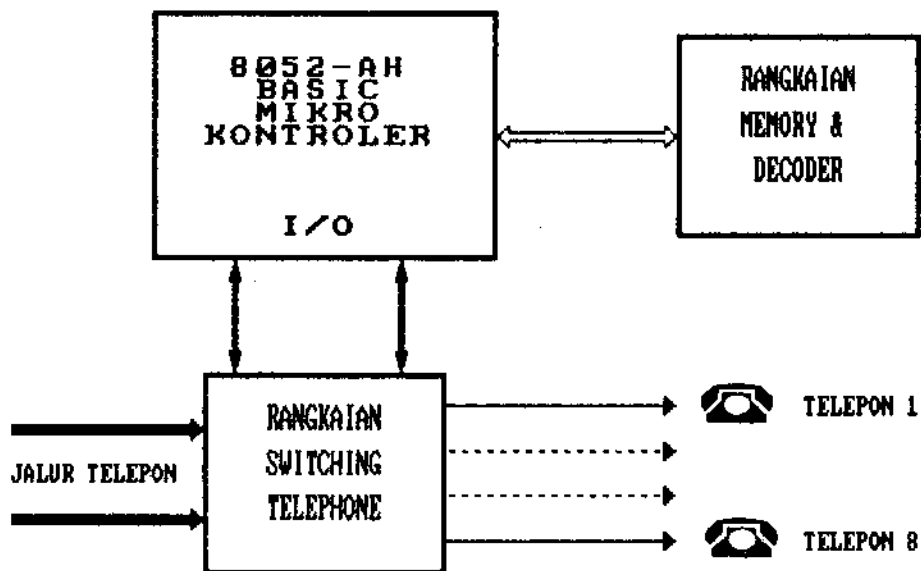


III . PERENCANAAN

1. PENDAHULUAN

Sistem telephone exchange yang akan direncanakan ini dikontrol oleh suatu BASIC komputer dengan mikrokontroler 8052-AH BASIC. BASIC computer ini tidak memiliki keyboard dan layar sendiri, karena itu fungsi komunikasi ini dilakukan oleh suatu terminal secara eksternal yang dihubungkan dengan komputer secara bidireksional, serial I/O port.

Fungsi-fungsi sistem telephone exchange ini dapat diganti atau diubah dengan mudah dengan cara mengganti program kontrolnya. Untuk memodifikasi program kontrolnya hanya dibutuhkan suatu personal computer dengan program komunikasi dan hubungan tiga jalur dengan BASIC komputer tersebut. Setelah program kontrol sudah ditest dan dapat berjalan dengan baik maka terminal atau personal computer tersebut tidak lagi diperlukan dan BASIC computer dapat berfungsi sebagai kontroler yang berdiri sendiri.



GAMBAR 3.1

BLOK DIAGRAM SISTEM YANG DIRENCANAKAN

Interface board dari sistem telephone exchange menghasilkan serangkaian pulsa "call charge" dan sinyal deringan sementara fungsi kontrolnya dilakukan oleh BASIC computer. Fungsi dari interface board tersebut adalah untuk mengkonversi sinyal digital yang dihasilkan oleh board komputer menjadi sinyal-sinyal jaringan telepon, dan sebaliknya.

2. PERENCANAAN SISTEM

Sistem ini dirancang untuk menghubungkan delapan buah pesawat telepon dengan pulse-dialing. Semua pesawat telepon tersebut dihubungkan dengan jalur tunggu (WAIT

line) dan jalur suara (VOICE line). Untuk melakukan percakapan telepon suatu pesawat harus dihubungkan dengan voice line yang memberikan catu daya, sedangkan pesawat yang tidak terlibat dalam percakapan dihubungkan dengan wait line yang akan menimbulkan nada sibuk jika gagangnya diangkat.

Beberapa relay digunakan untuk melakukan switching pada pesawat telepon antara wait line dan voice line serta untuk mentransfer sambungan telepon yang diterima dari jalur eksternal menuju pesawat telepon lainnya dalam jaringan.

Jika ada sambungan telepon dari luar, pesawat telepon pertama yang menjawab sambungan tersebut akan dihubungkan dengannya dan sambungan dari luar ini dapat ditransfer ke pesawat lainnya. Caranya yaitu telepon yang pertama menjawab sambungan meletakkan gagang teleponnya dan memutar nomer internal dari pesawat telepon yang diinginkan. Akan tetapi ada batas waktu yang diijinkan bagi telepon pertama meletakkan gagang teleponnya dan batas ini diatur dalam program kontrolnya.

3. PERENCANAAN BASIC COMPUTER

BASIC computer yang merupakan pusat pengendali sistem ini terdiri dari komponen utama, seperti mikrokontroler 8052-AH BASIC, Latch 74LS373 untuk men-

demultiplex jalur alamat/data , RAM 6264, EPROM 27128 yang memiliki kapasitas penyimpanan program sebesar 16K bytes, dan sebuah dekoder 74LS138 untuk memilih lokasi memory yang akan dialamati, serta beberapa IC TTL.

Selain itu BASIC computer juga dilengkapi dengan rangkaian-rangkaian penunjang seperti rangkaian pembangkit sinyal clock dan rangkaian reset.

3.1 Rangkaian Clock

Rangkaian clock yang direncanakan ini menggunakan sebuah kristal oscillator dua buah kapasitor yang bernilai 10 pF - 30 pF untuk kristal dan 10 pF - 40 pF untuk resonator keramik. Karena oscillator yang digunakan adalah dari jenis kristal, maka diambil nilai tengahnya dari data book sebesar 20 pF.

Sedangkan kristal yang digunakan bernilai 11,0592 MHz dengan alasan frekuensi oscillator juga dipakai sebagai pembangkit baud rate sehingga dipilih frekuensi yang merupakan kelipatan dari nilai baud rate. Akan tetapi karena kristal dengan nilai tersebut sulit diperoleh di pasaran maka sebagai pendekatannya dipakai kristal senilai 11 MHz. Frekuensi kristal yang tepat dibutuhkan untuk timing yang tepat bagi serial channel, real time clock, dan pulsa-pulsa pemrograman EPROM.

Rangkaian clock yang dianjurkan oleh Intel,Co. selaku pembuat mikrokontroler ini menggunakan fasilitas 'internal clock generator' yang terdapat di dalam mikrokontroler 8052-AH BASIC. Gambar 3.2 menunjukkan rangkaian clock dan reset yang direncanakan dalam BASIC computer.

3.2 Rangkaian Reset

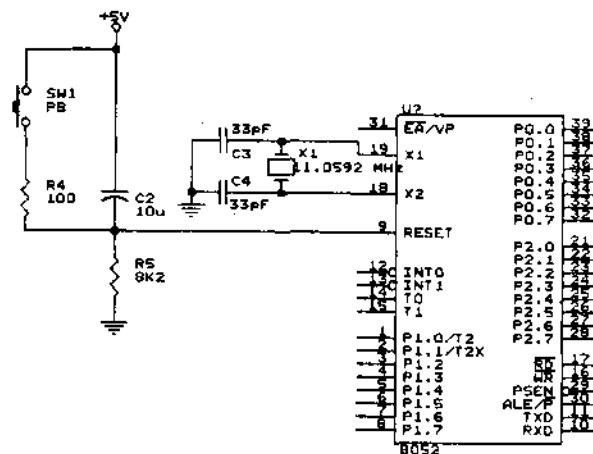
Rangkaian reset yang direncanakan harus mempunyai kemampuan 'power-on reset' dan reset secara manual dengan menggunakan saklar push-on. Syarat yang harus dipenuhi untuk mendapatkan kedua kemampuan itu adalah:

- a. Lebar pulsa untuk sinyal reset harus lebih besar dari 24 periode oscillator (2 siklus mesin).
- b. Rise time dari Vcc tidak melebihi 1 ms.
- c. Waktu start-up dari oscillator tidak melebihi 10 ms.

Syarat nomer 2 dan 3 hanya berlaku jika digunakan 'power on reset', yaitu reset yang dilakukan pada saat power (Vcc) diberikan.

Cara kerja dari rangkaian reset yang ditunjukkan oleh gambar 3.2 adalah sebagai berikut: bila tegangan supply Vcc mulai diberikan maka akan mengalir arus pada kapasitor yang menimbulkan beda tegangan pada resistor. Tegangan pada pin 9

(RST) merupakan beda tegangan antara V_{cc} dan tegangan pada kapasitor (V_c). Karena sifat kapasitor yang berlaku sebagai hubungan singkat pada saat awal pengisian, maka tegangan pada pin RST akan menurun sampai mencapai tegangan 'lower threshold' dari internal Schmitt Trigger sebesar 2,5 Volt. Untuk memberikan efek 'power-on reset' maka besarnya waktu penurunan tegangan dari V_{cc} sampai menjadi 2,5 Volt harus lebih besar dari 2 siklus mesin ditambah dengan waktu 'start-up' dari oscillator.



GAMBAR 3.2

RANGKAIAN CLOCK DAN RESET

Untuk keperluan reset secara manual digunakan saklar push-on yang bila ditekan maka pin RST akan memperoleh tegangan sebesar V_{cc} yang menyebabkan mikrokontroler berada pada keadaan

reset.

Jika diasumsikan waktu start-up oscillator 10 ms dan frekuensi oscillator 11 MHz, maka lebar pulsa untuk pin RST :

$$t > (10 \cdot 10^{-3}) + (2 \cdot 12 \cdot 10^{-6} / 11)$$

$$t > 10,0021 \text{ ms}$$

Dengan menggunakan ketentuan bahwa tegangan 'lower threshold' dari pin RST sebesar 2,5 Volt dan lebar pulsa (t) sebesar 10,0021 ms, maka dengan rumus:

$$V_{\text{reset}} = V_{\text{cc}} (1 - e^{-t/RC}) \dots\dots\dots (3.1)$$

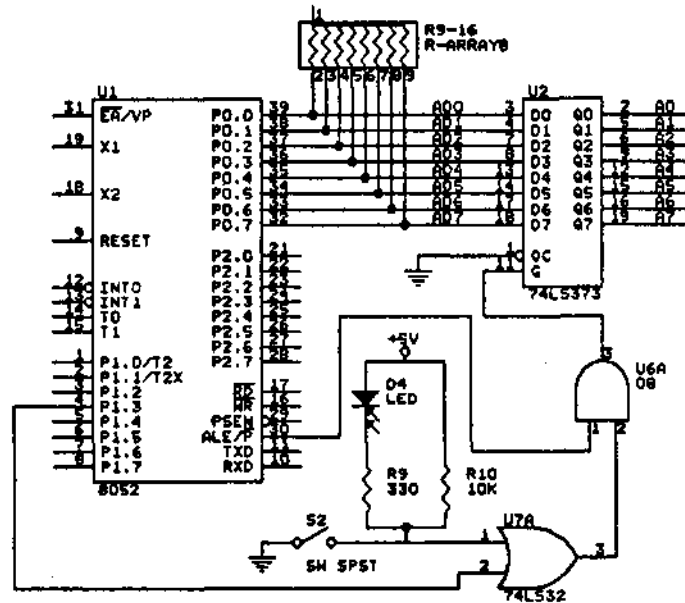
maka akan diperoleh nilai RC sebesar 14,42998 ms. Nilai R diambil sebesar 8K2 dan kapasitor sebesar 10 pF berdasarkan pertimbangan nilai-nilai komponen yang banyak terdapat di pasaran.

3.3 Address Latch

Karena port 0 merupakan multiplex antara alamat byte rendah dan data byte maka untuk memperoleh byte alamat dan byte data secara terpisah harus dilakukan proses demultiplexing. Untuk keperluan ini digunakan suatu octal latch 74LS373 yang rangkaianannya ditunjukkan pada gambar 3.3.

Proses demultiplex dilakukan dengan bantuan sinyal ALE (Address Latch Enable) dari mikrokontroler 8052-AH BASIC. Pada saat melakukan

pemrograman EPROM, pin P1.3 (ALE Disable) men-disable sinyal ALE yang menuju address latch eksternal.



GAMBAR 3.3
ADDRESS LATCH

Perangkat lunak yang dibuat tidak dapat mengontrol lebar pulsa ALE secara langsung, karena itu hal ini diatasi dengan bantuan gerbang OR U1 dan gerbang AND U6 serta level logika "0" pada output mikrokontroler pin P1.3.

3.4 Rangkaian Decoding

Struktur memory dari 8052-AH BASIC tidak mengikuti ketentuan model von Neumann's, yaitu program memory berbeda dengan data memory. Struktur memory 8052-AH BASIC ditunjukkan pada

gambar 3.4 dimana RAM dialamati mulai dari alamat 0000H dan EPROM dialamati mulai dari alamat 8000H pada mode pemrograman EPROM.

FFFF H	INTERFACE
E000 H	BOARD
DFFF H	K O S O N G
C000 H	
BFFF H	E P R O M 2 7 1 2 8
8000 H	
7FFF H	R A M EXTENTION
4000 H	
3FFF H	R A M (I) 6 2 6 4
2000 H	
1FFF H	R A M (II) 6 2 6 4
0000 H	

GAMBAR 3.4
PETA ALAMAT MEMORY

Dekoder 74LS138 membagi lokasi memory ke dalam blok 8 KByte. Input alamat A, B dan C dari 74LS138 adalah bit alamat A₁₃, A₁₄ dan A₁₅ dari mikrokontroler 8052-AH BASIC. Agar dekoder tetap dalam keadaan aktif maka input enable G₁ dihu-

bungkan dengan tegangan supply V_{CC} sedangkan kedua input enable lainnya yaitu G_{2A} dan G_{2B} dihubungkan ke ground karena kedua pin tersebut 'active low'.

Output dari dekoder digunakan untuk mengaktifkan memory yang diberikan pada pin CS dari masing-masing memory. Output Y_0 akan mengamati RAM (I) pada alamat 0000H-1FFFH sedangkan Y_1 mengamati RAM (II) pada alamat 2000H-3FFFH. Karena dekoder menggunakan interval memory 8 KByte, sedangkan EPROM yang digunakan berkapasitas 16K Byte maka output Y_4 dan Y_5 dimasukkan ke gerbang AND untuk mengkombinasi dua blok dan output dari gerbang AND dihubungkan dengan pin CE dari EPROM 27128.

TABEL 3.1

TABEL KEBENARAN DEKODER

A15 A14 A13	OUTPUT 74LS138								DEVICE
	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0	
0 0 0	1	1	1	1	1	1	1	0	RAM (I) 6264
0 0 1	1	1	1	1	1	1	0	1	RAM (II) 6264
0 1 0	1	1	1	1	1	0	1	1	RAM EXTENTION
0 1 1	1	1	1	1	0	1	1	1	RAM EXTENTION
1 0 0	1	1	1	0	1	1	1	1	EPROM 27128
1 0 1	1	1	0	1	1	1	1	1	EPROM 27128
1 1 0	1	0	1	1	1	1	1	1	KOSONG
1 1 1	0	1	1	1	1	1	1	1	INTERFACE BOARD

Output dekoder lainnya tidak digunakan dalam perencanaan ini tetapi ditujukan untuk keperluan ekspansi memory.

3.5 Hubungan Mikrokontroler 8052-AH BASIC dengan Memory

Pada saat port 0 digunakan pada mode I/O dibutuhkan resistor pull-up pada output open drain yang bernilai $10K\Omega$. Biasanya port ini berfungsi sebagai jalur data dan alamat, tetapi berfungsi sebagai port I/O pada mode pemrograman EPROM. Suatu octal latch 74LS373 memisahkan kedua jenis informasi itu dan sinyal ALE digunakan untuk mengaktifkan latch sehingga outputnya berupa alamat bagian bawah ($A_0 - A_7$) yang dihubungkan dengan jalur alamat byte rendah yang sesuai dengan memory RAM 6264 / EPROM 27128. Sedangkan jalur alamat byte tinggi disediakan oleh mikrokontroler melalui port 2.

Byte alamat $A_8 - A_{12}$ dihubungkan dengan RAM 6264 sedangkan $A_8 - A_{13}$ diletakkan pada EPROM 27128 yang berkapasitas 16K byte. Sisanya berupa byte alamat $A_{14} - A_{15}$ dimasukkan ke dalam dekoder yang memilih lokasi memory yang dialamati.

Byte data $AD_0 - AD_7$ dari port 0 yang sudah dide-multiplex dapat langsung dihubungkan dengan jalur data $D_0 - D_7$ dari piranti memory.

Pin 20 (CS1) dari RAM dihubungkan dengan output dari dekoder Y_0 dan Y_1 sesuai dengan peta memory yang telah ditentukan. Pin 26 (CS2) dari RAM yang bersifat 'active high' dihubungkan dengan tegangan V_{CC} . Karena memory RAM dapat ditulisi dan dibaca maka operasi tulis akan dilakukan dengan sinyal WR dari mikrokontroler sedangkan operasi baca dengan sinyal RD yang dihubungkan dengan pin 22(OE) dari RAM.

Pin 20 (CE) dari EPROM dihubungkan dengan output dari gerbang AND yang menggabungkan Y_4 dan Y_5 , sedangkan pin OE dihubungkan dengan output gerbang AND yang menggabungkan sinyal PSEN (Program Store Enable), yaitu kontrol untuk operasi baca pada program memory eksternal dan sinyal RD. Hal ini disebabkan struktur memory dari 8052-AH BASIC yang tidak mengikuti model von Neumann's: program memory berbeda dengan data memory. Pin 27 (PGM) dari EPROM dihubungkan dengan pin P1.4 (Program Pulse) yang digunakan untuk pemrograman EPROM.

Hubungan antara mikrokontroler 8052-AH BASIC dengan memory-memory ditunjukkan oleh gambar 3.5.

3.6 Rangkaian Pemrograman EPROM

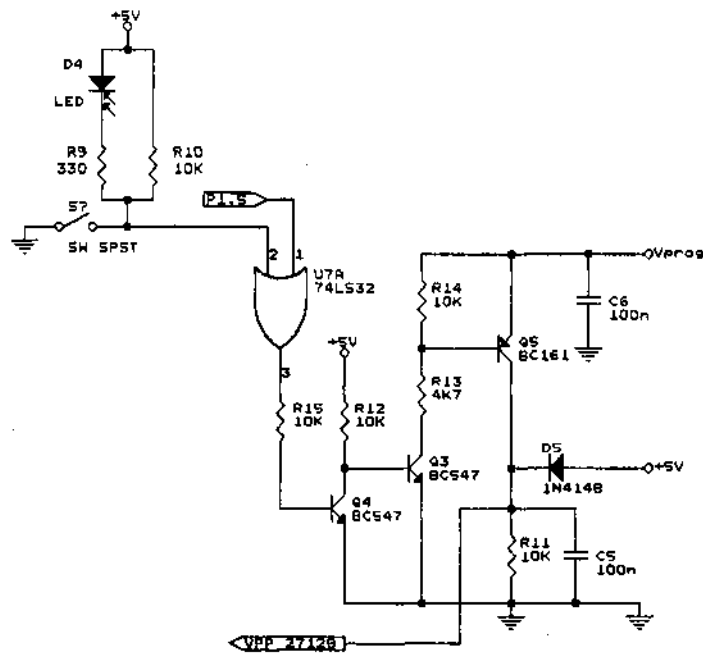
BASIC computer memiliki fasilitas 'on-board EPROM Programmer' yang dikendalikan secara langsung

oleh mikrokontroler 8052-AH BASIC. Pada saat berkomunikasi dengan pemakai melalui terminal, 8052-AH BASIC dapat menyimpan, mengedit, mendebug, dan mengetest subrutin-subrutin BASIC di dalam EPROM. Beberapa parameter program akan disimpan bersama-sama dengan programnya, seperti parameter baud rate, variabel MTOP, autoexecute flag, dll.

Beberapa jalur dari port 1 mikrokontroler 8052-AH BASIC digunakan untuk keperluan pemrograman ini, yaitu jalur P1.3, P1.4, dan P1.5. Jalur P1.3 berfungsi untuk men-disable sinyal ALE yang menuju ke address latch selama pemrograman. P1.3 biasanya adalah suatu TTL yang berlevel HIGH, akan tetapi pada saat pemrograman akan menjadi LOW. Dalam penggunaannya P1.3 harus dimasukkan ke dalam gerbang AND bersama-sama dengan sinyal ALE. Jalur P1.4 (Program Pulse) menyediakan pulsa pemrograman untuk menuliskan data pada EPROM, sedangkan port P1.5 (Program Enable) digunakan untuk mengontrol tegangan pemrograman, pin ini berlogika LOW pada waktu pemrograman.

Selama pemrograman, port 0 dan port 2 tidak digunakan sebagai data/address port seperti biasanya. Mula-mula mikrokontroler 8052-AH BASIC menset pin P1.5 menjadi LOW. Pada saat ini, transistor T3, T4, dan T5 memastikan bahwa

tegangan pemrograman sudah mencapai terminal V_{PP} dari EPROM.



GAMBAR 3.6

RANGKAIAN PEMROGRAMAN EPROM

Mikrokontroler akan menempatkan byte alamat rendah pada jalur $AD_0 - AD_7$ (port 0) dan men-disable ALE dengan membuat jalur P1.3 berlogika LOW. Byte alamat akan tetap di-latch pada IC 74LS373 selama sisa siklus pemrograman. Byte alamat tinggi diberikan pada jalur $A_8 - A_{15}$ (port 2) dan byte data pada jalur $D_0 - D_7$ dari EPROM yang akan diprogram. Selanjutnya output P1.4 dibuat berlogika LOW dan EPROM mulai diprogram sebab sinyal PGM akan menjadi LOW jika diberikan tegangan pemrograman V_{PP} . Panjang pulsa PGM ditentukan oleh frekuensi clock dari mikrokon-

troler. Switch S2 berfungsi untuk mem-blok tiga sinyal pemrograman tersebut dengan alasan keamanan, karena port P1 dapat digunakan untuk keperluan lainnya selain pemrograman EPROM.

Gambar 3.6 menunjukkan rangkaian pemrograman EPROM ini.

4. PERENCANAAN SISTEM TELEPHONE EXCHANGE

Sistem yang direncanakan memiliki kemampuan menghubungkan delapan buah pesawat telepon pulse-dialing. Interface board ini dihubungkan dengan BASIC computer melalui sebuah konektor 40 pin yang membawa sinyal-sinyal yang dibutuhkan untuk komunikasi antara kedua unit ini, antara lain sinyal Y₇, PWM, INT, jalur alamat, jalur data, dan sinyal baca RD serta sinyal tulis WR.

Sinyal Y₇ yang dihasilkan dekoder 74LS138 dari BASIC computer digunakan untuk memilih logika pada interface board yang dialamati mulai dari address E000H - FFFFH. Range alamat ini dibagi menjadi tiga bagian untuk tiga buah buffer 74LS377 (I), 74LS377 (II), dan 74LS245 dengan bantuan jalur alamat A₁₀, A₁₁, dan A₁₂.

Sinyal INT yang 'active low' diberikan oleh interface board yang berfungsi untuk mengaktifkan kembali mikrokontroler dari keadaan stand-by pada saat sinyal deringan dideteksi pada jalur eksternal. Mikrokontroler 8052-AH BASIC menghasilkan nada sibuk

(engaged tone) pada jalur PWM.

Latch 74LS377 berfungsi sebagai register I/O tambahan untuk mengontrol fungsi switching pada exchange. Untuk fungsi switching ini dilakukan dengan bantuan relay. Register 8-bit 74LS377 (I) mengontrol relay Re1 - Re8 melalui power driver ULN2803. Relay-relay ini digunakan untuk men-switch perangkat telepon yang bersangkutan antara jalur VOICE dan jalur WAIT. Sedangkan ketiga jalur data LS dari IC 74LS377 (II) berfungsi untuk men-switch relay Re9, Re10, dan Re11. Relay Re9 berfungsi untuk membangkitkan sinyal deringan sedangkan relay Re10 digunakan untuk mentransfer sambungan telepon yang diterima pada jalur eksternal menuju ke pesawat telepon lainnya di dalam exchange. Dengan menswitch Re10 maka jalur eksternal dilepaskan pada impedansi tertentu dan exchange dapat digunakan untuk sambungan internal.

Relay Re11 berfungsi untuk menghubungkan jalur eksternal sistem exchange dengan dengan telepon lainnya yang berada di luar exchange.

4.1 Rangkaian Pembangkit Sinyal Deringan

Relay Re9 digunakan untuk membangkitkan sinyal deringan (ringing signal). Pada keadaan stand-by, transistor T1 dihubungkan dengan jalur VOICE dan memberikan tegangan supply bagi semua perangkat telepon melalui jalur VOICE. Pada saat relay Re9 diswitch, jalur VOICE akan memperoleh tegangan

dari transformer tegangan Tr1 sebesar 24 Volt AC. Akibatnya bell di dalam pesawat telepon yang dihubungkan dengan jalur VOICE akan berbunyi. Pihak yang melakukan sambungan telepon akan mendengar sinyal deringan ini sebagai serangkaian nada desis (buzzing tone). Rangkaian ini ditunjukkan oleh gambar 3.7.

4.2 Melakukan Sambungan Telepon

Pada saat salah satu gagang telepon diangkat untuk melakukan sambungan telepon, maka akan mengalir arus yang menyebabkan optocoupler aktif. Hal ini akan mengakibatkan jalur INP yang bersesuaian ditarik menjadi logika LOW. Mikrokontroler 8052-AH BASIC mengidentifikasi pesawat yang melakukan sambungan telepon dengan membaca logika "0" yang dihasilkan pada IC 74LS245 pada alamat EFFFH. Selanjutnya mikrokontroler akan memberikan perintah tulis kepada IC 74LS377 (I) dan 74LS377 (II) pada alamat FBFFH dan F7FFH untuk menghubungkan semua pesawat telepon lainnya ke jalur WAIT. Pesawat-pesawat telepon ini di-disable dan akan menghasilkan nada sibuk (engaged tone) pada saat gagangnya diangkat.

Mikrokontroler menghitung pulsa putaran (dialing pulse) yang dihasilkan oleh pesawat yang

melakukan sambungan melalui IC 74LS245. Nomer pertama yang diputar menentukan proses yang akan terjadi selanjutnya dan hal ini diatur di dalam program kontrolnya. Misalnya karena nomer 0 merupakan nomer awal untuk sambungan eksternal maka pada saat nomer ini diputar, relay Re11 diaktifkan dan jalur eksternal dihubungkan dengan telepon luar. Transformer jalur Tr2 dihubungkan dengan jalur eksternal dan semua pulsa putaran yang mengikuti angka 0 diberikan pada jalur eksternal dengan bekerjanya relay Re11. Kontak relay menswitch antara impedansi rendah (transformer jalur Tr2) dan impedansi tinggi (detektor pulsa deringan). Pulsa-pulsa putaran diberikan kepada exchange melalui gerbang-gerbang OR U11, U12, dan U13, serta gerbang NAND U14 setelah gerbang OR U12a-U12d atau U13a-U13d telah dienable melalui IC 74LS245. Gerbang U11d memastikan bahwa pulsa-pulsa putaran yang dihasilkan oleh salah satu pesawat internal tidak dilewatkan pada jalur eksternal pada waktu exchange digunakan secara internal.

4.3 Menerima Sambungan Dari Luar

Sambungan yang diterima dari luar melalui jalur eksternal dideteksi oleh detektor pulsa deringan yang terdiri dari D3, D4, D8, R4, C3, dan

optocoupler ISO10. Jika suatu pulsa deringan dideteksi, optocoupler ISO10 menarik jalur INT1 dari mikrokontroler 8052-AH BASIC menjadi logika LOW. Hanya pesawat yang diijinkan menerima sambungan dari luar yang tetap berada pada jalur VOICE, pesawat lainnya dihubungkan pada jalur WAIT. Sinyal deringan diberikan pada jalur VOICE dengan bantuan relay Re9.

Setelah sambungan tersebut dijawab, jika diinginkan sambungan ini dapat ditransfer menuju pesawat telepon lainnya dengan bantuan relay Re10 yang jika diswitch akan melepaskan jalur eksternal untuk sementara pada impedansi tertentu, yaitu sebesar R5 (680 Ω) dan exchange dapat dipakai secara internal, yaitu untuk mentransfer sambungan telepon ke pesawat lainnya.

5. PERENCANAAN PERANGKAT LUNAK

Perangkat lunak untuk sistem ini ditulis di dalam bahasa pemrograman BASIC yang telah banyak dikenal dan disimpan di dalam EPROM. Di dalam program kontrol ini ditetapkan beberapa alamat yang penting untuk proses baca-tulis, antara lain:

- address 0EFFFH untuk IC 74LS245 (input data address)
- address 0F7FFH untuk IC buffer 74LS377 (I) yang mengendalikan relay Re1 - Re8

- address 0FBFFH untuk IC buffer 74LS377 (II) yang mengontrol relay Re9 - Re11.

Di dalam program kontrol terdapat beberapa main rutin yang memiliki fungsi-fungsi, seperti:

- nomer "0" digunakan sebagai kode untuk sambungan eksternal yang diikuti dengan nomer telepon eksternal
- nomer "1" adalah nomer awal dari sambungan internal antar pesawat telepon di dalam exchange, yang diikuti dengan nomer dari setiap pesawat telepon yang dimulai dari nomer 1 sampai dengan 8
- nomer "2" berfungsi sebagai kode awal untuk 'shortcut' bagi nomer-nomer eksternal
- kode yang digunakan untuk 'shortcut dialing' disimpan dengan pertama-tama memutar nomer "3" dan diikuti dengan kode (0 - 9) dan selanjutnya nomer dari telepon eksternal. Mikrokontroler akan menyimpan kode shortcut ini beserta nomer teleponnya jika gagang telepon diletakkan
- nomer "5" berfungsi untuk men-disable suatu pesawat telepon di dalam exchange sehingga pesawat yang bersangkutan tidak dapat menerima sambungan telepon dari luar
- untuk meng-enable kembali pesawat telepon tersebut agar dapat menerima sambungan dari luar, diputar angka "6".

Jika ada sambungan telepon yang masuk ke exchange,

akan dideteksi oleh detektor pulsa deringan. Jumlah deringan yang berbunyi diset di dalam program kontrol sebanyak maksimum 13 deringan dengan jeda (pause) 2,5 detik. Pesawat pertama yang mengangkat gagang telepon yang akan dihubungkan dengan sambungan dari luar tersebut.

Jika suatu sambungan eksternal akan ditransfer ke pesawat telepon lainnya di dalam exchange maka pesawat yang pertama kali menerima sambungan tersebut harus meletakkan gagang teleponnya untuk sementara dan memutar nomer dari pesawat telepon yang ingin dihubungkan. Jalur eksternal tidak diputuskan sampai salah satu pesawat tersebut meletakkan gagangnya lebih dari 5 detik. Jika tidak ada pesawat yang menjawab transfer sambungan tersebut maka penelpon dari luar dapat dihubungkan kembali dengan memutar nomer pesawat yang pertama menerima sambungan tersebut. .

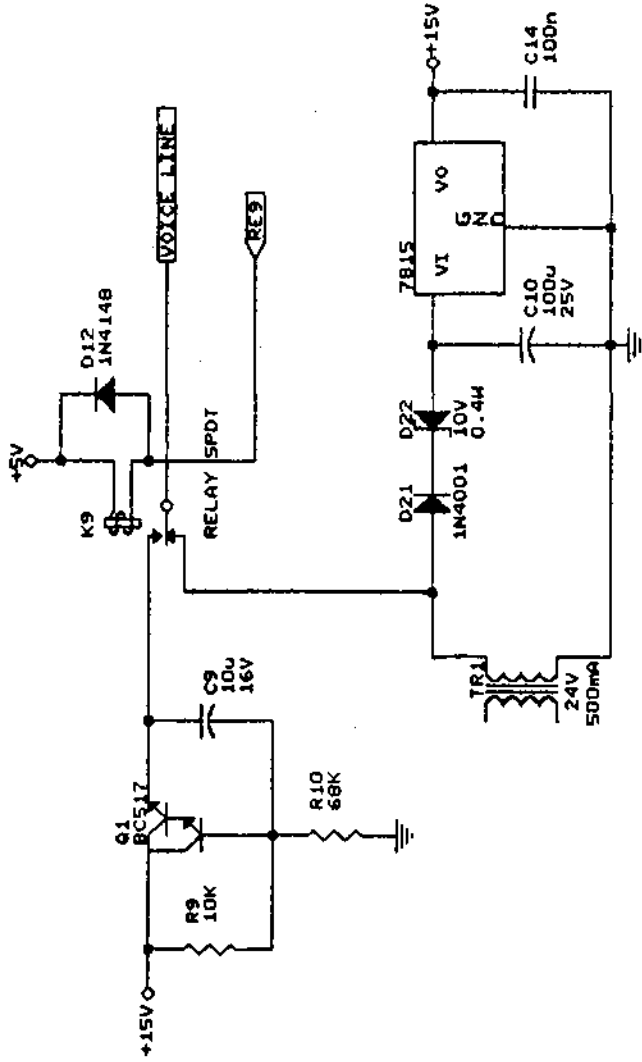
Pesawat yang melakukan sambungan ke luar dapat diidentifikasi oleh mikrokontroler dengan membaca logika "0" yang dihasilkan oleh IC 74LS245 pada alamat OEFFFFH dan nomer yang diputar dapat diketahui dengan menghitung jumlah pulsa putaran yang dihasilkan pada alamat yang sama. Selanjutnya mikrokontroler akan memberikan perintah tulis kepada IC 74LS377 (I) pada alamat OFBFFH serta IC 74LS377 (II) pada alamat OF7FFH untuk menghubungkan pesawat-pesawat lainnya ke jalur WAIT. Pesawat-pesawat ini di-disable (tidak dapat melakukan

sambungan ke luar) dan jika gagangnya diangkat akan terdengar nada sibuk (engaged tone). Nada sibuk ini dihasilkan oleh mikrokontroler pada jalur PWM (Pulse Width Modulation) yang durasi cyclenya sebesar $1/12$ frekuensi oscillator. Jika digunakan kristal 11.0592 maka frekuensi minimumnya sebesar 7,03 Hz (periode 142 ms) dan nilai maksimumnya sebesar 18,43 KHz (periode 54,25 μ s). Pada saat PWM sedang dieksekusi, mikrokontroler 8052-AH BASIC tidak dapat diinterupsi.

Untuk keperluan tertentu, real-time clock dapat diaktifkan dengan instruksi CLOCK1. Instruksi ini akan menambah nilai operator TIME setiap 5 milidetik. CLOCK1 mengeset bit-bit 7 dan 1 dari IE dan menggunakan interrupt overflow Timer0. Instruksi TIME berfungsi untuk membaca/menulis nilai dari real-time clock 8052-AH BASIC dengan menggunakan Timer0. Setelah mikrokontroler direset, TIME akan bernilai 0.

Suatu interrupt dapat ditangani dengan subrutin BASIC dengan menggunakan instruksi ONEX1. Pada saat pin 13 (INT1) dari 8052-AH BASIC berlogika LOW, mikrokontroler akan mengeksekusi perintah yang diberikan pada nomer baris sesudah instruksi ONEX1. Instruksi ini mengeset bit-bit 2 dan 7 di dalam register IE. Untuk kembali dari subrutin interrupt digunakan instruksi RET1.

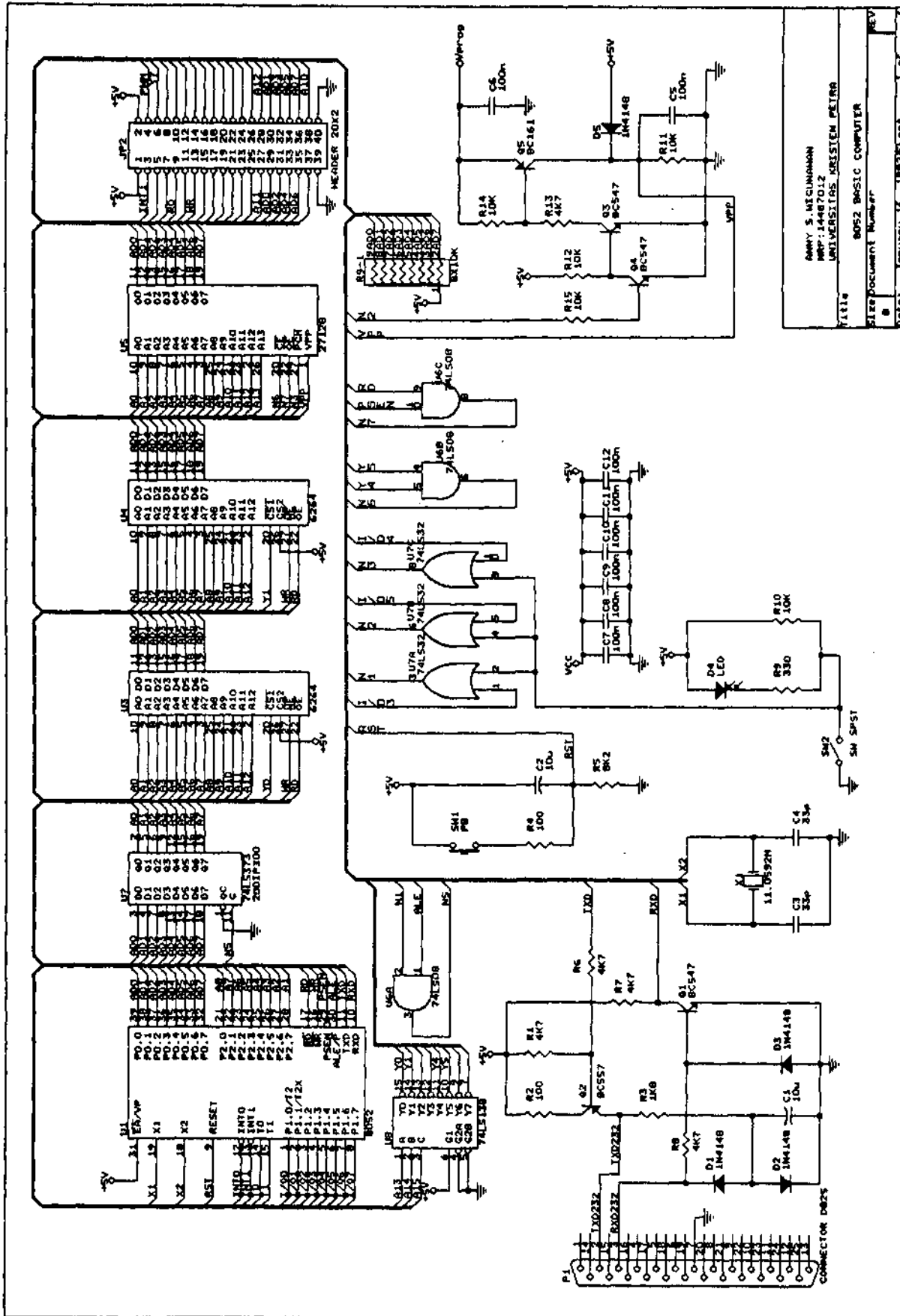
Flowchart dari program ini ditunjukkan oleh gambar 3.8.



ANNY S. HIGUNAHAN 14487012 UNIVERSITAS KRISTEN PETRA	
Title	RANGKAIAN PEMBANGKIT SINYAL DERINGAN
Size	document
Number	REV
Date:	January 17, 1992
Sheet	of

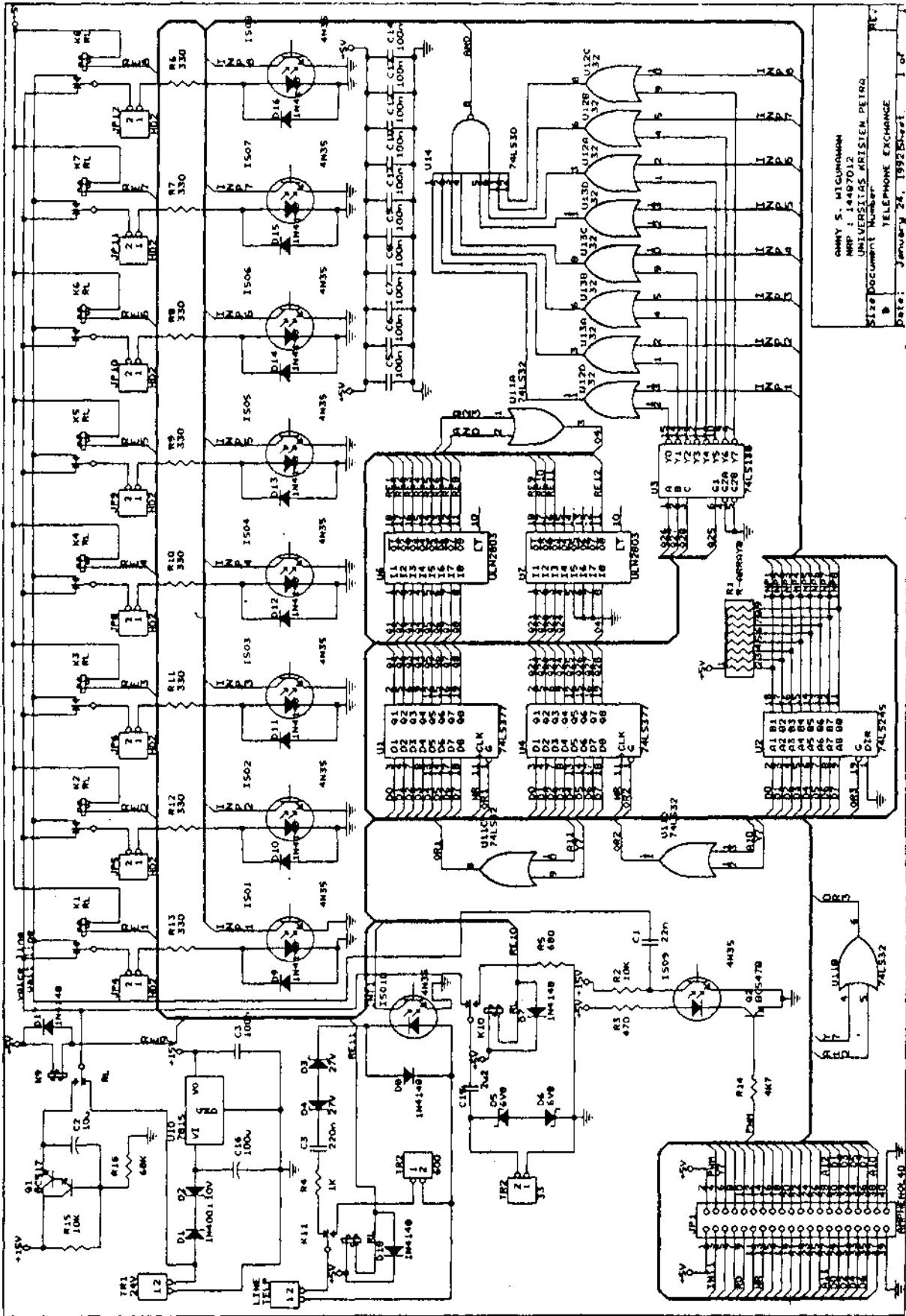
GAMBAR 3.7

RANGKAIAN PEMBANGKIT SINYAL DERINGAN



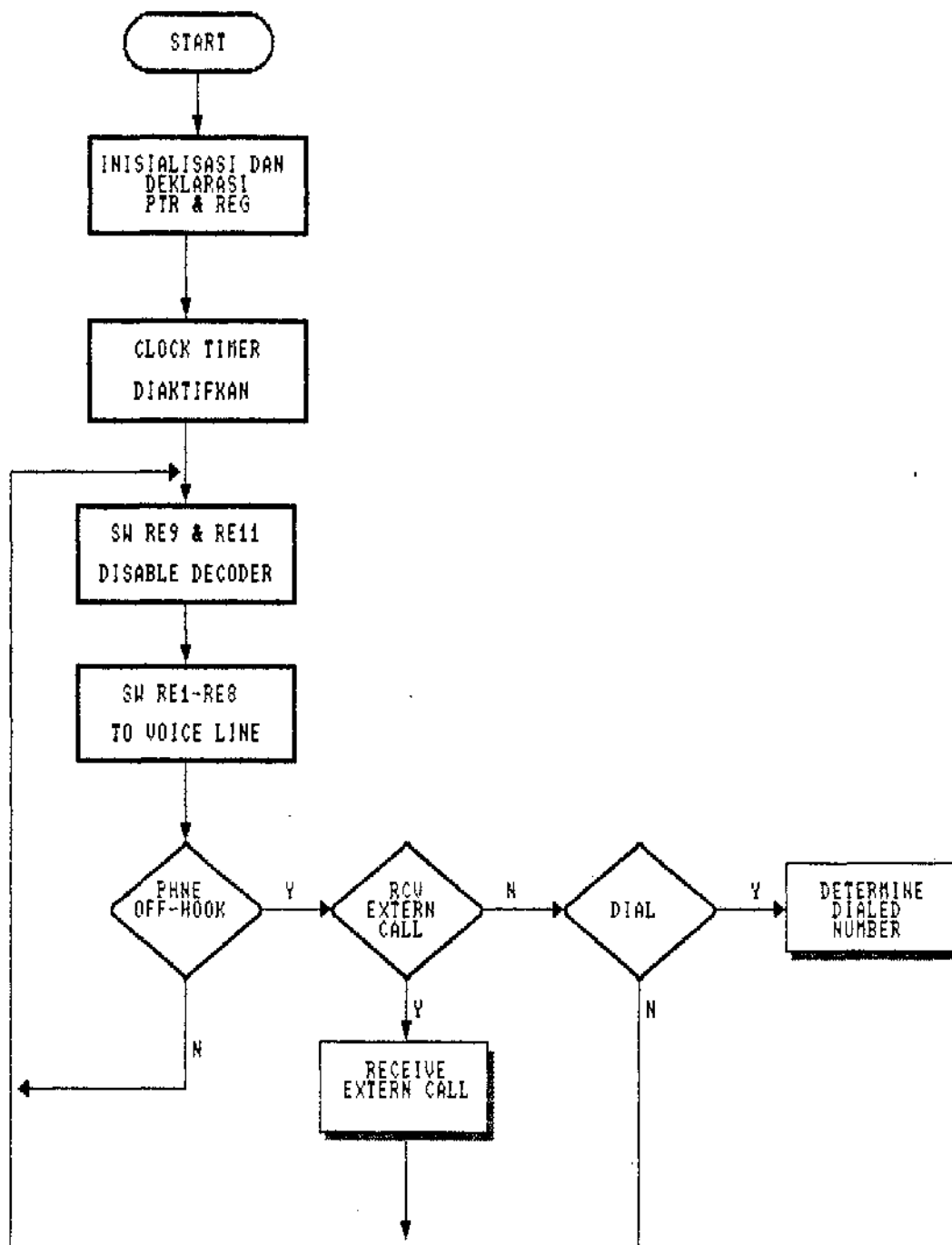
FILE # _____
 NAME: S052 BASIC COMPUTER
 DATE: January 16, 1992
 SHEET: 1 of 3

GAMBAR 3.8
 RANGKAIAN BASIC COMPUTER

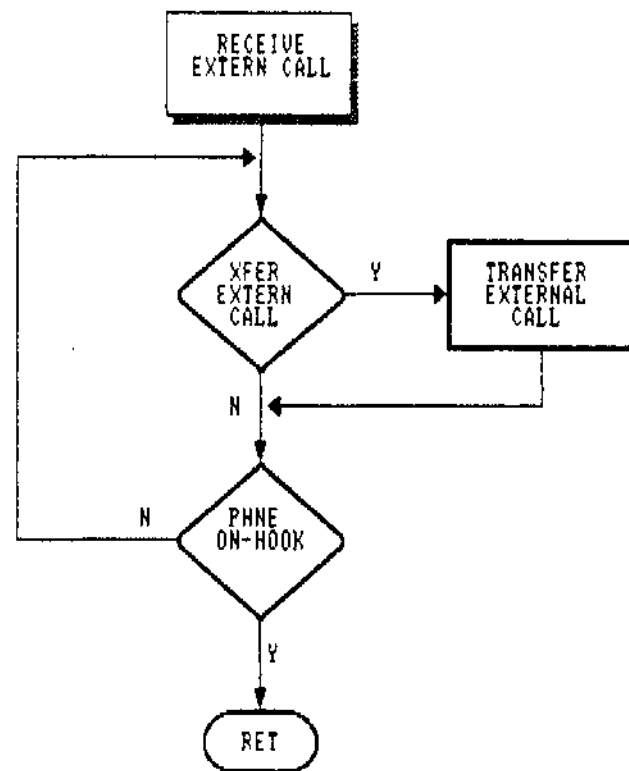


GEMMY S. MUGNANAN
 NRP : 14407012
 UNIVERSITAS ARIEN PERI
 Size Document Number
 B TELEPHONE EXCHANGE
 Date: January 24, 1992

GAMBAR 3.9
RANGKAIAN TELEPHONE KXCHANGE

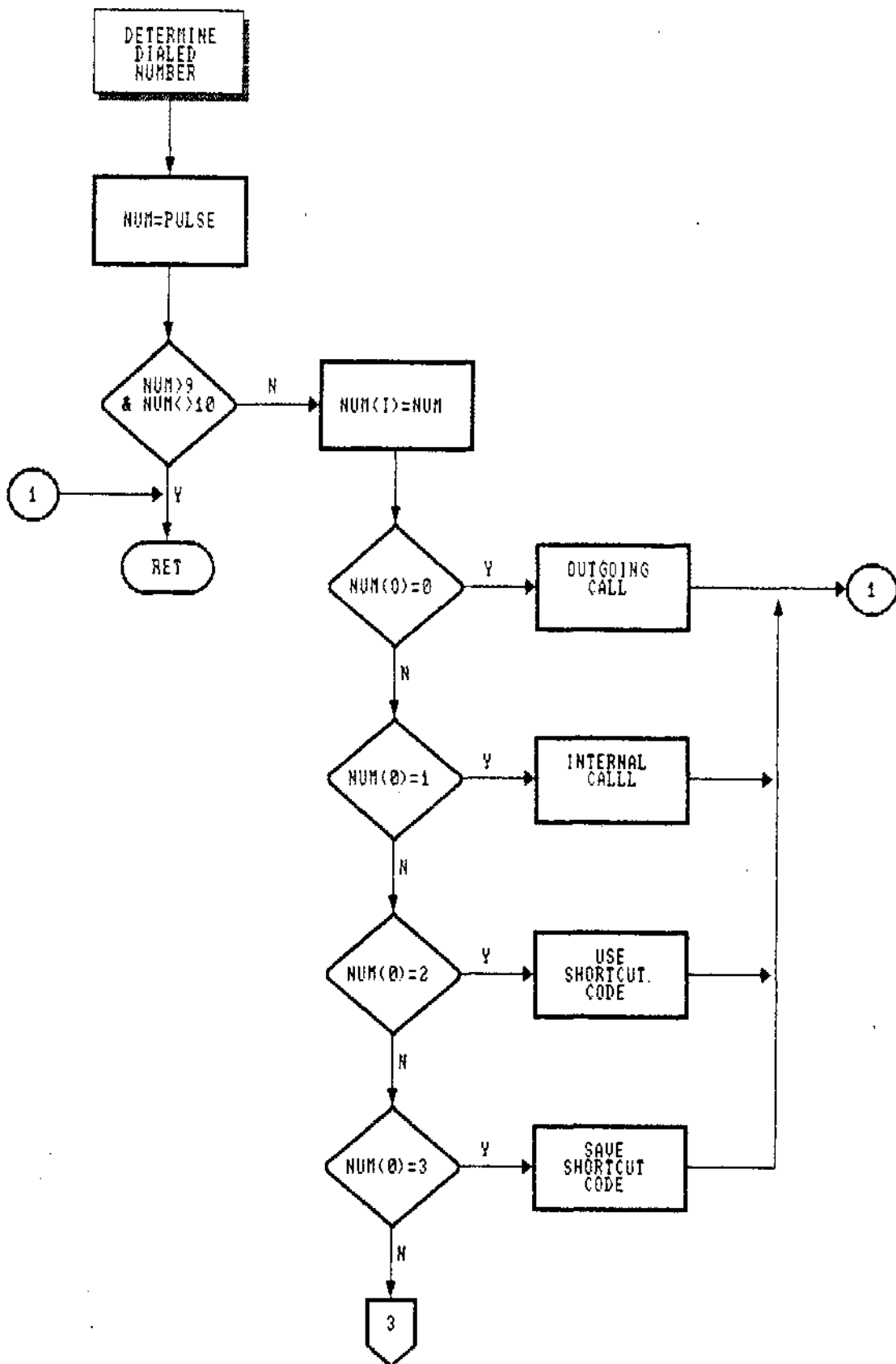


GAMBAR 3.10
DIAGRAM ALIR PROGRAM UTAMA



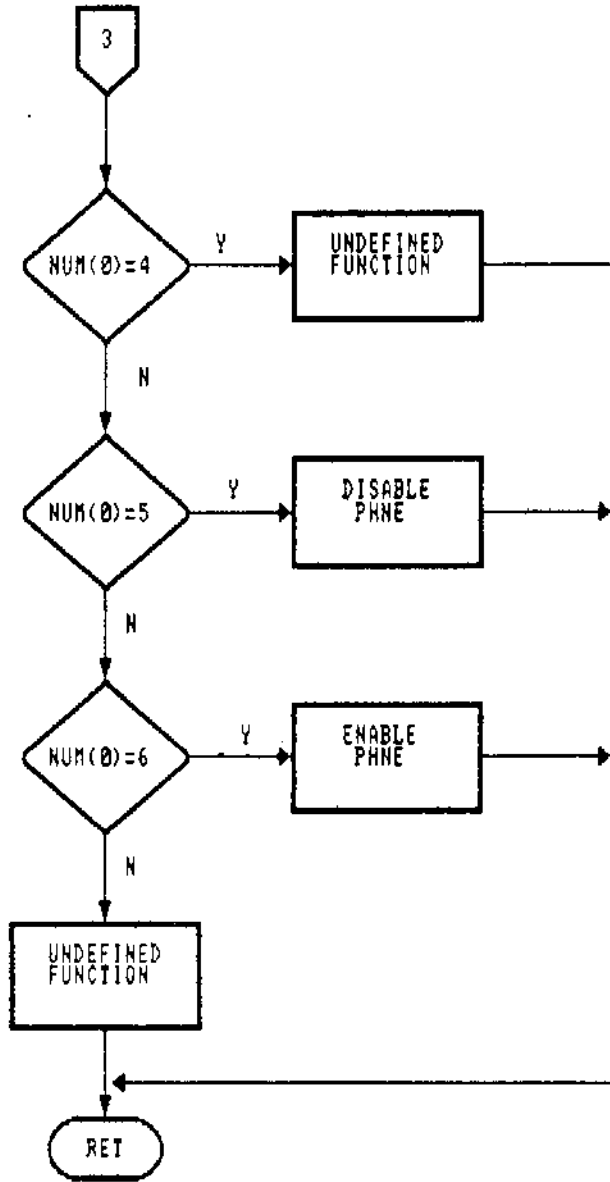
GAMBAR 3.11

DIAGRAM ALIR RUTIN RECEIVE EXTERN CALL



GAMBAR 3.12

DIAGRAM ALIR RUTIN DETERMINE DIALED NUMBER



Listing Perangkat Lunak Sistem Telephone Exchange

```

10  REM PROGRAM KONTROL UNTUK SISTEM TELEPHONE EXCHANGE
20  REM *****
30  REM NAMA : ANNY S. WIGUNAWAN
50  REM NRP  : 14487012
70  REM *****
100 REM INISIALISASI DAN DEKLARASI
110 DIM NUM(12) : REM LIST NOMOR
120 DIM SEL(10) : REM LIST EXTERNAL SWITCHING
130 INPADDR=0EFFFH : REM INPUT DATA ADDRESS
140 REL1=0F7FFH : REM ADDRESS UNTUK RELAY 1-6
150 REL2=0FBFFH : REM ADDRESS UNTUK RELAY 9-11
170 SEL(1)=010H : SEL(2)=030H : SEL(3)=050H : SEL(4)=070H
175 SEL(5)=090H : SEL(6)=0B0H : SEL(7)=0D0H : SEL(8)=0F0H
180 SEL(9)=04H : ENPH=0
190 INEN=0C6H : REM NILAI INTERRUPT ENABLE
200 INDI=0C2H : REM NILAI INTERRUPT DISABLE
210 CLOCK 1 : REM MENGAKTIFKAN CLOCK TIMER
212 IRU=0 : IE=INDI : ENABLE=0
214 ONEX1 10000 : REM INTERRUPT ROUTINE ADDRESS
216 IE=INEN : ENABLE=1
220 XBY(REL2)=0 : REM SWITCH RELAY 9 DAN 11 SERTA DISABLE DECODER
230 XBY(REL1)=0 : REM SEMUA RELAY DISWITCH KE VOICE LINE
240 REM DELAY UNTUK MEN-SWITCH RELAY
250 T=TIME
260 IF TIME<T+0.01 GOTO 260
400 REM *****
500 REM MAIN PROGRAM
505 REM *****
510 REM LOOP UNTUK MENDETEKSI OFF-HOOK
520 DO
530 INP_OH=XBY(INPADDR)
540 UNTIL INP_OH<>0FFH
550 REM CHECK TELPON YANG MENJAWAB SAMBUNGAN
560 GOSUB 900
570 IF FAILED<>1 THEN XBY(REL1)=0FFH-2***(PHNE-1)
575 IF FAILED=1 THEN GOTO 800
580 IF IRU<>1 THEN 590
585 IF TIME<PZ+3.0 THEN GOSUB 6000 ELSE IRU=0 : ENABLE=1
590 REM CHECK NOMOR YANG SEDANG DIPUTAR
600 GOSUB 1150
610 IF FAILED=1 THEN GOTO 800
620 PRINT " Nomor_nomor yang telah diputar adalah: "
650 FOR Z=1 TO MAXNUM : PRINT NUM(Z) : NEXT
660 ON NUM(1) GOSUB 2200,2500,3000,3300,3550,3600,3800,4000,4200,4400
800 GOTO 220

```

```

900  REM *****
1000 REM SUBROUTINE UNTUK MENENTUKAN TELPON YANG OFF-HOOK
1005 REM *****
1010 IF IRU=1 THEN ENABLE=0
1020 PHNE=0 : VWD=0 : FAILED=0 : COUNTER=1
1030 DO
1040 PHNE=PHNE+1
1050 INPDAT=XBY(INPADDR)
1060 IF (INPDAT=OFFH-COUNTER) THEN VWD=1
1070 COUNTER=COUNTER*2
1080 IF COUNTER>256 THEN FAILED=1 : VWD=1
1090 WHILE VWD<>1
1100 XBY(REL1)=OFFH
1110 PWM 1000,1000,100
1120 RETURN : REM PHNE BERISI NOMER PESAWAT YANG MELAKUKAN SAMBUNGAN

```

```

1150 REM *****
1200 REM SUBROUTINE NOMER-NOMER YANG DIPUTAR
1205 REM *****
1210 REM LOOP MAXIMUM MENUNGGU 20 DETIK
1220 FLANKO=0 : FAILED=0 : MAX_I=12
1230 COD=2**(PHNE-1)
1240 T=TIME
1250 DO
1260 INPO=XBY(INPADDR)
1270 IF (COD.AND.INPO)=COD THEN FLANKO=1 : T=T-20
1280 WHILE TIME<T+20.00
1290 IF FLANKO<>1 THEN FAILED=1 : GOTO 1660
1300 REM MENGISI LIST NUM(I) DENGAN NOMER-NOMER
1310 I=1 : FLANK1=0 : VWD1=0
1320 DO
1330 GOSUB 1800 : REM FETCH NOMER TELEPON
1340 IF FAILED=1 THEN VWD1=1 : GOTO 1650
1350 NUM(I)=NUMB
1360 ON NUM(1) GOTO 1370,1390,1410,1430,1450,1470,1490,1510,1530,1550
1370 REM SAMBUNGAN TELEPHONE EXTERNAL
1380 MAX_I=1 : GOTO 1560
1390 REM SAMBUNGAN TELEPHONE INTERNAL
1400 MAX_I=2 : GOTO 1560
1410 REM MEMILIH LIST
1420 MAX_I=2 : GOTO 1560
1430 REM MENGISI LIST
1440 MAX_I=12 : GOTO 1560
1450 REM FUNGSI YANG BELUM DIGUNAKAN
1460 GOTO 1560
1470 MAX_I=1
1480 GOTO 1560
1490 MAX_I=1
1500 GOTO 1560
1510 REM FUNGSI YANG BELUM DIGUNAKAN
1520 GOTO 1560
1530 REM FUNGSI YANG BELUM DIGUNAKAN

```

```

1540 GOTO 1560
1550 REM FUNGSI YANG BELUM DIGUNAKAN
1560 IF I>=MAX_I THEN VWD1=1 : GOTO 1640
1570 FLANK1=0 : REM MASIH ADA NOMER LAIN?
1580 T=TIME
1590 DO
1600 INP1=XBY(INPADDR)
1610 IF (COD.AND.INP1)=COD THEN FLANK1=1 : T=T-3
1620 WHILE TIME<T+3.00
1630 IF FLANK1<>1 THEN VWD1=1 : REM TIDAK ADA NOMER
1640 I=I+1
1650 WHILE VWD1=0
1655 MAXNUM=I-1
1660 RETURN
1800 REM MENENTUKAN NOMER
1810 FLANK2=0 : FAILED=0 : VWD_NR=0
1820 FLANK3=0 : NUMB=0
1830 DO
1840 FLANK2=0 : T=TIME
1850 DO
1860 INP2=XBY(INPADDR)
1870 IF (COD.AND.INP2)=0 THEN FLANK2=1 : T=T-0.07
1880 WHILE TIME<T+0.0700
1890 IF FLANK2<>1 THEN FAILED=1 : VWD_NR=1
1900 FLANK3=0
1910 T=TIME
1920 DO
1930 INP3=XBY(INPADDR)
1940 IF (COD.AND.INP3)=COD THEN FLANK3=1 : T=T-0.04
1950 WHILE TIME<T+0.0400
1960 IF FLANK3<>1 THEN VWD_NR=1
1970 NUMB=NUMB+1
1980 IF NUMB=10 THEN NUMB=0
1990 IF NUMB>9 THEN FAILED=1 : VWD_NR=1
2000 UNTIL VWD_NR=1
2010 RETURN

2100 REM *****
2200 REM SUBROUTINE SAMBUNGAN TELEPHONE EKSTERNAL
2205 REM *****
2210 ENABLE=0 : REM DISABLE INTERRUPT
2220 XBY(REL2)=SEL(PHNE) : REM SWITCH TELPON KE EKSTERNAL
2230 VWD2=0
2240 DO
2250 PWM 1000,2000,100 : REM NADA SIBUK PADA WAIT LINE
2260 REM LOOP UNTUK MENGETES AKHIR DARI SAMBUNGAN TELPON
2270 T=TIME
2280 DO
2290 INP4=XBY(INPADDR)
2300 IF (INP4.AND.(2**{PHNE-1}))<>0 THEN VWD2=1 : T=T-1.0
2310 WHILE TIME<T+1.0
2320 IF VWD2=0 THEN GOTO 2390

```



```

2330 REM APAKAH SAMBUNGAN BENAR-BENAR TELAH SELESAI?
2340 T=TIME
2350 DO
2360 INP5=XBY(INPADDR)
2370 IF (INP5.AND.(2**(PHNE-1)))=0 THEN VWD2=0 : T=T-0.1
2380 WHILE TIME<T+0.1
2390 WHILE VWD2=0
2400 GOSUB 5200
2410 IF DV=1 THEN DV=0 : GOTO 2220
2414 T=TIME
2416 DO
2418 WHILE TIME<T+2
2420 ENABLE=1 : REM ENABLE INTERRUPT
2430 RETURN

2450 REM *****
2500 REM SUBROUTINE UNTUK SAMBUNGAN TELEPHONE INTERNAL
2505 REM *****
2510 IF PHNE=NUM(2) THEN GOSUB 5000 : GOTO 2870
2520 IF NUM(2)>8 THEN GOSUB 5100 : GOTO 2870
2540 OPG_N=0 : OPG_L=0
2550 BEL_T=2**(PHNE-1)
2560 ANW_T=2**(NUM(2)-1)
2570 MSK_IN=OFFH.XOR.(BEL_T+ANW_T)
2580 XBY(REL1)=MSK_IN : REM BEL_T DAN ANW_T DINUBUNGAN KE VOICE LINE
2590 T=TIME : REM DELAY UNTUK MENSWITCH RELAY
2600 IF TIME<T+0.01 THEN GOTO 2600
2610 FOR A=1 TO 13 : REM MAX. 13 KALI DERINGAN
2620 T=TIME
2630 DO
2640 INP6=XBY(INPADDR)
2650 IF (INP6.AND.ANW_T)=0 THEN OPG_N=1 : A=14 : T=T-2.5
2670 IF INP6=OFFH THEN OPG_L=1 : A=14 : T=T-2.5
2680 WHILE TIME<T+2.5
2690 IF A=14 THEN GOTO 2770
2700 XBY(REL2)=05H : REM RELAY DERINGAN ON
2710 REM TUNGGU DELAY
2720 T=TIME
2730 IF TIME<T+1.5 THEN GOTO 2730
2740 XBY(REL2)=04H : REM RELAY DERINGAN OFF
2750 T=TIME : REM DELAY UNTUK MENSWITCH RELAY
2760 IF TIME<T+0.01 THEN GOTO 2760
2770 NEXT A
2780 REM TUNGGU SAMPAI SAMBUNGAN TELPON DIJAWAB
2790 DO
2800 PWM 2000,1000,100 : REM NADA SIBUK PADA WAIT LINE
2810 T=TIME
2820 DO
2830 INP7=XBY(INPADDR)
2840 IF ((INP7.AND.(OFFH-BEL_T-ANW_T))<>INP7) THEN OPG_L=1 : T=T-1.0
2850 WHILE TIME<T+1.0
2860 WHILE OPG_L<>1

```

```

2870 RETURN
3000 XBY(RRL2)=4
3005 ON NUM(2) GOTO 3010,3020,3030,3040,3050,3060,3070,3080,3090,3100
3010 FOR I=1 TO MAXNA : VAR(I)=TA(I) : NEXT
3015 MAX_BL=MAXNA-2 : GOTO 3110
3020 FOR I=1 TO MAXNB : VAR(I)=TB(I) : NEXT
3025 MAX_BL=MAXNB-2 : GOTO 3110
3030 FOR I=1 TO MAXNC : VAR(I)=TC(I) : NEXT
3035 MAX_BL=MAXNC-2 : GOTO 3110
3040 FOR I=1 TO MAXND : VAR(I)=TD(I) : NEXT
3045 MAX_BL=MAXND-2 : GOTO 3110
3050 FOR I=1 TO MAXNE : VAR(I)=TE(I) : NEXT
3055 MAX_BL=MAXNE-2 : GOTO 3110
3060 FOR I=1 TO MAXNF : VAR(I)=TF(I) : NEXT
3065 MAX_BL=MAXNF-2 : GOTO 3110
3070 FOR I=1 TO MAXNG : VAR(I)=TG(I) : NEXT
3075 MAX_BL=MAXNG-2 : GOTO 3110
3080 FOR I=1 TO MAXNH : VAR(I)=TH(I) : NEXT
3085 MAX_BL=MAXNH-2 : GOTO 3110
3090 FOR I=1 TO MAXNI : VAR(I)=TI(I) : NEXT
3095 MAX_BL=MAXNI-2 : GOTO 3110
3100 FOR I=1 TO MAXNJ : VAR(I)=TJ(I) : NEXT
3105 MAX_BL=MAXNJ-2
3110 XBY(RRL1)=OFFH
3120 VERB_A=4
3130 VERB_U=0
3134 T=TIME
3136 IF TIME<T+1.0 THEN GOTO 3136
3140 FOR I=1 TO MAX_BL
3145 IF VAR(I)=0 THEN VAR(I)=10
3150 FOR K=1 TO VAR(I)
3160 T=TIME
3170 XBY(RRL2)=VERB_U
3180 DO
3190 WHILE TIME<T+0.054
3195 T=TIME
3200 XBY(RRL2)=VERB_A
3210 DO
3220 WHILE TIME<T+0.021
3230 NEXT K
3240 T=TIME
3250 DO
3260 WHILE TIME<T+0.85
3270 NEXT I
3272 PWM 2000,1000,100
3274 XBY(RRL1)=OFFH-2** (PHNE-1)
3276 FOR P=0 TO 100 : NEXT
3278 GOSUB 2200
3280 RETURN
3300 REM FUNGSI NOMER 3
3310 ON NUM(2) GOTO 3320,3340,3360,3380,3400,3420,3440,3460,3480,3500
3320 MAXNA=MAXNUM : FOR I=2 TO MAXNA
3330 TA(I-2)=NUM(I) : NEXT : GOTO 3520
3340 MAXNB=MAXNUM : FOR I=2 TO MAXNB

```

```

3350 TB(I-2)=NUM(I) : NEXT : GOTO 3520
3360 MAXNC=MAXNUM : FOR I=2 TO MAXNC
3370 TC(I-2)=NUM(I) : NEXT : GOTO 3520
3380 MAXND=MAXNUM : FOR I=2 TO MAXND
3390 TD(I-2)=NUM(I) : NEXT : GOTO 3520
3400 MAXNE=MAXNUM : FOR I=2 TO MAXNE
3410 TE(I-2)=NUM(I) : NEXT : GOTO 3520
3420 MAXN5=MAXNUM : FOR I=2 TO MAXN5
3430 TF(I-2)=NUM(I) : NEXT : GOTO 3520
3440 MAXNG=MAXNUM : FOR I=2 TO MAXNG
3450 TG(I-2)=NUM(I) : NEXT : GOTO 3520
3460 MAXNH=MAXNUM : FOR I=2 TO MAXNH
3470 TH(I-2)=NUM(I) : NEXT : GOTO 3520
3480 MAXN8=MAXNUM : FOR I=2 TO MAXN8
3490 TI(I-2)=NUM(I) : NEXT : GOTO 3520
3500 MAXNJ=MAXNUM : FOR I=2 TO MAXNJ
3510 TJ(I-2)=NUM(I) : NEXT
3520 RETURN
3550 REM FUNGSI NOMER 4 BELUM DIGUNAKAN
3560 RETURN
3600 REM FUNGSI NOMER 5 UNTUK MENDISABLE TELEPHONE
3610 COD2=2**(PHNE-1)
3620 ENPH=ENPH.OR.COD2
3640 RETURN
3800 REM FUNGSI NOMER 6 UNTUK MENGENABLE KEMBALI TELEPHONE
3810 COD3=2**(PHNE-1)
3820 ENPH=ENPH.AND.(OFFH-COD3)
3840 RETURN
4000 REM FUNGSI NOMER 7 BELUM DIGUNAKAN
4010 RETURN
4200 REM FUNGSI NOMER 8 BELUM DIGUNAKAN
4210 RETURN
4400 REM FUNGSI NOMER 9 BELUM DIGUNAKAN
4410 RETURN

4500 REM *****
5000 REM SUBROUTINE UNTUK MELAKUKAN SAMBUNGAN TELPON INTERNAL
5005 REM *****
5010 XBY(REL1)=2**(PHNE-1) : REM SWITCH PENELPON DARI LUAR KE WAIT LINE
5020 PWM 1000,1000,50
5030 T=TIME
5040 IF TIME<T+0.5 THEN GOTO 5040
5050 PWM 500,500,200
5060 RETURN

5080 REM *****
5100 REM SUBROUTINE UNTUK MENDETEKSI NOMER INTERNAL YANG SALAH (>8)
5105 REM *****
5110 XBY(REL1)=2**(PHNE-1) : REM SWITCH PENELPON DARI LUAR KE WAIT LINE
5120 PWM 500,500,200

```

```

5130 T=TIME
5140 IF TIME<T+0.5 THEN GOTO 5140
5150 PWM 1000,1000,50
5160 RETURN

```

```

5180 REM *****
5200 REM SUBROUTINE UNTUK MENTRANSFER SAMBUNGAN DARI LUAR
5205 REM *****
5210 DV=0 : FLANK4=0 : FLANK5=0
5220 VWD4=0 : VWD5=0 : VWD6=0 : OPG_N=0
5224 XBY(REL2)=SEL(9).OR.SEL(PHNE)
5226 T=TIME
5228 IF TIME<T+0.2 THEN 5228
5230 XBY(REL2)=SEL(9) : REM TELEPHONE YANG KELIRU
5240 DO
5250 COD1=2**(PHNE-1)
5260 T=TIME : REM TUNGGU JIKA TELPON OFF-HOOK LAGI DALAM 5 DETIK
5270 DO
5280 INP10=XBY(INPADDR)
5290 IF ((INP10.AND.COD1)=0) THEN VWD4=1 : T=T-5.0
5300 WHILE TIME<T+5.0
5310 IF VWD4=0 THEN DV=0 : VWD5=1 : GOTO 5500
5320 XBY(REL1)=OFFH
5330 PWM 1000,2000,100 : REM NADA SELAMA TRANSFER DILAKUKAN
5340 XBY(REL1)=OFFH-COD1
5345 FOR Z=1 TO 100 : NEXT
5350 REM TUNGGU SAMPAI NOMER TELAH DIBERIKAN
5360 DO
5365 FLANK4=0
5370 INP11=XBY(INPADDR)
5380 IF ((COD1.AND.INP11)=COD1) THEN FLANK4=1
5390 WHILE FLANK4=0
5400 COD=COD1 : GOSUB 1800 : REM FETCH NOMER
5410 IF FAILED=1 THEN VWD4=0 : VWD5=1 : GOTO 5500
5415 IF NUMB=PHNE THEN OPG_N=0 : VWD5=1 : GOTO 5500
5420 PHNE2=NUMB : NUM(2)=NUMB : GOSUB 2500
5430 IF OPG_N=1 THEN VWD5=1
5500 WHILE VWD5<>1
5510 IF VWD4=0 THEN GOTO 5550
5520 IF OPG_N=1 THEN PHNE=PHNE2 : XBY(REL1)=OFFH-2**(PHNE-1)
5530 IF OPG_N<>1 THEN XBY(REL1)=OFFH-COD1
5540 DV=1
5550 RETURN

```

```

5900 REM *****
6000 REM SUBROUTINE PADA WAKTU SAMBUNGAN TELPON EKSTERNAL DITERIMA
6002 REM *****
6005 ENABLE=0
6010 VWD3=0 : IRU=0
6020 XBY(REL2)=SEL(PHNE)

```

```

6030 DO
6040 PWM 2500,500,100
6050 T=TIME
6060 DO
6070 INP9=XBY(INPADDR)
6080 IF (INP9.AND.(2*(PHNE-1)))<>0 THEN VWD3=1 : T=T-1.0
6090 WHILE TIME<T+1.0
6091 IF VWD3=0 THEN GOTO 6100
6092 REM APAKAH SAMBUNGAN BENAR-BENAR TELAH SELASAI?
6093 T=TIME
6094 DO
6095 INP9=XBY(INPADDR)
6096 IF (INP9.AND.(2*(PHNE-1)))=0 THEN VWD3=0 : T=T-0.1
6097 WHILE TIME<T+0.1
6100 WHILE VWD3=0
6110 GOSUB 5200
6120 IF DV=1 THEN DV=0 : GOTO 6010
6122 T=TIME : XBY(REL2)=0
6124 DO
6126 WHILE TIME<T+1
6130 ENABLE=1
6140 RETURN 800

```

```

9000 REM *****
10000 REM INTERRUPT SUBROUTINE
10002 REM *****
10005 IF ENABLE=0 THEN GOTO 10160
10010 IRU=1 : INP8=0 : T=0 : PZ=TIME
10020 INP8=XBY(INPADDR)
10030 IF INP8<>OFFH THEN GOTO 10110
10040 XBY(REL1)=ENPH
10050 XBY(REL2)=01H
10060 T=TIME : DO
10070 WHILE TIME<T+1.5
10080 XBY(REL2)=0H
10085 XBY(REL1)=0
10090 FOR P=1 TO 100 : NEXT P
10100 GOTO 10160
10110 XBY(REL1)=OFFH
10120 PWM 2000,500,100
10130 PWM 500,1000,500
10140 XBY(REL1)=INP8
10150 FOR P=1 TO 100 : NEXT
10160 RETI

```