

Tampilan sudah merupakan hal pokok pada hampir setiap aplikasi. Dalam artikel ini akan membahas penggunaan salah satu modul tampilan Innovative Electronics yaitu SPC Serial LCD. Penggunaan SPC Serial LCD akan menghemat port mikrokontroler karena komunikasi serial modul ini hanya membutuhkan maksimum 3 pin I/O saja. Sedangkan modul mikrokontroler yang digunakan adalah DT-AVR Low Cost Nano System, tetapi tidak menutup kemungkinan untuk menggunakan DT-AVR Low Cost Micro System. Serta untuk mempercepat pemrograman mikrokontrolernya digunakan bahasa BASIC dengan compiler BASCOM-AVR©. Artikel ini akan menampilkan semua kemampuan SPC Serial LCD dalam komunikasi secara serial I²C, UART TTL, UART RS-232, maupun *synchronous serial*.

Komponen yang diperlukan:

- 1 DT-AVR Low Cost Nano System / Low Cost Micro System.
- 1 SPC Serial LCD /w character LCD 16x2.

Adapun blok diagram sistem secara keseluruhan adalah sebagai berikut:



Gambar 1
Blok Diagram AN82

Hubungan antara modul-modul tersebut adalah sebagai berikut:

DT-AVR Low Cost Nano System / Low Cost Micro System	SPC Serial LCD
+5VDC	VCC (J5)
GND	GND (J5)
PB.4*	SCL (J1)
PB.5*	SDA (J1)

* Pin ini tidak mutlak dan dapat diganti pin lain, tetapi harus mengubah program.

Tabel 1
Hubungan DT-AVR Low Cost Nano System / Low Cost Micro System dengan SPC Serial LCD secara I²C

Gunakan kabel serial DT-AVR Low Cost Nano System / Low Cost Micro System untuk menghubungkan modul dengan komputer. Pasanglah *jumper* J2 SPC Serial LCD untuk memberi resistor *pull up* pada jalur komunikasi I²C. Aturlah *jumper* A2-A0 (J3) SPC Serial LCD untuk alamat terprogram 111b (*default*). Setelah semua rangkaian dan sumber tegangan terhubung dengan benar, programlah i2cLCD.BIN ke dalam DT-AVR Low Cost Nano System (berbasis AT90S2313) dengan menggunakan DT-HiQ AVR In System Programmer atau divais AVR ISP programmer lain yang memiliki konektor 10-pin standar Atmel.

DT-AVR Low Cost Nano System / Low Cost Micro System	SPC Serial LCD
+5VDC	VCC (J5)
GND	GND (J5)
PB.0*	SerClr (J4-pin1)
PB.2*	SerClk (J4-pin2)
PB.4*	SerIn (J4-pin3)

* Pin ini tidak mutlak dan dapat diganti pin lain, tetapi harus mengubah program.

Tabel 2
Hubungan DT-AVR Low Cost Nano System / Low Cost Micro System dengan SPC Serial LCD
secara Synchronous serial

Gunakan kabel serial DT-AVR Low Cost Nano System / Low Cost Micro System untuk menghubungkan modul dengan komputer. Pasanglah *jumper* J12 SPC Serial LCD untuk memberi resistor *pull up* pada jalur SerClr, SerClk, dan SerIn. Aturlah *jumper* J9, J10, & J11 SPC Serial LCD agar komunikasi *Synchronous Serial* dapat digunakan. Dan aturlah *jumper* A2-A0 (J3) SPC Serial LCD untuk alamat terprogram 000b.

Setelah semua rangkaian dan sumber tegangan terhubung dengan benar, programlah syncLCD.BIN ke dalam DT-AVR Low Cost Nano System (berbasis AT90S2313) dengan menggunakan DT-HiQ AVR In System Programmer atau divais AVR ISP programmer lain yang memiliki konektor 10-pin standar Atmel.

DT-AVR Low Cost Nano System / Low Cost Micro System	SPC Serial LCD
+5VDC	VCC (J5)
GND	GND (J5)
PD.1*	RX (J4-pin3)
PD.0*	TX (J4-pin2)

* Pin ini tidak mutlak dan dapat diganti pin lain, tetapi harus mengubah program.

Tabel 3
Hubungan DT-AVR Low Cost Nano System / Low Cost Micro System dengan SPC Serial LCD
secara UART TTL

Aturlah *jumper* J9, J10, dan J11 SPC Serial LCD agar komunikasi serial UART TTL dapat digunakan. Aturlah *jumper* J4 & J5 DT-AVR Low Cost Nano System / Low Cost Micro System pada posisi 2-3, untuk komunikasi UART TTL.

Setelah semua rangkaian dan sumber tegangan terhubung dengan benar, programlah 232LCD.BIN ke dalam DT-AVR Low Cost Nano System (berbasis AT90S2313) dengan menggunakan DT-HiQ AVR In System Programmer atau divais AVR ISP programmer lain yang memiliki konektor 10-pin standar Atmel.

DT-AVR Low Cost Nano System / Low Cost Micro System	SPC Serial LCD
+5VDC	VCC (J5)
GND	GND (J5)
TX (J3/J1)	RX (J14)
RX (J3/J1)	TX (J14)

Tabel 4
Hubungan DT-AVR Low Cost Nano System / Low Cost Micro System dengan SPC Serial LCD
secara UART RS-232

Aturlah *jumper* J9, J10, dan J11 SPC Serial LCD agar komunikasi serial UART RS-232 dapat digunakan. Aturlah *jumper* J4 & J5 DT-AVR Low Cost Nano System / Low Cost Micro System pada posisi 1-2, untuk komunikasi UART RS-232.

Setelah semua rangkaian dan sumber tegangan terhubung dengan benar, programlah 232LCD.BIN ke dalam DT-AVR Low Cost Nano System (berbasis AT90S2313) dengan menggunakan DT-HiQ AVR In System Programmer atau divais AVR ISP programmer lain yang memiliki konektor 10-pin standar Atmel. Program yang digunakan untuk komunikasi secara UART TTL dan UART RS-232 adalah sama, yang membedakan hanya setting *jumper* dan hubungan antar modul.

Catatan:

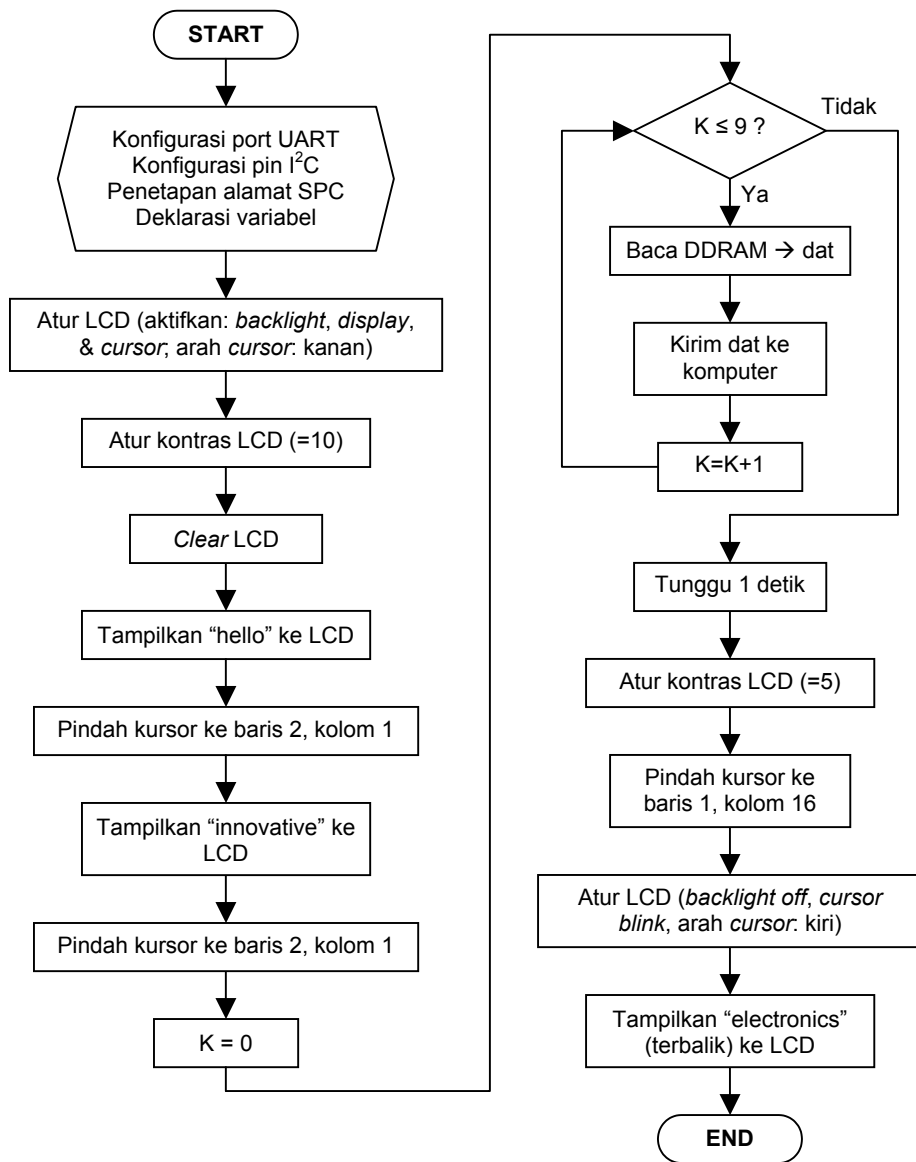
Dalam aplikasi ini menggunakan DT-AVR Low Cost Nano System (dengan mikrokontroler AT90S2313) sehingga i2cLCD.BIN, syncLCD.BIN, maupun 232LCD.BIN yang disertakan dalam **AN82.ZIP** hanya berlaku untuk AT90S2313.

Apabila menggunakan mikrokontroler tipe lain, bukalah i2cLCD.BAS, syncLCD.BAS, atau 232LCD.BAS menggunakan BASCOM-AVR© dan ubahlah baris pertama dalam program tersebut:

- \$regfile = "2313def.dat" '→ untuk AT90S2313
- \$regfile = "8535def.dat" '→ untuk AT90S8535
- \$regfile = "m8535.dat" '→ untuk ATmega8535 (DT-AVR Low Cost Micro System)

agar sesuai dengan tipe mikrokontroler yang akan digunakan. Lalu *compile* ulang program sehingga menghasilkan *file* dengan ekstensi .bin atau .hex yang sesuai dan dapat diprogram ke dalam modul mikrokontroler.

Flowchart program yang menggunakan komunikasi I²C adalah sebagai berikut:

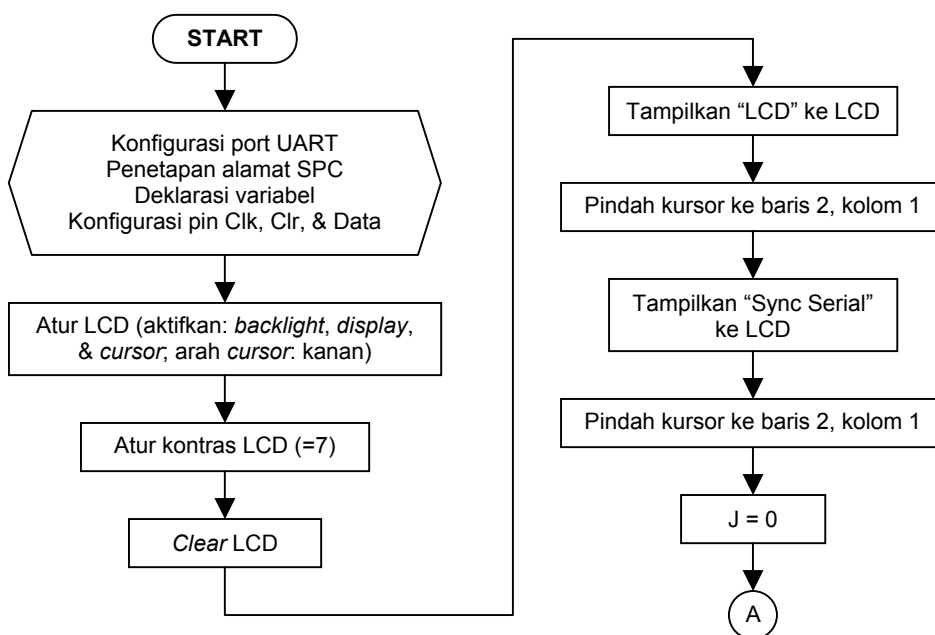


Gambar 2
Flowchart Program yang menggunakan Komunikasi I²C

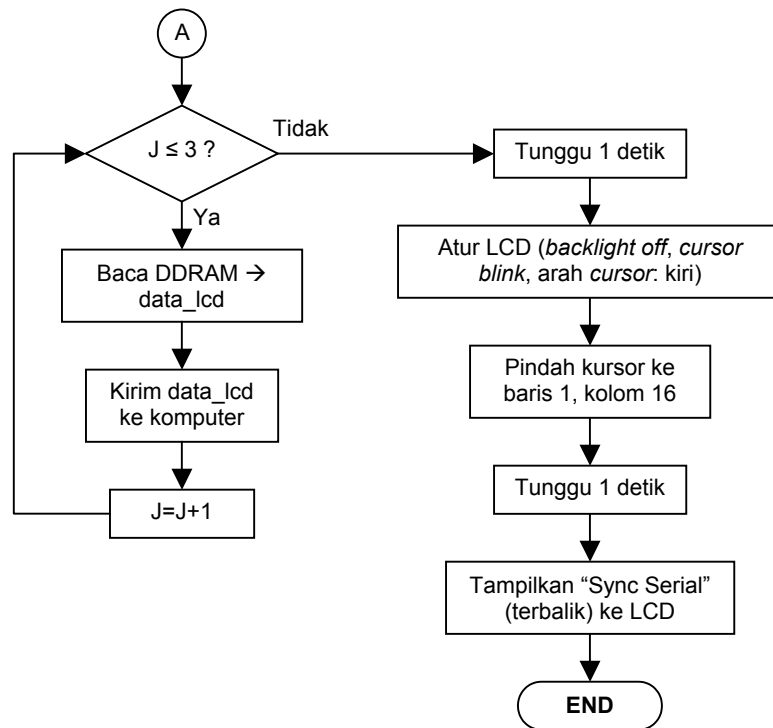
Program utama yang menggunakan komunikasi I²C akan diproses sebagai berikut:

1. Proses yang pertama dilakukan adalah konfigurasi port UART (*baudrate* 9600 bps) dan menentukan definisi pin I/O (SDA & SCL) untuk komunikasi I²C.
2. Penetapan alamat SPC Serial LCD yaitu EEh.
3. Kemudian deklarasi variabel *Dat*, *Ddram_data*, *Read_addr*, *Data_kolom*, *Data_baris*, *I*, *J*, *K*.
Dat adalah variabel I/O yang digunakan dalam prosedur *high level* (misalnya: *I2c_kontrol*).
Ddram_data adalah variabel I/O yang digunakan dalam prosedur *low level* (misalnya: *I2c_wbyte*).
Read_addr adalah variabel yang berisi alamat SPC untuk proses baca (bit ke-0 = *high*).
Data_kolom adalah variabel yang berisi posisi kolom DDRAM.
Data_baris adalah variabel yang berisi posisi baris DDRAM.
I, *J*, *K* adalah variabel yang digunakan dalam proses pengulangan.
4. Pada tahap berikutnya program akan mengirimkan data 8Eh menggunakan prosedur *I2c_kontrol* untuk mengaktifkan *backlight*, *display*, dan *cursor*, serta menetapkan arah *cursor* yaitu bergeser ke kanan.
5. Setelah itu program akan mengubah tingkat kontras LCD menjadi 10.
6. Program menghapus tampilan LCD dengan mengirimkan fungsi 01h ke LCD menggunakan prosedur *I2c_command_lcd*.
7. Langkah berikutnya adalah penulisan "hello" pada *display* dengan metode penulisan *single*.
8. Kemudian program memindahkan posisi kursor ke kolom pertama baris ke-2.
9. Segera setelah pemindahan posisi kursor, program menuliskan "innovative" dengan menggunakan metode penulisan *string*.
10. Pada tahap berikutnya, posisi kursor akan dikembalikan ke kolom pertama baris ke-2.
11. Setelah itu program akan membaca huruf demi huruf yang ada pada posisi kursor hingga 9 kolom berikutnya dan mengirimkan data hasil pembacaan ke komputer melalui komunikasi serial. Data ini dapat ditampilkan ke layar komputer dengan bantuan Terminal© atau HyperTerminal©. (Pada *baudrate* 9600 bps, 8 bit data, tanpa bit *parity*, 1 bit *stop*, dan tanpa *flow control*)
12. Setelah menunggu selama 1 detik, program mengubah tingkat kontras LCD menjadi 5 dan meletakkan posisi kursor pada kolom terakhir baris pertama.
13. Program mengirimkan fungsi 07h menggunakan prosedur *I2c_kontrol* untuk mematikan *backlight* LCD, mengubah arah kursor menjadi bergeser ke kiri, serta membuat kursor berkedip.
14. Setelah itu, program menuliskan "electronics" menggunakan metode *string* dan akan tampil pada LCD secara terbalik dari kanan ke kiri.

Flowchart program yang menggunakan komunikasi *synchronous serial* adalah sebagai berikut:



Gambar 3
Flowchart Program yang menggunakan Komunikasi Synchronous Serial (bagian 1)

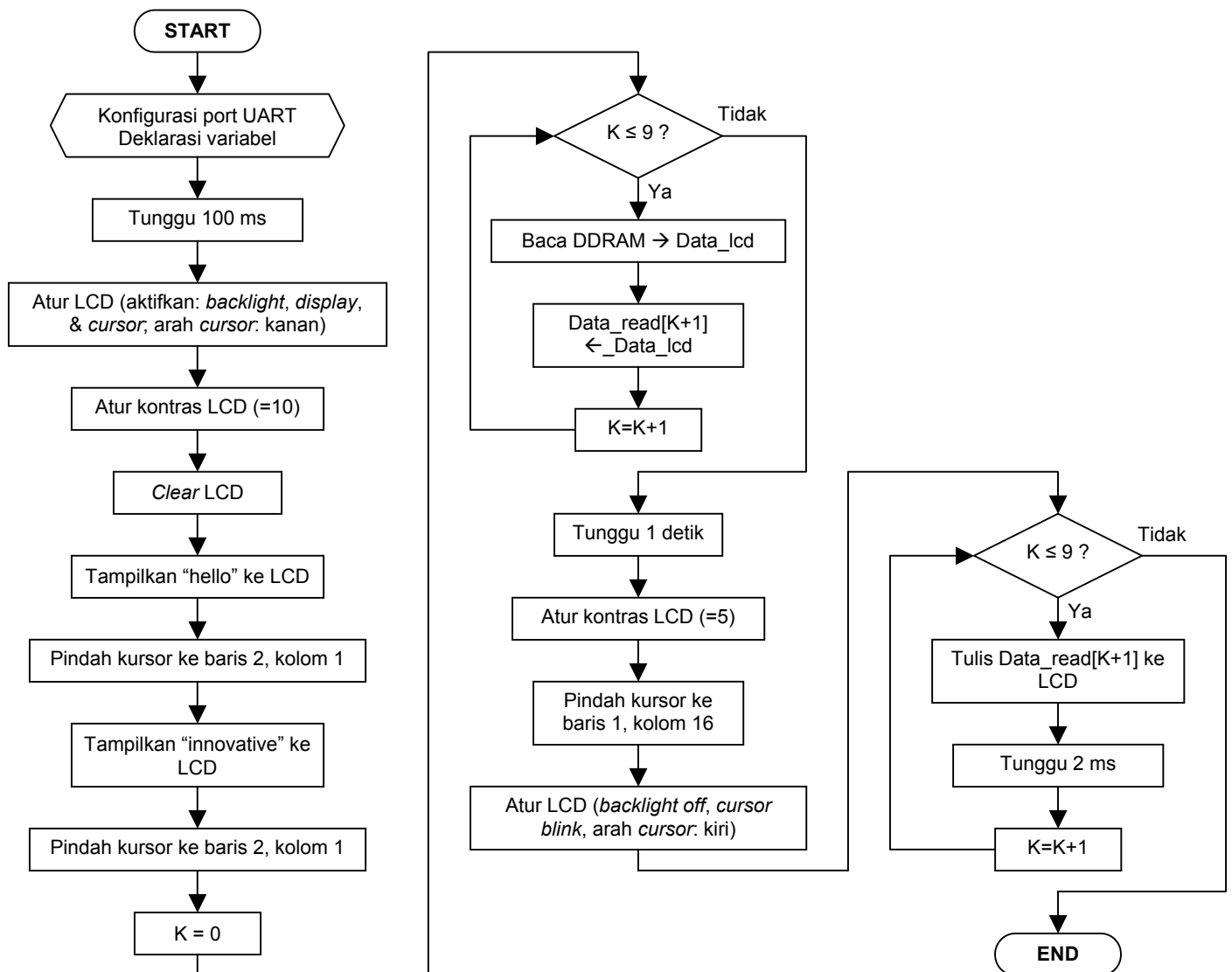


Gambar 4
Flowchart Program yang menggunakan Komunikasi Synchronous Serial (bagian 2)

Program utama yang menggunakan komunikasi *synchronous serial* akan diproses sebagai berikut:

1. Proses yang pertama dilakukan adalah konfigurasi port UART (*baudrate* 9600 bps) dan penetapan alamat SPC Serial LCD yaitu E0h.
2. Kemudian deklarasi variabel *Dat*, *Data_lcd*, *Data_kolom*, *Data_baris*, *I*, *J*.
Dat adalah variabel I/O yang digunakan dalam prosedur *low level* (mis: *Write_sync_byte*).
Data_lcd adalah variabel I/O yang digunakan dalam prosedur *high level* (mis: *Lcd_kontrol_ss*).
Data_kolom adalah variabel yang berisi posisi kolom DDRAM.
Data_baris adalah variabel yang berisi posisi baris DDRAM.
I, *J* adalah variabel yang digunakan dalam proses pengulangan.
3. Menentukan definisi pin I/O (*Clk*, *Clr*, dan *Data*) untuk komunikasi *synchronous serial*.
4. Pada tahap berikutnya program akan mengirimkan data 8Eh menggunakan prosedur *Lcd_kontrol_ss* untuk mengaktifkan *backlight*, *display*, dan *cursor*, serta menetapkan arah *cursor* yaitu bergeser ke kanan.
5. Setelah itu program akan mengubah tingkat kontras LCD menjadi 7.
6. Program menghapus tampilan LCD dengan mengirimkan fungsi 01h ke LCD menggunakan prosedur *Lcd_command_ss*.
7. Langkah berikutnya adalah penulisan "LCD" pada *display* dengan metode penulisan *single*.
8. Kemudian program memindahkan posisi kursor ke kolom pertama baris ke-2.
9. Segera setelah pemindahan posisi kursor, program menuliskan "Sync Serial" dengan menggunakan metode penulisan *string*.
10. Pada tahap berikutnya, posisi kursor akan dikembalikan ke kolom pertama baris ke-2.
11. Setelah itu program akan membaca huruf demi huruf yang ada pada posisi kursor hingga 3 kolom berikutnya dan mengirimkan data hasil pembacaan ke komputer melalui komunikasi serial. Data ini dapat ditampilkan ke layar komputer dengan bantuan Terminal© atau HyperTerminal©. (Pada *baudrate* 9600 bps, 8 bit data, tanpa bit *parity*, 1 bit *stop*, dan tanpa *flow control*)
12. Program menunggu selama 1 detik. Dan kemudian mengirimkan fungsi 07h menggunakan prosedur *Lcd_kontrol_ss* untuk mematikan *backlight* LCD, mengubah arah kursor menjadi bergeser ke kiri, serta membuat kursor berkedip.
13. Program meletakkan posisi kursor pada kolom terakhir baris pertama. Lalu menunggu lagi selama 1 detik.
14. Setelah itu, program menuliskan "Sync Serial" menggunakan metode *string* dan akan tampil pada LCD secara terbalik dari kanan ke kiri.

Flowchart program yang menggunakan komunikasi UART TTL/RS-232 adalah sebagai berikut:



Gambar 5
Flowchart Program yang menggunakan Komunikasi UART TTL/RS-232

Program utama yang menggunakan UART TTL/RS-232 akan diproses sebagai berikut:

1. Proses yang pertama dilakukan adalah konfigurasi port UART (*baudrate* 9600 bps).
2. Kemudian deklarasi variabel *Data_read*, *Data_lcd*, *Data_kolom*, *Data_baris*, *I*, *J*, *K*.
Data_read adalah variabel *array* untuk menyimpan karakter hasil pembacaan DDRAM LCD.
Data_lcd adalah variabel I/O yang digunakan dalam prosedur *high level* (mis: *Cmd_kontrol*).
Data_kolom adalah variabel yang berisi posisi kolom DDRAM.
Data_baris adalah variabel yang berisi posisi baris DDRAM.
I, *J*, *K* adalah variabel yang digunakan dalam proses pengulangan.
3. Program menunggu selama 100 mili detik.
4. Pada tahap berikutnya program akan mengirimkan data 8Eh menggunakan prosedur *Cmd_kontrol* untuk mengaktifkan *backlight*, *display*, dan *cursor*, serta menetapkan arah *cursor* yaitu bergeser ke kanan.
5. Setelah itu program akan mengubah tingkat kontras LCD menjadi 10.
6. Program menghapus tampilan LCD dengan mengirimkan fungsi 01h menggunakan prosedur *Cmd_command* ke LCD.
7. Langkah berikutnya adalah penulisan "hello" pada *display* dengan metode penulisan *single*.
8. Kemudian program memindahkan posisi kursor ke kolom pertama baris ke-2.

9. Segera setelah pemindahan posisi kursor, program menuliskan "innovative" dengan menggunakan metode penulisan *string*.
10. Pada tahap berikutnya, posisi kursor akan dikembalikan ke kolom pertama baris ke-2.
11. Setelah itu program akan membaca huruf demi huruf yang ada pada posisi kursor hingga 9 kolom berikutnya dan menyimpan data hasil pembacaan ke dalam variabel *array Data_read*.
12. Program menunggu selama 1 detik.
13. Lalu program mengubah tingkat kontras LCD menjadi 5 dan meletakkan posisi kursor pada kolom terakhir baris pertama.
14. Program mengirimkan fungsi 07h menggunakan prosedur *Cmd_kontrol* untuk mematikan *backlight* LCD, mengubah arah kursor menjadi bergeser ke kiri, serta membuat kursor berkedip.
15. Setelah itu, program menuliskan 10 karakter yang terdapat dalam variabel *Data_read* dan akan tampil pada LCD secara terbalik dari kanan ke kiri.

Listing program terdapat pada **AN82.ZIP**.

Selamat berinovasi!

BASCOM-AVR is a copyright by MCS Electronics.

HyperTerminal is a copyright by Hilgraeve Inc.

Terminal is a copyright by Bray++.

I²C is a registered trademark of Philips Semiconductors.