

## DT-AVR AN210 – Temperature Logger berbasis DT-AVR Inoduno dengan SD Card sebagai Media Penyimpanan Data

Oleh: Tim IE

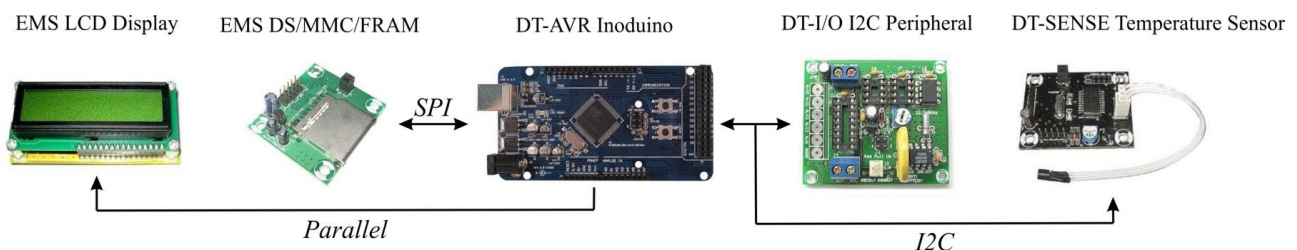
**S**D Card saat ini merupakan jenis memori *non-volatile* yang sudah umum digunakan pada berbagai *device* seperti *handphone*, *music player*, kamera digital, *video recorder*, *smart TV*, dll. Sifat universal tersebut merupakan salah satu alasan mengapa penggunaan SD Card juga populer pada berbagai *embedded system*. Selain sifat universal, alasan lain penggunaan SD Card adalah tingkat ketersediaan yang tinggi serta harga yang relatif lebih ekonomis dibandingkan dengan memori *non-volatile* yang masih dalam kemasan IC. *Application Note* kali ini akan menyampaikan contoh penggunaan SD Card pada aplikasi *Temperature Logger*. Aplikasi ini akan melakukan pembacaan data suhu ruangan kemudian menyimpan hasil pembacaan tersebut ke dalam SD Card dengan format *file XML*. Tiap data suhu akan disertai dengan informasi mengenai nomor pencatatan serta waktu pembacaannya.

Berikut adalah perlengkapan yang digunakan dalam aplikasi ini :

- 1x DT-AVR Inoduno
- 1x DT-I/O I2C Peripheral
- 1x DT-SENSE Temperature Sensor
- 1x EMS SD/MMC/FRAM
- 1x EMS LCD Display
- 1x SD Card
- 1x Komputer
- 1x *Breadboard*
- 1x Kabel USB A-B
- 1x Beberapa kabel *jumper*
- 1x *Power supply* 12 VDC

Modul mikrokontroler yang digunakan kali ini adalah DT-AVR Inoduno, yaitu modul Arduino *compatible* berbasis *chip* AT90USB1286. DT-AVR Inoduno akan difungsikan untuk melakukan pembacaan nilai suhu serta informasi waktu dari DT-SENSE Temperature Sensor dan DT-I/O I2C Peripheral. Antarmuka yang digunakan untuk pengaksesan kedua modul tersebut oleh DT-AVR Inoduno adalah I2C. Sedangkan untuk keperluan penamplan data suhu dan waktu, akan digunakan modul EMS LCD Display yang dapat diakses dengan antarmuka *parallel*. Setelah nilai suhu dan informasi waktu sudah didapatkan, informasi-informasi tersebut akan dituliskan pada SD Card dengan interval waktu 1 menit. Adapun tegangan operasional dari SD Card adalah 3.3 VDC. Karena tegangan operasi DT-AVR Inoduno adalah 5 VDC, akan digunakan EMS SD/MMC/FRAM sebagai *adapter* untuk mengubah nilai tegangan tersebut agar sesuai dengan nilai tegangan operasi SD Card. Antarmuka yang digunakan untuk mengakses SD Card adalah SPI.

**A**dapun blok diagram dari aplikasi ini adalah sebagai berikut :



**Gambar 1**  
Blok diagram AN210

**H**ubungan antar modul adalah sebagai berikut :

DT-AVR Inoduino	DT-I/O I2C Peripheral	DT-SENSE Temperature Sensor
GND / DGND	GND (J1)	GND (J1 pin 1)
5V / VCC	VCC (J1)	VCC (J1 pin 2)
Digital pin 5 / SCL (J4)	SCL (J4)	SCL (J1 pin 6)
Digital pin 6 / SDA (J4)	SDA (J4)	SDA (J1 pin 5)
Digital pin 32 (J5)	SQWOUT (J2)	-

**Tabel 1**  
**Hubungan DT-AVR Inoduino dengan DT-I/O I2C Peripheral dan DT-SENSE Temperature Sensor**

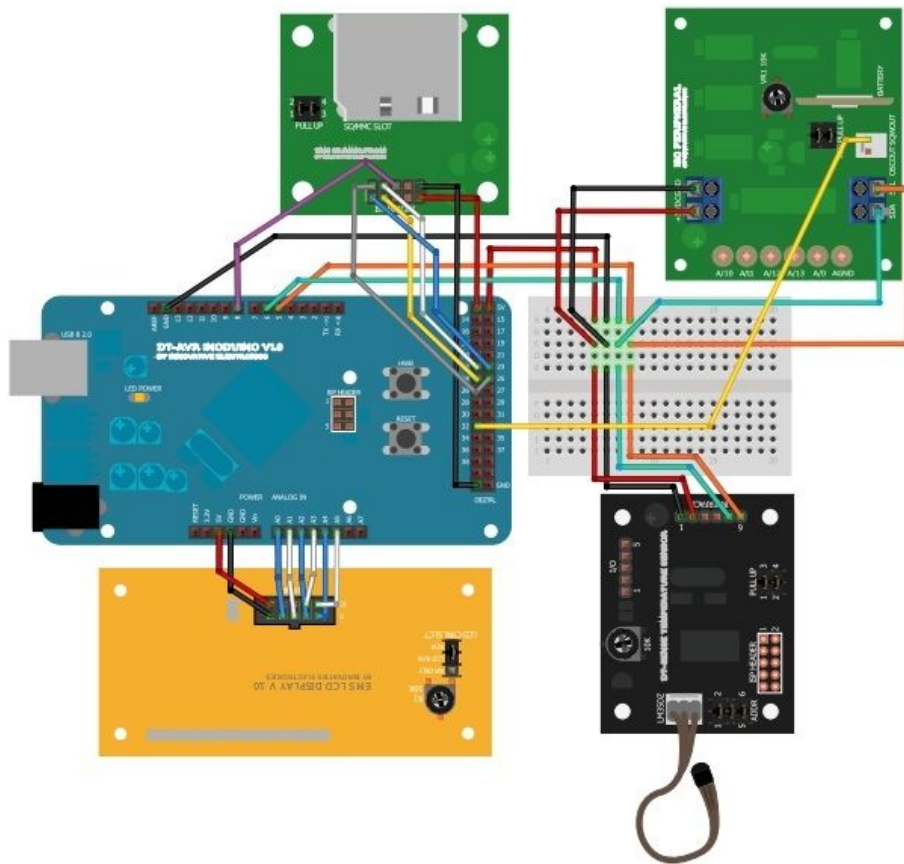
DT-AVR Inoduino	EMS SD/MMC/FRAM
GND / DGND	GND (J2 pin 1)
5V / VCC	+5 V (J2 pin 2)
Digital pin 8 (J3)	CD (J2 pin 5)
Digital pin 22 (J5)	CSSD (J2 pin 7)
Digital pin 23 (J5)	SCK (J2 pin 10)
Digital pin 24 (J5)	MOSI (J2 pin 8)
Digital pin 25 (J5)	MISO (J2 pin 9)

**Tabel 2**  
**Hubungan DT-AVR Inoduino dengan EMS SD/MMC/FRAM**

DT-AVR Inoduino	EMS LCD Display
GND / DGND	GND (J3 pin 1)
5V / VCC	+5 V (J3 pin 2)
A0 / Digital pin 39 (J7)	RS (J3 pin 3)
A1 / Digital pin 40 (J7)	E (J3 pin 5)
A2 / Digital pin 41 (J7)	DB4 (J3 pin 7)
A3 / Digital pin 42 (J7)	DB5 (J3 pin 8)
A4 / Digital pin 43 (J7)	DB6 (J3 pin 9)
A5 / Digital pin 44 (J7)	DB7 (J3 pin 10)

**Tabel 3**  
**Hubungan DT-AVR Inoduino dengan EMS LCD Display**

Setelah menghubungkan modul-modul tersebut menggunakan kabel *jumper*, lakukan pengecekan kembali menggunakan *multimeter*, untuk memastikan apakah koneksi antar modul sudah benar atau tidak. Pastikan juga bahwa tidak terjadi hubungan singkat antara VCC dan GND, karena hal tersebut dapat menyebabkan kerusakan pada modul. Ilustrasi koneksi antar modul terdapat pada **Gambar 2**.



**Gambar 2**  
**Hubungan antar modul pada AN210**

**M**odul-modul diatas perlu dikonfigurasi terlebih dahulu agar dapat bekerja sesuai dengan yang diharapkan. Berikut adalah beberapa langkah konfigurasi yang perlu dilakukan :

- **DT-I/O I2C Peripheral**

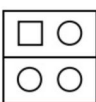
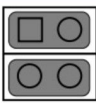
Antarmuka I2C membutuhkan 2 buah resistor *pull-up* yang masing-masing diletakkan pada jalur SDA dan SCL. Kedua modul I2C yang digunakan pada aplikasi ini memiliki resistor *pull-up* yang dapat diaktifkan menggunakan *jumper*. Maka dari itu hanya resistor *pull-up* pada salah satu modul saja yang perlu diaktifkan. Agar konfigurasi *hardware* seragam, *jumper pull-up* yang diaktifkan pada aplikasi kali ini adalah *jumper pull-up* pada DT-SENSE Temperature Sensor. Non-aktifkan resistor *pull-up* pada DT-I/O I2C Peripheral dengan cara melepas *jumper* J3 seperti **Gambar 3** di bawah.

<b>Jumper PULL-UP J3</b>	<b>Fungsi</b>
	<i>Pull-up</i> tidak aktif ( <i>jumper</i> terlepas)
	<i>Pull-up</i> aktif ( <i>jumper</i> terpasang)

**Gambar 3**  
**Pengaturan *jumper* pada DT-I/O I2C Peripheral**

- **DT-SENSE Temperature Sensor**

Aktifkan resistor *pull-up* dengan cara memasang *jumper* J2. Setelah itu, konfigurasi *jumper* J3 agar alamat I2C modul bernilai E0H untuk operasi tulis dan E1H untuk operasi baca. Pengaturan *jumper* J2 dan J3 dapat dilakukan sesuai **Gambar 4** dan **Gambar 5**.

Jumper PULL-UP J2	Fungsi
	Pull-up tidak aktif (jumper terlepas)
	Pull-up aktif (jumper terpasang)

**Gambar 4**  
Pengaturan jumper resistor pull-up pada DT-SENSE Temperature Sensor

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


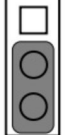
Keterangan:

■ : jumper terpasang

**Gambar 5**  
Pengaturan jumper alamat I2C pada DT-SENSE Temperature Sensor

- **EMS LCD Display**

Prosedur menampilkan data suhu dan waktu pada LCD hanya dilakukan dengan operasi penulisan saja. Maka dari itu, sesuaikan jumper J2 pada EMS LCD Display agar bekerja pada mode "WR Only" seperti pada Gambar 6.

Jumper J2	Operasi yang dapat dilakukan pada modul LCD
	<b>Operasi write only</b> (Pin R/W LCD terhubung ke GND, pin R/W pada J3 tidak terhubung ke mana-mana)
	<b>Operasi read/write</b> (Pin R/W LCD terhubung ke pin R/W pada J3)

**Gambar 6**  
Pengaturan jumper pada EMS LCD Display

Setelah proses konfigurasi modul di atas selesai dilakukan, lakukan langkah-langkah berikut ini :

1. Jika belum memiliki Arduino IDE versi 1.0.1 dari Innovative Electronics, unduh terlebih dahulu *software* tersebut melalui *website* Innovative Electronics. Arduino IDE tersebut telah dimodifikasi agar dapat memberikan dukungan penuh terhadap penggunaan DT-AVR Inoduino.
2. Extract *file* "arduinoIDE-1.0.1.zip".
3. Extract *file* "AN210Lib.zip" ke *folder* "arduino-1.0.1\libraries".
4. Buka *file* "TemperatureLogger.ino" yang terdapat pada *folder* "TemperatureLogger" menggunakan *software* Arduino IDE.
5. Sesuaikan jenis *board* pada Arduino IDE menjadi DT-AVR Inoduino, kemudian lakukan proses *verify*. Apabila tidak terdapat kesalahan, maka akan keluar peringatan *Done Compiling* pada akhir proses.
6. Hubungkan DT-AVR Inoduino dengan komputer menggunakan kabel USB A-B.
7. Berikan catu daya 12 VDC dari *power supply* ke DT-AVR Inoduino. Kemudian masukkan DT-AVR Inoduino ke mode *bootloader*.
8. Komputer akan mendeteksi koneksi USB DT-AVR Inoduino sebagai COM *port* pada *device manager*. Sesuaikan nilai COM *port* pada *device manager* dengan nilai COM pada Arduino IDE, kemudian lakukan proses *upload*.
9. Apabila konfigurasi *hardware* telah dilakukan dengan benar dan tidak terdapat kesalahan pada proses *upload* program, tampilan LCD akan tampak seperti pada **Gambar 5**.



**Gambar 5**  
Tampilan LCD jika SD Card tidak terdeteksi pada saat program pertama kali dieksekusi

10. Masukkan SD Card (pada aplikasi ini, format SD Card yang digunakan adalah FAT16 dengan ukuran *allocation unit* sebesar 8192 bytes) pada EMS SD/MMC/FRAM sehingga tampilan LCD akan berubah seperti **Gambar 6**.



**Gambar 6**  
Tampilan LCD saat SD Card sudah terdeteksi

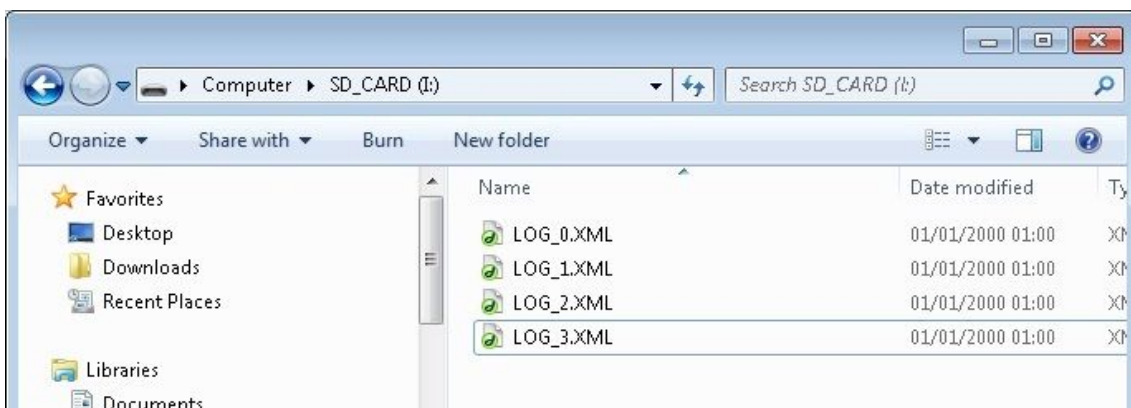
11. Pada saat ini proses *logging* data akan dilakukan tiap satu menit sekali. Jika SD Card dilepas, maka proses pemantauan suhu akan tetap dilakukan dan ditampilkan pada LCD, namun proses *logging* tidak akan dilakukan. **Gambar 7** merupakan tampilan LCD saat SD Card dilepas dari EMS SD/MMC/FRAM.



**Gambar 7**

Tampilan LCD jika SD Card tidak terdeteksi setelah program sudah dieksekusi beberapa saat

- Lakukan proses *logging* data selama beberapa menit, kemudian lepaskan SD Card dari EMS SD/MMC/FRAM. Periksa isi SD Card menggunakan bantuan komputer. Adapun *file logging* data oleh sistem akan diletakkan pada bagian terluar dari SD Card seperti pada **Gambar 8**.



**Gambar 8**

*File hasil logging suhu yang terdapat pada SD Card*

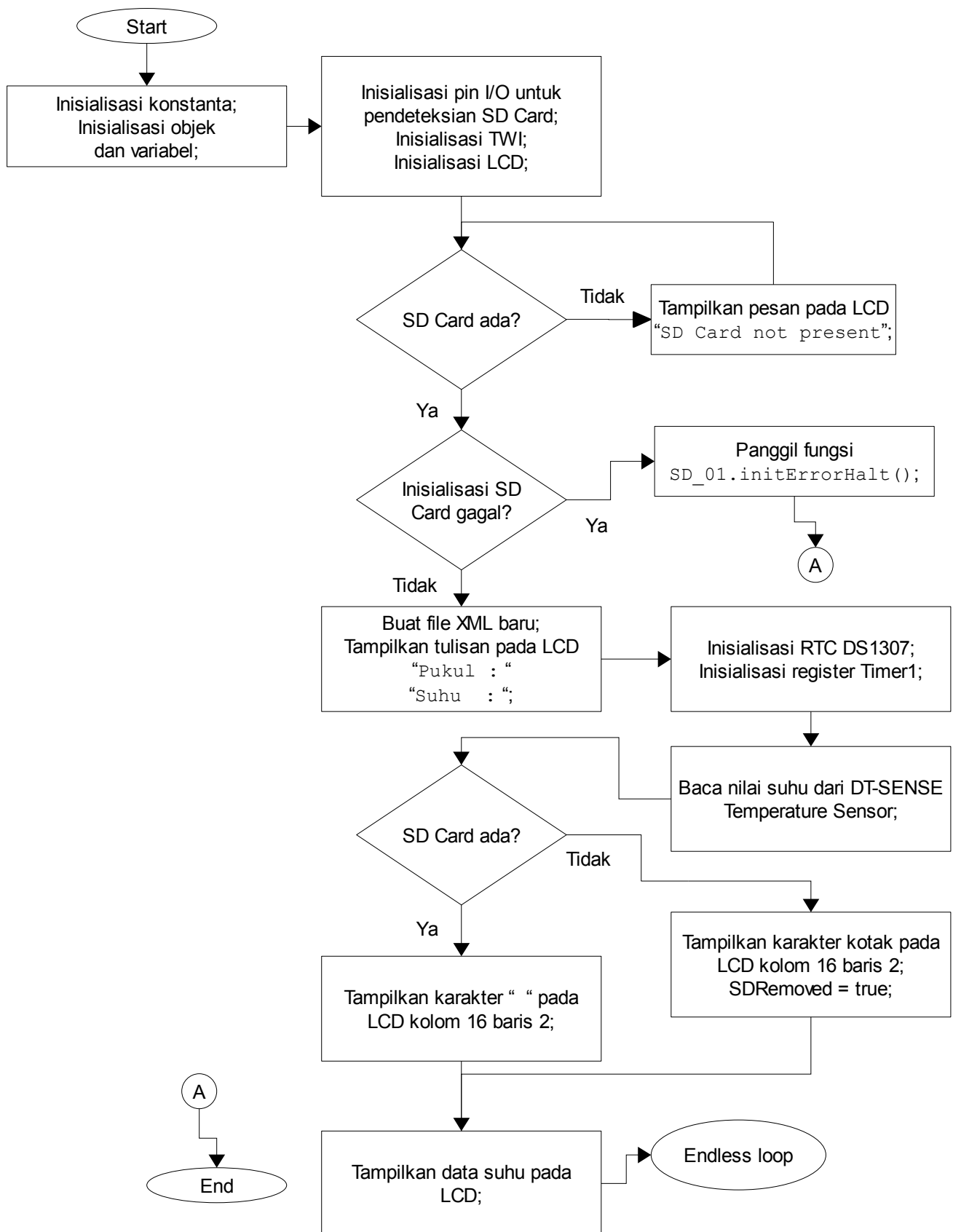
- Pada **Gambar 8** terdapat 4 buah *file XML*. *File XML* baru akan diciptakan setiap sistem ini baru dinyalakan. Adapun dalam contoh ini, *file XML* yang terbaru adalah "LOG\_3.XML". Gunakan *web browser*, Microsoft Excel, Notepad, atau program lain yang mendukung format XML untuk membaca data dari *file* tersebut. **Gambar 9** merupakan tampilan dari "LOG\_3.XML" yang dibuka menggunakan *software XML Editor* yang dapat diunduh gratis melalui website <http://www.asaapplications.com/>.

NO.	TIME.	DATA
1	0:45:43	26
2	0:46:43	26
3	0:47:43	26
4	0:48:43	26
5	0:49:43	26
6	0:50:43	26
7	0:51:43	26
8	0:52:43	26
9	0:53:43	26
10	0:54:43	26

**Gambar 9**

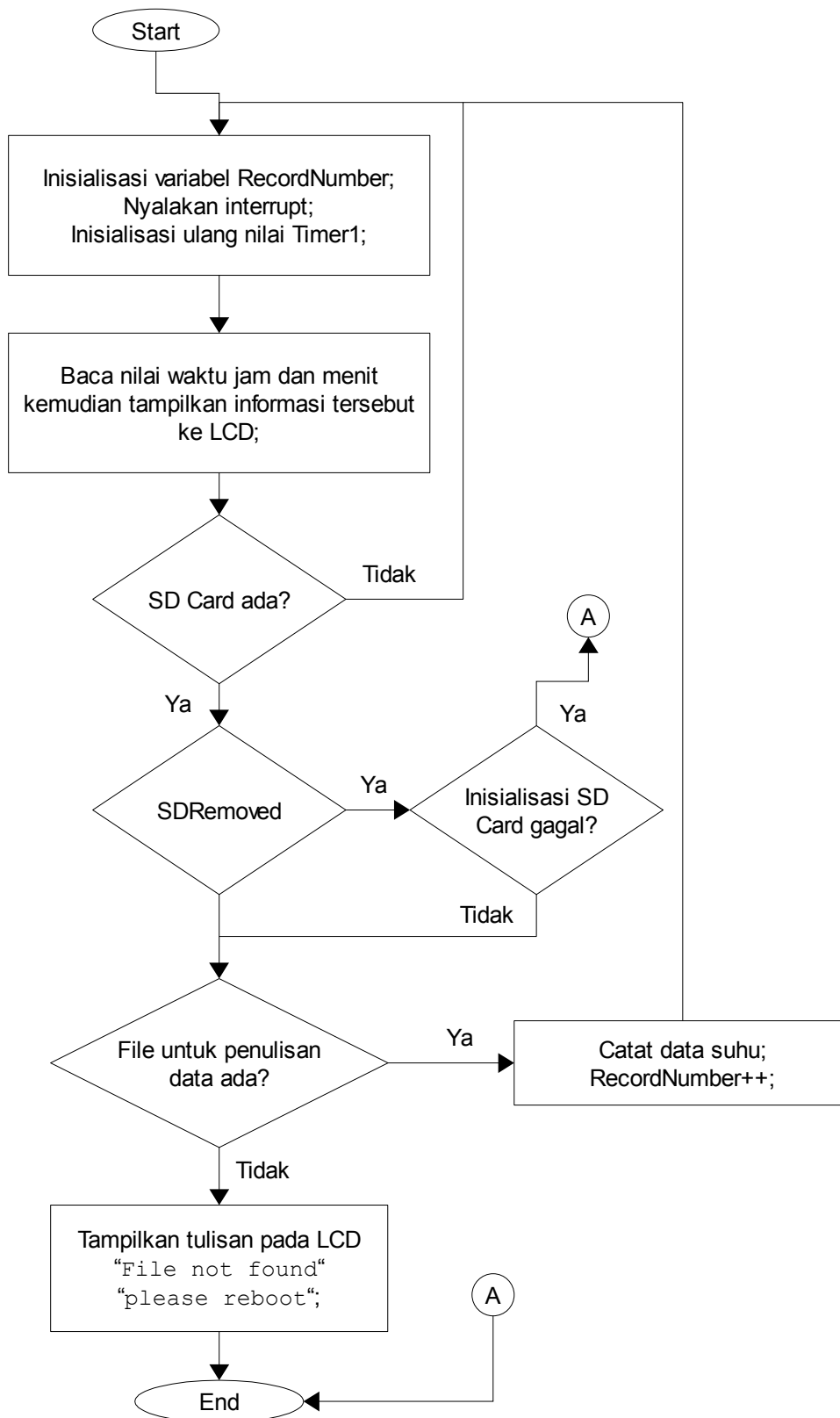
Tampilan LOG\_3.XML pada software XML Editor

**A**lur program dari “TemperatureLogger.ino” adalah sebagai berikut :



**Gambar 10**  
Alur program utama pada TemperatureLogger.ino





**Gambar 11**  
Alur program *interrupt* Counter1 pada TemperatureLogger.ino



Penjelasan urutan kerja dari program utama pada "TemperatureLogger.ino" adalah sebagai berikut :

1. Pertama-tama program akan melakukan inisialisasi konstanta, objek, serta variabel yang diperlukan pada proses eksekusi program. Kemudian proses akan dilanjutkan dengan inisialisasi awal protokol I2C, SPI, serta pin I/O DT-AVR Inoduino yang akan digunakan.
2. Program akan mendeteksi apakah SD Card sudah dimasukkan pada EMS SD/MMC/FRAM. Apabila belum dimasukkan, program akan menampilkan tulisan "SD Card not present" pada LCD dan akan menunggu sampai SD Card dimasukkan.
3. Apabila SD Card sudah dimasukkan, program akan melakukan inisialisasi awal SD Card. Jika proses inisialisasi gagal, maka proses program akan dihentikan. Jika proses inisialisasi berhasil, program akan berlanjut ke langkah 4.
4. Program akan membuat file XML baru pada SD Card untuk keperluan logging data. Proses penamaan file akan dilakukan mengikuti nomor file terakhir yang terdapat pada SD Card. Apabila nama file terakhir yang ada pada SD Card adalah "LOG\_3.XML", pada proses ini akan dibuat file baru dengan nama "LOG\_4.XML".
5. Program akan menampilkan tulisan "Pukul : " dan "Suhu : " pada LCD yang kemudian akan diikuti dengan inisialisasi RTC DS1307, register Counter1, serta DT-SENSE Temperature Sensor.
6. Proses akan dilanjutkan dengan pembacaan nilai suhu pada DT-SENSE Temperature Sensor.
7. Jika SD Card ada, tampilkan karakter " " pada LCD kolom 16 baris 2. Namun jika SD Card tidak ada, tampilkan karakter "■", kemudian ubah nilai variabel SDRemoved menjadi true.
8. Setelah itu, program akan menuliskan nilai pembacaan suhu pada langkah 6 ke LCD. Proses kembali ke langkah 6.

Penjelasan urutan kerja dari program interrupt Counter1 pada "TemperatureLogger.ino" adalah sebagai berikut :

1. Program akan melakukan inisialisasi variabel RecordNumber yang akan digunakan sebagai nomor pencatatan data. Program kemudian akan menyalakan interrupt dan melakukan inisialisasi ulang nilai Counter1.
2. Program akan membaca informasi jam dan menit dari RTC untuk ditampilkan pada LCD.
3. Program akan memeriksa apakah SD Card ada atau tidak. Jika SD Card ada maka proses berlanjut ke langkah 4. Apabila SD Card tidak ada proses akan kembali ke langkah 1.
4. Program akan memeriksa apakah variabel SDRemoved bernilai true atau false. Jika false, program akan berlanjut ke langkah 5. Sedangkan jika true, program akan melakukan inisialisasi ulang SD Card. Apabila SD Card berhasil diinisialisasi, nilai SDRemoved akan dibuat menjadi false kemudian program akan berlanjut ke langkah 5. Sedangkan jika SD Card tidak berhasil diinisialisasi, proses program akan dihentikan.
5. Proses program akan dilanjutkan dengan pemeriksaan apakah file untuk penulisan data (yang sudah diciptakan sebelumnya) ada atau tidak. Jika file tersebut ada, proses berlanjut ke langkah 6. Sedangkan jika file tersebut tidak ada, proses berlanjut ke langkah 7.
6. Jika file tersebut ada, maka proses penulisan data pada SD Card akan dilakukan yang kemudian akan dilanjutkan dengan penambahan variabel RecordNumber dengan angka 1. Proses program kembali ke langkah 1.
7. Program akan menuliskan "File not found please reboot" pada LCD dan proses program akan dihentikan.

Program interrupt Counter1 akan dijalankan setiap DT-AVR Inoduino telah mendapatkan 60 pulsa high-low pada digital pin 32. Pulsa tersebut berasal dari pin SQWOUT pada DT-I/O I2C Peripheral yang memiliki frekuensi 1 Hz sehingga program interrupt Counter1 akan dieksekusi tiap 1 menit sekali (60 detik).

Pada Application Note kali ini SD Card yang digunakan memiliki format FAT16 dengan ukuran allocation unit sebesar 8912 bytes. SD Card yang digunakan memiliki kapasitas 128 MB serta dilengkapi dengan antarmuka SPI. Sebagai informasi tidak semua SD Card dilengkapi dengan antarmuka SPI karena antarmuka ini bersifat opsional. Apabila SD Card yang digunakan tidak dapat diinisialisasi oleh sistem, coba terlebih dahulu untuk menyesuaikan kode program "TemperatureLogger.ino" pada bagian "#define INO\_SPI\_SPEED SPI\_SIXTEENTH\_SPEED" dengan konstanta kecepatan komunikasi SPI yang lain seperti "#define INO\_SPI\_SPEED SPI\_HALF\_SPEED". Informasi mengenai konstanta tersebut terdapat pada file "arduino-1.0.1\libraries\SdFat\Sd2Card.h".

Antarmuka I2C pada DT-SENSE Temperature Sensor mendukung frekuensi SCL dengan nilai maksimal 50 kHz. Maka dari itu, perlu dilakukan penyesuaian pada file "arduino-1.0.1\libraries\Wire\utility\twi.h" bagian "#define TWI\_FREQ 100000L" menjadi "#define TWI\_FREQ 40000L".

**L**isting program aplikasi ini terdapat pada **AN210.ZIP**

**S**elamat 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.*