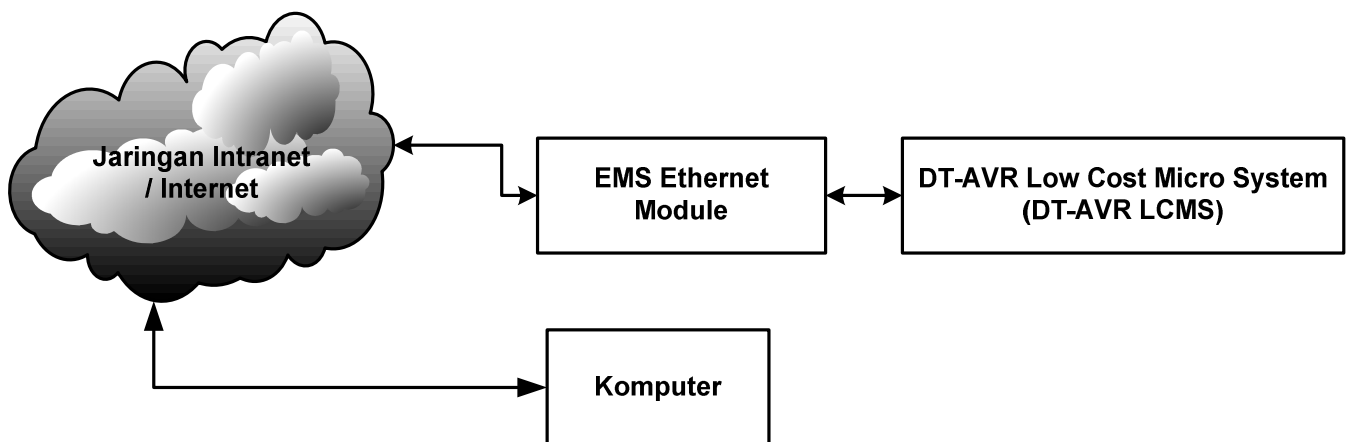


Web Server merupakan perangkat keras atau perangkat lunak yang menyediakan layanan akses kepada pengguna melalui protokol komunikasi HTTP atau HTTPS untuk mengakses berkas-berkas yang terdapat pada suatu situs web menggunakan aplikasi tertentu seperti *web browser*. Sebuah halaman web dapat terdiri atas berkas teks, gambar, *link*, dan video. EMS Ethernet Module yang berfungsi sebagai *ethernet controller module* untuk menjembatani komunikasi antara DT-AVR Low Cost Micro System (DT-AVR LCMS) dengan suatu jaringan komputer melalui antarmuka SPI. Konten web sederhana ditanamkan ke dalam memori *flash* ATmega8535 pada DT-AVR LCMS. Halaman web yang ditanamkan berisi konten web sederhana yang terdiri dari teks dan *link* ke situs tertentu. Program pada DT-AVR LCMS ditulis dalam bahasa C menggunakan *editor* Programmer Notepad dan *compiler* AVR GCC yang terintegrasi pada WinAVR.

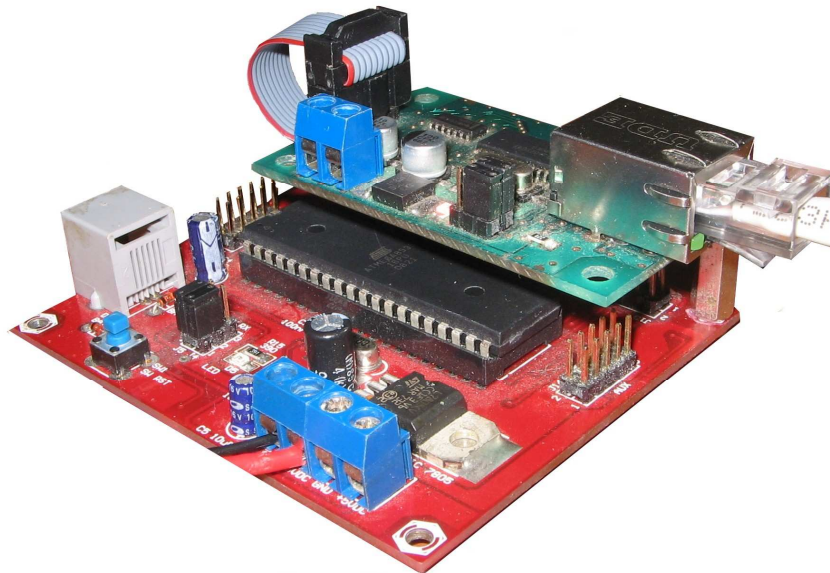
Modul yang digunakan pada aplikasi ini adalah:

- 1 unit DT-AVR Low Cost Micro System (DT-AVR LCMS).
- 1 unit EMS Ethernet Module.
- 1 set kabel LAN dengan konfigurasi *cross over* (umumnya disertakan pada paket pembelian EMS Ethernet Module).

Blok diagram sistem secara keseluruhan sebagai berikut:



Gambar 1
Blok Diagram AN180 Melalui 1 Komputer



Gambar 2
Modul Web Server (EMS Ethernet Module + DT-AVR LCMS)

Hubungan antar modul pada aplikasi ini sebagai berikut:

DT-AVR Low Cost Micro System	EMS Ethernet Module
GND (J11 pin 1)	GND (J1 pin 1)
VCC (J11 pin 2)	VCC (J1 pin 2)
PORTB.2 (J11 pin 3)	CLKOUT (J1 pin 3)
PORTB.3 (J11 pin 4)	- (J1 pin 4)
PORTB.4 (J11 pin 5)	INT (J1 pin 5)
PORTB.5 (J11 pin 6)	RESET (J1 pin 6)
SS (J11 pin 7)	CS (J1 pin 7)
MOSI (J11 pin 8)	MOSI (J1 pin 8)
MISO (J11 pin 9)	MISO (J1 pin 9)
SCK (J11 pin 10)	SCK (J1 pin 10)

Tabel 1
Hubungan Detil Antar Modul

Modul DT-AVR LCMS terhubung dengan EMS Ethernet Module melalui antarmuka SPI melalui pin SS, MISO, MOSI, dan SCK. EMS Ethernet Module dioperasikan pada mode *half duplex* sehingga perlu dilakukan pengaturan konfigurasi *jumper* J3 & J5 pada posisi 1-2. Catu daya modul EMS Ethernet modul diperoleh dari J11 pin 1&2 pada DT-AVR LCMS. Setelah semua modul siap, programlah main.hex ke dalam DT-AVR LCMS menggunakan DT-HiQ AVR USB ISP atau divais *programmer* lain yang kompatibel dan memiliki konektor ISP standard ATMEL 5x2. Hal yang perlu diingat, saat mengisikan program ke dalam DT-AVR LCMS, lepas koneksi EMS Ethernet Module yang terhubung ke port B DT-AVR LCMS. Setelah program main.hex berhasil diisikan ke dalam DT-AVR LCMS, hubungkan EMS Ethernet Module ke port B DT-AVR LCMS. Hubungkan kabel LAN tipe *cross over* antara EMS Ethernet Module dengan komputer kemudian hubungkan catu daya ke modul DT-AVR LCMS. Tekan tombol reset untuk memastikan program dijalankan dari awal. Konfigurasi IP komputer diatur antara 192.168.1.1 – 192.168.1.14 atau 192.168.1.16 - 192.168.1.253. Selanjutnya lakukan ping 192.168.1.15 melalui *command prompt*, jika modul EMS Ethernet Module telah terkoneksi dengan komputer maka komputer akan menerima pesan *reply from 192.168.1.15*

```

C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\support2>ping 192.168.1.15

Pinging 192.168.1.15 with 32 bytes of data:
Reply from 192.168.1.15: bytes=32 time=3ms TTL=64
Reply from 192.168.1.15: bytes=32 time=2ms TTL=64
Reply from 192.168.1.15: bytes=32 time=1ms TTL=64
Reply from 192.168.1.15: bytes=32 time=2ms TTL=64

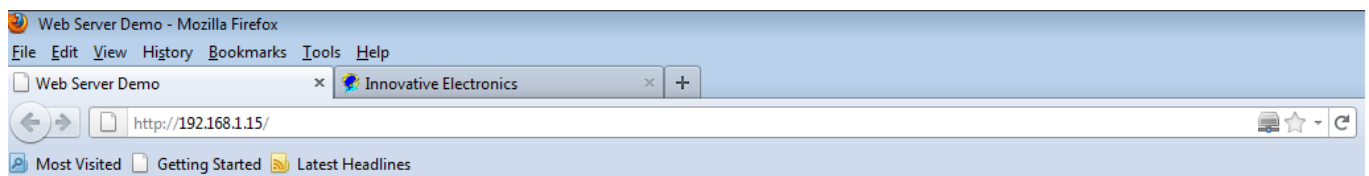
Ping statistics for 192.168.1.15:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 3ms, Average = 2ms

C:\Users\support2>

```

Gambar 3
Tampilan respon perintah ping 192.168.1.15 pada command prompt

Selanjutnya buka halaman *browser* yang sudah terinstal pada komputer yang digunakan. Jika *browser* yang digunakan terdapat pengaturan *proxy*, fitur *proxy* harus di-*disable* (dinonaktifkan) terlebih dahulu. Pada *address bar browser*, ketikkan alamat <http://192.168.1.15> maka akan ditampilkan konten halaman web yang disimpan pada memori *flash* ATmega8535 DT-AVR LCMS. Saat mencoba aplikasi ini digunakan *browser* Mozilla Firefox.



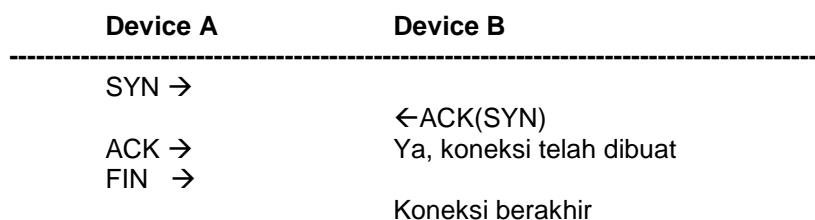
Simple Web Server

Selamat Datang di Web Server sederhana berbasis ENC28J60 & ATmega8535

IE.COM

Gambar 4
Tampilan *web server* pada *browser* Mozilla Firefox

Aplikasi “Simple Web Server” menggunakan protokol koneksi jenis TCP dan menggunakan *port* 80 (HTTP). TCP merupakan protokol yang membutuhkan adanya koneksi terlebih dahulu sebelum dilakukan proses tukar-menukar data antar *device*. Awalnya suatu paket dengan *SYN-flag* dikirim ke IP tujuan, tujuan akan memberikan respon dengan suatu *ACK(SYN) flag* atau suatu paket dengan *RST-flag*. *SYN* merupakan singkatan dari *SYN - (synchronisation)*, yang digunakan untuk ‘memberitahukan’ *device* tujuan suatu permintaan melakukan koneksi, kalau diterima, maka permintaan tersebut akan dijawab dengan suatu paket *ACK(SYN) flag*. *ACK* merupakan singkatan dari *ACK-(nowledgement)*. Setelah menerima paket dengan *ACK(SYN) flag*, *device* mengirim kembali suatu *ACK* memberitahukan *host* lain bahwa koneksi telah dibuat. Hal ini disebut sebagai “*Three-Way-Handshake*”. Jika koneksi telah dibuat dan salah satu *host* ingin melakukan *disconnect*, akan dikirim suatu paket dengan *FIN-flag* diaktifkan (*FIN* merupakan singkatan dari *FINish*). Proses tersebut dapat digambarkan sebagai berikut:



Protokol TCP dipilih karena aplikasi yang dikembangkan memerlukan akses yang sederhana melalui *web browser*. Tentunya HTTP merupakan pilihan yang tepat agar *device* dapat diakses secara langsung melalui *web browser* memanfaatkan *port* 80. Berikut ini tampilan lalu lintas paket data antara komputer dengan *web server* (EMS

Ethernet Module + DT-AVR LCMS) yang terekam oleh *Tool Network* Wireshark. Saat uji coba menggunakan Wireshark, IP komputer disatur pada 192.168.1.14 dan IP *web server* pada 192.168.1.15.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	QuantaCo_8c:80:6d	Broadcast	ARP	42	who has 192.168.1.15? Tell 192.168.1.14
2	0.001679	54:55:58:10:00:24	quantaCo_8c:80:6d	ARP	60	192.168.1.15 is at 54:55:58:10:00:24
3	0.001705	192.168.1.14	192.168.1.15	TCP	62	57516 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1
4	0.003727	192.168.1.15	192.168.1.14	TCP	60	http > 57516 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1408
5	0.003816	192.168.1.14	192.168.1.15	TCP	54	57516 > http [ACK] Seq=1 Ack=1 Win=16896 Len=0
6	0.011079	192.168.1.14	192.168.1.15	HTTP	383	GET / HTTP/1.1
7	0.016935	192.168.1.15	192.168.1.14	TCP	60	http > 57516 [ACK] Seq=1 Ack=330 Win=16896 Len=0
8	0.021628	192.168.1.15	192.168.1.14	TCP	387	[TCP segment of a reassembled PDU]
9	0.021709	192.168.1.14	192.168.1.15	TCP	54	57516 > http [ACK] Seq=330 Ack=335 Win=16563 Len=0
10	0.394014	192.168.1.14	192.168.1.15	TCP	54	57516 > http [FIN, ACK] Seq=330 Ack=335 Win=16563 Len=0
11	0.396003	192.168.1.15	192.168.1.14	TCP	60	http > 57516 [ACK] Seq=335 Ack=331 Win=16563 Len=0

Gambar 5
Tampilan lalu lintas data menggunakan Wireshark

User diberikan kebebasan untuk melakukan pengaturan konfigurasi MAC Address & IP Address dari *web server* ini. Pemilihan konfigurasi MAC Address & IP Address hendaknya disesuaikan dengan konfigurasi pada jaringan yang ada. Pemilihan konfigurasi MAC Address & IP Address tidak boleh sama dengan konfigurasi MAC Address & IP Address komputer/*device* lain yang aktif dalam satu jaringan. Konfigurasi MAC Address & IP Address disimpan dalam variabel *mymac* & *myip* yang ditulis pada *file main.c* seperti berikut ini:

- Konfigurasi MAC Address

```
static uint8_t mymac[6] = {0x54,0x55,0x58,0x10,0x00,0x24};
```
- Konfigurasi IP Address

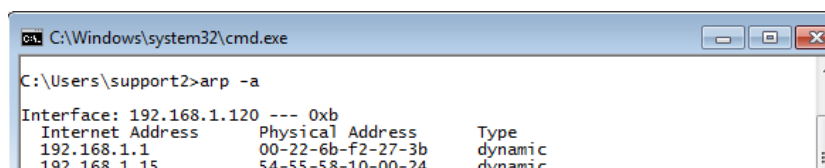
```
static uint8_t myip[4] = {192,168,1,15};
```

Script HTML yang digunakan untuk menampilkan konten halaman web yang disimpan dalam memori *flash* ATmega8535 DT-AVR LCMS sebagai berikut:

```
<html>
  <head>
    <meta http-equiv="Pragma" Content="no cache">
    <title>Web Server Demo</title>
  </head>
  <body>
    <center><h1>Simple Web Server</h1><br>
    Selamat Datang di Web Server sederhana berbasis ENC28J60 & ATmega8535<br>
    <p><a href="http://www.InnovativeElectronics.com">IE.COM</a></p>
  </body>
</html>
```

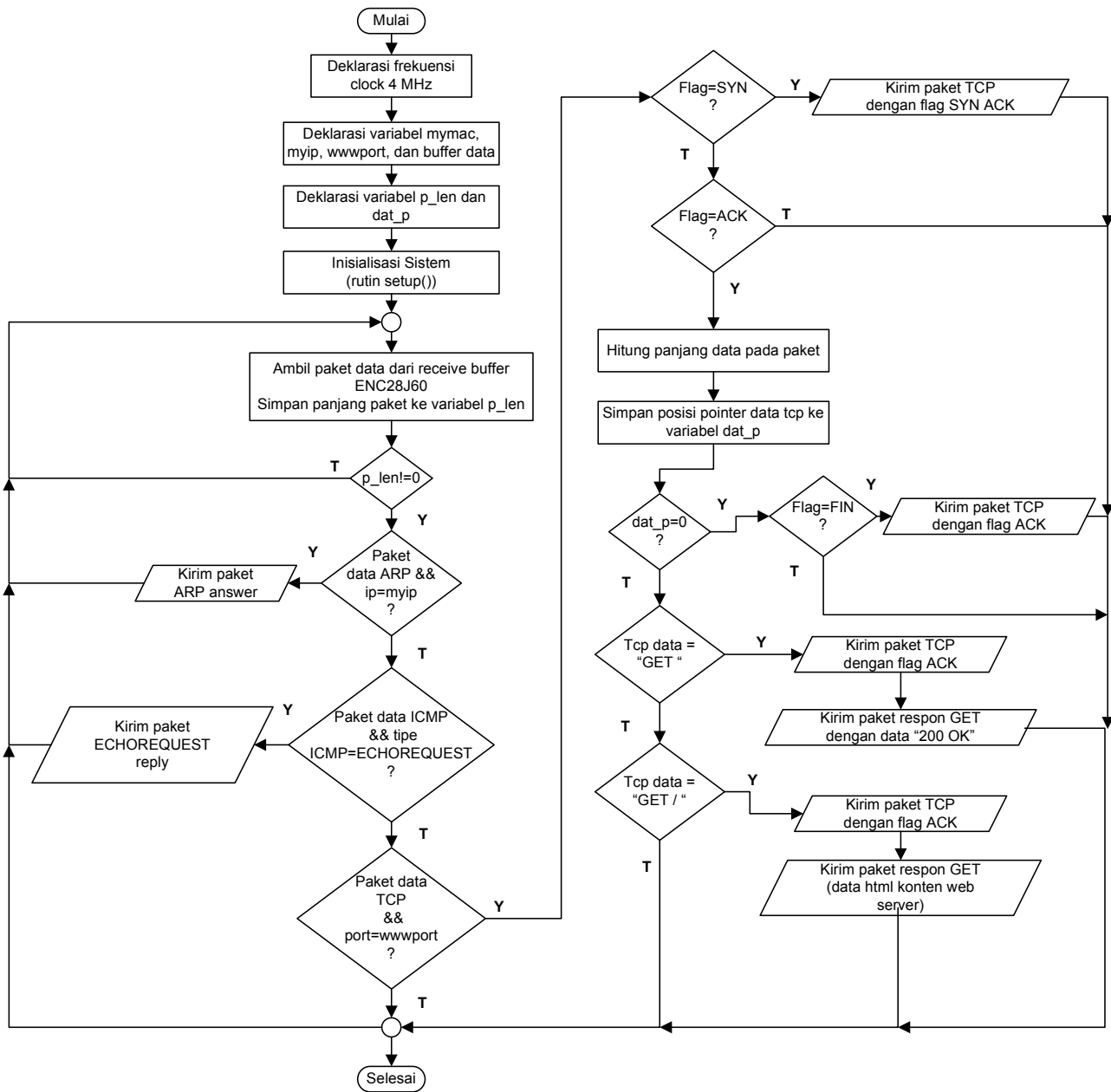
Pada *script* HTML tersebut ditambahkan *syntax* `<meta http-equiv="Pragma" Content="no cache">`, hal ini dimaksudkan untuk menghilangkan (*disable*) fitur *cache* pada *browser* yang digunakan. Sehingga saat kita menuliskan alamat <http://192.168.1.15> maka data (konten halaman web) yang ditampilkan benar-benar data dari *web server* bukan data memori *cache* yang masih disimpan oleh *browser*.

Jika dikirimkan perintah *arp* melalui *command prompt* dan *web server* telah terkoneksi dengan *host* (komputer), maka IP dan MAC Address *web server* akan terdaftar pada tabel ARP. Tampilan respon perintah *arp* melalui *command prompt* sebagai berikut:



Gambar 6
Tampilan tabel ARP pada command prompt

Flowchart program utama (main.c) pada aplikasi ini adalah sebagai berikut:



Gambar 7
Flowchart program main.c

Program main.c akan dijalankan sebagai berikut:

1. Pertama kali program mendefinisikan frekuensi *clock* mikrokontroler yang digunakan yaitu pada nilai 4 MHz (4000000 Hz).
2. Selanjutnya program mendeklarasikan variabel-variabel yang digunakan pada program seperti myip, mymac, wwwport, dat_p, dan plen. Variabel myip berisi data IP Address *web server*, variabel mymac berisi data MAC Address *web server*, dan variabel wwwport berisi alamat *port* HTTP yaitu 80. Pada aplikasi ini nilai MAC Address dan IP Address dikonfigurasi sebagai berikut:

```
MAC Address= 54:55:58:10:00:24
IP Address= 192.168.1.15.
Port= 80 (HTTP)
```

Konfigurasi MAC & IP Address dapat diubah sesuai dengan kebutuhan aplikasi, namun pastikan nilai konfigurasi tersebut tidak sama dengan *device* yang lain dalam satu jaringan.

3. Kemudian program melakukan inialisasi sistem yang ditangani oleh rutin setup(). Inialisasi yang dilakukan meliputi:
 - Inialisasi *chip* ENC28J60.
 - Inialisasi lampu LED yang terhubung dengan MAG JACK pada EMS Ethernet Module.
 - Inialisasi *layer* Ethernet atau IP sesuai dengan konfigurasi mymac, myip, dan wwwport.
4. Program melakukan pengecekan isi *receive buffer* ENC28J60. Panjang isi data pada *receive buffer* ENC28J60 disimpan dalam variabel plen.
5. Jika *receive buffer* ENC28J60 kosong maka program akan mengecek kembali isi *receive buffer* ENC28J60.
6. Jika paket data berupa paket ARP dan alamat IP tujuan sama dengan nilai konfigurasi myip maka program akan mengirimkan paket data *ARP answer*.
7. Jika paket data yang diterima berupa paket ICMP dan tipe ICMP adalah ECHOREQUEST (atau biasanya disebut PING) maka program akan mengirimkan paket data ICMP berupa ECHOREPLY.
8. Jika paket data berupa paket TCP dan *port* tujuan adalah wwwport (port 80), maka program akan melakukan pengecekan *flag* paket data sebagai berikut:

Jika *flag*=SYN maka program akan mengirimkan paket TCP dengan *flag* SYN ACK.

- Jika *flag*=ACK maka program akan menghitung panjang data (*payload*) pada paket TCP. Selanjutnya *pointer* posisi awal data (*payload*) disimpan ke dalam variabel *dat_p*. Variabel *dat_P* digunakan untuk proses pengecekan isi data (*payload*).

9. Kemudian program akan melakukan pengecekan isi data (*payload*) paket TCP.
 - Jika data (*payload*) kosong dan *flag*=FIN maka program akan mengirimkan paket TCP dengan *flag* =ACK.
 - Jika data (*payload*) berisi selain "GET " maka program akan mengirimkan paket TCP dengan *flag* =ACK dan kemudian mengirimkan paket TCP dengan data "200 OK".
 - Jika data (*payload*) berisi "GET /" maka program akan mengirimkan paket TCP dengan *flag* =ACK dan kemudian mengirimkan konten HTML *web server* yang disimpan pada memori flash ATmega8535 DT-AVR LCMS.
10. Selanjutnya program akan kembali melakukan pengecekan isi *receive buffer* ENC28J60 untuk menentukan proses selanjutnya (kembali ke nomor 4).

Konten halaman web yang ditampilkan pada aplikasi ini cukup sederhana yaitu hanya terdiri dari *header*, *teks*, dan *link* yang merujuk ke situs Innovative Electronics (<http://www.InnovativeElectronics.com>). Hal ini disebabkan keterbatasan memori RAM ATmega8535 pada DT-AVR LCMS. Aplikasi *web server* yang sederhana ini dapat dikembangkan lebih lanjut untuk *monitoring* sensor maupun kendali aktuator jarak jauh melalui jaringan ethernet. Hal yang perlu diperhatikan adalah kapasitas memori mikrokontroler yang digunakan khususnya memori RAM.

Listing program aplikasi ini terdapat pada **AN180.zip**

Selamat berinovasi!

All trademarks, trade names, company names, and product names are the property of their respective owners.
All softwares are copyright by their respective software publishers and/or creators.