

# Simulasi Pemodelan Procedure-Consumer Problem pada Sistem Operasi

Maradu Sihombing<sup>1</sup>, Jaidup Banjarnahor<sup>2</sup>

<sup>1,2</sup>AMIK Medan Business Polytechnic

Jl. Jamin Ginting No. 285-287, Padang Bulan, Medan Baru, Kota Medan, Sumatera Utara, Indonesia - 20155

<sup>1</sup>maradu@amikmbp.ac.id, <sup>2</sup>jaidup@amikmbp.ac.id

DOI: xx.xxxx/j.ccs.xxxx.xx.xxx

---

## Abstrak

Sistem operasi yang menggunakan lebih dari satu proses untuk dapat bekerja bersama mencapai tujuan yang diinginkan. Agar tujuan tercapai secara benar, proses-proses tersebut harus mensinkronkan kegiatan-kegiatannya sehingga terkendali dengan baik untuk menghindari kondisi deadlock, salah satu metode untuk menyelesaikan masalah Deadlock adalah Procedure-Consumer. Procedure-Consumer menggunakan Perangkat lunak metode sleep and wake-up untuk mencegah masalah yang terjadi ketika buffer penuh, sementara producer ingin meletakkan item ke buffer dan consumer ingin mengambil item sementara buffer telah kosong. Perangkat lunak menggunakan semaphore untuk untuk mem-blocked producer atau consumer lain ketika salah satu producer atau consumer sedang berada dalam buffer. Perangkat lunak simulasi Producer-Consumer ini merupakan ilustrasi dari proses sinkronisasi, yaitu bagaimana cara mengatur beberapa proses yang mengakses beberapa variabel secara bersamaan.

**Kata Kunci:** Sistem Operasi, Deadlock, ProcEDURE-Consumer, Buffer.

---

## 1. Pendahuluan

Dalam sistem operasi, lebih dari satu proses dapat bekerja bersama untuk mencapai tujuan yang diinginkan. Agar tujuan tercapai secara benar, proses-proses tersebut harus mensinkronkan kegiatan-kegiatannya sehingga terkendali dengan baik untuk menghindari kondisi deadlock.

Kasus producer-consumer digunakan sebagai ilustrasi pembahasan sinkronisasi. Kasus producer-consumer berisi masalah mutual-exclusion dan sinkronisasi. Kasus ini sering juga disebut sebagai bounded-buffer problem (masalah buffer dengan jumlah terbatas). Kasus ini dapat diilustrasikan sebagai berikut, terdapat produsen (ilustrasi dari proses yang menyimpan informasi ke buffer) menghasilkan barang (ilustrasi dari informasi) dan konsumen (ilustrasi dari proses yang mengambil informasi dari buffer) yang akan menggunakan barang yang dihasilkan produsen. Keduanya mempunyai market (ilustrasi dari buffer) bersama dan berukuran tetap. Karena ukuran market terbatas, petaka (bencana) dapat terjadi untuk producer dan consumer. Masalah bagi producer terjadi ketika market telah penuh, sementara producer ingin meletakkan barang ke market yang telah penuh itu. Sedangkan masalah bagi consumer terjadi ketika consumer ingin

mengambil barang sementara market telah/sedang kosong.

Berdasarkan uraian di atas, peneliti akan merancang suatu perangkat lunak yang mampu untuk mensimulasikan proses producer-consumer dengan menerapkan metode sleep and wake-up untuk mencegah deadlock dan semaphore dan menampilkannya dalam bentuk animasi agar mudah dipahami pengguna terutama pada perkuliahan dengan mata kuliah sistem operasi.

## 2. Landasan Teori

Untuk dapat memahami penelitian ini lebih dalam, peneliti memaparkan teori-teori yang digunakan.

### 2.1. Sistem Operasi

Sistem operasi merupakan sebuah penghubung antara pengguna dari komputer dengan perangkat keras komputer. Sebelum ada sistem operasi, orang hanya dapat mengoperasikan komputer dengan menggunakan sinyal analog dan sinyal digital. Seiring dengan berkembangnya pengetahuan dan teknologi, muncullah sistem operasi yang menyediakan lingkungan untuk mengoperasikan komputer secara lebih nyaman. Untuk lebih memahami sistem operasi maka sebaiknya perlu diketahui terlebih dahulu

beberapa konsep dasar mengenai sistem operasi itu sendiri.

Pengertian sistem operasi secara umum ialah pengelola seluruh sumber daya yang terdapat pada sistem komputer dan menyediakan sekumpulan layanan (system calls) ke pemakai sehingga memudahkan dan menyamankan penggunaan serta pemanfaatan sumber-daya sistem komputer.

Sistem operasi mempunyai tiga sasaran utama yaitu:

1. Kenyamanan, membuat penggunaan komputer menjadi lebih nyaman.
2. Efisiensi, penggunaan sumber daya sistem komputer secara efisien.
3. Evolusi, sistem operasi harus dibangun sehingga memungkinkan dan memudahkan pengembangan, pengujian serta pengajuan sistem-sistem yang baru.

## 2.2. Komponen Sistem Operasi

Menurut Avi Silberschatz, Peter Galvin, dan Greg Gagne, pada umumnya sebuah sistem operasi modern mempunyai komponen sebagai berikut:

### 1. Manajemen Proses.

Proses adalah program yang sedang di eksekusi. Suatu proses membutuhkan satu atau beberapa sumber daya untuk menyelesaikan tugasnya. Sumber daya tersebut dapat berupa CPU time, memori, berkas-berkas, atau perangkat-perangkat I/O. Sistem operasi bertanggung jawab atas aktivitas-aktivitas yang berkaitan dengan manajemen proses seperti:

- a. Pembuatan dan penghapusan proses pengguna dan sistem proses.
- b. Menunda atau melanjutkan proses.
- c. Menyediakan mekanisme untuk proses sinkronisasi.
- d. Menyediakan mekanisme untuk proses komunikasi.
- e. Menyediakan mekanisme untuk penanganan deadlock.

### 2. Manajemen Memori Utama.

Memori utama atau lebih dikenal sebagai memori adalah suatu array besar yang terdiri dari word atau byte, yang ukurannya dapat mencapai ratusan, ribuan, atau bahkan jutaan. Setiap word atau byte mempunyai alamat tersendiri. Memori utama berfungsi sebagai tempat penyimpanan yang akses datanya digunakan oleh CPU atau perangkat I/O. Memori utama merupakan tempat penyimpanan data yang sementara (volatile), artinya data dapat hilang begitu sistem dimatikan. Sistem operasi bertanggung jawab atas aktivitas-aktivitas yang berkaitan dengan manajemen memori seperti:

- a. Menjaga track dari memori yang sedang digunakan dan siapa yang menggunakannya.
- b. Memilih program yang akan di-load ke memori.
- c. Mengalokasikan dan meng-dealokasikan ruang memori sesuai kebutuhan.

### 3. Manajemen Secondary Storage.

Data yang disimpan dalam memori utama bersifat sementara dan ukurannya sangat kecil jika dibandingkan dengan keseluruhan data yang terdapat dalam komputer. Oleh karena itu, untuk menyimpan keseluruhan data dan program komputer dibutuhkan secondary storage yang bersifat non volatil dan mampu menampung banyak data. Contoh dari secondary storage adalah harddisk, disket, dan lain-lain. Sistem operasi bertanggung jawab atas aktivitas-aktivitas yang berkaitan dengan disk-management seperti free-space management, alokasi penyimpanan, penjadualan disk.

### 4. Manajemen Sistem I/O.

Fungsi ini sering disebut device manager, dimana sistem operasi menyediakan "device driver" yang umum sehingga operasi I/O dapat seragam (membaca atau menuliskan data tanpa mempedulikan mekanisme kerja yang berbeda dari perangkat-perangkat I/O yang ada). Contoh: pengguna menggunakan operasi yang sama untuk membaca berkas pada hard-disk, CD-ROM dan floppy disk. Komponen untuk sistem I/O, yaitu:

- a. Buffer: penampungan sementara data dari/ ke perangkat I/O.
- b. Spooling: melakukan penjadualan pemakaian I/O sistem supaya pemakaian I/O lebih efisien (antrian dan sebagainya).
- c. Driver: penerjemah instruksi antara sistem operasi dan I/O untuk dapat melakukan operasi tertentu. Setiap perangkat keras I/O memiliki driver yang berbeda-beda.

### 5. Manajemen File.

File adalah kumpulan informasi yang dibuat dengan tujuan tertentu. File disimpan dalam struktur yang bersifat hirarkis, seperti direktori. Dalam manajemen file, sistem operasi bertanggung jawab dalam melakukan:

- a. Pembuatan dan penghapusan berkas.
- b. Pembuatan dan penghapusan direktori.
- c. Pemanipulasian berkas dan direktori.
- d. Pemetaan berkas ke secondary storage.
- e. Backup berkas ke media penyimpanan yang permanen (non-volatile).

### 6. Sistem Proteksi.

Proteksi mengacu pada mekanisme untuk mengontrol akses yang dilakukan oleh program,

- prosesor, atau pengguna ke sistem sumber daya. Mekanisme proteksi harus:
- membedakan antara penggunaan yang sudah diberi izin dan yang belum.
  - menetapkan pembatasan atau pengaturan yang telah ditentukan (specify the controls to be imposed).
  - menyediakan tata cara pelaksanaan (provide a means of enforcement).
7. Sistem Jaringan.  
Sistem ini untuk mendukung penggunaan jaringan. Sistem ini umumnya kini telah terpadu dalam sistem operasi karena kebutuhan kinerjanya serta kebutuhan komputasi telah menghendaki kemampuan ini terdapat di dalam sistem komputer. Sistem ini menyediakan akses pengguna ke bermacam sumber-daya sistem. Sistem ini membawa keuntungan dalam hal:
- Kecepatan komputer yang meningkat (computation speed-up).
  - Ketersediaan data meningkat (increased data availability).
  - Realibilitas dapat ditingkatkan (enhanced reliability).
8. User Interface (Shell).  
Sistem Operasi menunggu instruksi dari pengguna (command driven). Program yang membaca instruksi dan mengartikan control statements disebut control-card interpreter atau command-line interpreter. User Interface sangat bervariasi dari satu sistem operasi ke sistem operasi yang lain dan disesuaikan dengan tujuan dan teknologi I/O devices yang ada. Contohnya: Command Line Interface (CLI), Graphical User Interface (GUI), dan lain-lain.

### 2.3. Producer-Consumer

Kasus producer-consumer digunakan sebagai ilustrasi pembahasan sinkronisasi. Masalah producer-consumer disebut juga bounded-buffer problem (masalah buffer dengan jumlah terbatas).

Asumsi dalam producer-consumer problem adalah sebagai berikut:

- Dua proses menggunakan suatu buffer yang dipakai bersama dan berukuran tetap.
- Satu proses adalah producer yang meletakkan informasi ke buffer.
- Proses lain adalah consumer yang mengambil informasi dari buffer.

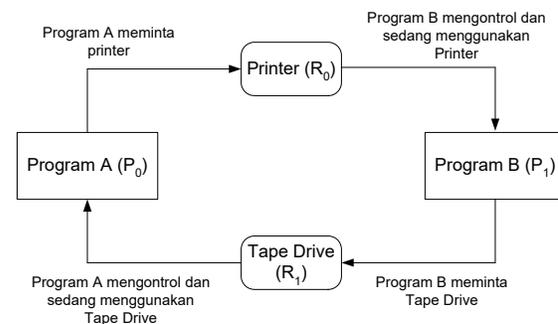
Masalah producer-consumer dapat dikembangkan menjadi masalah yang memiliki  $m$  buah produsen dan  $n$  buah konsumen. Karena buffer terbatas, masalah berikut dapat terjadi, yaitu:

- Masalah untuk producer.  
Masalah terjadi ketika buffer telah penuh, sementara producer ingin meletakkan informasi ke buffer yang telah penuh itu.
- Masalah untuk consumer.  
Masalah terjadi ketika consumer ingin mengambil informasi sementara buffer telah/sedang kosong.

Kedua proses perlu sinkronisasi agar keduanya dapat menghindari masalah.

### 2.4. Deadlock

Definisi deadlock dapat diperhatikan pada ilustrasi berikut. Misalkan pada suatu komputer terdapat dua buah program, sebuah tape drive dan sebuah printer. Program A mengontrol tape drive, sementara program B mengontrol printer. Setelah beberapa saat, program A meminta printer, tapi printer masih digunakan. Berikutnya, B meminta tape drive, sedangkan A masih mengontrol tape drive. Dua program tersebut memegang kontrol terhadap sumber daya yang dibutuhkan oleh program yang lain. Tidak ada yang dapat melanjutkan proses masing-masing sampai program yang lain memberikan sumber dayanya, tetapi tidak ada yang mengalah. Kondisi inilah yang disebut Deadlock. Gambar graph deadlock dapat dilihat pada Gbr. 1. berikut.



Gbr. 1. Graph Deadlock

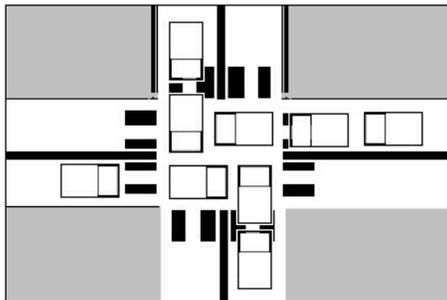
Deadlock mungkin dapat terjadi pada suatu proses disebabkan proses itu menunggu suatu kejadian tertentu yang tidak akan pernah terjadi. Dua atau lebih proses dikatakan berada dalam kondisi deadlock, bila setiap proses yang ada menunggu suatu kejadian yang hanya dapat dilakukan oleh proses lain dalam himpunan tersebut.

Terdapat kaitan antara overhead dari mekanisme koreksi dan manfaat dari koreksi deadlock itu sendiri. Pada beberapa kasus, overhead atau ongkos yang harus dibayar untuk membuat sistem bebas deadlock menjadi hal yang terlalu mahal dibandingkan jika mengabaikannya. Sementara pada kasus lain, seperti pada real-time process control, mengizinkan deadlock

akan membuat sistem menjadi kacau dan membuat sistem tersebut tidak berguna.

Contoh berikut ini terjadi pada sebuah persimpangan jalan. Beberapa hal yang dapat membuat deadlock pada suatu persimpangan, yaitu:

1. Terdapat satu jalur pada jalan.
2. Mobil digambarkan sebagai proses yang sedang menuju sumber daya.
3. Untuk mengatasinya beberapa mobil harus preempt (mundur).
4. Sangat memungkinkan untuk terjadinya starvation (kondisi proses tak akan mendapatkan sumber daya).



Gbr. 2. Ilustrasi *Deadlock*

### 3. Metodologi Penelitian

Adapun lokasi penelitian ini berada pada sistem operasi komputer dan dilakukan pada laboratorium komputer.

#### 3.1. Pengumpulan Data

Adapun metode pengumpulan data yang digunakan dalam penelitian ini adalah sebagai berikut:

1. Studi lapangan  
Dengan metode ini peneliti mengamati serta mempelajari bagaimana cara kerja dari sistem operasi khususnya pada hal-hal yang mengakibatkan deadlock terjadi.
2. Studi Kepustakaan  
Dengan melakukan studi pustaka, peneliti mendapatkan data-data yang bersifat teori ilmiah yang dipergunakan sebagai dasar dalam melakukan penulisan dan analisa terhadap kendala-kendala yang ada sehingga kendala tersebut dapat diselesaikan dengan baik.

#### 3.2. Langkah dalam pembuatan perangkat lunak

Langkah-langkah pembuatan perangkat lunak ini antara lain:

1. Membaca dan mempelajari buku-buku yang berhubungan dengan Sistem Operasi.
2. Mempelajari Producer-Consumer Problem.
3. Mempelajari teknik-teknik dasar pemrograman.
4. Merancang interface untuk perangkat lunak simulasi.
5. Merancang suatu perangkat lunak simulasi Producer-Consumer Problem.
6. Menguji perangkat lunak dan memperbaiki kesalahan (error) yang muncul.

### 4. Hasil dan Pembahasan

Adapun hasil dan pembahasan dalam penelitian ini akan dibahas disini.

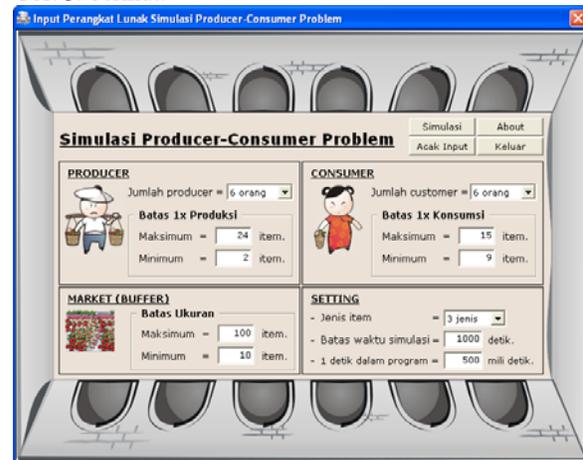
#### 4.1. Hasil

Adapun perangkat lunak (software) yang digunakan untuk menjalankan aplikasi ini adalah lingkungan sistem operasi Microsoft Windows 98, 98 Second Edition atau Microsoft Windows NT / 2000 / XP.

Untuk menguji hasil proses eksekusi perangkat lunak, dimasukkan input sebagai berikut:

1. Jumlah producer = 6 orang
2. Batas satu kali produksi, maksimum= 24 item dan minimum = 2 item.
3. Jumlah consumer = 6 orang.
4. Batas maksimum satu kali konsumsi = 15 item dan minimum = 9 item.
5. Batas ukuran maksimum market = 100 item.
6. Batas ukuran minimum market = 10 item.
7. Banyak jenis item = 3 jenis.
8. Batas waktu simulasi = 1000 detik.

Pada perangkat lunak, form Input terlihat seperti pada Gbr. 3. berikut.



Gbr. 3. Tampilan form Input (Penguujian Program)

Tampilan awal form Simulasi dapat dilihat pada Gbr. 4. berikut.



Gbr. 4. Tampilan form Simulasi sebelum proses simulasi dimulai

Pada saat  $t = 202$  detik, producer berada dalam keadaan tertidur (sleep). Tampilan prosesnya dapat dilihat pada Gbr. 5. berikut.



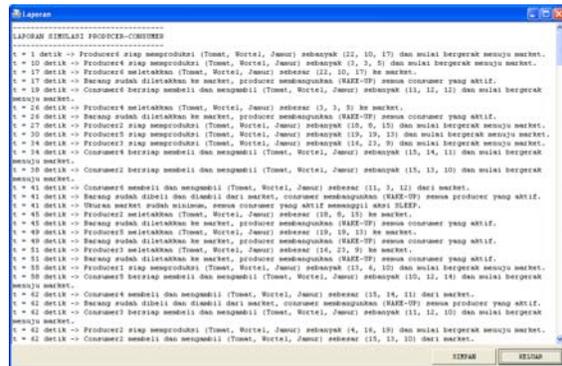
Gbr. 5. Tampilan simulasi ketika producer berada dalam keadaan tertidur (sleep)

Pada saat  $t = 452$  detik, consumer berada dalam keadaan tertidur (sleep). Tampilan prosesnya dapat dilihat pada Gbr. 6. berikut.



Gbr. 6. Tampilan simulasi ketika consumer berada dalam keadaan tertidur (sleep)

Tampilan form History dapat dilihat pada Gbr. 7. berikut.



Gbr. 7. Tampilan form History

Tampilan form Tabel Simulasi dapat dilihat pada Gbr. 8. berikut.

Produker	Waktu Mulai Prod/Kon	Waktu Selesai Prod/Kon	Waktu Tida & Makan	Alas	MARKET (Tomat, Wortel, Jambu, Kentang, D...
Produker1	1	13	20	+11, 7, 10, 9	(39, 21, 15, 20, 10)
Produker2	1	47	61	+0, 5, 15, 10, 9	(34, 23, 31, 20, 24)
Produker3	1	27	27	+0, 7, 15, 15	(42, 27, 36, 41, 40)
Produker1	1	5	23	+0, 12, 6, 10, 9	(5, 12, 6, 10, 9)
Produker2	1	45	45	+0, 15, 11, 11, 7	(47, 41, 42, 39, 21)
Produker3	1	25	25	+0, 11, 11, 12, 9	(34, 30, 27, 28, 27)
Produker2	33	40	55	+0, 15, 15, 13, 10	(46, 31, 40, 46, 35)
Produker3	40	55	55	+0, 8, 10, 10, 10	(46, 31, 34, 41, 29)
Produker1	49	70	85	+0, 10, 6, 13, 15	(37, 41, 34, 31, 44)
Produker2	32	48	55	+0, 13, 6, 15, 14	(35, 37, 28, 44, 30)
Consumer1	49				
Produker1	76				
Produker2	78				
Consumer2	87				

Gbr. 8. Tampilan form Tabel Simulasi

#### 4.2. Pembahasan

Dalam sub bab ini, akan dibahas mengenai alur kerja perangkat lunak simulasi, penggambaran objek simulasi dan proses simulasi Producer-Consumer Problem, sebagai berikut ini.

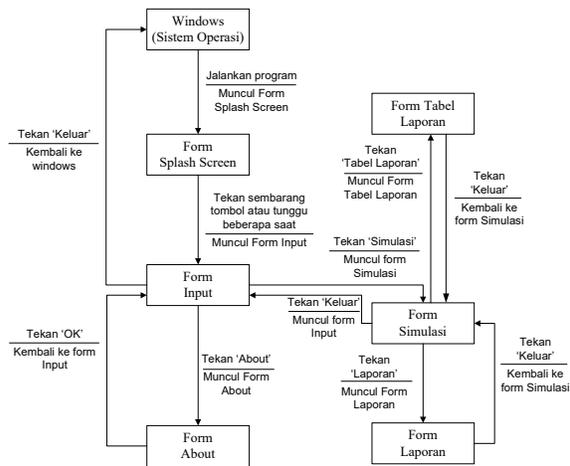
##### 1. Alur Kerja Perangkat Lunak Simulasi

Perangkat lunak simulasi Producer-Consumer Problem dimulai dengan tampilan splash screen. Form splash screen berisi nama / judul perangkat lunak dan nama pembuat perangkat lunak. Beberapa saat kemudian, form input akan tampil. Form input berfungsi untuk mengatur kondisi simulasi. Komponen yang dapat diatur adalah sebagai berikut:

- Pengaturan pada Producer, yaitu jumlah producer, batas maksimum dan minimum bagi producer dalam satu kali produksi.
- Pengaturan pada Consumer, yaitu jumlah consumer, batas maksimum dan minimum bagi consumer dalam satu kali konsumsi.
- Pengaturan pada Market, yaitu batas ukuran maksimum dan minimum market.
- Pengaturan lain, yaitu jenis item dan kecepatan proses simulasi.

Setelah pengaturan pada form input, proses simulasi dapat dimulai dengan menekan tombol 'Simulasi'. Selanjutnya, form simulasi akan muncul. Ketika proses simulasi sedang berjalan, user dapat menghentikan untuk sementara proses simulasi dengan menekan tombol 'Hentikan'. User juga dapat melihat laporan proses yang terjadi dalam simulasi dengan menekan tombol 'Laporan'. Laporan juga disediakan dalam bentuk tabel.

Alur kerja perangkat lunak simulasi ini dapat digambarkan dalam bentuk State Transition Diagram (STD), seperti terlihat pada gambar 9 berikut.



Gbr. 9. State Transition Diagram (STD)

##### 2. Penggambaran Objek Simulasi

Objek simulasi atau variabel yang terdapat dalam simulasi Producer-Consumer Problem diilustrasikan dalam bentuk objek gambar. Variabel dalam simulasi ini adalah producer, consumer dan market. Penggambaran variabel dilakukan dengan menggunakan aplikasi photo editor Adobe Photoshop CS.

##### 3. Proses Simulasi Producer-Consumer Problem

Proses simulasi Producer-Consumer Problem adalah sebagai berikut:

- Producer aktif memproduksi dan meletakkan item ke market (buffer). Aksi ini akan menambah jumlah item di dalam market.
- Consumer aktif mengambil item dari market (buffer) dan mengonsumsi item. Aksi ini akan mengurangi jumlah item di dalam market.
- Apabila market telah penuh atau jumlah item di dalam market telah mencapai batas maksimum, maka producer akan tidur (memanggil aksi sleep).
- Apabila consumer mengambil item dari market dan producer dalam keadaan sleep, maka consumer akan membangunkan (wake-up) producer.
- Apabila market kosong atau jumlah item di dalam market telah mencapai batas minimum, maka consumer akan tidur (memanggil aksi sleep).
- Apabila producer meletakkan item ke market dan consumer dalam keadaan sleep, maka producer akan membangunkan (wake-up) consumer.

Keenam poin di atas merupakan inti dari simulasi Producer-Consumer Problem. Pencegahan kondisi deadlock (producer ingin meletakkan item ke market sedangkan market telah penuh atau consumer ingin mengambil item dari market

sedangkan market telah kosong) dihindari dengan metode sleep dan wake up. Masing-masing variabel akan memanggil aksi sleep untuk menghindari kondisi deadlock dan akan dibangunkan variabel lainnya, ketika keadaan sudah tidak menyebabkan deadlock.

Critical section dalam perangkat lunak simulasi ini adalah ketika producer ingin meletakkan item ke dalam market dan consumer ingin mengambil item dari market. Dalam hal ini, semaphore digunakan untuk mengatur producer dan consumer supaya tidak meletakkan atau mengambil item dari market atau mengakses critical section secara bersamaan. Ketika salah satu producer atau consumer meletakkan atau mengambil item, maka variabel lain di-blocked agar tidak memasuki critical section, sehingga jumlah item di dalam market tidak diakses secara bersamaan, nilainya tetap konsisten dan terjaga kebenarannya.

Proses simulasi yang dirancang dalam perangkat lunak menggunakan komponen timer yang terdapat dalam bahasa pemrograman Microsoft Visual Basic 6.0. Fungsi dari komponen timer adalah untuk memeriksa keadaan suatu objek setiap satu interval waktu yang diatur kepadanya. Timer akan memeriksa dan memajukan keadaan objek ke keadaan berikutnya, sehingga terbentuk proses animasi. Misalkan, keadaan sekarang adalah producer berjalan ke bawah dengan kaki kiri, maka pada keadaan berikutnya, timer akan mengubah gambar tersebut dengan gambar producer berjalan ke bawah dengan kaki kanan. Demikian seterusnya.

## 5. Kesimpulan dan Saran

Setelah menyelesaikan perangkat lunak simulasi Producer-Consumer Problem, penulis menarik kesimpulan sebagai berikut:

1. Perangkat lunak menggunakan metode sleep and wake-up untuk mencegah masalah yang terjadi ketika buffer penuh, sementara producer ingin meletakkan item ke buffer dan consumer ingin mengambil item sementara buffer telah kosong.
2. Perangkat lunak menggunakan semaphore untuk untuk mem-blocked producer atau consumer lain ketika salah satu producer atau consumer sedang berada dalam buffer.
3. Perangkat lunak simulasi Producer-Consumer ini merupakan ilustrasi dari proses sinkronisasi, yaitu bagaimana cara mengatur beberapa proses yang mengakses beberapa variabel secara bersamaan.

## Referensi

- [1] Silberschatz, A., Peterson, L.J., Operating System Concepts, Alternate Edition, Addison-Wesley Publishing Company, Inc., Juni 1978.
- [2] Hariyanto, B., MT, Sistem Operasi, Edisi2, Informatika, Bandung, 1999.
- [3] Hariyanto, B., MT, Sistem Operasi Lanjut, Informatika, Bandung, 2003.
- [4] Kusumadewi, S., Sistem Operasi, Edisi2, Graha Ilmu, Jakarta, 2000.
- [5] Hadi, R., Pemrograman Microsoft Visual Basic dengan menggunakan Windows API, PT. Elex Media Komputindo, Jakarta, 2001.
- [6] Suryokusumo, A., Microsoft Visual Basic 6.0, PT. Elex Media Komputindo, 2001.
- [7] Supardi, Y., Microsoft Visual Basic 6.0 Untuk Segala Tingkat, PT. Elex Media Komputindo, Jakarta, 2006.
- [8] [www.cs.umd.edu/~shankar/412-Notes/12-BoundedBufferProdCons.html](http://www.cs.umd.edu/~shankar/412-Notes/12-BoundedBufferProdCons.html) (tanggal akses: 22 Agustus 2006).
- [9] [cis.poly.edu/muller/CS623/consumer\\_producer.htm](http://cis.poly.edu/muller/CS623/consumer_producer.htm) (tanggal akses: 22 Agustus 2006).