

**TUGAS AKHIR**  
**EC7O30 – Metoda Formal**

**Dosen : Budi Rahardjo, MSc, PhD**

**Pemodelan Sistem Telepon  
menggunakan Promela/Spin**

**Oleh:**  
**Agus Heri S**  
**232 01 060**



**PROGRAM STUDI TEKNIK ELEKTRO**  
**BIDANG KHUSUS TEKNIK KOMPUTER**  
**PROGRAM PASCASARJANA**  
**INSTITUT TEKNOLOGI BANDUNG**  
**2004**

# Pemodelan Sistem Telepon menggunakan Promela/Spin

## Abstrak

Tugas ini membahas pemodelan sistem telepon dengan menggunakan bahasa Promela dan menggunakan perangkat lunak SPIN untuk memverifikasi desainnya. Spin menunjukkan bahwa desain telah benar dan telah dibuktikan dengan diverifikasi.

### 1. Pendahuluan : Spin

SPIN adalah suatu paket software yang dikembangkan di Laboratorium Bell group "*formal specification and verification*". SPIN dapat digunakan untuk men"*trace*" logical error di dalam desain *distributed systems*. Tool memeriksa konsistensi logical dari suatu spesifikasi. Sehingga dapat ditemukan *deadlock, unspecified receptions, flag incompleteness, race conditions, dan unwarranted assumptions* tentang kecepatan proses.

SPIN dapat menganalisa sebuah spesifikasi dengan tiga cara : simulasi, *exhaustive search* (pencarian yang menyeluruh), dan *bit space (partial order reduction) search* untuk spesifikasi yang besar.

Cara yang paling mudah untuk menggunakan SPIN adalah dengan simulasi. Simulasi akan berhenti ketika spesifikasi diakhiri/ *terminate* atau ketika terjadi fail/kegagalan. SPIN dapat melakukan suatu *strict analysis* dari spesifikasi dengan *exhaustively searching* pada semua state.

SPIN menghasilkan suatu custom (pan.c) program C, ketika dicompile dan di"run" akan melaksanakan *exhaustive search*. Ketika program gagal/fail untuk menemukan spesifikasinya, *custom analyzer* akan menghasilkan path yang memaksa sistem ke invalid state. Ini adalah kemampuan untuk membuktikan ketepatan dari spesifikasi yang dibuat oleh user.

## 2. Latar Belakang Masalah

Dalam suatu perancangan sistem dibutuhkan sebuah desain yang baik dan benar. Desain yang baik benar akan menghindarkan munculnya hal hal yang tidak diinginkan. Sistem telepon juga memerlukan desain yang baik dan benar. Sehingga diharapkan tidak terjadinya sebuah kesalahan, misalnya kita tidak dapat menerima telepon, padahal sudah terdengar dering. Atau kita tidak dapat melakukan panggilan keluar, sedangkan nada tone sudah terdengar. Dan masalah masalah yang lain yang dapat muncul karena kurang tepatnya desain.

## 3. Deskripsi Sistem

Sistem telepon ini adalah versi sederhana dari *fixed phone* yang dapat kita temukan di rumah. Pada *user level interface* telepon dapat digambarkan sebagai *finite state machine* dengan 9 state, 3 operasi user dan 5 hasil operasi user. Operasi user dan hasil operasi user seperti terlihat pada tabel 1. State ditampilkan langsung dalam sistem transisi seperti pada gambar 1. Initial state dari telepon ditunjukkan dengan lingkaran yang diberi label STANDBY. Pada status tersebut hook telepon dalam posisi diletakkan dan menunggu telepon masuk atau dipergunakan oleh user. User menandai sebuah telepon masuk dengan bunyi dering yang dikeluarkan telepon. Ketika telpon berdering ada dua kemungkinan, diangkat user dan mulai berbicara atau setelah sekian lama tidak ada yang menerima maka dering berhenti dan telepon kembali dalam mode standby. Dalam spesifikasi ini, lama dering telepon tidak diangkat tidak dimodelkan.

Ketika user selesai melakukan pembicaraan di telepon maka user akan meletakkan atau mengembalikan hook pada telepon. Dan setelah diletakkan maka telepon kembali pada mode standby.

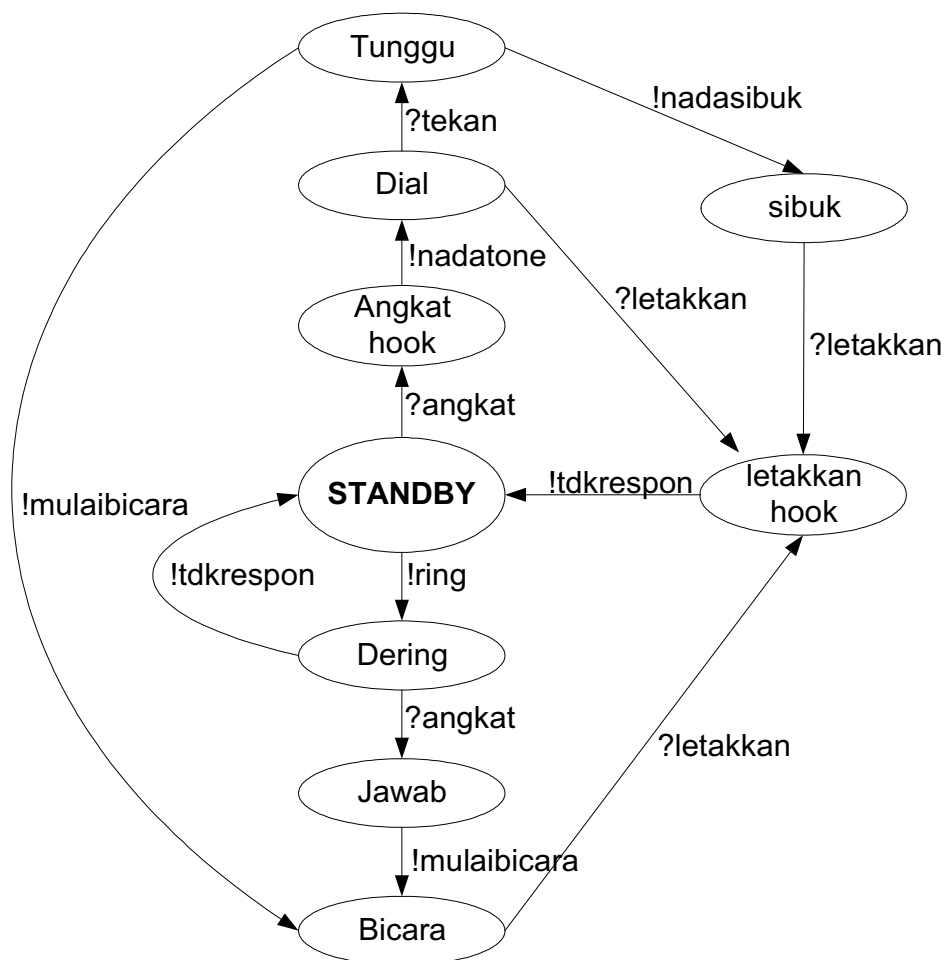
Ketika user ingin melakukan panggilan keluar, user perlu mengangkat hook dan kemudian akan mendengar nadatone. Setelah mendengar nadatone ada dua kemungkinan yang dapat terjadi. Yang umum user tinggal menekan nomer tujuan, kemudian menunggu koneksi tersambung dan setelah koneksi tersambung maka pembicaraan dapat dilakukan (mulaibicara). Kemungkinan yang lain user mendengar nada sibuk sehingga user perlu untuk meletakkan kembali hook.

Transisi sistem ditunjukkan pada gambar 1, yang juga sekaligus menunjukkan pemodelan state dan transisi. Transisi ditandai oleh operasi user dan *device respon*.

Keduanya ditandai dengan tanda seru (!) yang digenerate oleh telepon. Sedangkan yang ditandai dengan tanda tanya (?) adalah operasi yang dapat dilakukan oleh user.

Operasi User		Hasil Operasi	
angkat	Angkat hook	Nadatone	Terdengar nada tone
letakkan	Letakkan hook	mulaibicara	Mulai terjadi percakapan
tekan	Tekan nomer	nadasibuk	Terdengar nada sibuk
		tdkrespon	Tidak ada respon
		ring	berdering

**Table 1.** Operasi User dan Hasil Operasi



**Gambar 1.** Spesifikasi Interface dari sistem telepon

#### 4. Bahasa Promela

Promela (PROtocol MEta LAnguage) adalah bahasa pemodelan untuk verifikasi suatu desain yang digunakan sebagai input dalam SPIN verifikasi. Model Promela dapat memuat tiga tipe objek yang berbeda : processes, variables, dan message channels.

Semua proses adalah global objek. variables, and message channels dapat di deklarasikan pada lokal atau global di dalam sebuah proses. Tidak ada perbedaan antara statement dan kondisi dalam Promela. Eksekusi dari setiap statement adalah kondisional bedasar kepada executability-nya. Dan executability dari sebuah statement adalah cara utama dari sinkronisasi. Sebuah kondisi hanya dapat dieksekusi ketika kondisi adalah *true*. Jika kondisi tidak *true*, eksekusi diblok sampai kondisi menjadi *true*. Scope dari variabel adalah global jika didefinisikan diluar deklarasi proses dan lokal jika didefinisikan didalam deklarasi proses.

#### 5. Pemodelan Menggunakan Promela

Pada bagian ini akan dilakukan translate dari model ke dalam bentuk formal yang dapat digunakan sebagai input model checker. Model checker yang dipergunakan adalah SPIN karena selain open source juga memiliki grafik user interface.

Pertama akan diterjemahkan sistem transisi sistem telepon ke dalam format Promela, bahasa pemodelan dari SPIN. Sistem telepon dimodelkan sebagai SISPON dalam Promela. Dan juga dimodelkan proses yang lain yaitu, user yang interaktif dengan sistem telepon dengan menerima atau memicu dari beberapa kemungkinan interaksi dengan telepon.

Dibawah ini spesifikasi Promela dari sistem telepon dan operasi user. Proses SISPON merepresentasikan struktur dari sistem transisi pada gambar 1, dimana setiap state direpresentasikan sebagai sebuah label di dalam Promela, dan kemungkinan transisi pada struktur pilihan yang mengikuti label. Interaksi antara operasi user dan SISPON dimodelkan dengan *shared channel* yang diberi nama '*a*' for *synchrononous (rendez-vous) communication*.

Kedua proses berjalan didalam atomic yang paralel, men-generate semua kemungkinan state yang akan memberikan hasil dari *interleaving execution*. Atomic

statement memastikan kedua proses dimulai secara bersamaan. Berikut ini adalah listing program yang digunakan:

```
mtype = {angkat,letakkan,tidakrespon,ring,tekan,mulaibicara,nadatone,nadasibuk};
chan a = [0] of {mtype};

proctype SISPON (chan u)
{
STANDBY:    if
            :: u?angkat -> goto ANGKATHOOK
            :: u!ring -> goto DERING
            fi;
DERING:     if
            :: u!tidakrespon -> goto STANDBY
            :: u?angkat -> goto JAWAB
            fi;
JAWAB:     u!mulaibicara; goto BICARA;
BICARA:    u?letakkan -> goto LETAKKANHOOK;
LETAKKANHOOK: u!tidakrespon; goto STANDBY;
ANGKATHOOK: u!tekan; goto DIAL;
DIAL:     if
            :: u?tekan -> goto TUNGGU
            :: u!letakkan -> goto LETAKKANHOOK
            fi;
TUNGGU:    if
            :: u!nadasibuk -> goto SIBUK
            :: u!mulaibicara -> goto BICARA
            fi;
SIBUK:     u?letakkan; goto LETAKKANHOOK;
}
proctype user (chan u)
{
do
            :: u?ring
            :: u?tidakrespon
            :: u?mulaibicara
            :: u?nadatone
            :: u?nadasibuk
            :: u!angkat
            :: u!letakkan
            :: u!tekan
od
}
init
{atomic {run user (a); run SISPON (a)}}
```

## 6. Verifikasi Menggunakan SPIN

Pada tugas ini akan dilakukan dua macam verifikasi yaitu : verifikasi untuk mencapai keadaan bicara (dapat melakukan panggilan keluar) dan verifikasi untuk keadaan sewaktu ada dering (ring) telepon masuk.

### 6.1 verifikasi untuk mencapai keadaan bicara (dapat melakukan panggilan keluar).

Kita ingin melakukan verifikasi bahwa desain sistem telepon yang kita rancang dapat mencapai keadaan untuk bicara (dapat melakukan panggilan keluar). Sehingga perlu untuk menambahkan temporal Claim sebagai berikut :

```
#define p (SISPON[2]@BICARA)

never {
  T0_init:
    if
      :: ((p)) -> goto accept_all
      :: (1) -> goto T0_init
    fi;
  accept_all:
    skip
}
```

Dan verifikasi outputnya adalah sebagai berikut :

preparing trail, please wait...done

spin: couldn't find claim (ignored)

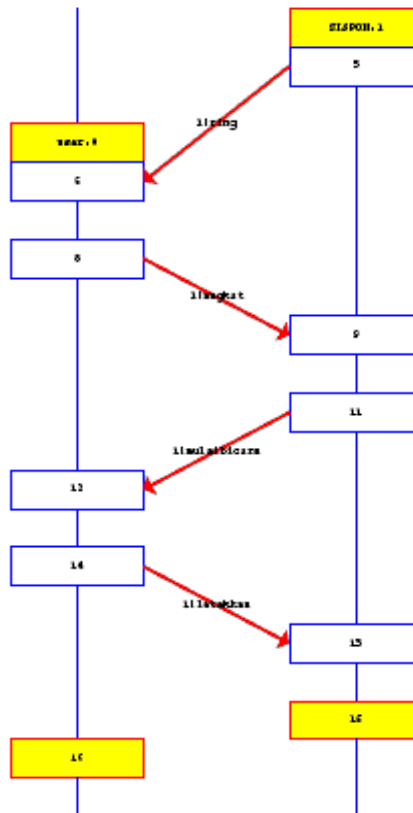
```
2:  proc 0 (:init:) line 42 "pan_in" (state 1)          [(run user(a))]
3:  proc 0 (:init:) line 42 "pan_in" (state 2)          [(run SISPON(a))]
5:  proc 1 (SISPON) line 8 "pan_in" (state -) [values: 1!ring]
5:  proc 1 (SISPON) line 8 "pan_in" (state 3) [u!ring]
6:  proc 0 (user) line 31 "pan_in" (state -)            [values: 1?ring]
6:  proc 0 (user) line 31 "pan_in" (state 1)           [u?ring]
8:  proc 0 (user) line 36 "pan_in" (state -)            [values: 1!angkat]
8:  proc 0 (user) line 36 "pan_in" (state 6)           [u!angkat]
9:  proc 1 (SISPON) line 12 "pan_in" (state -) [values: 1?angkat]
9:  proc 1 (SISPON) line 12 "pan_in" (state 9) [u?angkat]
11: proc 1 (SISPON) line 14 "pan_in" (state -) [values: 1!mulaibicara]
11: proc 1 (SISPON) line 14 "pan_in" (state 13) [u!mulaibicara]
12: proc 0 (user) line 33 "pan_in" (state -)           [values: 1?mulaibicara]
12: proc 0 (user) line 33 "pan_in" (state 3)           [u?mulaibicara]
14: proc 0 (user) line 37 "pan_in" (state -)           [values: 1!letakkan]
14: proc 0 (user) line 37 "pan_in" (state 7)           [u!letakkan]
```

```

15: proc 1 (SISPON) line 15 "pan_in" (state -) [values: 1?letakkan]
15: proc 1 (SISPON) line 15 "pan_in" (state 15) [u?letakkan]
spin: trail ends after 16 steps
#processes: 3
16: proc 1 (SISPON) line 16 "pan_in" (state 17)
16: proc 0 (user) line 30 "pan_in" (state 9)
16: proc 0 (:init:) line 42 "pan_in" (state 4)
3 processes created

```

Dengan hasil visualisasi Message Sequence Chartnya :



Gambar 2. MSC untuk keadaan mencapai bicara

Dari hasil verifikasi output dan visualisasi MSC dapat disimpulkan bahwa sistem dapat mencapai keadaan untuk melakukan panggilan keluar (bicara).

## 6.2 Verifikasi untuk untuk keadaan sewaktu ada dering telepon masuk.

Sewaktu ada dering telepon masuk, terdapat dua kemungkinan yaitu, begitu terdengar dering kemudian telepon diangkat dan selanjutnya melakukan pembicaraan

(bicara) atau begitu terdengar dering kemudian tidak ada yang menerima, sehingga status kembali standby. Perlu ditambahkan temporal claim sebagai berikut :

```
#define p (SISPON[2]@BICARA)
#define q (SISPON[2]@RING)

never {
T0_init:
    if
        :: ((p) && (q)) -> goto accept_all
        :: ((q)) -> goto T0_S4
        :: (1) -> goto T0_init
    fi;
T0_S4:
    if
        :: ((p)) -> goto accept_all
        :: (1) -> goto T0_S4
    fi;
accept_all:
    skip
}

```

Dengan hasil verifikasi outputnya :

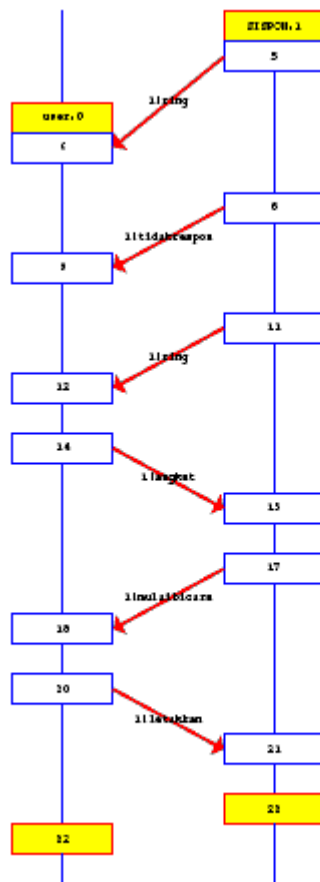
```
preparing trail, please wait...done
spin: couldn't find claim (ignored)
2:  proc 0 (:init:) line 42 "pan_in" (state 1)          [(run user(a))]
3:  proc 0 (:init:) line 42 "pan_in" (state 2)          [(run SISPON(a))]
5:  proc 1 (SISPON) line 8 "pan_in" (state -) [values: 1!ring]
5:  proc 1 (SISPON) line 8 "pan_in" (state 3) [u!ring]
6:  proc 0 (user) line 31 "pan_in" (state -)            [values: 1?ring]
6:  proc 0 (user) line 31 "pan_in" (state 1)            [u?ring]
8:  proc 1 (SISPON) line 11 "pan_in" (state -) [values: 1!tidakrespon]
8:  proc 1 (SISPON) line 11 "pan_in" (state 7) [u!tidakrespon]
9:  proc 0 (user) line 32 "pan_in" (state -)            [values: 1?tidakrespon]
9:  proc 0 (user) line 32 "pan_in" (state 2)            [u?tidakrespon]
11: proc 1 (SISPON) line 8 "pan_in" (state -) [values: 1!ring]
11: proc 1 (SISPON) line 8 "pan_in" (state 3) [u!ring]
12: proc 0 (user) line 31 "pan_in" (state -)            [values: 1?ring]
12: proc 0 (user) line 31 "pan_in" (state 1)            [u?ring]
14: proc 0 (user) line 36 "pan_in" (state -)            [values: 1!angkat]
14: proc 0 (user) line 36 "pan_in" (state 6)            [u!angkat]
15: proc 1 (SISPON) line 12 "pan_in" (state -) [values: 1?angkat]
15: proc 1 (SISPON) line 12 "pan_in" (state 9) [u?angkat]
17: proc 1 (SISPON) line 14 "pan_in" (state -) [values: 1!mulaibicara]
17: proc 1 (SISPON) line 14 "pan_in" (state 13) [u!mulaibicara]
18: proc 0 (user) line 33 "pan_in" (state -)            [values: 1?mulaibicara]
```

```

18:  proc 0 (user) line 33 "pan_in" (state 3           [u?mulaibicara]
20:  proc 0 (user) line 37 "pan_in" (state -)         [values: 1!letakkan]
20:  proc 0 (user) line 37 "pan_in" (state 7)         [u!letakkan]
21:  proc 1 (SISPON) line 15 "pan_in" (state -) [values: 1?letakkan]
21:  proc 1 (SISPON) line 15 "pan_in" (state 15)     [u?letakkan]
spin: trail ends after 22 steps
#processes: 3
22:  proc 1 (SISPON) line 16 "pan_in" (state 17)
22:  proc 0 (user) line 30 "pan_in" (state 9)
22:  proc 0 (:init:) line 42 "pan_in" (state 4)
3 processes created

```

Dengan visualisasi Message Sequence Chartnya :



Gambar 3. MSC keadaan sewaktu ada dering telepon masuk.

Dapat dilihat dari verifikasi output dan pada Visualisasi MSC bahwa sewaktu ada dering (ring) telepon masuk dan tidak ada yang merespon maka status kembali standby. Dari stand by bisa dering (ring) kembali dan kemudian diterima (angkat) kemudian melakukan pembicaraan (bicara) dan kemudian diakhiri (letakkan) .

## 7. Kesimpulan

Telah digambarkan penggunaan Promela untuk model sistem telepon dan menggunakan perangkat lunak SPIN untuk meverifikasinya. Dan dapat ditunjukkan pula bahwa sistem berjalan dengan baik. Hal ini dapat dibuktikan dengan melakukan verifikasi, yaitu verifikasi untuk mencapai keadaan bicara (dapat melakukan panggilan keluar) dan Verifikasi untuk untuk keadaan sewaktu ada dering telepon masuk.

## 8. Saran

Dengan semakin kompleks dan besarnya sistem yang ada, menuntut pengujian model yang lebih teliti dan lebih cermat. Untuk itu kita harus melakukan desain suatu sistem dengan baik dan benar. Tahapan yang dilakukan dalam proses desain adalah membuat spesifikasi sistem secara formal, implementasi, validasi, dan verifikasi.

## 9. Acknowledgements

Terima kasih kepada Bapak Budi yang telah banyak membimbing penulis dalam mata kuliah Metode Formal. Dan juga kepada Eric (Mahasiswa The University of Electro-Communications Tokyo Japan) yang telah membantu penulis, dalam bentuk diskusi online mengenai Promela dan SPIN.

## Daftar Pustaka

1. Budi Rahardjo, Paper Metoda Formal, Institut Teknologi Bandung, [br@paume.itb.ac.id](mailto:br@paume.itb.ac.id) 2002
2. Gang Liu, *CIS 841 Verification and Validation*, informasi online terdapat <http://www.cis.ksu.edu/~hankley/d841/Fa99/chap4a.html>
3. M. Masak and D. Latella, *Deriving Manual from formal specifications*, Institut C.N.R.- ISTI Via Moruzzi.

4. SPIN Verifier's Roadmap: *Building and Verifying SPIN Models*, 1997, informasi online terdapat <http://cm.bell-labs.com/cm/cs/what/spin/Man/Roadmap.html>