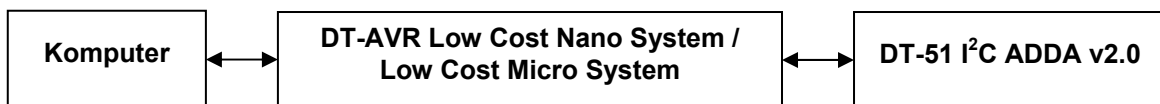


**K**onversi tegangan analog - digital banyak sekali digunakan dalam aplikasi pemrosesan sinyal, aplikasi yang menggunakan sensor analog, atau aplikasi digital lainnya yang berhubungan dengan dunia analog. Pada AN kali ini akan menggunakan DT-51 I<sup>2</sup>C ADDA v2.0 yang memiliki antarmuka serial I<sup>2</sup>C untuk mengkonversi tegangan analog - digital. Agar aplikasi ini menjadi lebih mudah dan cepat, maka digunakan DT-AVR Low Cost Nano System / Low Cost Micro System sebagai modul mikrokontrolernya serta bahasa pemrograman C yaitu dengan CodeVisionAVR© C *cross-compiler*. Aplikasi ini dapat menjadi contoh sederhana penggunaan DT-51 I<sup>2</sup>C ADDA v2.0 dan penggunaan rutin-rutin CodeVisionAVR© untuk komunikasi serial I<sup>2</sup>C.

Komponen yang diperlukan:

- 1 DT-AVR Low Cost Nano System / Low Cost Micro System
- 1 DT-51 I<sup>2</sup>C ADDA v2.0

**A**dapun blok diagram sistem secara keseluruhan adalah sebagai berikut:



**Gambar 1**  
**Blok Diagram AN89**

**H**ubungan antara modul-modul tersebut adalah sebagai berikut:

DT-AVR Low Cost Nano System / Low Cost Micro System	DT-51 I <sup>2</sup> C ADDA v2.0
GND	- (JP2)
+5VDC	+ (JP2)
PB.0*	SDA (JP3)
PB.1*	SCL (JP3)

\* pin ini tidak mutlak dan dapat diganti pin lain tetapi harus mengubah konfigurasi pin pada program

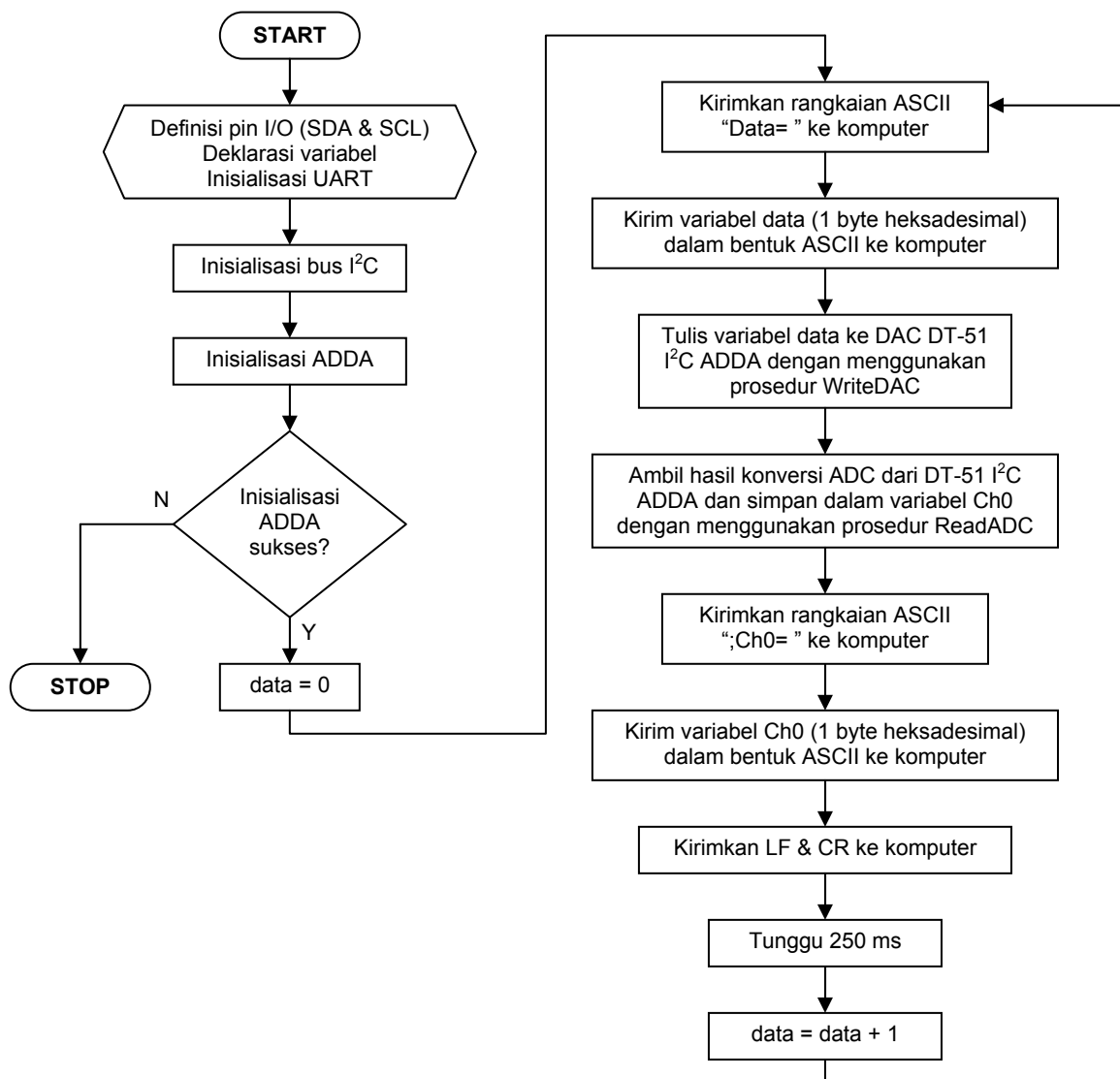
**Tabel 1**  
**Hubungan DT-AVR Low Cost Nano System / Low Cost Micro System dengan DT-51 I<sup>2</sup>C ADDA v2.0**

Pasang semua *jumper* "I2C Address" DT-51 I<sup>2</sup>C ADDA v2.0 (J3[A2], J2[A1], & J1[A0]) pada posisi 2-3 sehingga alamat terprogram DT-51 I<sup>2</sup>C ADDA v2.0 menjadi 0 (*default*). Serta pasang semua *jumper* J5 DT-51 I<sup>2</sup>C ADDA v2.0 untuk memberi resistor *pull up* pada jalur komunikasi I<sup>2</sup>C (SDA dan SCL). Pada terminal JP1 DT-51 I<sup>2</sup>C ADDA v2.0 hubungkan pin AO dengan pin AI3, pin AI2, pin AI1, dan pin AI0 secara paralel.

Gunakan kabel serial DT-AVR Low Cost Nano System / Low Cost Micro System untuk menghubungkan modul ke komputer. Kabel ini digunakan untuk mengirimkan data ke komputer dan ditampilkan pada layar monitor dengan bantuan program Terminal© atau *tool* Terminal pada CodeVisionAVR©. Konfigurasi UART pada *baud rate* 9600 bps, 8 bit data, tanpa bit *parity*, 1 bit stop, dan tanpa *flow control*.

Setelah semua rangkaian terhubung dengan tepat, berilah catu daya pada rangkaian dan bukalah proyek I2C\_ADDA.prj menggunakan CodeVisionAVR© (versi 1.24.5 atau lebih tinggi). Lalu kopikan listing "Contoh 1.C" atau "Contoh 2.C" ke Program.c pada proyek tersebut. Gunakan menu Project → Make atau tekan Shift+F9 untuk membuat *image file*. Kemudian programlah hasilnya (program.rom atau program.hex) ke dalam DT-AVR Low Cost

**F**lowchart dari program “Contoh 1.C” adalah sebagai berikut:



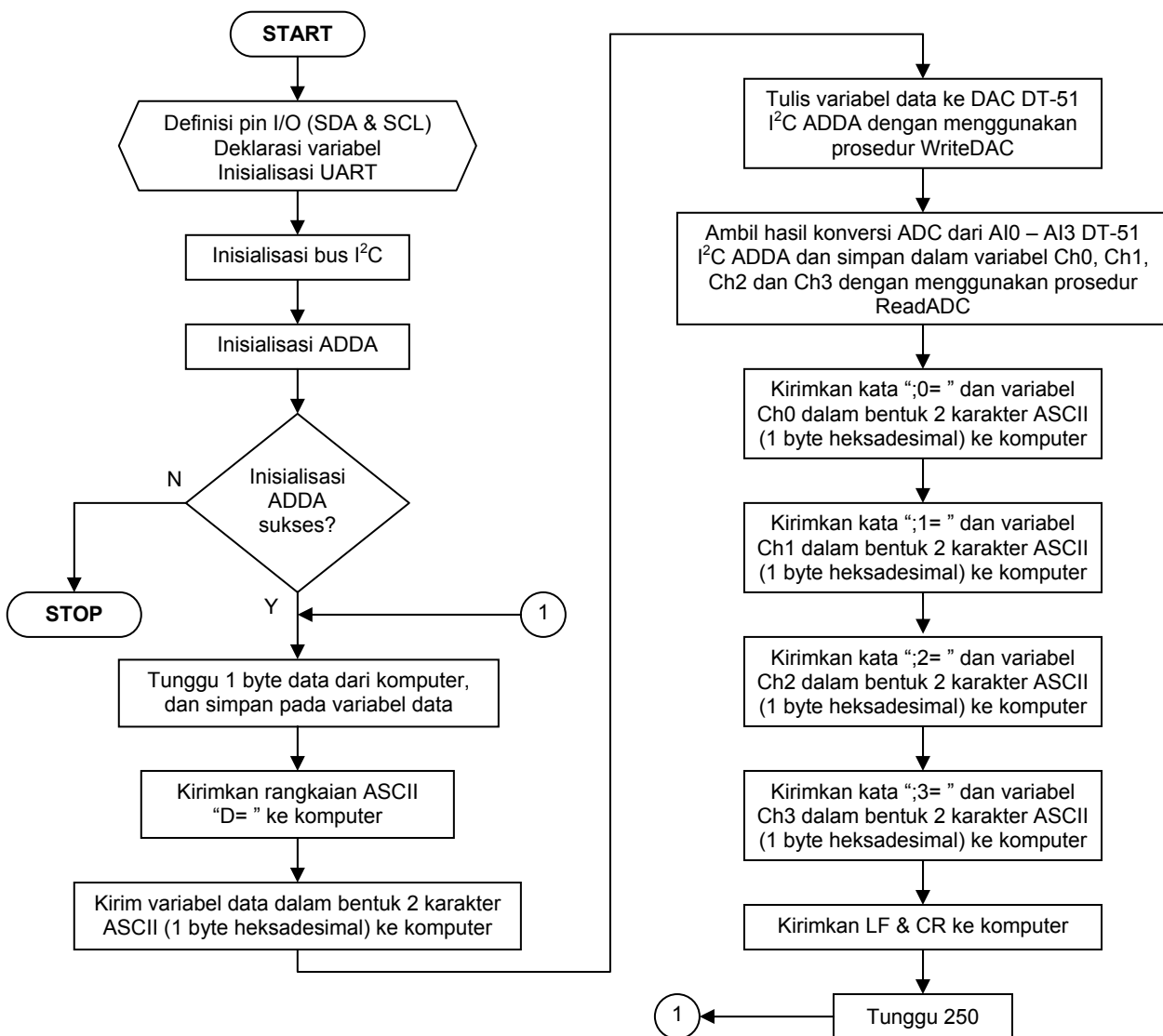
**Gambar 2**  
**Flowchart Program untuk “Contoh 1.c”**

**P**rogram utama (Contoh 1.c) akan diproses sebagai berikut:

- Proses yang pertama kali dilakukan adalah menentukan definisi pin I/O SDA dan SCL untuk komunikasi serial I<sup>2</sup>C. Kemudian inisialisasi port B sebagai input dan deklarasi variabel :
  - data, berisi nilai yang akan dikonversikan ke output analog (AO) oleh DAC.
  - Ch0, digunakan untuk menampung hasil konversi ADC dari input analog *channel* 0 (AI0).
- Proses berikutnya adalah inisialisasi komunikasi serial UART yaitu pada *baud rate* 9600 bps, 8 bit data, tanpa bit *parity*, 1 bit stop. Kemudian dilakukan inisialisasi bus komunikasi I<sup>2</sup>C.
- Setelah itu program akan melakukan inisialisasi ADDA dengan nilai Mode=0, Channel=0, AutoInc=0, dan OutputEnb=1 menggunakan prosedur InItADDA. Nilai Mode=0 berarti dipilih mode 4 *channel single ended input*, nilai Channel=0 berarti dipilih input channel 0 (pembacaan ADC pada pin AI0 DT-51 I<sup>2</sup>C ADDA), nilai AutoInc=0 berarti prosedur ReadADC hanya membaca satu analog input sesuai dengan nilai Channel, dan nilai OutputEnb=1 berarti hasil konversi DAC dikeluarkan pada pin AO DT-51 I<sup>2</sup>C ADDA.
- Jika inisialisasi ADDA tidak berhasil maka sistem akan berhenti. Tetapi jika berhasil maka program akan melanjutkan ke langkah berikutnya.

5. Kemudian variabel data diisi dengan nilai 0.
6. Setelah itu program akan mengirimkan kata "Data=" ke komputer serta mengirimkan variabel data dalam bentuk 2 karakter ASCII yang merepresentasikan 1 byte heksadesimal ke komputer.
7. Proses berikutnya yaitu mengirimkan variabel data ke DT-51 I<sup>2</sup>C ADDA dengan menggunakan prosedur WriteDAC. Sehingga besar tegangan output pada pin AO DT-51 I<sup>2</sup>C ADDA akan sesuai dengan rumus:  $AO = (2,5 \times (\text{data} \div 256)) \text{ volt}$ .
8. Setelah itu, program akan membaca tegangan input analog pada pin AI0 dengan menggunakan prosedur ReadADC. Hasil pengukuran ADC dari AI0 (sesuai nilai Channel=0 dan AutoInc=0) disimpan pada variabel Ch0.
9. Kemudian program akan mengirimkan kata ";Ch0=" dan variabel Ch0 (dalam bentuk 2 karakter ASCII yang merepresentasikan 1 byte heksadesimal) ke komputer. Sehingga dapat diketahui besar tegangan input pada pin AI0 di komputer. Rumus hasil konversi ADC (*single-ended input*) yaitu:  $Ch0 = (V_{AI0} \div 2,5) \times 256$ .
10. Program mengirimkan karakter LF dan CR, lalu berhenti selama 250 ms. Setelah itu nilai variabel data ditambah dengan 1 dan program kembali ke langkah 6. Nilai dari variabel data akan berputar dari 00h – FFh karena lebar variabel data adalah 8 bit.

**F**lowchart dari program "Contoh 2.C" adalah sebagai berikut:



**Gambar 3**  
**Flowchart Program untuk "Contoh 2.C"**

**P**rogram utama (Contoh 2.C) akan diproses sebagai berikut:

1. Proses yang pertama kali dilakukan adalah menentukan definisi pin I/O SDA dan SCL untuk komunikasi serial I<sup>2</sup>C. Kemudian inisialisasi port B sebagai input dan deklarasi variabel :
  - o data, berisi nilai yang akan dikonversikan ke output analog (AO) oleh DAC.
  - o Ch0, digunakan untuk menampung hasil konversi ADC dari input analog *channel* 0 (AI0).
  - o Ch1, digunakan untuk menampung hasil konversi ADC dari input analog *channel* 1 (AI1).
  - o Ch2, digunakan untuk menampung hasil konversi ADC dari input analog *channel* 2 (AI2).
  - o Ch3, digunakan untuk menampung hasil konversi ADC dari input analog *channel* 3 (AI3).
2. Proses berikutnya adalah inisialisasi komunikasi serial UART yaitu pada *baud rate* 9600 bps, 8 bit data, tanpa bit *parity*, 1 bit stop. Kemudian dilakukan inisialisasi bus komunikasi I<sup>2</sup>C.
3. Setelah itu program akan melakukan inisialisasi ADDA dengan nilai Mode=0, Channel=0, AutoInc=1, dan OutputEnb=1 menggunakan prosedur InitADDA. Nilai Mode=0 berarti dipilih mode 4 *channel single ended input*, nilai AutoInc=1 berarti prosedur ReadADC akan membaca semua analog input (AI0 – AI3) tanpa memperhatikan nilai channel, dan nilai OutputEnb=1 berarti hasil konversi DAC dikeluarkan pada pin AO DT-51 I<sup>2</sup>C ADDA.
4. Jika inisialisasi ADDA tidak berhasil maka sistem akan berhenti. Tetapi jika berhasil maka program akan melanjutkan ke langkah berikutnya.
5. Kemudian program akan menunggu 1 byte data dari komputer dengan menggunakan fungsi getch dan data yang diterima akan disimpan pada variabel data.
6. Setelah itu program akan mengirimkan kata "D= " ke komputer serta mengirimkan variabel data dalam bentuk 2 karakter ASCII yang merepresentasikan 1 byte heksadesimal ke komputer.
7. Proses berikutnya yaitu mengirimkan variabel data ke DT-51 I<sup>2</sup>C ADDA dengan menggunakan prosedur WriteDAC. Sehingga besar tegangan output pada pin AO DT-51 I<sup>2</sup>C ADDA akan sesuai dengan rumus:  $AO = (2,5 \times (data \div 256))$  volt.
8. Setelah itu, program akan membaca tegangan input analog pada pin AI0 – AI3 dengan menggunakan prosedur ReadADC. Hasil pengukuran ADC dari AI0 – AI3 (karena nilai AutoInc=1) disimpan pada variabel Ch0 – Ch3.
9. Kemudian program akan mengirimkan hasil konversi ADC satu-persatu. Dimulai dari pengiriman kata ";Ch0= " dan variabel Ch0 (dalam bentuk 2 karakter ASCII yang merepresentasikan 1 byte heksadesimal) ke komputer, pengiriman kata ";Ch1= " dan variabel Ch1 ke komputer, dan seterusnya sampai pengiriman variabel Ch3 ke komputer. Sehingga dapat diketahui besar tegangan input pada masing-masing pin AI0 – AI3 di komputer. Rumus hasil konversi ADC (*single-ended input*) yaitu:  $Ch0 = (V_{AI0} \div 2,5) \times 256$ .
10. Program mengirimkan karakter LF & CR lalu berhenti selama 250 ms. Kemudian program kembali ke langkah 5.

**L**isting program "Contoh 1.C" dan "Contoh 2.C" terdapat pada **AN89.ZIP**.

**S**elamat berinovasi!

CodeVisionAVR is copyright by Pavel Haiduc, HP InfoTech s.r.l.

Terminal is copyright by Bray++.

I<sup>2</sup>C is a registered trademark of Philips Semiconductors.