

# DT-ARM

## DT-ARM *Application Note*

### AN220 – Simple Web Server

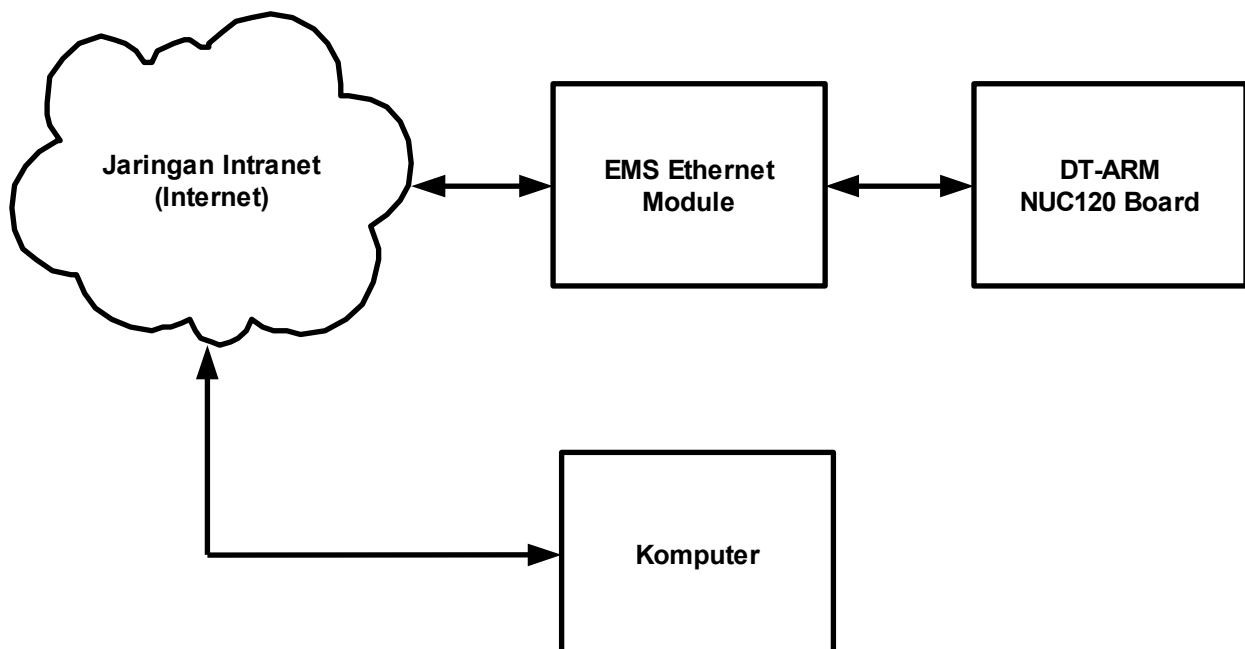
Oleh: Tim IE

Informasi dari website tidak selalu harus menggunakan server komputer yang cepat dan mahal. Bisa saja informasi konten website tersebut hanya memerlukan *resource* yang minim dan tentunya dengan biaya yang minimalis pula. Pada aplikasi ini kita mencoba membangun web server sederhana atau website dengan menggunakan EMS Ethernet Module dan DT-ARM NUC120 Board. EMS Ethernet Module berfungsi sebagai *ethernet controller module* untuk menjembatani komunikasi antara DT-ARM NUC120 Board dengan suatu jaringan komputer melalui antarmuka SPI. Konten web sederhana ditanamkan ke dalam memori *flash* mikrokontroler NUC120RD2BN pada DT-ARM NUC120 Board. Halaman web yang ditanamkan berisi konten web sederhana yang terdiri dari teks dan *link* ke situs tertentu. Program pada DT-ARM NUC120 Board ditulis dalam bahasa C menggunakan *editor Programmer* dan *compiler* CooCox ColDE.

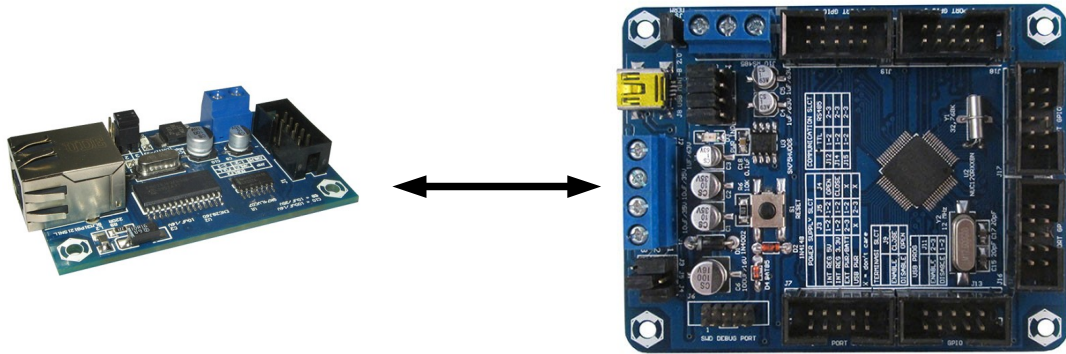
Modul yang digunakan pada aplikasi ini adalah:

- 1 unit DT-ARM NUC120 Board.
- 1 unit EMS Ethernet Module.
- 1 unit adaptor 12VDC
- 1 set kabel LAN dengan konfigurasi *cross over* (umumnya disertakan pada paket pembelian EMS Ethernet Module).
- 1 unit komputer.

Diagram blok sistem secara keseluruhan sebagai berikut:



Gambar 1. Diagram Blok Sistem untuk AN220



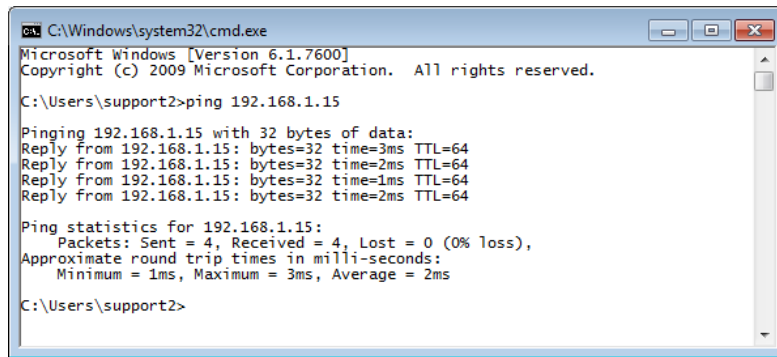
**Gambar 2. Modul Web Server (EMS Ethernet Module + DT-ARM NUC120 Board)**

**H**ubungan antar modul pada aplikasi ini sebagai berikut:

**Tabel 1. Hubungan Detil Antar Modul**

DT-ARM NUC120 Board	EMS Ethernet Module
GND (J19 pin 1)	GND (J1 pin 1)
VCC (J19 pin 2)	VCC (J1 pin 2)
-	CLKOUT (J1 pin 3)
-	- (J1 pin 4)
-	- (J1 pin 5)
PC.15 (J19 pin 10)	RESET (J1 pin 6)
PC.8 (J19 pin 3)	CS (J1 pin 7)
PC.11 (J19 pin 6)	MOSI (J1 pin 8)
PC.10 (J19 pin 5)	MISO (J1 pin 9)
PC.9 (J19 pin 4)	SCK (J1 pin 10)

**M**odul DT-ARM NUC120 Board terhubung dengan EMS Ethernet Module melalui antarmuka SPI melalui pin CS, 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-ARM NUC120 Board, pada J11 ubah ke posisi 2-3 untuk pemrograman melalui USB, hubungkan port usb *micro* pada modul DT-ARM NUC120 Board ke port USB komputer menggunakan kabel USB. Setelah semua modul siap, programlah WebSv\_simple.bin ke dalam DT-ARM NUC120 Board menggunakan program *tools* NuMicro ISP Programming Tool.exe. Setelah program WebSv\_simple.bin berhasil diisikan ke dalam DT-ARM NUC120 Board, hubungkan kabel LAN tipe *cross over* antara EMS Ethernet Module dengan komputer kemudian hubungkan catu daya ke modul DT-ARM NUC120 Board. Tekan tombol reset atau matikan adaptor lalu hidupkan lagi untuk memastikan program dijalankan dari awal. Pastikan juga konfigurasi IP komputer diatur satu grup jaringan IP antara 192.168.1.1 – 192.168.1.14 atau 192.168.1.16 - 192.168.1.253. Sedangkan IP 192.168.1.15 akan digunakan sebagai nomer IP alat aplikasi kita. Selanjutnya lakukan ping 192.168.1.15 melalui *command prompt* komputer, jika EMS Ethernet Module telah terkoneksi dengan komputer maka komputer akan menerima pesan reply from 192.168.1.15 seperti contoh yang ditampilkan pada Gambar 3.



```
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

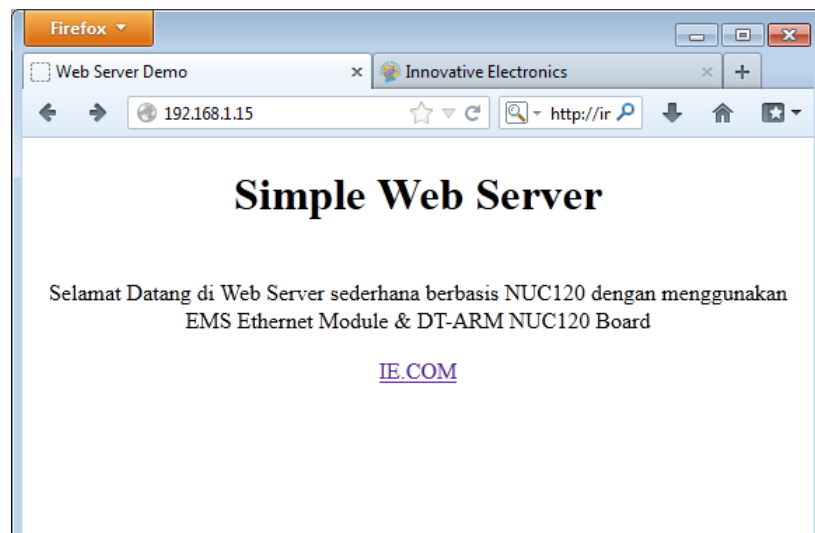
Pinging 192.168.1.15 with 32 bytes of data:
Reply from 192.168.1.15: bytes=32 time=3ms TTL=64
Reply from 192.168.1.15: bytes=32 time=2ms TTL=64
Reply from 192.168.1.15: bytes=32 time=1ms TTL=64
Reply from 192.168.1.15: bytes=32 time=2ms TTL=64

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. Pada contoh aplikasi ini, jika *browser* yang digunakan terdapat pengaturan *proxy*, maka 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* DT-ARM NUC120 Board. Contoh tampilan aplikasi ini menggunakan *browser* Mozilla Firefox dapat dilihat pada Gambar 4.



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

**A**plikasi “Simple Web Server” ini menggunakan protokol koneksi jenis TCP (*Transmission Control Protocol*) dan menggunakan *port* 80 (HTTP). Akan tetapi EMS Ethernet Module bisa kita gunakan pada beberapa layer protokol TCP/IP ataupun hanya layer protokol tertentu tergantung pada program yang kita buat pada modul kontroler. Pada aplikasi ini kita memanfaatkan protokol lapisan antar-host dan protokol lapisan *internetwork*. Protokol lapisan antar-host untuk membuat komunikasi menggunakan sesi koneksi yang bersifat *connection-oriented* seperti TCP. Protokol lapisan *internetwork* untuk melakukan pemetaan (*routing*) dan enkapsulasi paket-paket data jaringan menjadi paket-paket IP seperti Internet Protocol (IP) dan *Address Resolution Protocol* (ARP). Berikut model koneksi dan komunikasi satu paket data web-server:

- Menerima SYN
- Kirim SYN,ACK
- Menerima ACK (ada koneksi)
- Menerima ACK dengan HTTP GET command
- Kirim ACK
- Kirim FIN,ACK dengan HTTP data (contoh: 200 OK)
- Terima FIN,ACK
- Kirim ACK

TCP/IP adalah standar komunikasi data yang digunakan oleh komunitas internet dalam proses tukar-menukar data dari satu komputer ke komputer lain di dalam jaringan Internet. Protokol ini tidaklah dapat berdiri sendiri, karena memang protokol ini berupa kumpulan protokol (*protocol suite*). Protokol ini juga merupakan protokol yang paling

banyak digunakan saat ini. Data tersebut diimplementasikan dalam bentuk perangkat lunak (software) di sistem operasi. Istilah yang diberikan kepada perangkat lunak ini adalah TCP/IP *stack*. TCP/IP mempunyai beberapa protokol untuk menangani komunikasi data:

- IP (*internet protocol*) komunikasi antar komputer atau suatu perangkat jaringan yang berperan dalam petransmisi paket data dari *node* ke *node*.
- TCP (*Transmission Control Protocol*) komunikasi antara aplikasi. TCP dapat mendeteksi *error* atau data yang hilang dan kemudian melakukan transmisi ulang sampai data diterima dengan benar dan lengkap.
- UDP (*User Datagram Protocol*) komunikasi sederhana antara aplikasi.
- IP (*Internet Protocol*) komunikasi antar komputer.
- ICMP (*Internet Control Message Protocol*) untuk kesalahan dan statistik.
- DHCP (*Dynamic Host Configuration Protocol*) untuk pengalamatan dinamis.

Layer TCP IP terdiri atas empat lapis kumpulan protokol yang bertingkat. Keempat layer TCP IP tersebut adalah:

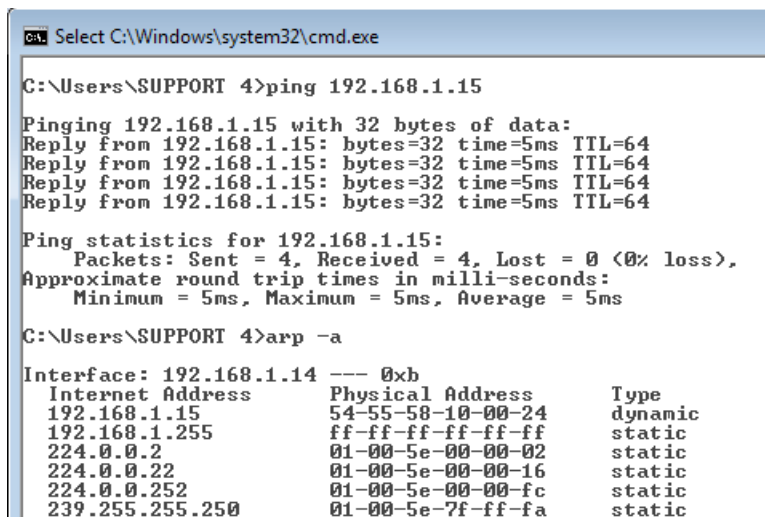
1. *Network Interface Layer*, bertanggungjawab mengirim dan menerima data ke dan dari media fisik.
2. *Internet Layer*, bertanggungjawab dalam proses pengiriman paket ke alamat yang tepat.
3. *Transport Layer*, bertanggungjawab untuk mengadakan komunikasi antara dua host/komputer.
4. *Application Layer*, pada layer inilah terletak semua aplikasi yang menggunakan protokol TCP/IP ini.

Dalam TCP IP, terjadi penyampaian data dari protokol yang berada di satu layer ke protokol yang berada di layer yang lain. Setiap protokol memperlakukan informasi yang diterimanya dari protokol lain sebagai data.

Pada aplikasi ini 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. Keuntungan istimewa pada aplikasi ini *User* diberikan kebebasan untuk melakukan pengaturan konfigurasi MAC Address & IP Address. 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};

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* dapat dilihat pada Gambar 5.



```
C:\Users\SUPPORT 4>ping 192.168.1.15

Pinging 192.168.1.15 with 32 bytes of data:
Reply from 192.168.1.15: bytes=32 time=5ms TTL=64
Reply from 192.168.1.15: bytes=32 time=5ms TTL=64
Reply from 192.168.1.15: bytes=32 time=5ms TTL=64
Reply from 192.168.1.15: bytes=32 time=5ms TTL=64

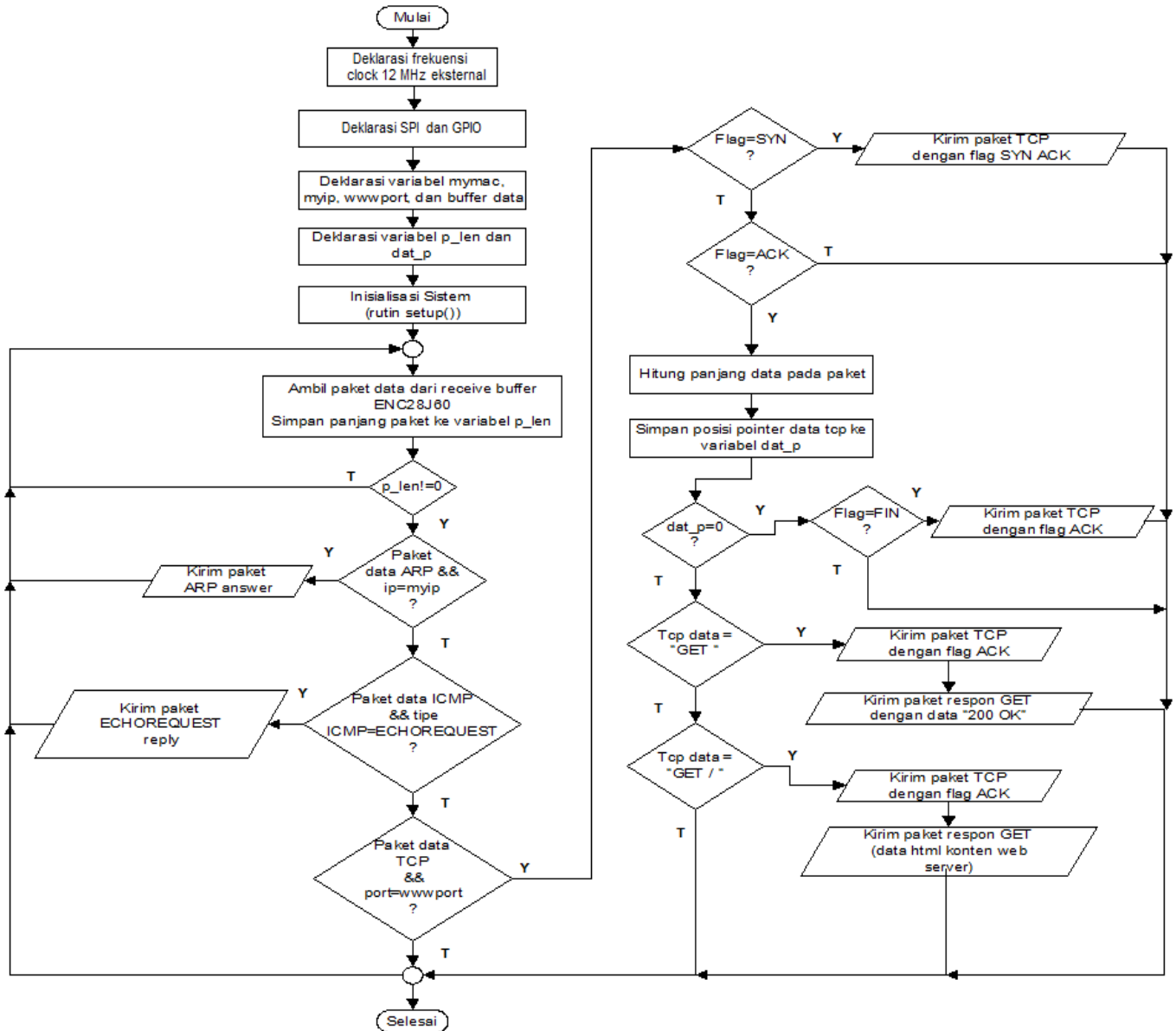
Ping statistics for 192.168.1.15:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 5ms, Maximum = 5ms, Average = 5ms

C:\Users\SUPPORT 4>arp -a

Interface: 192.168.1.14 --- 0xb
Internet Address      Physical Address      Type
192.168.1.15          54-55-58-10-00-24    dynamic
192.168.1.255         ff-ff-ff-ff-ff-ff    static
224.0.0.2             01-00-5e-00-00-02    static
224.0.0.22            01-00-5e-00-00-16    static
224.0.0.252           01-00-5e-00-00-fc    static
239.255.255.250       01-00-5e-7f-ff-fa    static
```

Gambar 5. Tampilan tabel ARP pada command prompt

**F**lowchart program utama (main.c) pada aplikasi ini adalah sebagai berikut:



**Gambar 6. Flowchart program main.c**

Program main.c akan dijalankan sebagai berikut:

1. Pertama kali program mendefinisikan frekuensi *clock* mikrokontroler yang digunakan yaitu pada osilator eksternal 12 MHz (12000000 Hz).
2. Mendefinisikan pin-pin SPI dan GPIO yang digunakan.
3. 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.

4. Kemudian program melakukan inisialisasi sistem yang ditangani oleh rutin setup(). Inisialisasi yang dilakukan meliputi:

- Inisialisasi *chip* ENC28J60.
  - Inisialisasi lampu LED yang terhubung dengan MAG JACK pada EMS Ethernet Module.
  - Inisialisasi *layer* Ethernet atau IP sesuai dengan konfigurasi mymac, myip, dan wwwport.
5. Program melakukan pengecekan isi *receive buffer* ENC28J60. Panjang isi data pada *receive buffer* ENC28J60 disimpan dalam variabel *plen*.
  6. Jika *receive buffer* ENC28J60 kosong maka program akan mengecek kembali isi *receive buffer* ENC28J60.
  7. Jika paket data berupa paket ARP dan alamat IP tujuan sama dengan nilai konfigurasi myip maka program akan mengirimkan paket data *ARP answer*.
  8. 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.
  9. 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*).
  10. 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 DT-ARM NUC120 Board.
  11. Selanjutnya program akan kembali melakukan pengecekan isi *receive buffer* ENC28J60 untuk menentukan proses selanjutnya (kembali ke nomor 5).

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>).

**L**isting program aplikasi ini terdapat pada **AN220.zip**

**S**elamat 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.