

DT-AVR AN214 – Simple Web Server menggunakan DT-AVR Inoduino dan WIZ820io

Oleh: Tim IE

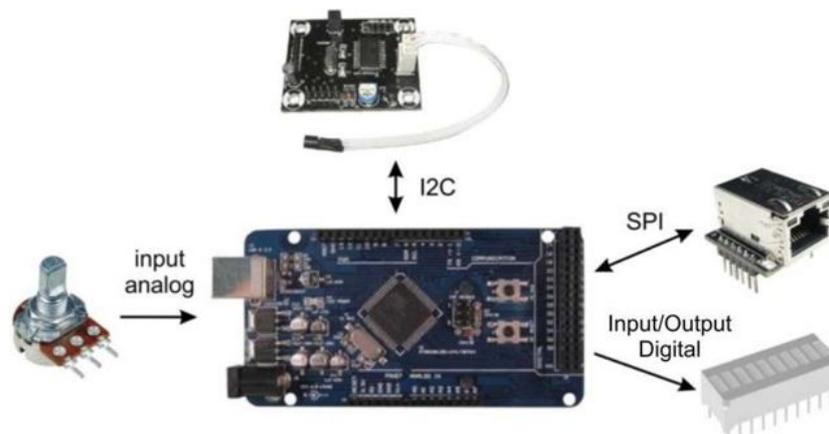
Aplikasi kali akan membahas pembuatan *web server* sederhana menggunakan DT-AVR Inoduino sebagai kontroler utama dan WIZ820io sebagai kontroler *ethernet*. Adapun AN214 ini merupakan pengembangan lanjutan dari AN213 sebelumnya. Apabila proses komunikasi pada AN213 hanya berupa transfer *text* dengan protokol yang kita definisikan sendiri, maka pada AN214 proses transfer *text* tersebut harus disesuaikan dengan protokol HTTP agar dapat dikenali dan ditampilkan oleh *web browser*.

Berikut adalah modul/komponen yang akan digunakan dalam aplikasi ini:

- 1x DT-AVR Inoduino
- 1x DT-Sense Temperature Sensor
- 1x WIZ820io
- 1x Kabel LAN tipe *cross*
- 1x Kabel USB tipe A-B
- Beberapa kabel *jumper*
- 1x *Power supply* 12V
- 1x Potensiometer 50 kOhm
- 4x LED
- 4x Resistor 220 Ohm
- 2x *Push button*
- Komputer

Pemrograman pada AN214 ini dapat dibagi menjadi 2 yaitu pada sisi *microcontroller* (DT-AVR Inoduino) yang bertindak sebagai *server* dan pada sisi program *web browser* yang bertindak sebagai *client*. Pemrograman pada sisi *server* ditujukan untuk melakukan pembacaan *input digital*, *input analog*, data temperatur dari DT-Sense Temperature Sensor, serta pengaturan nilai PWM yang akan dikeluarkan sesuai dengan permintaan *user* melalui *web browser*. Pemrograman pada sisi *client* ditujukan untuk mengatur tampilan data pada *user* serta sebagai media pengaturan nilai PWM *microcontroller*.

Adapun blok diagram dari aplikasi ini adalah sebagai berikut :



Gambar 1
Blok Diagram AN214

Hubungan antar modul adalah sebagai berikut :

DT-AVR Inoduino	WIZ820io *
GND	Pin 2 J1 (GND)
3.3V	Pin 3 J2 (3V3D)
Reset	Pin 5 J2 (nReset)
Pin 22 (SS)	Pin 5 J1 (SS)
Pin 23 (SCK)	Pin 4 J1 (SCK)
Pin 24 (MOSI)	Pin 3 J1 (MOSI)
Pin 25 (MISO)	Pin 6 J2 (MISO)

*) penomoran dan peletakan *pin* terdapat pada manual WIZ820io

Tabel 1
Hubungan DT-AVR Inoduino dengan WIZ820io

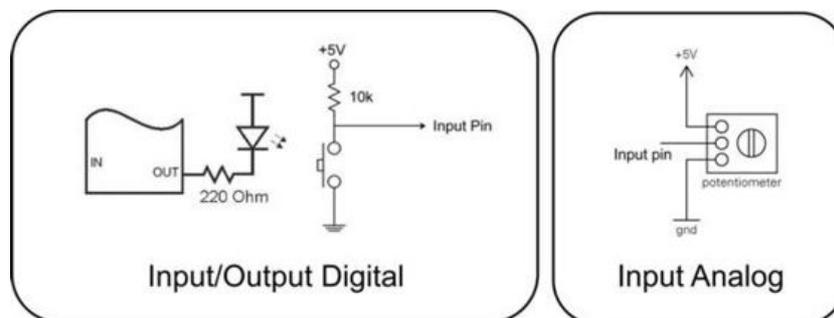
DT-AVR Inoduino	DT-Sense Temperature Sensor *
GND	Pin 1 J1 (GND)
5V	Pin 2 J1 (VCC)
Pin 5 (SCL)	Pin 5 J1 (SDA)
Pin 6 (SDA)	Pin 6 J1 (SCL)

*) penomoran dan peletakan *pin* terdapat pada manual DT-Sense Temperature Sensor

Tabel 2
Hubungan DT-AVR Inoduino dengan DT-Sense Temperature Sensor

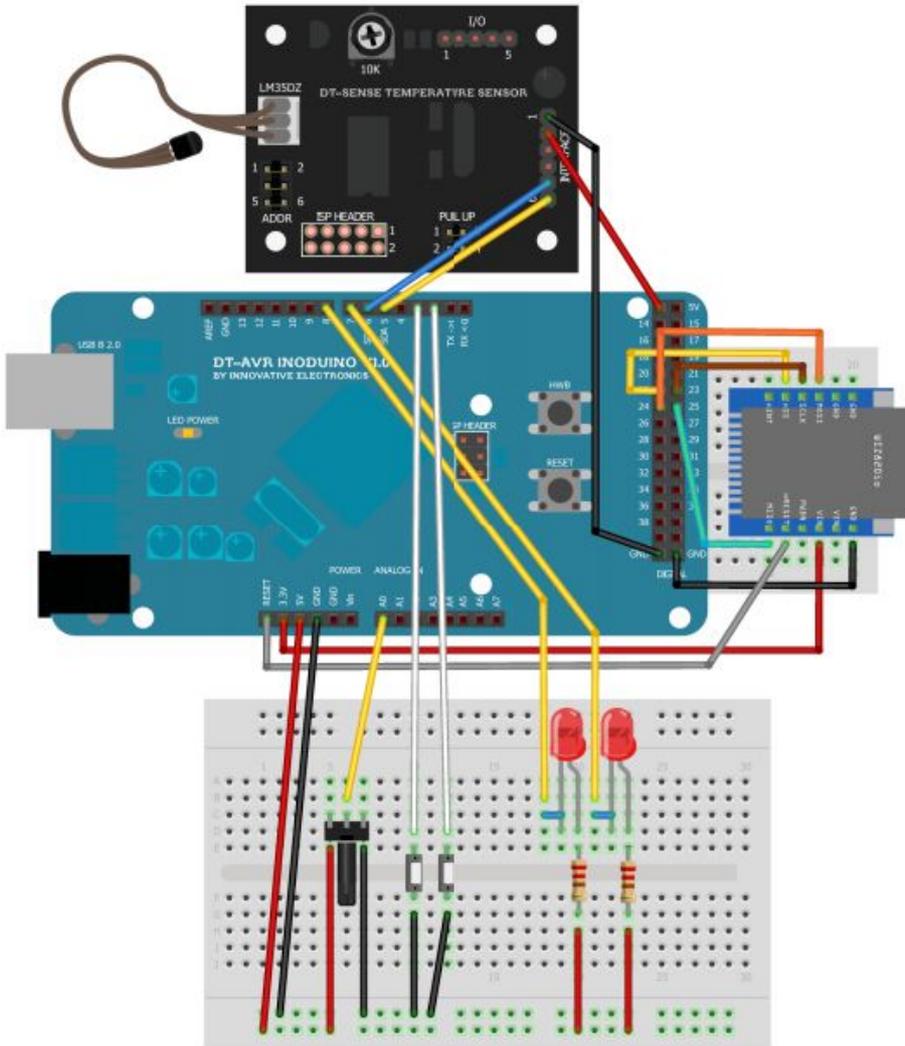
Adapun tujuan dari aplikasi ini adalah sama dengan AN213 sebelumnya yaitu untuk membuat sistem pengaturan serta pembacaan jalur *input/output* secara *remote*. Maka dari itu kita membutuhkan rangkaian perangkat *input/output* sebagai aktuator. Adapun rangkaian perangkat *input/output* yang digunakan pada aplikasi ini adalah seperti yang diilustrasikan pada **Gambar 2**.

Alokasi *pin* untuk *input digital* pada aplikasi ini adalah *pin* 0 – 3, *pin* 14 – 21, dan *pin* 26 – 38. Alokasi *pin* untuk *output* PWM adalah *pin* 7 – 13, sedangkan alokasi *pin* untuk *input analog* adalah *pin* A0 – A7. Pada *list* komponen di halaman sebelumnya, perangkat *input/ouput* yang digunakan adalah 4 buah LED, 2 buah *Push button*, dan 1 buah Potensiometer. Komponen-komponen tersebut dapat ditambah sesuai kebutuhan mengikuti contoh ilustrasi pada **Gambar 2**.



Gambar 2
Contoh rangkaian *input/output digital* dan *input analog*

Setelah menghubungkan modul-modul di atas menggunakan kabel *jumper*, lakukan pengecekan kembali menggunakan *multimeter*, apakah koneksi antar modul sudah benar atau tidak. Pastikan juga tidak terdapat hubungan singkat antara jalur VCC dan GND sebelum memberikan catu daya. Ilustrasi koneksi antar modul terdapat pada **Gambar 3**.



Gambar 3
Ilustrasi koneksi antar modul pada AN214

Agar modul DT-Sense Temperature Sensor dapat berkomunikasi melalui jalur I2C dengan baik, perlu dilakukan pengaturan *jumper* J2 untuk mengaktifkan resistor *pull-up* serta pengaturan *jumper* J3 untuk penyesuaian *address*. Konfigurasi *jumper-jumper* tersebut adalah seperti pada **Gambar 4** dan **Gambar 5**.

Jumper PULL-UP J2	Fungsi
	Pull-up tidak aktif (<i>jumper</i> terlepas)
	Pull-up aktif (<i>jumper</i> terpasang)

Gambar 4
Konfigurasi *jumper pull-up* pada DT-Sense Temperature Sensor

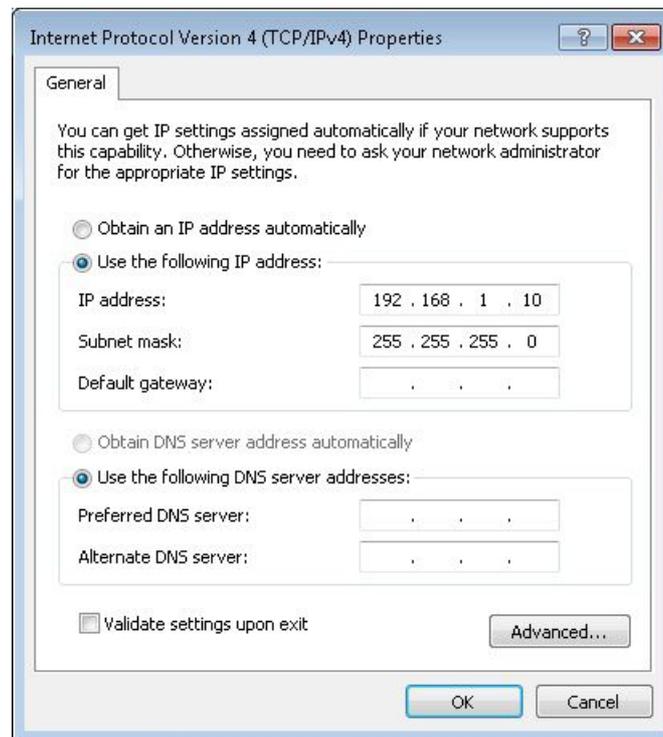
J3 (A2) Pin 5-6	J3 (A1) Pin 3-4	J3(A0) Pin 1-2	Alamat I ² C	
			Alamat Tulis I ² C	Alamat Baca I ² C
■	■	■	E0H	E1H
■	■		E2H	E3H
■		■	E4H	E5H
■			E6H	E7H
	■	■	E8H	E9H
	■		EAH	EBH
		■	ECH	EDH
			EEH	EFH

Keterangan:

■ : jumper terpasang

Gambar 5
Konfigurasi jumper address pada DT-Sense Temperature Sensor

Setelah konfigurasi jumper pada DT-Sense Temperature selesai, langkah selanjutnya adalah melakukan konfigurasi pada komputer yang akan digunakan. Proses konfigurasi pada komputer hanyalah menentukan alamat IP saja. Adapun nilai tersebut adalah seperti pada **Gambar 6**.

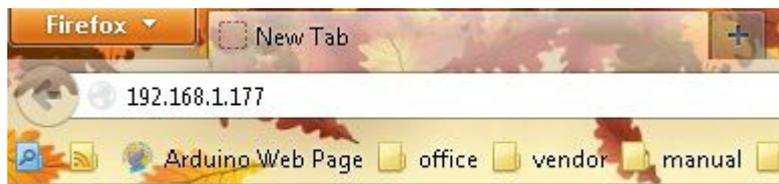


Gambar 6
Pengaturan alamat IP pada komputer

Apabila konfigurasi di atas sudah selesai, lanjutkan dengan mengikuti beberapa langkah berikut:

1. Copy dan extract Arduino IDE yang terdapat pada DVD produk DT-AVR Inoduino ke *local drive* komputer yang digunakan.
2. Buka *folder library* Arduino pada **folder_instalasi_arduino_IE\libraries**, hapus *folder* Ethernet. Apabila kedepannya *library* Ethernet diperlukan kembali, *library* tersebut dapat diambil melalui Arduino IDE pada DVD produk DT-AVR Inoduino.
3. Extract file AN214.zip, masuk ke *folder inoWebserver*, kemudian copy *folder Ethernet* ke **folder_instalasi_arduino_IE\libraries**.

4. Buka file **inoWebserver.ino** menggunakan Arduino IDE yang telah dimodifikasi oleh Innovative Electronics. Sesuaikan pilihan *board* Arduino, yaitu DT-AVR Induino, kemudian lakukan proses *compile*. Apabila tidak terdapat kesalahan, proses *compile* tidak akan memunculkan pesan *error*. Apabila muncul pesan *error*, teliti kembali apakah proses pada langkah sebelumnya sudah benar atau tidak.
5. Hubungkan DT-AVR Induino dengan *power supply* 12 VDC, lalu hubungkan DT-AVR Induino dengan komputer menggunakan kabel USB tipe A-B.
6. Sesuaikan nomor Serial Port yang akan digunakan untuk proses *upload* program.
7. Lakukan proses *upload sketch inoWebserver.ino*.
8. Agar komunikasi antara rangkaian sistem dengan komputer berjalan dengan lancar, matikan semua koneksi jaringan kecuali koneksi LAN yang akan digunakan.
9. Hubungkan WIZ820io dengan konektor LAN komputer menggunakan kabel LAN yang memiliki konfigurasi *cross*.
10. Buka program *web browser* yang biasa digunakan (dalam aplikasi ini, *web browser* yang digunakan adalah Mozilla Firefox versi 19.0.2), kemudian ketikkan "192.168.1.177" pada *address bar* kemudian tekan tombol *enter* pada *keyboard*.



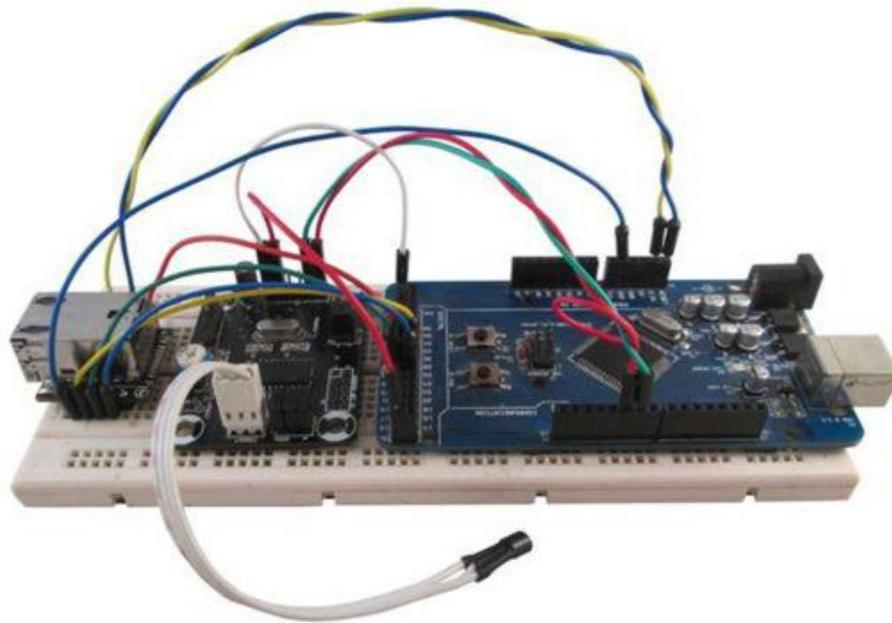
Gambar 7
Penulisan IP address pada address bar web browser

11. Apabila koneksi berhasil, maka akan ditampilkan halaman *web* dari DT-AVR Induino seperti pada **Gambar 8**.



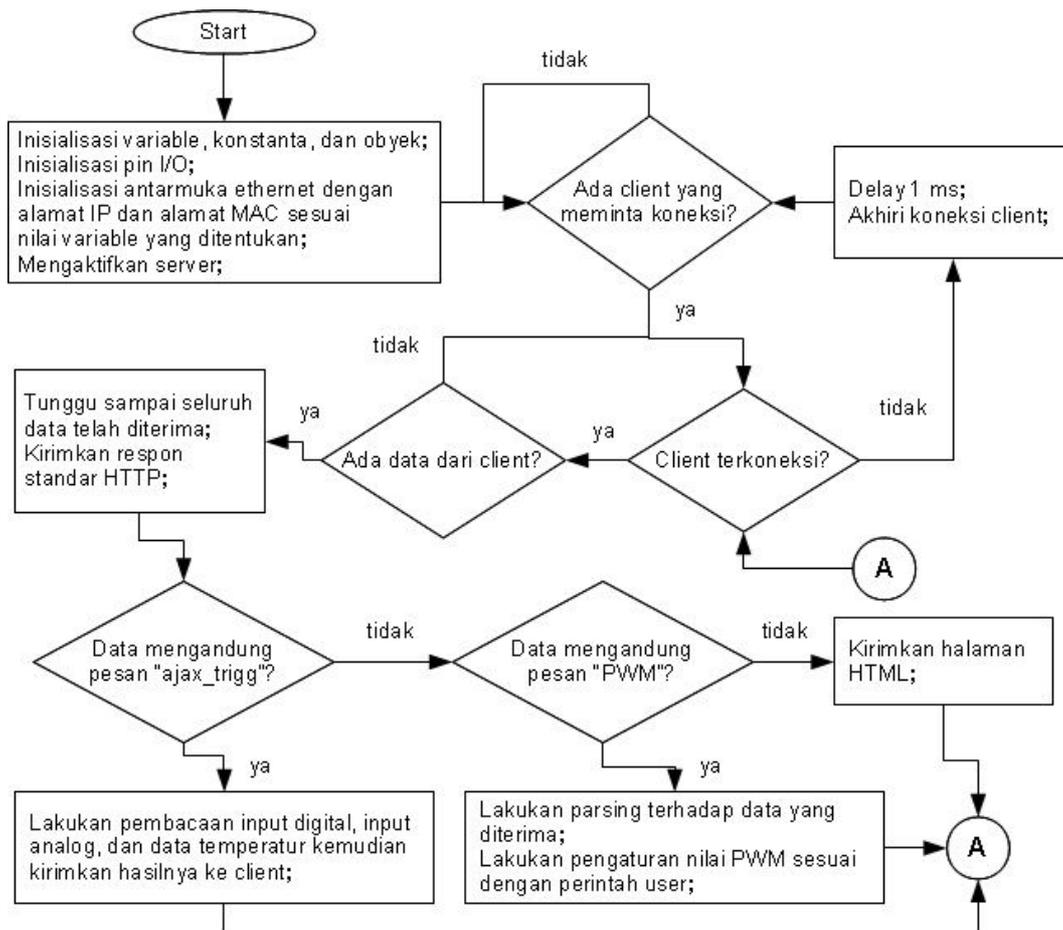
Gambar 8
Tampilan halaman web dari DT-AVR Induino

12. Pada halaman *web* yang ditampilkan terdapat tiga tabel yang masing-masing menginformasikan hasil pembacaan kondisi *pin input digital*, *pin input analog*, serta hasil pembacaan suhu oleh DT-Sense Temperature Sensor (dalam satuan celcius). Pengaturan nilai PWM pada *pin* 7 – 13 dilakukan melalui 7 buah *input box* yang tersedia dengan mengisi nilai terlebih dahulu (0 – 255) kemudian menekan tombol *enter* pada *keyboard*.



Gambar 9
Rangkaian antar modul pada AN214

Adapun alur kerja dari program pada server, **inoWebserver.ino**, adalah sebagai berikut :



Gambar 10
Flowchart program inoWebserver.ino

Penjelasan urutan kerja dari program **inoWebserver.ino** adalah sebagai berikut :

1. Program melakukan inisialisasi awal konstanta, *variable*, serta obyek yang akan digunakan.
2. Program melakukan inisialisasi *pin I/O* yang akan digunakan.
3. Program memulai koneksi *ethernet* dengan konfigurasi alamat **MAC** dan **IP** sesuai nilai *variable* yang ditentukan pada langkah 1.
4. Program menunggu sampai ada permintaan koneksi dari *client* (*web browser*). Apabila terdapat permintaan koneksi maka proses berlanjut ke langkah 5.
5. Program akan memeriksa apakah *client* dalam keadaan terkoneksi atau tidak. Jika *client* terkoneksi, maka program akan berlanjut ke langkah 6. Jika *client* tidak terkoneksi, maka *server* (DT-AVR Inoduino) akan menutup koneksi dan akan kembali ke langkah 4.
6. Program akan memeriksa apakah ada data dari *client* atau tidak. Apabila ada maka program akan berlanjut ke langkah 7. Jika tidak ada data dari *client*, maka program akan kembali ke langkah 5.
7. Program akan menunggu sampai data dari *client* selesai diterima. Apabila data telah selesai diterima, maka *server* akan mengirimkan respon standar protokol HTTP.
8. Program pada *server* akan melakukan pemeriksaan terhadap data yang diterima dari *client*. Apabila data tersebut mengandung *string* "ajax_trigg", program berlanjut ke langkah 9. Apabila data dari *client* mengandung *string* "PWM", program berlanjut ke langkah 10. Jika data dari *client* tidak mengandung kedua *string* yang telah disebutkan sebelumnya, *server* akan mengirimkan seluruh kode halaman HTML ke *client*.
9. *Server* akan melakukan pembacaan *pin I/O analog* dan *digital* serta data temperatur dari DT-Sense Temperature Sensor yang kemudian akan dikirimkan ke *client*. Alur program pada *server* kembali ke langkah 5.
10. *Server* melakukan pengaturan nilai PWM sesuai dengan perintah dari *client*. *Server* melakukan *parsing* data untuk mengetahui nilai yang diharapkan serta nomor *pin* yang dituju oleh *client*. Alur program pada *server* kembali ke langkah 5.

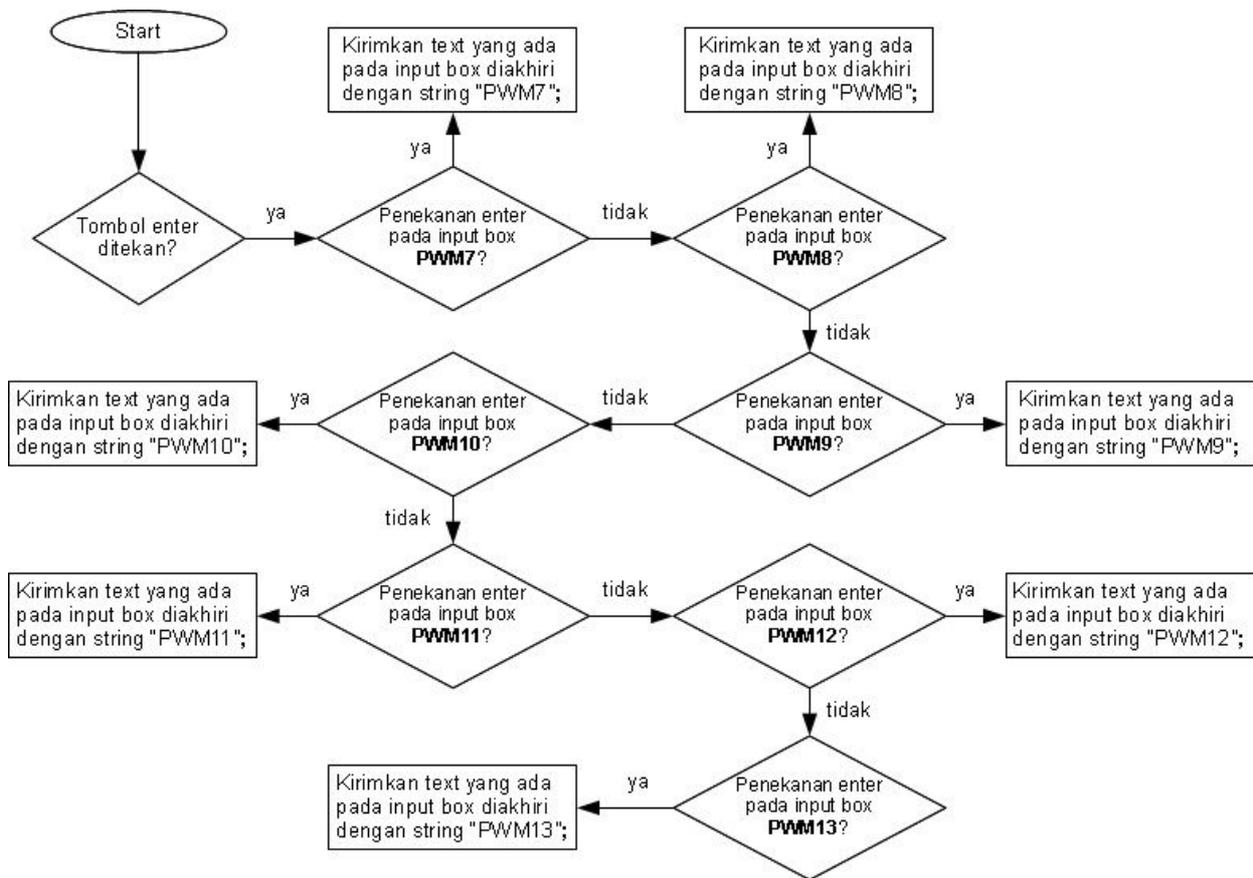
Terdapat dua bagian utama kode program/*script* pada halaman HTML yang dikirimkan oleh *server* ke *client*. Adapun *script* tersebut dijalankan sepenuhnya oleh *client*. Masing-masing bagian tersebut adalah fungsi **AjaxTrigger** dan fungsi **sendPWM**. Fungsi **AjaxTrigger** digunakan oleh *client* untuk meminta data hasil pembacaan *input digital*, *input analog*, serta data temperatur dari DT-Sense Temperature Sensor dari *server*. Fungsi **sendPWM** digunakan oleh *client* untuk mengirimkan perintah pengaturan nilai PWM ke *server*. Alur kerja dari kedua fungsi tersebut adalah seperti pada **Gambar 11** dan **Gambar 12**.



Gambar 11
Flowchart fungsi AjaxTrigger pada halaman HTML client

Penjelasan urutan kerja dari fungsi **AjaxTrigger** adalah sebagai berikut :

1. Fungsi terlebih dahulu memeriksa apakah ada balasan/respon dari *server*. Apabila *client* menerima respon dari *server*, maka data *string* yang dikirimkan oleh *server* ke *client* ditampung terlebih dahulu ke dalam suatu *variable array*. Data tersebut kemudian di-*parsing* untuk dipisah-pisah dan kemudian dituliskan pada halaman *web*.
2. *Client* mengirimkan *string* "ajax_trigg" ke *server* untuk meminta hasil pembacaan kondisi *pin I/O* dan temperatur oleh *server*. Fungsi secara otomatis kembali ke langkah 1 dengan jeda waktu 1 detik.



Gambar 12
Flowchart fungsi sendPWM pada halaman HTML client

Penjelasan urutan kerja dari fungsi **sendPWM** adalah sebagai berikut :

1. Fungsi **sendPWM** akan dijalankan oleh *client* saat terjadi penekanan tombol *keyboard* oleh *user* pada salah satu kolom *input box* di halaman *web*. Apabila terdapat penekanan tombol, fungsi ini akan memeriksa apakah tombol yang ditekan adalah tombol *enter* atau bukan. Jika tombol yang ditekan bukanlah tombol *enter*, maka fungsi tetap berjalan pada langkah 1. Apabila tombol yang ditekan adalah tombol *enter*, maka proses berlanjut ke langkah 2.
2. Fungsi ini selanjutnya akan mendeteksi *input box* mana yang digunakan oleh *user* untuk mengirimkan nilai PWM. Jika *input box* yang digunakan adalah *input box PWM 12*, maka fungsi ini akan mengirimkan *text* pada *input box* diikuti dengan *string* "PWM12". Hal yang sama dilakukan juga *input box* lainnya.
3. Fungsi akan kembali ke langkah 1 saat terjadi penekanan tombol *keyboard* pada *input box*.

Listing program aplikasi ini terdapat pada **AN214.ZIP**

Selamat berinovasi!

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