Km/Jam)
10	Sepeda luar ruangan ( 22.5 - 25.6 Km/Jam )
12	Sepeda luar ruangan ( 25.7 - 30.6 Km/Jam )
16	Sepeda luar ruangan ( > 32.2 Km/Jam )

Perhitungan pembakaran kalori akan dilakukan setiap satu kali putaran ban. Hasil dari perhitungan tersebut akan selalu ditambahkan dengan hasil perhitungan kalori setiap satu kali putaran ban berikutnya.

## PENGUJIAN SISTEM

### Pengujian Modul Minimum sistem ATMega 32

Untuk melakukan pengujian dengan melakukan serangkaian prosedur seperti memasang kabel isp programmer ke port parallel PC. Kemudian dengan software CVAVR dilakukan prosedur chip signature, maka didapatkan hasil pengujian seperti gambar berikut.



Gambar 7. Pesan error dari CVAVR

Gambar 8. Pesan *chip signature* berhasil

Pada gambar 7 menunjukkan adanya *error* yang terjadi pada saat melakukan *chip signature*. Ini disebabkan karena beberapa hal, seperti : Power tidak tersambung ke modul minimum sistem, sistem pengaturan pada software tidak tepat, *port* paralel PC rusak, kabel *isp downloader* rusak, atau chip dari AVR rusak.

Pada pengujian ini *error* tersebut dikarenakan kabel *isp downloader* yang rusak. Gambar 8 memperlihatkan keberhasilan prosedur *chip signature*. Ini menandakan minimum sistem siap diprogram melalui PC. Dengan melihat hasil *chip signature* yang berhasil membaca chip AVR, maka dapat diambil kesimpulan bahwa koneksi antara modul minimum sistem ATmega32 dengan PC telah berhasil dimana PC dapat membaca chip AVR yang ada pada modul minimum sistem ATmega32.

### Pengujian Modul Main Panel

Pada pengujian modul ini dilakukan serangkaian pengujian terhadap komponen-komponen yang ada di dalamnya seperti *driver seven segment*, dan rangkaian sensor magnet.

Download Potongan program di bawah ini ke modul minimum sistem ATmega32 untuk menguji *driver seven segment*. Program ini akan mengirimkan data 0x01 ke *port A* untuk menampilkan bilangan 01 pada *seven segment*.

```
while (1)
{
    PORTA = 0x01;
    delay_ms (100);
};
}
```

Untuk menguji sensor magnet yang ada pada modul *Main Panel*, *download* potongan program di bawah ini. Pada program ini, data dari sensor akan dibaca mikrokontrol melalui *port C.0*. Apabila terdapat data dari sensor maka mikrokontrol akan mengirimkan data 0x22 untuk menampilkan bilangan 22 pada *seven segment*.

```
while (1)
{
    if ( PINC.0 == 0)
    {
        PORTA = 0x22;
        delay_ms (100);
    }
};
}
```

Hasil pengujian modul ini terlihat pada gambar 9 dan gambar 10.

Gambar 9. Hasil pengujian pada *driver seven segment*

Gambar 10. Hasil pengujian pada sensor magnet

Gambar 9 merupakan hasil pengujian yang tampak pada *seven segment* setelah potongan program di atas di-*downloadkan* ke modul minimum sistem ATmega32.

Pada Gambar 10 menunjukkan hasil pengujian pada sensor magnet. Apabila magnet didekatkan pada sensor, maka minimum sistem akan mengirim data 0x22 ke *port A*.

Setelah dilakukan pengujian, maka dapat diambil kesimpulan bahwa hasil pengujian pada modul *Main Panel* terutama pada *driver seven segment* dapat membaca input dari mikrokontrol dan mengrimkannya ke *seven segment*. Pada sensor magnet juga dapat mengirimkan data ke mikrokontrol dan dapat di tampilkan ke *seven segment*.

### Pengujian Modul User Board

Untuk pengujian seven segment, *download* potongan program di bawah ini. Program ini akan mengirimkan data 0x84 ke *port A* dan data 0x19 ke *port B* untuk menampilkan bilangan 1984 ke *seven segment*.

```
while (1)
{
    PORTA = 0x84;
    PORTB = 0x19;
    delay_ms (100);
};
}
```

Untuk pengujian tombol *Push Button*, *download* potongan program di bawah ini. Program di bawah ini akan mendeteksi penekanan tombol. Apabila tombol 1 ditekan maka mikrokontrol akan menerima data pada *port C.1* dan menampilkan bilangan 01 pada *seven segment*. Bila tombol 2 ditekan akan menampilkan bilangan 02 ke *seven segment*. Jika tombol 3 yang ditekan maka akan menampilkan 03. Dan jika tombol 4 maka akan menampilkan bilangan 04 ke *seven segment*.

```
if ( (PINC.1 == 0) && (PINC.2 == 1) && (PINC.3 == 1) && (PINC.4 == 1) )
{
    PORTA = 0x01;
    delay_ms (10);
}
```

```

}
else if ( (PINC.1 == 1) && (PINC.2 == 0) && (PINC.3 == 1) && (PINC.4 == 1))
{
    PORTA = 0x02;
    delay_ms(10);
}
else if ( (PINC.1 == 1) && (PINC.2 == 1) && (PINC.3 == 1) && (PINC.4 == 0))
{
    PORTA = 0x03;
    delay_ms(10);
}
else if ( (PINC.1 == 1) && (PINC.2 == 1) && (PINC.3 == 0) && (PINC.4 == 1))
{
    PORTA = 0x04;
    delay_ms(10);
}

```

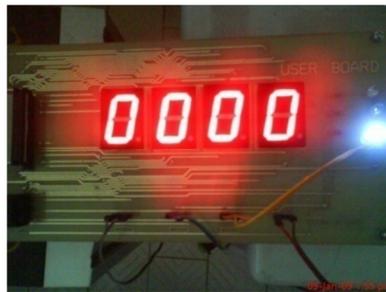
Untuk pengujian led indikator, *download* potongan program berikut ini. Program ini akan menyalakan led indikator.

```

while (1)
{
    PORTD = 0b00000111;
    delay_ms(100);
};
}

```

Setelah melakukan serangkaian prosedur di atas maka terlihat hasil seperti pada Gambar 4.9, Gambar 4.10, Gambar 4.11, Gambar 4.12, Gambar 4.13, Gambar 4.14, Gambar 4.15.



Gambar 11. Modul *User Board* setelah disambungkan dengan catu daya



Gambar 12. Tampilan *seven segment* setelah program di-*download*



Gambar 13. Hasil penekanan pada tombol *Push Button 1*



Gambar 14. Hasil penekanan pada tombol *Push Button 2*



Gambar 15. Hasil penekanan pada tombol *Push Button 3*



Gambar 16. Hasil penekanan pada tombol *Push Button 4*



Gambar 17. Tampilan led indikator

Gambar 11 memperlihatkan *seven segment* dan led indikator menyala ini membuktikan adanya sumber daya yang masuk ke modul *User Board*. Gambar 12 adalah hasil dari program yang memperlihatkan *seven segment* dapat menyala sesuai dengan program.

Gambar 13 menunjukkan *seven segment* yang menampilkan bilangan 01. Tampilan ini sesuai dengan program yang di-download ke *chip* mikrokontrol apabila terdapat penekanan pada tombol 1.

Gambar 14 menampilkan hasil apabila tombol 2 ditekan maka pada *seven segment* akan ditampilkan bilangan 02. Ini sesuai dengan program yang sudah di-download ke *chip* mikrokontrol.

Pada gambar 15 terlihat *seven segment* menampilkan bilangan 03. Sesuai dengan program apabila terdapat penekanan tombol 3 maka akan keluar tampilan bilangan 03 pada *seven segment*.

Gambar 16 menunjukkan hasil dari penekanan tombol 4 yang sesuai dengan program.

Gambar 17 memperlihatkan led indikator yang menyala. Hasil menunjukkan led dapat menyala sesuai program. Secara umum dapat disimpulkan bahwa modul *User Board* dapat melakukan komunikasi dengan dua modul yang lainnya.

## KESIMPULAN

1. Hasil rancangan untuk panel elektronik yang dipasang di depan sepeda dapat digunakan sebagai *user interfaces* atau sebagai sarana pemberi informasi jarak, kecepatan, dan pembakaran kalori ke *user*.
2. Modul minimum sistem ATmega32 dapat mengolah semua data – data yang diperlukan dan dapat berkomunikasi dengan modul *Main panel* dan modul *User Board*.
3. Modul Main Panel yang terdiri dari *driver seven segment* dan rangkaian sensor magnet dapat berkomunikasi dengan modul minimum sistem ATmega32 dan modul *User Board*. *Driver seven segment* dapat menyalakan *seven segment* dan data dari sensor magnet dapat dibaca di mikrokontrol.
4. Modul *User Board* dapat berkomunikasi dengan modul minimum sistem dan modul Main Panel. *Seven Segment display* dapat menampilkan data sesuai dengan program, led indikator dapat menyala sesuai dengan program, dan data dari tombol *push button* dapat dideteksi oleh mikrokontrol.

Perhitungan untuk mencari jarak tempuh, kecepatan sepeda, dan pembakaran kalori dapat diaplikasikan pada sistem di dalam *chip* mikrokontrol.

## DAFTAR RUJUKAN

- ALLEGRO Microsystem Inc. *Ratiometric, Linear Hall-Effect Sensor UGN3503*, (Online), ([www.alldatasheet.com](http://www.alldatasheet.com), diakses 2 September 2008).
- ATMEL Corporation. 2005. *8-bit AVR Microcontroller with 32K Bytes In-System Programmable Flash ATmega32*, (Online), (<http://www.atmel.com>, diakses 15 Desember 2008)
- FAIRCHILD Semiconductor. *BCD to 7-Segment Decoder/Driver DM7447A*, (Online), ([www.alldatasheet.com](http://www.alldatasheet.com), diakses 2 September 2008)
- Hadju, dr. Veni Ph.d, *Diktat Gizi Dasar*, Fakultas Kesehatan Masyarakat UNHAS, Makassar.
- Healthline, *Calculate your own calories, MET*, (Online), ([www.healthline.com/hlbook/dia-calculating-caloric-goals](http://www.healthline.com/hlbook/dia-calculating-caloric-goals), diakses 10 November 2008).
- Heryanto, M.Ary, & Adi, Ir.Wisnu, *Pemrograman bahasa C untuk mikrokontoler ATMEGA8535*. Yogyakarta : ANDI.

- Sediaoetama, Achmad Djaeni, 2000, *Ilmu Gizi*, Dian Rakyat, Jakarta.
- Wardhana, L. 2006. *Belajar Sendiri Mikrokontroler AVR Seri ATmega8535 Simulasi, Hardware, dan Aplikasi*. Yogyakarta : ANDI.
- Wapedia. *The Free Encyclopedia, sepeda*, (online) (<http://wapedia.mobi/id/Sepeda>, diakses 3 Januari 2009).
- Wapedia. *The Free Encyclopedia, sepeda gunung*, (online) ([http://wapedia.mobi/id/Sepeda gunung](http://wapedia.mobi/id/Sepeda_gunung), diakses 9 Januari 2009).
- Winoto, Ardi, 2008, *Mikrokontroler AVR ATmega8/32/16/8535 dan Pemrogramannya dengan Bahasa C pada WinAVR*, Informatika, Bandung.