

# Implementasi Penanganan Deadlock Menggunakan Metode Taskkill

*by* S Saifulloh

---

**Submission date:** 28-Jan-2019 06:18AM (UTC-0800)

**Submission ID:** 1069566999

**File name:** DEADLOCK.docx (373.72K)

**Word count:** 1837

**Character count:** 11757

# IMPLEMENTASI PENANGANAN DEADLOCK MENGUNAKAN METODE TASKKILL

<sup>1</sup>Saifulloh, <sup>2</sup>Hani Atun Mumtahana

20

<sup>1,2</sup> Program Studi Sistem Informasi, Universitas PGRI Madiun  
Email : saifulloh@unipma.ac.id, hanimumtahana@unipma.ac.id

## Abstract

*Purpose Operating system one of them to manage the processes that run in the same period of time without being able to collide with each other. The operating system must be able to control the use of resources. Processes that interact require synchronization to be controlled and avoid deadlock conditions. The process of not responding when the execution of a program process called Deadlock condition. Handling deadlock on windows operating system using task kill method. In the process of handling using task kill is making the shortcut in the windows facility by adding the source code with the aim to end the process that not responding because of the deadlock. Impacts that occur in applying this method is able to run the process of running execution without having to open the file again when experiencing not responding.*

**Keywords :** *deadlock, Operating System, taskkill, shortcut.*

## Abstrak

*Tujuan Sistem operasi salah satunya untuk mengatur proses-proses yang berjalan dalam jangka waktu yang sama tanpa boleh saling bertabrakan satu dengan yang lainnya. Sistem operasi harus mampu melakukan pengontrolan penggunaan resource. Proses-proses yang berinteraksi memerlukan sinkronisasi agar terkendali dan terhindar dari kondisi deadlock. Proses not responding ketika eksekusi suatu proses program disebut dengan kondisi Deadlock. Penanganan deadlock pada sistem operasi windows menggunakan metode task kill. Pada proses penanganan menggunakan task kill yaitu membuat shortcut yang ada di fasilitas windows dengan menambahkan source code dengan tujuan untuk mengakhiri proses yang not responding karena terjadinya deadlock. Dampak yang terjadi dalam menerapkan metode ini adalah dapat melanjutkan proses eksekusi yang sedang berjalan tanpa harus buka file kembali ketika mengalami not responding.*

**Kata kunci:** *deadlock, sistem operasi, taskkill, shortcut*

## 1. Pendahuluan

Sistem operasi komputer merupakan perangkat lunak komputer atau program yang bertugas untuk melakukan kontrol dan manajemen perangkat keras termasuk menjalankan *software* aplikasi seperti program pengolahan data yang bisa digunakan untuk mempermudah kegiatan manusia. Dalam proses perancangan sistem operasi terdapat lebih dari satu proses berada pada saat yang sama, keadaan ini disebut dengan *multitasking* dari sistem operasi [2]. Sistem operasi bertugas untuk mengatur proses-proses yang berjalan dalam jangka waktu yang sama tanpa boleh saling bertabrakan satu dengan yang lainnya. Proses-proses yang berinteraksi memerlukan sinkronisasi agar terhindar dari kondisi *deadlock*. Dalam melakukan pengaturan diperlukan mekanisme khusus untuk mencegah terjadinya masalah [4].

*Deadlock* adalah keadaan sistem dimana setiap proses di beberapa kelompok meminta sumber daya dari proses lain dalam kelompok tersebut, dan kemudian menunggu tanpa batas waktu agar permintaan ini terpenuhi. Ada empat teknik yang secara teratur terlibat dalam mengatasi kebuntuan dalam sistem database yaitu

*mengabaikan masalah, deteksi deadlock, pencegahan kebuntuan dan penghindaran kebuntuan.* Mengabaikan kebuntuan adalah skema termudah untuk diterapkan. *Deadlock* terjadi karena sekumpulan proses yang di blok dikarenakan setiap proses membawa sumber daya dan menunggu mendapatkan sumber daya sehingga keadaan status menunggu tidak akan berakhir. Deteksi *deadlock* mencoba untuk menemukan dan menyelesaikan kebuntuan. Penghindaran kebuntuan menggambarkan teknik yang mencoba untuk menentukan apakah kebuntuan akan terjadi pada saat sumber daya diminta dan bereaksi terhadap permintaan dengan cara yang menghindari kebuntuan [1]. Pencegahan *deadlock* adalah penataan sistem sedemikian rupa sehingga salah satu kondisi yang diperlukan untuk kebuntuan tidak dapat terjadi. Setiap kategori solusi cocok untuk jenis lingkungan tertentu dan memiliki kelebihan dan kekurangan, untuk lebih jelasnya ketika kita membuka banyak aplikasi pada suatu *operating system windows* di waktu yang bersamaan, dan karena terlalu banyak aplikasi yang harus diproses oleh *Operating System* tersebut tanpa interval waktu maka salah satu atau lebih

aplikasi akan mengalami keadaan “Not Responding” merupakan suatu kejadian dimana sebuah komputer tidak dapat melakukan proses dikarenakan terjadi kegagalan pembacaan perintah yang berakibat komputer berhenti.

Penelitian ini membahas tentang cara penanganan deadlock dengan menggunakan perangkat lunak *taskkill.exe* sebuah aplikasi bawaan Windows yang sudah disediakan. Dan dengan begitu dapat menutup aplikasi yang mengalami “Not Responding” dengan cepat dan mudah.

## 2. Kajian Literatur

### 2.1 Definisi Sistem Operasi

Sistem operasi secara umum adalah pengelola seluruh sumber daya yang terdapat pada sistem komputer dan menyediakan sekumpulan layanan ke pemakai sehingga memudahkan penggunaan serta pemanfaatan sumber daya pada sistem komputer. Sistem operasi bertugas untuk mengatur proses-proses yang berjalan dalam jangka waktu yang sama tanpa boleh saling bertabrakan satu dengan yang lainnya. Proses-proses yang berinteraksi memerlukan sinkronisasi agar terhindar dari kondisi *deadlock* [2].

Sistem operasi berfungsi sebagai penghubung antara lapisan hardware dan software serta menjamin aplikasi-aplikasi yang berbeda fungsi dapat berjalan lancar secara bersama tanpa hambatan. Sehingga sistem operasi akan mengatur jadwal sesuai perintah dan dapat melakukan eksekusi sesuai state proses program.

### 2.2. Deadlock

Deadlock adalah suatu keadaan yang mengakibatkan tidak ada proses yang dapat diselesaikan sampai tuntas karena adanya saling tunggu antar proses. Saling tunggu dapat dikarenakan prasyarat dari setiap proses adalah proses lain, ataupun saling menunggu sumber daya yang sedang digunakan proses lain [4]. Ada empat penyebab *deadlock* :

1. Mutual exclusion : Keadaan di mana setiap sumber daya hanya bisa digunakan untuk satu proses saja pada satu periode tertentu.
2. Hold and wait : Suatu keadaan di mana proses dapat masuk ke dalam status hold dan menunggu *resource* lain yang sedang digunakan proses lain.
3. No preemption : Suatu sumber daya tidak bisa diambil setiap saat dari suatu proses. Sumber daya hanya dapat diambil apabila proses tersebut telah selesai digunakan.
4. Circular wait : Keadaan dua proses saling menunggu secara circular karena proses saling menunggu sumber daya. Jika terjadi deadlock pada suatu proses maka dideteksi sistem mana yang terlibat di dalamnya. Setelah diketahui sistem mana saja yang terlibat maka diadakan proses untuk memperbaiki dan menjadikan sistem berjalan kembali.

Jika sebuah sistem tidak memastikan deadlock akan terjadi, dan juga tidak didukung dengan pendeteksian

deadlock serta pencegahannya, maka kita akan sampai pada kondisi deadlock yang dapat berpengaruh terhadap performance sistem karena sumber daya tidak dapat digunakan oleh proses sehingga proses-proses yang lain juga terganggu. Akhirnya sistem akan berhenti dan harus direstart.

Hal yang terjadi dalam mendeteksi adanya deadlock adalah permintaan sumber daya dikabulkan selama memungkinkan, sistem operasi memeriksa adakah kondisi circular wait secara periodic, pemeriksaan adanya deadlock dapat dilakukan setiap ada sumber daya yang hendak digunakan oleh sebuah proses, memeriksa dengan algoritma tertentu.

### 3. Metode Penelitian

Pada penelitian ini menggunakan metode penelitian tindakan (*action research*). Suatu pendekatan *systematic inquiry* yang memiliki ciri, prinsip, pedoman dan prosedur yang harus memenuhi kriteria [5]. Kondisi suatu proses yang beragam ketika dieksekusi merupakan suatu ciri dimana untuk terjadinya kondisi deadlock pada sistem khususnya antara proses hal ini merupakan suatu kriteria yang harus terjadi.

Selanjutnya penelitian tindakan selalu berhubungan dengan tindakan untuk mencapai hasil yang praktis dan menciptakan bentuk pemahaman baru dikarenakan tindakan tanpa pengetahuan sama halnya buta dan teori tanpa tindakan tidak berarti [3].

Tujuan dasar penelitian tindakan lebih ditujukan untuk meningkatkan praktik ketimbang memproduksi pengetahuan, berfokus pada praktik sosial bertujuan untuk meningkatkan perbaikan keadaan

## 4. Hasil dan Pembahasan

### 4.1. Analisis Permasalahan

Pada tahap ini dilakukan analisa permasalahan yang telah dirumuskan, yaitu menangani kasus deadlock pada sistem operasi windows. Kemudian menganalisa kebutuhan simulasi dan menganalisa metode yang akan digunakan.

### 4.2. Analisa Simulasi

Deadlock dapat dianalogikan dalam kondisi ketika sedang mengerjakan tugas dimana terdapat 1 orang berperan sebagai notulis dan dua orang bertugas mendiktekan hasil diskusi yang sudah selesai dikerjakan dan harus diketik. Ketika dua orang pendikte secara bersamaan mendiktekan sesuatu kepada notulis maka notulis akan kesulitan untuk memilih yang mana yang harus dituliskan terlebih dahulu. Dalam kondisi “bingung” tersebut inilah yang dinamakan deadlock

#### 4.2.1. Analisa Metode

Deadlock pada Sistem Operasi adalah keadaan dimana 2 atau lebih proses saling menunggu meminta resources untuk waktu yang tidak terbatas lamanya. Analoginya seperti pada kondisi jalan raya dimana terjadi kemacetan parah. Deadlock adalah efek samping dari sinkronisasi, dimana satu variabel digunakan oleh 2 proses.

Solusinya yaitu, dengan menggunakan *taskkill.exe* sebuah aplikasi bawaan Windows yang sudah disediakan. Dan dengan begitu dapat menutup aplikasi yang mengalami “Not Responding” dengan cepat dan mudah. Metode dalam kasus ini menggunakan Mutual exclusion, dimana keadaan setiap sumber daya hanya bisa digunakan untuk satu proses saja pada satu periode tertentu.

#### 4.3. Perancangan

Untuk mengatasi permasalahan tersebut, maka perlu dirancang antar muka (interface). Dalam perancangan interface hal terpenting yang ditekankan adalah bagaimana menciptakan tampilan yang baik dan mudah dimengerti oleh pengguna.

#### 4.1. Implementasi

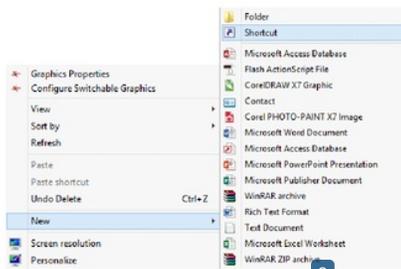
Implementasi merupakan kelanjutan dari tahap perancangan yang telah di buat. Implementasi juga merupakan tahap pembuatan aplikasi penanganan deadlock. Tujuan implementasi antara lain :

- 1) Menyelesaikan aplikasi yang di buat untuk mengatasi deadlock
- 2) Menguji dan mendokumentasikan aplikasi tersebut.
- 3) Memastikan bahwa pemakai dapat menjalankan aplikasi tersebut.
- 4) Memastikan bahwa aplikasi berjalan dengan benar

#### 4.2. Hasil Implementasi

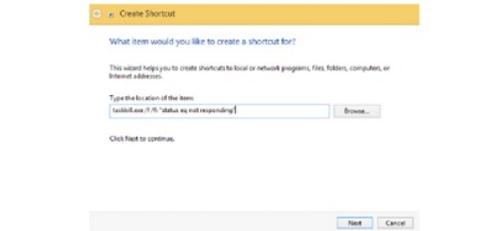
Dari analisa yang dijelaskan pada metode penelitian, pada bab ini kami akan membuat cara penanganan deadlock dengan menggunakan *taskkill.exe* sebuah aplikasi bawaan Windows yang sudah disediakan. Dan dengan begitu dapat menutup aplikasi yang mengalami “Not Responding” dengan cepat dan mudah. Dan untuk pembuatan *taskkill.exe* caranya adalah :

- 1) Klik kanan desktop Anda, Pilih New > Shortcut

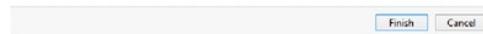
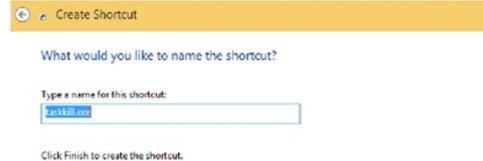


Gambar 1.Langkah Pembuatan Shortcut

- 2) Copy-paste kode di bawah ini ke field location shortcut `taskkill.exe /f /fi "status eq not responding"`



Gambar 2. Memberikan Kode Perintah



Gambar 3. Memberikan nama pada shortcut



Gambar 4.Hasil Pembuatan Shortcut

Setelah aplikasi *taskkill.exe* berhasil dibuat, maka kita akan menguji coba aplikasi tersebut apakah berfungsi dengan baik atau tidak. Kami mendapat peristiwa ata kejadian saat deadlock seperti pada gambar dibawah ini. Pada gambar tersebut, terjadi not responding pada saat membuka aplikasi Adobe Flash CS 6.

Penyebab terjadinya kasus deadlock tersebut, mungkin dikarenakan sistem terlalu banyak melakukan proses sehingga aplikasi tersebut tiba-tiba terhenti atau not responding. Dan pada waktu not responding, biasanya akan muncul notifikasi seperti gambar dibawah ini. Sebenarnya kita bisa saja mengklik *Close Program* agar bisa langsung menghentikan proses tersebut. Tapi di sini, kami ingin menguji cobakan aplikasi *taskkill* ini, apakah berfungsi dengan baik atau tidak. Dan ternyata setelah saya mengklik aplikasi *taskkill*, sistem tidak langsung menutup aplikasi Adobe Flash tersebut, tapi justru membuka aplikasi tersebut dan dapat dijalankan dengan baik.



Gambar 5. Contoh kasus deadlock

## 5. Kesimpulan

Deadlock merupakan suatu keadaan yang mengakibatkan tidak ada proses yang dapat diselesaikan sampai tuntas karena adanya saling tunggu antar proses baik dikarenakan dari proses lain, ataupun saling menunggu sumber daya yang sedang digunakan untuk proses lain. Salah satu proses yang mengalami kondisi deadlock yakni proses *not responding*.

Deadlock bisa ditangani dengan cara taskkill dengan memanfaatkan fitur shortcut yang berisikan kode untuk mengakhiri proses yang not responding dengan mode *shortcut*. Kondisi *not responding* termasuk *circular wait* dikarenakan proses yang dieksekusi berbenturan satu proses dengan proses lainnya yang mengakibatkan deadlock dan dapat berpengaruh terhadap performance sistem

## Daftar Pustaka

- [1] Darwish S.M, et all. 2015. Soft Computing FOR Database Deadlock Resolution. International Journal Of Model And Optimization, Vol.5 No.1 February 2015.
- [2] Khusna A.N, et all. 2017. Perangkat Lunak Simulasi Deadlock Menggunakan Ilustrasi Dining Philosophers Problem. Jurnal Informatika dan Komputer (JIKO), Vol.2 No.1 Februari 2017. Universitas Ahmad Dahlan : Yogyakarta.
- [3] Koshy, Valsa. Action Research for Improving Practice : A practical Guide. London: Sage Publication Ltd. 2005
- [4] Safei T.T. 2012. Pencegahan deadlock pada alokasi resource dalam sistem operasi menggunakan algoritma Greedy. Program Studi Teknik Informatika, Institut Teknologi Bandung : 2012.
- [5] Semiawan, Conny R. Catatan kecil tentang penelitian dan pengembangan ilmu pengetahuan. Jakarta: Kencana Prenada Media Group, 2007.

# Implementasi Penanganan Deadlock Menggunakan Metode Taskkill

## ORIGINALITY REPORT

16%

SIMILARITY INDEX

14%

INTERNET SOURCES

1%

PUBLICATIONS

6%

STUDENT PAPERS

## PRIMARY SOURCES

1	<a href="http://pemikiranperasaanku.blogspot.com">pemikiranperasaanku.blogspot.com</a> Internet Source	2%
2	<a href="http://5302414039.blogspot.com">5302414039.blogspot.com</a> Internet Source	2%
3	<a href="http://langit-it.blogspot.com">langit-it.blogspot.com</a> Internet Source	1%
4	Submitted to Boston Latin Academy Student Paper	1%
5	Submitted to CVC Nigeria Consortium Student Paper	1%
6	<a href="http://amminessalafy.blogspot.com">amminessalafy.blogspot.com</a> Internet Source	1%
7	<a href="http://repository.uinjkt.ac.id">repository.uinjkt.ac.id</a> Internet Source	1%
8	<a href="http://www.pradana.net">www.pradana.net</a> Internet Source	1%
9	<a href="http://eltineregiena.blogspot.com">eltineregiena.blogspot.com</a>	

	Internet Source	1%
10	<a href="http://ejournal.iainpurwokerto.ac.id">ejournal.iainpurwokerto.ac.id</a> Internet Source	1%
11	<a href="http://blogagki.files.wordpress.com">blogagki.files.wordpress.com</a> Internet Source	1%
12	<a href="http://ririnbelajarlinier.blogspot.com">ririnbelajarlinier.blogspot.com</a> Internet Source	1%
13	<a href="http://ejournal.akakom.ac.id">ejournal.akakom.ac.id</a> Internet Source	1%
14	<a href="http://scholar.uad.ac.id">scholar.uad.ac.id</a> Internet Source	<1%
15	<a href="http://keys17.wordpress.com">keys17.wordpress.com</a> Internet Source	<1%
16	<a href="http://ajengnareswari.blog.student.eepis-its.edu">ajengnareswari.blog.student.eepis-its.edu</a> Internet Source	<1%
17	<a href="http://catatan-helpdesk.blogspot.com">catatan-helpdesk.blogspot.com</a> Internet Source	<1%
18	<a href="http://tugaskami25.blogspot.com">tugaskami25.blogspot.com</a> Internet Source	<1%
19	<a href="http://djulkiplie.blogspot.com">djulkiplie.blogspot.com</a> Internet Source	<1%
20	Hadi Kusnanto, Yudi Hartono. "MASJID	<1%

# TEGALSARI JETIS PONOROGO (MAKNA SIMBOLIK DAN POTENSINYA SEBAGAI SUMBER PEMBELAJARAN SEJARAH LOKAL)", Gulawentah:Jurnal Studi Sosial, 2017

Publication

---

---

Exclude quotes      On

Exclude matches      Off

Exclude bibliography      On