

DT-51

DT-51 *Application Note* AN78 – Komunikasi Mikrokontroler Melalui Jalur Telepon II

Oleh: Tim IE

Telepon telah menjadi sarana komunikasi jarak jauh yang cukup vital bagi manusia. Kali ini DT-51 Low Cost Micro System juga akan menggunakan telepon untuk komunikasi. Dengan bantuan modul DT-I/O Phone Interface Ver 2.0, DT-51 Low Cost Micro System dapat dengan mudah berkomunikasi dengan jalur telepon. Ditambah lagi menggunakan bahasa pemrograman C dengan *compiler* $\mu C/51^{\circ}$ sehingga aplikasi ini tampak sederhana. Meskipun sederhana, aplikasi ini dapat dikembangkan menjadi suatu sistem home automation / remote control.

Komponen yang diperlukan:

- 1 DT-51 Low Cost Micro System
- 1 DT-I/O Phone Interface Ver 2.0

Adapun blok diagram sistem secara keseluruhan adalah sebagai berikut:



Gambar 1
Blok Diagram AN78

Hubungan antara modul-modul tersebut adalah sebagai berikut:

DT-51 Low Cost Micro System	DT-I/O Phone Interface Ver 2.0
VCC	+ 5V (J4)
GND	- GND (J4)
P1.0*	D0 (pin 1 J1)
P1.1*	D1 (pin 2 J1)
P1.2*	D2 (pin 3 J1)
P1.3*	D3 (pin 4 J1)
P0.1*	CS (pin 9 J1)
P0.0*	Int0 (pin 3 J2)
P0.2*	WR (pin 7 J2)
P0.3*	RD (pin 8 J2)
P0.4*	RI (pin 9 J3)
P0.5*	HK (pin 10 J3)
P0.6*	RE (pin 11 J3)
P0.7*	RS0 (pin 12 J3)
P2.0*	T0 (pin 5 J2)

* Pin ini tidak mutlak dan dapat diganti pin lain dengan cara mengubah program

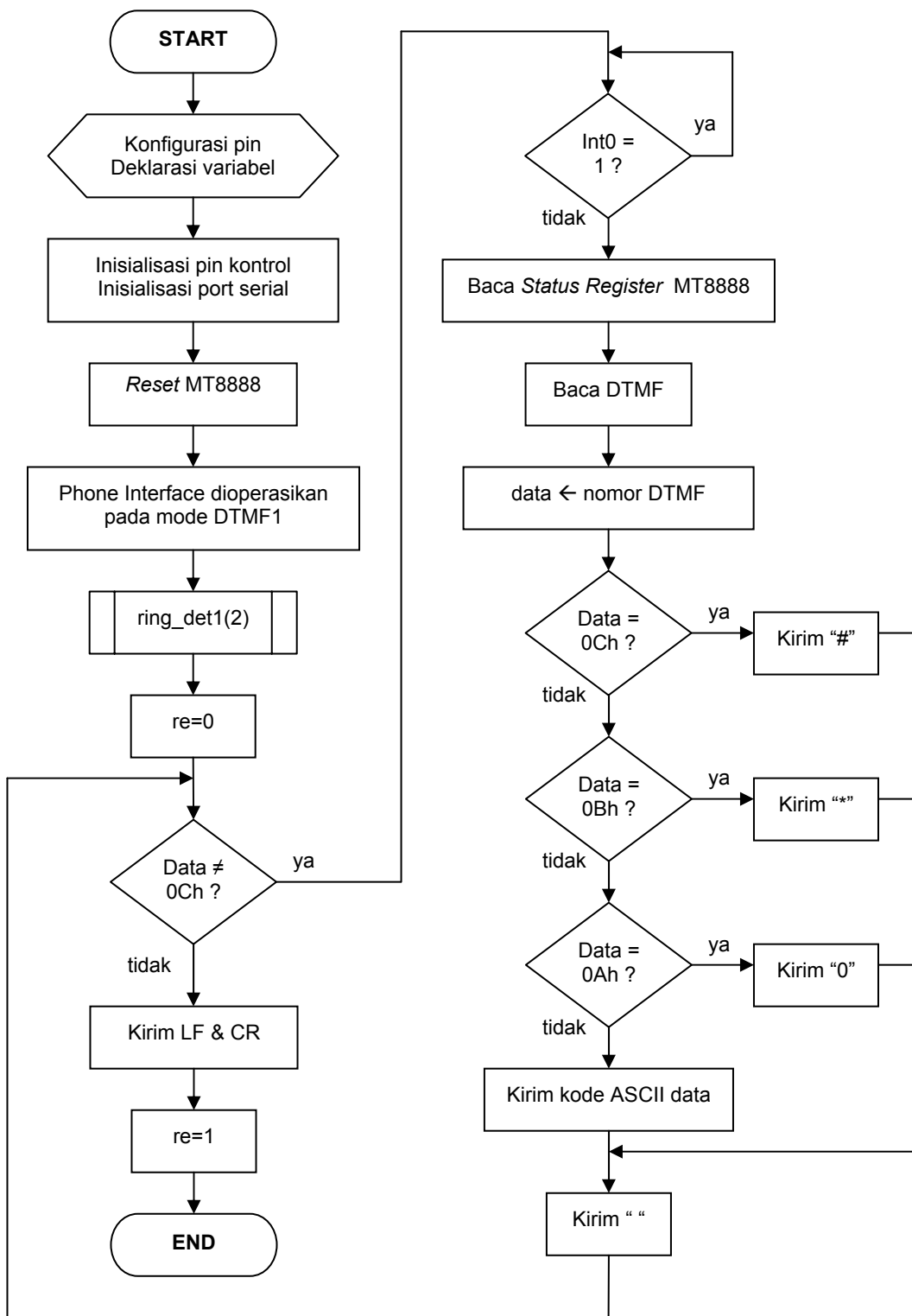
Tabel 1
Hubungan DT-51 Low Cost Micro System dengan DT-I/O Phone Interface Ver 2.0

Atur jumper JMP MODE pada mode 2 (jumper terhubung), hubungkan jalur telepon ke konektor RJ11 (LINE), dan hubungkan pesawat telepon (pespon) ke konektor RJ11 (PESPON)

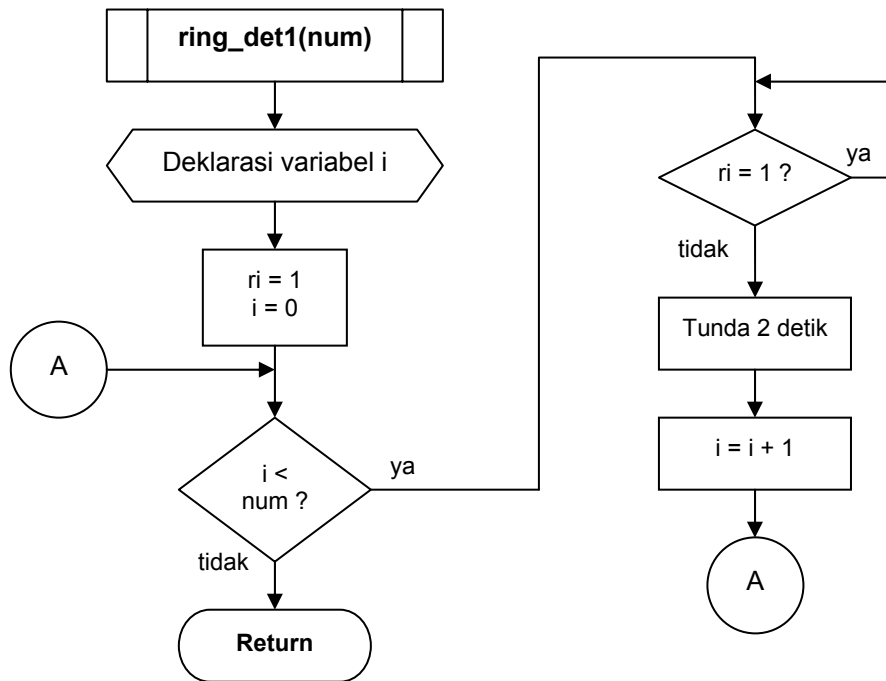
Gunakan kabel serial DT-51 Low Cost Micro System untuk menghubungkan modul DT-51 Low Cost Micro System dengan komputer. Setelah semua rangkaian dan sumber tegangan terhubung dengan benar, programlah test1.hex,

tst2.hex, test5.hex, test5b.hex, test6.hex, atau test9.hex ke dalam DT-51 Low Cost Micro System. Gunakanlah DT-HiQ AT89S In System Programmer dengan bantuan Atmel Microcontroller ISP Software untuk memprogram file-file .hex tersebut ke dalam DT-51 Low Cost Micro System.

Program test1.c akan menunggu adanya dering pada pespon yang terhubung ke Phone Interface. Setelah terdeteksi adanya 2 kali dering, program akan mengangkat telepon (*off-hook*). Dan selanjutnya menunggu sinyal DTMF dari line telepon, untuk dikirimkan ke komputer. *Flowchart* program test1.c adalah sebagai berikut:



Gambar 2
Flowchart Program test1.c



Gambar 3
Flowchart Rutin ring_det1

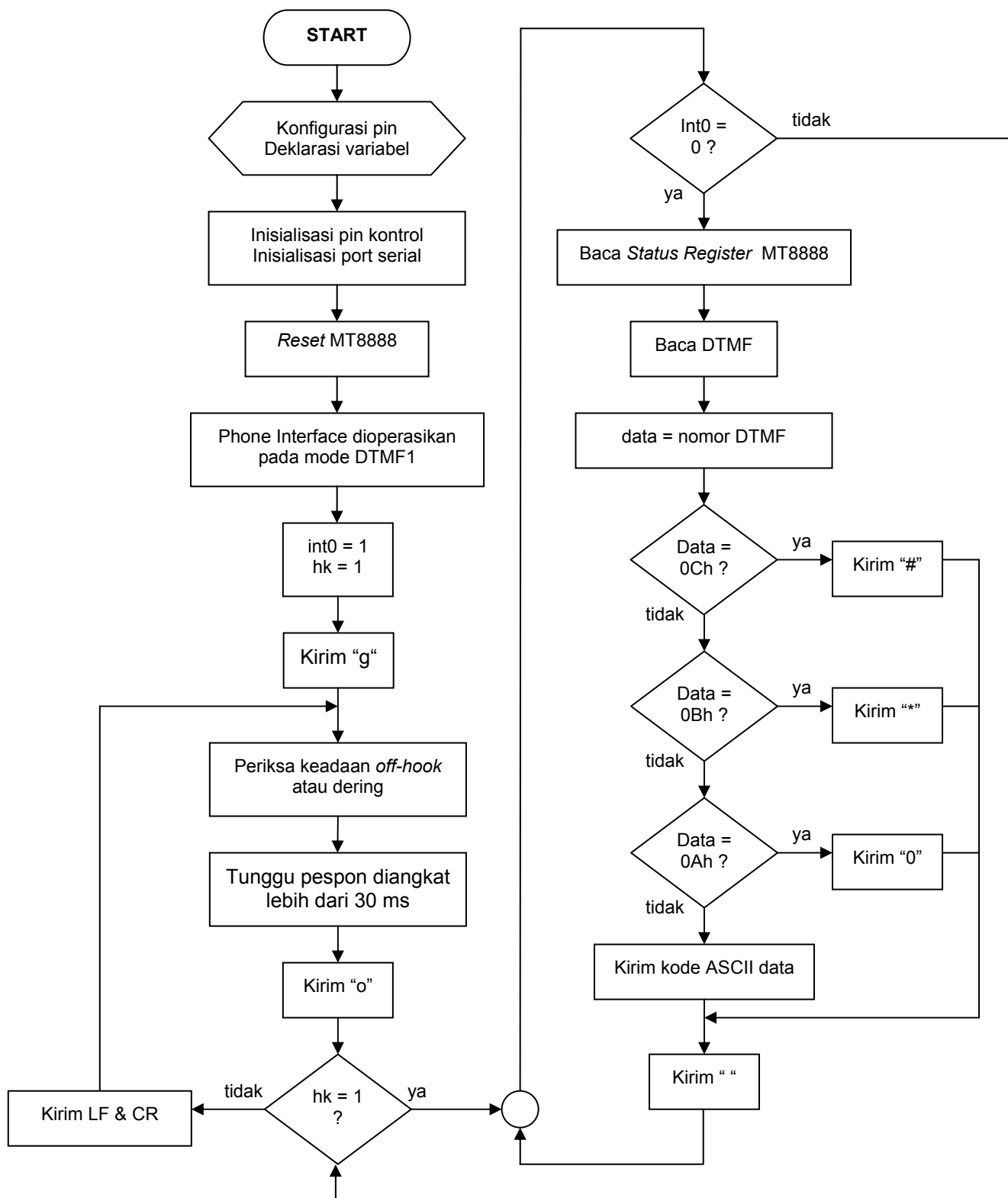
Program test1.c secara garis besar akan diproses sebagai berikut:

- Proses yang pertama dilakukan adalah konfigurasi pin untuk jalur komunikasi dengan Phone Interface, yaitu:
 - mt8888 = jalur mengirim/menerima data,
 - rs0 = *register select*,
 - re = untuk menghubungkan pespon dengan jalur telepon,
 - hk = status *on-hook / off-hook*,
 - ri = status dering,
 - int0 = interupsi penekanan tombol,
 - cs = *chip select*,
 - wr = pin kontrol untuk proses tulis,
 - rd = pin kontrol untuk proses baca.
- Kemudian deklarasi variabel mode dan data, yaitu:
 - data = variabel untuk menampung data DTMF yang diterima dari mt8888,
 - mode = variabel yang dipakai untuk menginisialisasi mt8888.
- Program melakukan inisialisasi pin kontrol ($re = wr = rd = rs0 = cs = int0 = hk = 1$) dan port serial (*baudrate* 9600 bps, 8 bit data, 1 bit *stop*, tanpa bit *parity*).
- Setelah itu dilakukan prosedur *reset* pada mt8888 sebelum rutin-rutin yang lain dijalankan.
- Kemudian dilakukan inisialisasi terhadap mt8888 dengan mode DTMF1 (0Dh).
- Dilakukan pemeriksaan jumlah dering pada jalur telepon sebanyak dua kali (status dering akan aktif apabila ada penelpon dari luar).
- Dilakukan pemutusan hubungan antara pespon dengan jalur telepon ($re=0$, mengangkat telepon).
- Setelah itu program akan membaca *status register* mt8888 dan menunggu penekanan tombol (0-9, *, #) oleh penelpon dari luar.
- Kemudian program membaca digit DTMF yang diterima dan menampilkannya pada komputer dengan ketentuan sebagai berikut:

DTMF	DISPLAY
0Ch	'#'
0Bh	'*'
0Ah	'0'
1h-9h	'1'..'9'
- Apabila DTMF digit yang diterima adalah 0Ch maka program akan keluar dari proses pembacaan digit DTMF. Lalu program akan mengirimkan karakter *Line Feed* (LF) dan *Carriage Return* (CR) ke komputer serta menyambung kembali hubungan antara pespon dengan jalur telepon.

Komunikasi serial pada komputer dapat menggunakan Terminal[®] atau HyperTerminal[®]. Pengaturan serial adalah *baudrate* 9600 bps, 8 bit data, 1 bit stop, tanpa bit *parity*, tanpa *flow control*.

Program `tst2.c` akan mendeteksi sinyal DTMF yang dihasilkan oleh pespon yang terhubung ke Phone Interface. Dan mengirimkan data DTMF tersebut ke komputer. *Flowchart* program `tst2.c` adalah sebagai berikut:



Gambar 4
Flowchart Program `tst2.c`

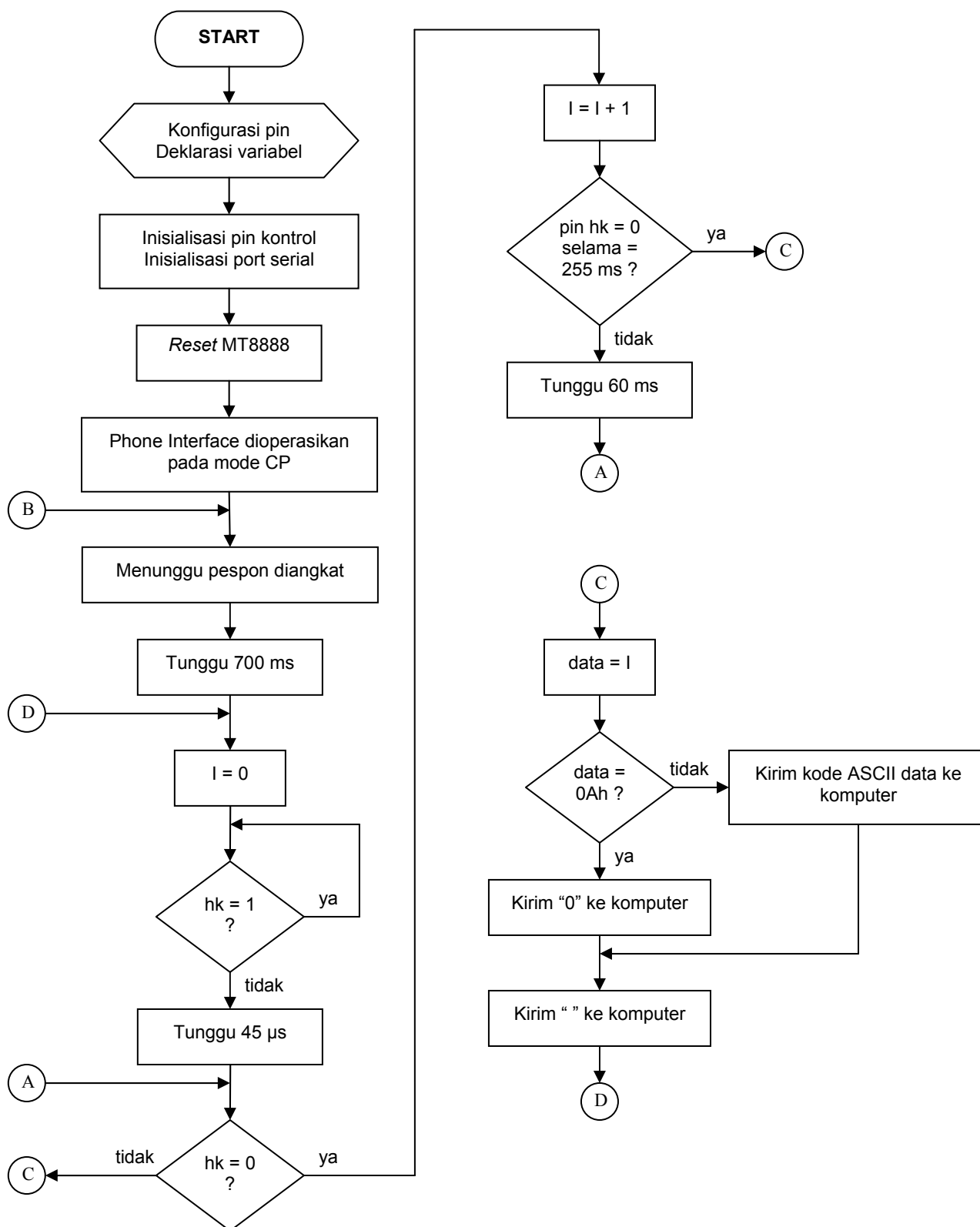
Program `tst2.c` secara garis besar akan diproses sebagai berikut:

1. Proses yang pertama dilakukan adalah konfigurasi pin untuk jalur komunikasi dengan Phone Interface, yaitu:
 - `mt8888` = jalur mengirim/menerima data,
 - `rs0` = *register select*,
 - `re` = untuk menghubungkan pespon dengan jalur telepon,
 - `hk` = status *on-hook / off-hook*,
 - `ri` = status dering,
 - `int0` = interupsi penekanan tombol,
 - `cs` = *chip select*,
 - `wr` = pin kontrol untuk proses tulis,
 - `rd` = pin kontrol untuk proses baca.
2. Kemudian deklarasi variabel mode dan data, yaitu:
 - `data` = variabel untuk menampung data DTMF yang diterima dari `mt8888`,
 - `mode` = variabel yang dipakai untuk menginisialisasi `mt8888`.
3. Program melakukan inisialisasi pin kontrol (`re = wr = rd = rs0 = cs = 1`) dan port serial (*baudrate* 9600 bps, 8 bit data, 1 bit *stop*, tanpa bit *parity*).
4. Setelah itu dilakukan prosedur *reset* pada `mt8888` sebelum rutin-rutin yang lain dijalankan.
5. Kemudian dilakukan inisialisasi terhadap `mt8888` dengan mode DTMF1 (0Dh). Dan memberi logika *high* pada pin `int0` dan `hk`. Lalu program mengirim karakter 'g' ke komputer sebagai tanda dimulainya proses *looping*.
6. Dilakukan pemeriksaan keadaan apakah *off-hook* atau dering, dan program akan menunggu pespon diangkat (*off-hook*) lebih dari 30 ms. Lalu program mengirim karakter 'o' ke komputer sebagai tanda keadaan pespon *off-hook*.
7. Program melihat keadaan pin `hk`. Bila *high* (*off-hook*) maka program menjalankan langkah selanjutnya, tetapi bila tidak (*on-hook*) maka program akan lompat ke langkah 12.
8. Program melihat keadaan pin `int0`. Bila *low* maka program akan melanjutkan ke langkah berikutnya, tetapi bila tidak maka program akan kembali ke langkah 7.
9. Setelah itu program akan membaca status register `mt8888` dan menunggu penekanan tombol (0-9, *, #) pada pespon.
10. Kemudian program membaca digit DTMF yang diterima dan menampilkannya pada komputer dengan ketentuan sebagai berikut:

DTMF	DISPLAY
0Ch	'#'
0Bh	'*'
0Ah	'0'
1h-9h	'1'..'9'
11. Program kembali ke langkah 8.
12. Program akan mengirimkan karakter *Line Feed* (LF) dan *Carriage Return* (CR) ke komputer.
13. Kembali ke langkah 6.

Komunikasi serial pada komputer dapat menggunakan Terminal[®] atau HyperTerminal[®]. Pengaturan serial adalah *baudrate* 9600 bps, 8 bit data, 1 bit stop, tanpa bit *parity*, tanpa *flow control*.

Program test5.c akan mendeteksi kondisi jalur telepon. Jika kondisinya adalah *off-hook*, program akan menunggu adanya sinyal *dial pulse* yang dihasilkan oleh pespon yang terhubung pada Phone Interface. Data *dial pulse* tersebut akan dikirimkan ke komputer secara serial. *Flowchart* program test5.c adalah sebagai berikut:



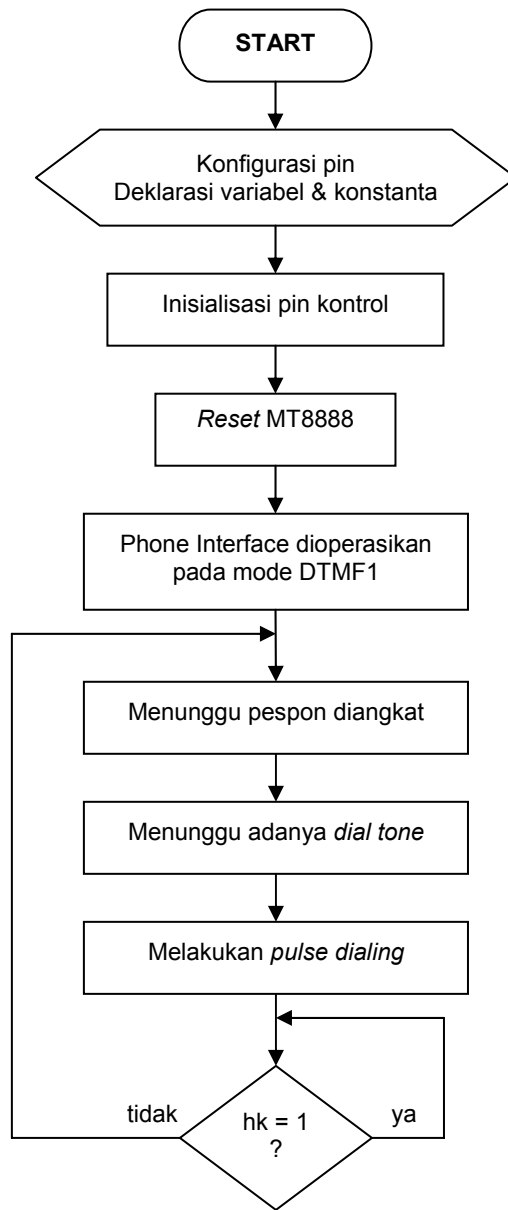
Gambar 5
Flowchart program test5.c

Program test5.c akan diproses sebagai berikut:

1. Proses yang pertama dilakukan adalah konfigurasi pin untuk jalur komunikasi dengan Phone Interface, yaitu:
 - mt8888 = jalur mengirim/menerima data,
 - rs0 = *register select*,
 - re = untuk menghubungkan pespon dengan jalur telepon,
 - hk = status *on-hook / off-hook*,
 - ri = status dering,
 - int0 = interupsi penekanan tombol,
 - cs = *chip select*,
 - wr = pin kontrol untuk proses tulis,
 - rd = pin kontrol untuk proses baca.
 - t0 = status *dial tone*
2. Kemudian deklarasi variabel mode dan data, yaitu:
 - data = variabel untuk menampung banyaknya *dial pulse* yang diterima dari mt8888,
 - mode = variabel yang dipakai untuk menginisialisasi mt8888.
3. Program melakukan inisialisasi pin kontrol (re = wr = rd = rs0 = cs = int0 = hk = 1) dan port serial (*baudrate* 9600 bps, 8 bit data, 1 bit *stop*, tanpa bit *parity*).
4. Setelah itu dilakukan prosedur *reset* pada mt8888 sebelum rutin-rutin yang lain dijalankan.
5. Kemudian dilakukan inisialisasi terhadap mt8888 dengan mode CP (0Eh).
6. Program menunggu gagang telepon diangkat dan terdengar *dial tone*.
7. Setelah terdeteksi adanya *dial tone*, program akan membaca penekanan tombol (pastikan dahulu bahwa mode pespon adalah Pulse) dan mengirimkan digit tombol yang ditekan ke komputer. Proses ini dilakukan berulang-ulang hingga gagang telepon ditutup.
8. Jika gagang telepon ditutup, maka program akan kembali ke langkah 6.

Komunikasi serial pada komputer dapat menggunakan Terminal[®] atau HyperTerminal[®]. Pengaturan serial adalah *baudrate* 9600 bps, 8 bit data, 1 bit *stop*, tanpa bit *parity*, tanpa *flow control*.

Program test5b.c akan mendeteksi kondisi jalur telepon. Jika kondisinya adalah *off-hook*, program akan melakukan *pulse dialing* ke nomor telepon yang telah diprogram. *Flowchart* program test5b.c adalah sebagai berikut:



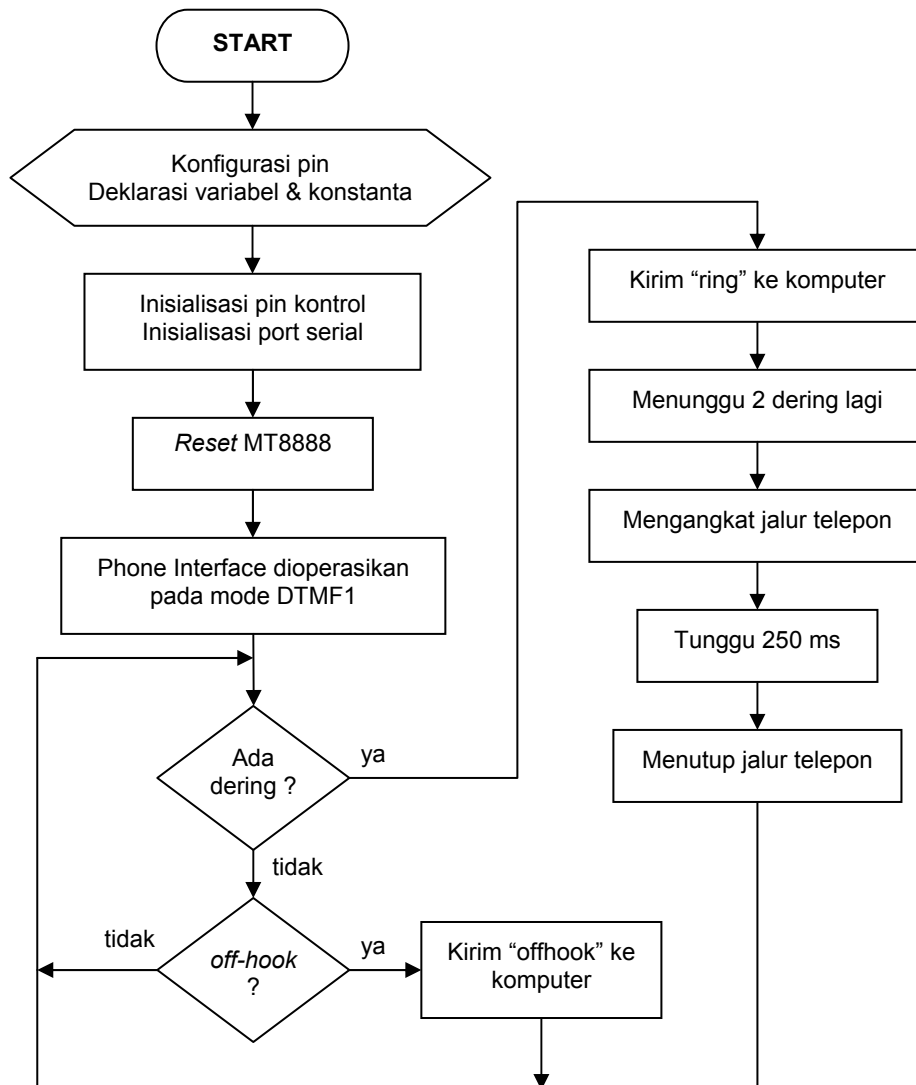
Gambar 6
Flowchart program test5b.c

Program test5b.c akan diproses sebagai berikut:

- Proses yang pertama dilakukan adalah konfigurasi pin untuk jalur komunikasi dengan Phone Interface, yaitu:
 - mt8888 = jalur mengirim/menerima data,
 - rs0 = register select,
 - re = untuk menghubungkan pespon dengan jalur telepon,
 - hk = status *on-hook* / *off-hook*,
 - ri = status dering,
 - int0 = interupsi penekanan tombol,
 - cs = chip select,
 - wr = pin kontrol untuk proses tulis,
 - rd = pin kontrol untuk proses baca.

- t0 = status *dial tone*
2. Kemudian deklarasi variabel `init_mode`, `data`, dan konstanta number yaitu:
 - `data` = variabel yang dipakai pada rutin `dialdp` untuk melakukan *pulse dialing*,
 - `init_mode` = variabel yang dipakai untuk menginisialisasi `mt8888`,
 - `number` = konstanta yang berisi nomor telepon tujuan.
 3. Program melakukan inisialisasi pin kontrol (`re = wr = rd = rs0 = cs = int0 = hk = 1`).
 4. Setelah itu dilakukan prosedur *reset* pada `mt8888` sebelum rutin-rutin yang lain dijalankan.
 5. Kemudian dilakukan inisialisasi terhadap `mt8888` dengan mode DTMF1 (0Dh).
 6. Program menunggu gagang telepon diangkat dan terdengar *dial tone*.
 7. Setelah terdeteksi adanya *dial tone*, program akan melakukan *pulse dialing* ke nomor telepon tujuan yang telah diprogram.
 8. Jika gagang telepon ditutup, maka program akan kembali ke langkah 6.

Program `test6.c` akan mendeteksi kondisi pada jalur telepon, *off-hook* atau ada sinyal dering. Kondisi ini akan dikirimkan ke komputer secara serial. Setelah mendeteksi adanya 3 dering, program akan mengangkat telepon (*off-hook*) secara otomatis. Dan setelah 250 ms program akan menutup telepon (*on-hook*) kembali. *Flowchart* program `test6.c` adalah sebagai berikut:



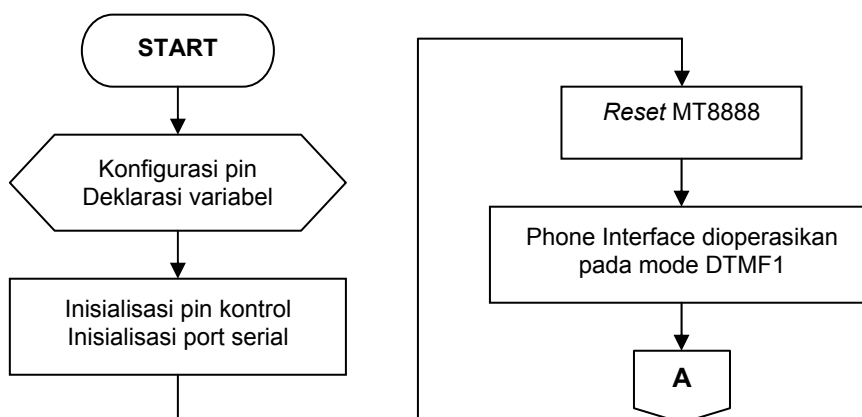
Gambar 7
Flowchart program `test6.c`

Program test6.c akan diproses sebagai berikut:

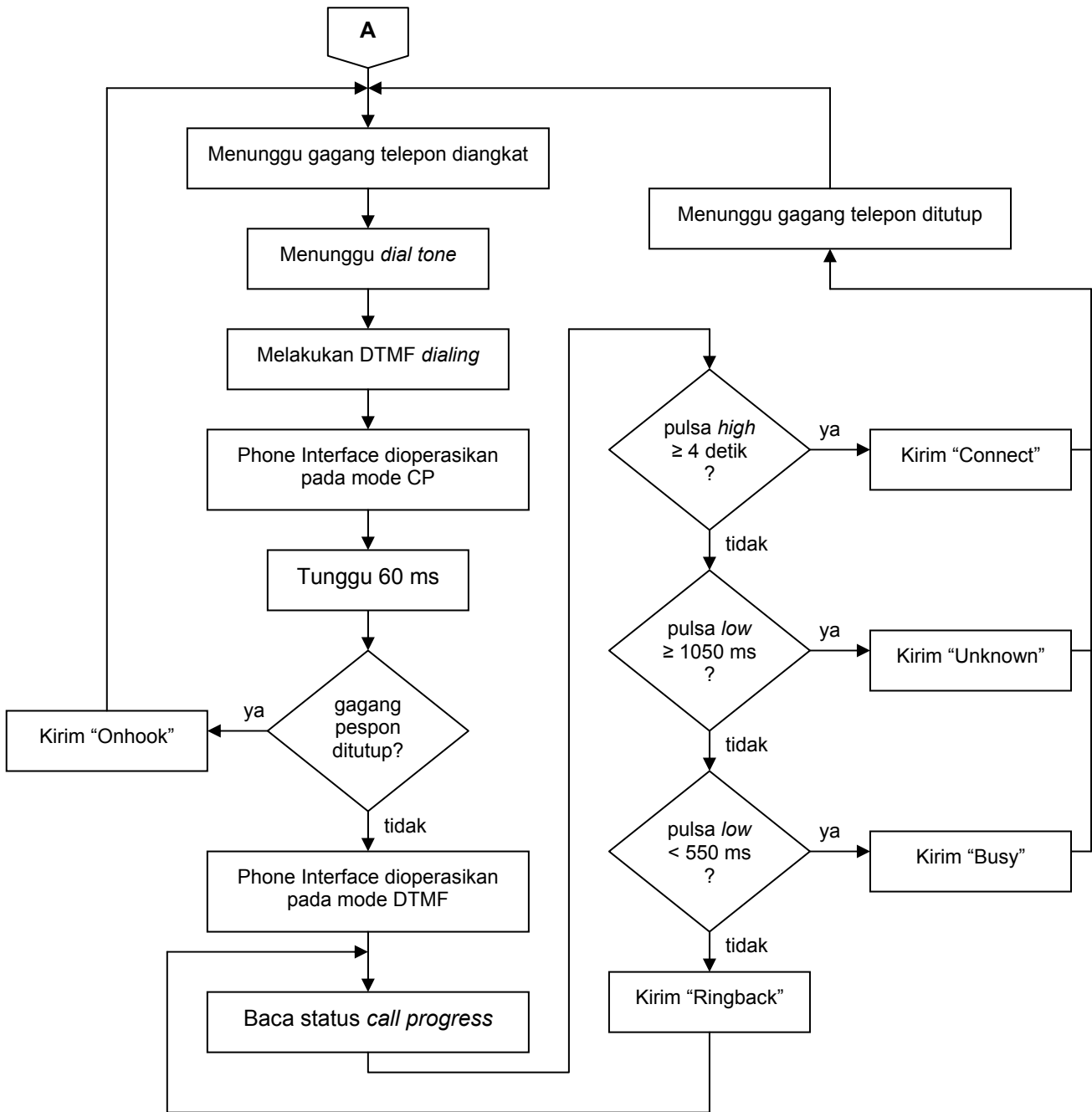
1. Proses yang pertama dilakukan adalah konfigurasi pin untuk jalur komunikasi dengan Phone Interface, yaitu:
 - mt8888 = jalur mengirim/menerima data,
 - rs0 = *register select*,
 - re = untuk menghubungkan pespon dengan jalur telepon,
 - hk = status *on-hook / off-hook*,
 - ri = status dering,
 - int0 = interupsi penekanan tombol,
 - cs = *chip select*,
 - wr = pin kontrol untuk proses tulis,
 - rd = pin kontrol untuk proses baca.
 - t0 = status *dial tone*
2. Kemudian deklarasi variabel *init_mode* serta konstanta *ring* dan *offhook*, yaitu:
 - *init_mode* = variabel yang dipakai untuk menginisialisasi mt8888,
 - *ring* = konstanta yang berisi tulisan “ring” yang akan dikirim ke komputer bila terdeteksi sinyal dering pada jalur telepon,
 - *offhook* = konstanta yang berisi tulisan “offhook” yang akan dikirim ke komputer bila keadaan pespon *off-hook*.
3. Program melakukan inialisasi pin kontrol (*re = wr = rd = rs0 = cs = int0 = hk = 1*) dan port serial (*baudrate* 9600 bps, 8 bit data, 1 bit *stop*, tanpa bit *parity*).
4. Setelah itu dilakukan prosedur *reset* pada mt8888 sebelum rutin-rutin yang lain dijalankan.
5. Kemudian dilakukan inialisasi terhadap mt8888 dengan mode DTMF1 (0Dh).
6. Program akan memeriksa status dering dan *hook*.
7. Jika terdeteksi dering pertama, program akan mengirim “ring” ke komputer untuk ditampilkan lalu menunggu 2 dering lagi. Setelah itu Phone Interface akan mengangkat jalur telepon selama 250 ms lalu menutupnya.
8. Jika gagang telepon diangkat, maka program akan mengirim “offhook” ke komputer untuk ditampilkan. Lalu program kembali ke langkah 6.

Komunikasi serial pada komputer dapat menggunakan Terminal© atau HyperTerminal©. Pengaturan serial adalah *baudrate* 9600 bps, 8 bit data, 1 bit stop, tanpa bit *parity*, tanpa *flow control*.

Program test9.c akan mendeteksi kondisi jalur telepon. Jika kondisinya *off-hook*, program akan mendeteksi adanya nada sambung (*dial tone*). Jika *dial tone* terdeteksi, program akan melakukan DTMF *dialing* ke nomor tujuan yang sudah diprogram. Langkah selanjutnya adalah mendeteksi *call progress*. Jika nada sibuk terdeteksi, program akan mengirimkan “busy” ke komputer. Jika nada tunggu terdeteksi, program akan mengirimkan “ringback”. Jika sambungan terhubung, program akan mengirimkan “connect”. Jika pespon yang terhubung ke Phone Interface ditutup, program akan mengirimkan “on hook”. *Flowchart* program test9.c adalah sebagai berikut:



Gambar 8
Flowchart program test9.c (bagian 1)



Gambar 9
Flowchart program test9.c (bagian 2)

Program test9.c akan diproses sebagai berikut:

- Proses yang pertama dilakukan adalah konfigurasi pin untuk jalur komunikasi dengan Phone Interface, yaitu:
 - mt8888 = jalur mengirim/menerima data,
 - rs0 = *register select*,
 - re = untuk menghubungkan pespon dengan jalur telepon,
 - hk = status *on-hook / off-hook*,
 - ri = status dering,
 - int0 = interupsi penekanan tombol,
 - cs = *chip select*,
 - wr = pin kontrol untuk proses tulis,
 - rd = pin kontrol untuk proses baca.

- t0 = status *dial tone*
2. Kemudian deklarasi variabel `init_mode` dan `flag`, yaitu:
 - `init_mode` = variabel yang dipakai untuk menginisialisasi `mt8888`,
 - `flag` = variabel yang berisi status dari *call progress*.
 3. Program melakukan inisialisasi pin kontrol (`re = wr = rd = rs0 = cs = int0 = hk = 1`) dan port serial (*baudrate* 9600 bps, 8 bit data, 1 bit *stop*, tanpa bit *parity*).
 4. Setelah itu dilakukan prosedur *reset* pada `mt8888` sebelum rutin-rutin yang lain dijalankan.
 5. Kemudian dilakukan inisialisasi terhadap `mt8888` dengan mode DTMF1 (0Dh).
 6. Program menunggu gagang telepon diangkat dan terdengar *dial tone*.
 7. Setelah terdeteksi adanya *dial tone*, program melakukan DTMF *dialing* ke nomor telepon yang telah diprogram.
 8. Kemudian dilakukan inisialisasi terhadap `MT8888` dengan mode CP (0Eh).
 9. Jika gagang telepon ditutup, maka program akan mengirim "Onhook" ke komputer lalu kembali ke langkah 6.
 10. Jika gagang telepon tidak ditutup, maka akan dilakukan inisialisasi terhadap `MT8888` dengan mode DTMF1 (0Dh).
 11. Program membaca status *call progress*.
 12. Jika terdapat pulsa *high* pada pin `t0` selama lebih dari / sama dengan 4 detik, maka program akan mengirim "Connect" ke komputer lalu menunggu gagang telepon ditutup (ke langkah 16).
 13. Jika terdapat pulsa *low* pada pin `t0` selama lebih dari / sama dengan 1050 ms, maka program akan mengirim "Unknown" ke komputer lalu menunggu gagang telepon ditutup (ke langkah 16).
 14. Jika terdapat pulsa *low* pada pin `t0` selama kurang dari 550 ms, maka program akan mengirim "Busy" ke komputer lalu menunggu gagang telepon ditutup (ke langkah 16).
 15. Jika terdapat pulsa *low* pada pin `t0` selama antara 550 ms hingga 1050 ms, maka program akan mengirim "Ringback" ke komputer lalu kembali ke langkah 11.
 16. Jika gagang telepon ditutup, maka program akan kembali ke langkah 6.

Komunikasi serial pada komputer dapat menggunakan Terminal[®] atau HyperTerminal[®]. Pengaturan serial adalah *baudrate* 9600 bps, 8 bit data, 1 bit *stop*, tanpa bit *parity*, tanpa *flow control*.

Listing program terdapat pada **AN78.ZIP**.

Selamat berinovasi!

μC/51 is copyright by Wickenhäuser Elektrotechnik.
Hyper Terminal is copyright by Hilgraeve Inc.
Terminal is copyright by Bray++.