

Jurnal 1 - Achmad Arrosyidi - Implementasi Program Simulasi Page Replacement

by Achmad Arrosyidi

Submission date: 06-Jan-2020 03:45PM (UTC+0700)

Submission ID: 1239492549

File name: d_Arrosyidi_-_Implementasi_Program_Simulasi_Page_Replacement.pdf (341.54K)

Word count: 2262

Character count: 14352

Implementasi Program Simulasi *Page Replacement*

8 Achmad Arrosyidi

Program Studi DIII Manajemen Informatika, Institut Bisnis dan Informatika Stikom Surabaya

Email: achmad@stikom.edu

4 Pembelajaran di Perguruan Tinggi yang belum optimal disebabkan diantaranya oleh pembelajar menggunakan konsep-konsep 4 pembelajaran yang tidak relevan dengan perkembangan teknologi pembelajaran dan belum diarahkan ke proses aktif pembelajar. Salah satu teori yang solutif berkenaan dengan masalah diatas adalah dengan metode pembelajaran aktif dan ditunjang dengan sarana pembelajaran aktif, yaitu dengan menggunakan media pembelajaran berbasis komputer.

Kendala yang dialami oleh mahasiswa program studi dibidang komputer yang wajib lulus yaitu memahami teori optimalisasi manajemen virtual memori dengan menggunakan *page replacement* pada mata kuliah Sistem Operasi yang didalamnya terdapat banyak algoritma, secara umum menggunakan algoritma *first in first out*, optimal dan *least recently used*. Dibutuhkan media pembelajaran berbasis komputer yang bertujuan untuk membantu mahasiswa belajar secara aktif dalam memahami materi *page replacement* yang berbentuk *software* komputer yaitu program simulasi *page replacement*.

Pada penelitian sebelumnya telah dibuat program simulasi *page replacement* dengan mempunyai fitur yang mampu mengimplementasikan alur data *page replacement* menggunakan algoritma *first in first out*, optimal dan *least recently used* serta mampu menyediakan *interface* grafis yang berguna membentuk program simulasi *page replacement*. Program simulasi *page replacement* sudah melalui tahapan pengujian *white box*, dengan hasil yang valid dan siap untuk dilakukan pengujian *black box* oleh calon pengguna.

Program simulasi *Page Replacement* belum diimplementasikan, sehingga belum dapat diketahui kelayakan agar dapat digunakan oleh mahasiswa yang menempuh mata kuliah sistem operasi dengan pembahasan *page reploment*. Dari permasalahan tersebut maka perlu dilakukan penelitian agar dapat diketahui kelayakan, mudah digunakan oleh penggunanya, dapat membantu memahami teori *page replacement* serta dapat membantu belajar mahasiswa digunakan oleh mahasiswa yang menempuh mata kuliah sistem operasi dengan pembahasan *page reploment* untuk dapat belajar secara mandiri.

TUJUAN DAN RUANG LINGKUP PEMBAHASAN

Berdasarkan latar belakang di atas, tulisan ini untuk mengetahui program simulasi *page replacement* dapat:

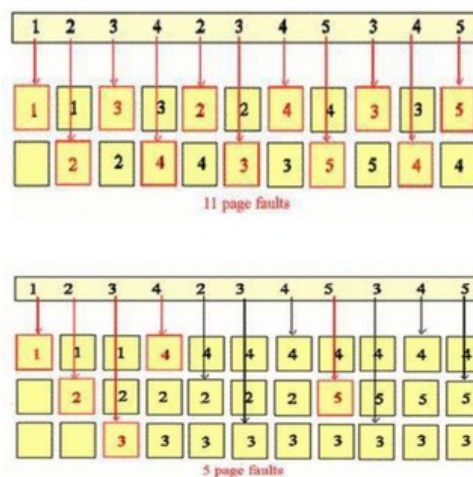
1. Valid dan layak untuk digunakan.
2. 5udah untuk digunakan.
3. Sangat membantu mahasiswa dalam memahami materi kuliah pada pembahasan *page replacement*.
4. Sangat membantu mahasiswa dalam belajar secara mandiri.

SISTEM OPERASI

Terdapat berbagai algoritma *Page Replacement* dalam Sistem Operasi komputer secara umum dalam materi pembelajaran adalah algoritma *First In First Out* (FIFO), algoritma Optimal dan algoritma *Least Recently Used* (LRU).

Algoritma FIFO (*First In First Out*)

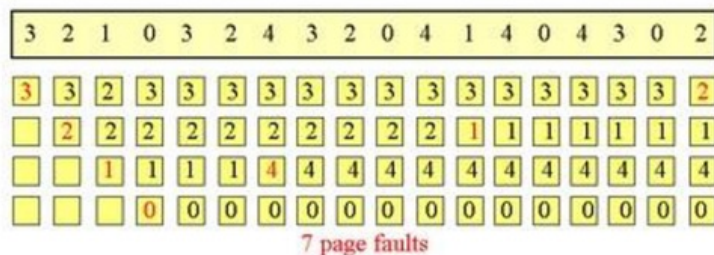
Algoritma FIFO merupakan algoritma paling sederhana. Algoritma FIFO diasosiasikan dengan sebuah *page* bila *page* tersebut dibawa ke memori. Bila ada suatu *page* yang akan ditempatkan, maka posisi *page* yang paling lama akan digantikan. Algoritma ini tidak perlu menyimpan waktu pada saat sebuah *page* dibawa ke memori.



Gambar 1. *Page Replacement* Dengan Menggunakan Algoritma FIFO

ALGORITMA OPTIMAL

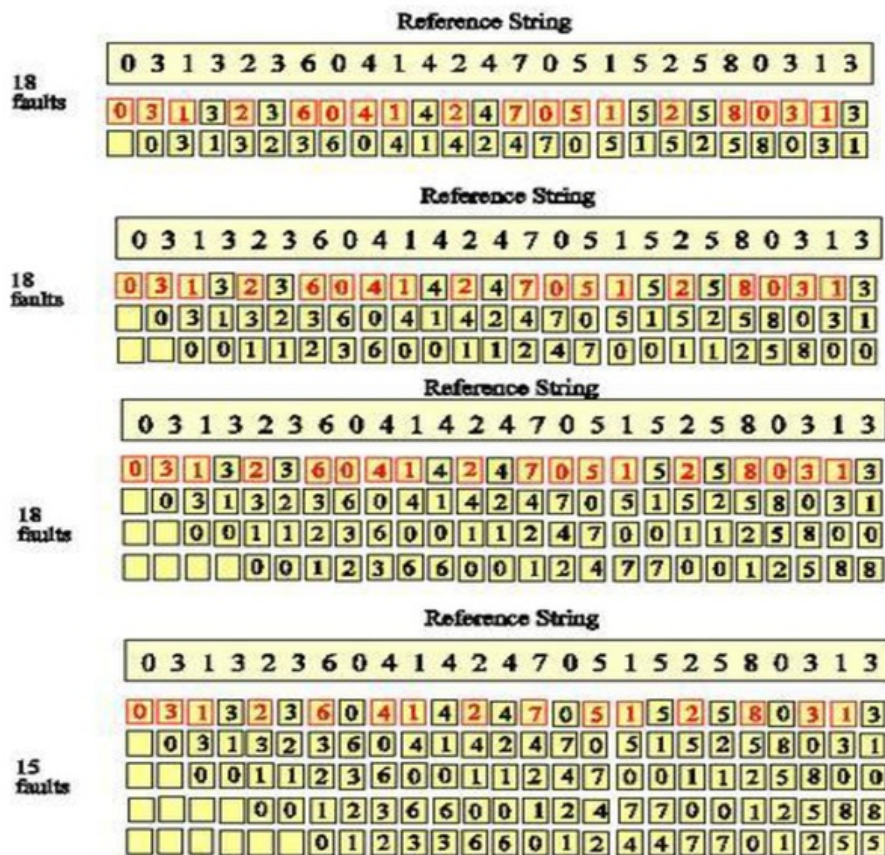
Algoritma Optimal merupakan hasil penemuan dari anomaly Belady. Algoritma ini mempunyai rata-rata *page fault* terendah. Algoritma optimal akan mengganti *page* yang tidak akan digunakan untuk periode waktu terlalu lama. Algoritma ini menjamin rata-rata *page fault* terendah untuk jumlah frame tetap, tetapi sulit implementasinya. Ilustrasi dari algoritma seperti berikut:



Gambar 2. *Page Replacement* Dengan Menggunakan Algoritma Optimal

ALGORITMA LRU (LEAST RECENTLY USED)

Dikarenakan algoritma optimal sangat sulit dalam pengimplementasinya, maka dibuat algoritma lain yang *performance*-nya mendekati algoritma optimal dengan sedikit *cost* yang lebih besar. Algoritma ini mengganti halaman yang paling lama tidak dibutuhkan. Asumsinya, halaman yang sudah lama tidak digunakan sudah tidak dibutuhkan lagi dan kemungkinan besar, halaman yang baru di-load akan digunakan kembali.



Gambar 3. Page Replacement Dengan Menggunakan Algoritma LRU

Sama seperti algoritma optimal, algoritma LRU tidak mengalami anomali Belady. Algoritma ini menggunakan *linked list* untuk mendata halaman mana yang paling lama tidak terpakai. *Linked list* inilah yang membuat *cost* membesar, karena harus meng-*update linked list* tiap saat ada halaman yang di akses. Halaman yang berada di *linked list* paling depan adalah halaman yang baru saja digunakan. Semakin lama tidak dipakai, halaman akan berada semakin belakang dan di posisi terakhir adalah halaman yang paling lama tidak digunakan dan siap untuk di-*swap*.

1

PENGUJIAN (TESTING)

Black box testing, dilakukan tanpa pengetahuan detil struktur internal dari sistem atau komponen yang dites. juga disebut sebagai *behavioral testing*, *specification-based testing*, *input/output testing* atau *functional testing*.

Black box testing berfokus pada kebutuhan fungsional pada *software*, berdasarkan pada spesifikasi kebutuhan dari *software*. Dengan adanya *black box testing*, perencana *software* dapat menggunakan sekumpulan kondisi masukan yang dapat secara penuh memeriksa keseluruhan kebutuhan fungsional pada suatu program.

Black box testing bukan teknik alternatif daripada *white box testing*. Lebih daripada itu, ia merupakan pendekatan pelengkap dalam mencakup *error* dengan kelas yang berbeda dari metode *white box testing*.

2

Kategori *error* yang akan diketahui melalui *black box testing*:

- Fungsi yang hilang atau tak benar
- *Error* dari antar-muka
- *Error* dari struktur data atau akses eksternal *database*
- *Error* dari kinerja atau tingkah laku
- *Error* dari inisialisasi dan terminasi

Tak seperti *white box testing*, yang 9) tipa digunakan pada tahap akhir dan berfokus pada domain informasi. Tes didisain untuk menjawab pertanyaan sebagai berikut:

- Bagaimana validasi fungsi yang akan dites?
- Bagaimana tingkah laku dan kinerja sistem dites?
- Kategori masukan apa saja yang bagus digunakan untuk test cases?
- Apakah sebagian sistem sensitif terhadap suatu nilai masukan tertentu?
- Bagaimana batasan suatu kategori masukan ditetapkan?
- Sistem mempunyai toleransi jenjang dan volume data apa saja?
- Apa saja akibat dari kombinasi data tertentu yang akan terjadi pada operasi sistem?

1

Dengan menerapkan teknik *black box*, dapat dibuat sekumpulan *test cases* yang memuaskan kriteria-kriteria sebagai berikut [MYE79]:

- *Test cases* yang mengurangi jumlah *test cases* (lebih dari satu) yang didesain untuk mencapai testing yang masuk akal.
- *Test cases* yang dapat memberikan informasi tentang kehadiran kelas-kelas dari *error*.

Dekomposisi Kebutuhan Untuk Dites Secara Sistematis

Kebanyakan tester, saat memulai proyek testing, akan menghadapi masalah untuk memutuskan *test cases* apa yang akan mereka eksekusi untuk melakukan tes sistem mereka. Untuk dapat membuat *test cases* yang efektif, harus dilakukan dekomposisi dari tugas-tugas testing suatu sistem ke aktivitas-aktivitas yang lebih kecil dan dapat dimanajementi, hingga tercapai *test case* individual. Tentunya, dalam disain *test case* juga digunakan mekanisme untuk memastikan bahwa *test case* yang ada telah cukup mencakup semua aspek dari sistem. Pendisainan *test case* dilakukan secara manual, Tidak ada alat bantu otomatis guna menentukan *test cases* yang dibutuhkan oleh sistem, karena tiap sistem berbeda, dan alat bantu tes tak dapat mengetahui aturan benar-salah dari suatu operasi. Disain tes membutuhkan pengalaman, penalaran dan intuisi dari seorang tester.

Spesifikasi Sebagai Tuntunan Testing

Spesifikasi atau model sistem adalah titik awal dalam memulai disain tes. Spesifikasi atau model sistem dapat berupa spesifikasi fungsional, spesifikasi kinerja atau keamanan, spesifikasi skenario pengguna, atau spesifikasi berdasarkan pada resiko sistem. Spesifikasi menggambarkan kriteria yang digunakan untuk menentukan operasi yang benar atau dapat diterima, sebagai acuan pelaksanaan tes.

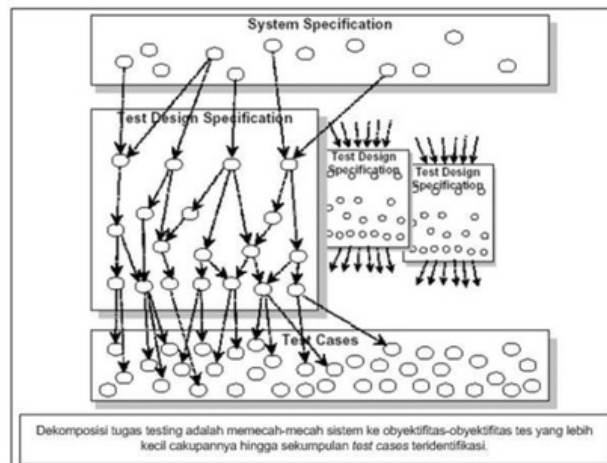
Banyak kasus, biasanya berhubungan dengan sistem lama, hanya terdapat sedikit atau bahkan tidak ada dokumentasi dari spesifikasi sistem. Dalam hal ini sangat dibutuhkan peran dari pengguna akhir yang mengetahui sistem untuk diikutsertakan ke dalam disain tes, sebagai ganti dari dokumen spesifikasi sistem. Walaupun demikian, harus tetap ada dokumentasi spesifikasi, yang bisa saja dibuat dalam bentuk sederhana, yang berisi sekumpulan obyektifitas tes di level atas.

Dekomposisi Obyektifitas Tes

Disain tes berfokus pada spesifikasi komponen yang dites. Obyektifitas tes tingkat atas disusun berdasarkan pada spesifikasi komponen. Tiap obyektifitas tes ini untuk kemudian didekomposisikan ke dalam obyektifitas tes lainnya atau *test cases* menggunakan teknik disain tes.

Terdapat banyak jenis teknik disain tes yang dapat dipilih berdasarkan pada tipe testing yang akan digunakan [BCS97A], yaitu:

- *Equivalence Class Partitioning*
- *Boundary Value Analysis*
- *State Transitions Testing*
- *Cause-Effect Graphing*



Gambar 4. Dekomposisi Obyektifitas Tes.

Pendokumentasian disain tes sangatlah penting. Disain tes direpresentasikan dalam suatdokumen yang disebut *Test Design Specification*, dimana format standarnya mengikuti [IEEE83A]. Keberadaan dokumen ini memudahkan dalam melakukan audit untuk melacak *test cases* yang diterapkan terhadap disain spesifikasi komponen. Pada umumnya dokumen disain tes dibutuhkan untuk menilai pihak lain dalam pemenuhan

dari tiap kebutuhan. Disain tes yang sistematis adalah kunci untuk melakukan dekomposisi tugas testing yang besar dan kompleks.

3

METODE PENELITIAN

Langkah-langkah yang digunakan dalam menyelesaikan penelitian ini adalah sebagai berikut:

1. Penentuan populasi, seluruh mahasiswa yang mengambil mata kuliah sistem operasi.
2. Penentuan sampel, menggunakan rumus Taro Yamane
3. Pembuatan instrumen penelitian, mengacu pada teori *black box*.
4. Pengambilan data, menggunakan instrumen penelitian dengan menggunakan skala linkert.
5. Rekapitulasi data, mengumpulkan data dan mengakumulasikan data berdasarkan item pengujian.
6. Analisis, menggunakan analisis deskriptif untuk mengetahui data hasil rekapitulasi.
7. Intepretasi, menterjemahkan hasil analisis data untuk diambil kesimpulan.

HASIL DAN PEMBAHASAN

Hasil implementasi Aplikasi *Page Replacement* pada responden adalah sebagai berikut:

Tabel 1. Hasil dan Pembahasan

No	Materi Implementasi	Hasil (%)	Skala	Pembahasan
1.	Tingkat validasi aplikasi pada setiap algoritma FIFO	100	Valid	Sesuai tujuan penelitian dan layak untuk digunakan
		0	Tidak valid	
2.	Tingkat validasi aplikasi pada setiap algoritma Optimal	100	Valid	Sesuai tujuan penelitian dan layak untuk digunakan
		0	Tidak valid	
3.	Tingkat validasi aplikasi pada setiap algoritma LRU	100	Valid	Sesuai tujuan penelitian dan layak untuk digunakan
		0	Tidak valid	
4.	Tampilan <i>Graphic User Interface</i> (GUI) pengisian jumlah antrian pada bagian <i>input/entry</i> data	23	Sangat mudah	Tampilan <i>Graphic User Interface</i> (GUI) pengisian jumlah antrian pada bagian <i>input/entry</i> data mudah untuk digunakan
		55	Mudah	
		23	Cukup Mudah	
		0	Sulit	
5.	Tampilan <i>Graphic User Interface</i> (GUI) pengisian jumlah <i>page</i> pada bagian <i>input/entry</i> data	17	Sangat mudah	Tampilan <i>Graphic User Interface</i> (GUI) pengisian jumlah <i>page</i> pada bagian <i>input/entry</i> data mudah untuk digunakan
		62	Mudah	
		21	Cukup Mudah	
		0	Sulit	
6.	Tampilan <i>Graphic User Interface</i> (GUI) pengisian	15	Sangat mudah	Tampilan <i>Graphic User Interface</i> (GUI)
		40	Mudah	

No	Materi Implementasi	Hasil (%)	Skala	Pembahasan
	<i>reference string</i> /urutan antrian pada bagian <i>input/entry</i> data	43	Cukup Mudah	pengisian <i>reference string</i> /urutan antrian pada bagian <i>input/entry</i> data cukup mudah untuk digunakan
		2	Sulit	
7.	Tampilan <i>Graphic User Interface</i> (GUI) pemilihan tombol algoritma FIFO pada bagian navigasi	28	Sangat mudah	Tampilan <i>Graphic User Interface</i> (GUI) pemilihan tombol algoritma FIFO pada bagian navigasi mudah untuk digunakan
		53	Mudah	
		19	Cukup Mudah	
		0	Sulit	
8.	Tampilan <i>Graphic User Interface</i> (GUI) pemilihan tombol algoritma Optimal pada bagian navigasi	30	Sangat mudah	Tampilan <i>Graphic User Interface</i> (GUI) pemilihan tombol algoritma Optimal pada bagian navigasi mudah untuk digunakan
		45	Mudah	
		25	Cukup Mudah	
		0	Sulit	
9.	Tampilan <i>Graphic User Interface</i> (GUI) pemilihan tombol algoritma LRU pada bagian navigasi	30	Sangat mudah	Tampilan <i>Graphic User Interface</i> (GUI) pemilihan tombol algoritma LRU pada bagian navigasi mudah untuk digunakan
		43	Mudah	
		26	Cukup Mudah	
		0	Sulit	
10.	Tampilan <i>Graphic User Interface</i> (GUI) pemilihan tombol Baru pada bagian navigasi	30	Sangat mudah	Tampilan <i>Graphic User Interface</i> (GUI) pemilihan tombol Baru pada bagian navigasi mudah untuk digunakan
		43	Mudah	
		26	Cukup Mudah	
		0	Sulit	
11.	Tampilan <i>Graphic User Interface</i> (GUI) pemilihan tombol Keluar pada bagian navigasi	45	Sangat mudah	Tampilan <i>Graphic User Interface</i> (GUI) pemilihan tombol Keluar pada bagian navigasi Sangat mudah untuk digunakan
		45	Mudah	
		9	Cukup Mudah	
		2	Sulit	
12.	Informasi yang ditampilkan <i>Graphic User Interface</i> (GUI) berupa hasil simulasi dengan menggunakan algoritma FIFO	30	Sangat mudah	Informasi yang ditampilkan <i>Graphic User Interface</i> (GUI) berupa hasil simulasi dengan menggunakan algoritma FIFO mudah untuk digunakan
		51	Mudah	
		19	Cukup Mudah	
		0	Sulit	

No	Materi Implementasi	Hasil (%)	Skala	Pembahasan
13.	Informasi yang ditampilkan <i>Graphic User Interface</i> (GUI) berupa hasil simulasi dengan menggunakan algoritma Optimal	30	Sangat mudah	Informasi yang ditampilkan <i>Graphic User Interface</i> (GUI) berupa hasil simulasi dengan menggunakan algoritma Optimal mudah untuk digunakan
		53	Mudah	
		17	Cukup Mudah	
		0	Sulit	
14.	Informasi yang ditampilkan <i>Graphic User Interface</i> (GUI) berupa hasil simulasi dengan menggunakan algoritma LRU	30	Sangat mudah	Informasi yang ditampilkan <i>Graphic User Interface</i> (GUI) berupa hasil simulasi dengan menggunakan algoritma LRU mudah untuk digunakan
		53	Mudah	
		15	Cukup Mudah	
		2	Sulit	
15.	Informasi yang ditampilkan <i>Graphic User Interface</i> (GUI) berupa hasil perbandingan tiga algoritma (FIFO, Optimal dan LRU)	23	Sangat mudah	Informasi yang ditampilkan <i>Graphic User Interface</i> (GUI) berupa hasil perbandingan tiga algoritma (FIFO, Optimal dan LRU) mudah untuk digunakan
		51	Mudah	
		25	Cukup Mudah	
		2	Sulit	
16.	Program simulasi <i>page replacement</i> dapat membantu dalam memahami materi kuliah pada pembahasan <i>page replacement</i>	55	Sangat mudah	Program simulasi <i>page replacement</i> dapat sangat membantu dalam memahami materi kuliah pada pembahasan <i>page replacement</i>
		36	Mudah	
		9	Cukup Mudah	
		0	Sulit	
17.	Program simulasi <i>page replacement</i> dapat membantu dalam belajar secara mandiri	51	Sangat mudah	Program simulasi <i>page replacement</i> dapat sangat membantu dalam belajar secara mandiri
		36	Mudah	
		13	Cukup Mudah	
		0	Sulit	

KESIMPULAN DAN SARAN

Didapatkan kesimpulan dan saran dari implementasi Aplikasi *Page Replacement* responden yang merupakan mahasiswa yang menempuh matakuliah dengan materi *page replacement*.

Kesimpulan

Secara keseluruhan implementasi Program Simulasi *Page Replacement* ini dapat disimpulkan bahwa:

1. Valid dan layak untuk digunakan.
2. Sudah untuk digunakan.
3. Sangat membantu mahasiswa dalam memahami materi kuliah pada pembahasan *page replacement*.
4. Sangat membantu mahasiswa dalam belajar secara mandiri.

Saran

Program Simulasi *Page Replacement* sudah relatif baik namun masih membutuhkan saran yaitu:

1. Tampilan *Graphic User Interface* (GUI) pengisian *reference string*/urutan antrian pada bagian *input/entry* data masih cukup sulit sehingga diperlukan cara *entry* data yang dapat secara langsung tanpa harus *entry* data secara satu persatu dari *combo box*.
2. Dibuat Aplikasi *Page Replacement* berbasis android sehingga dapat digunakan perangkat mobil yang secara umum digunakan oleh mahasiswa.

AFTAR PUSTAKA

- Dzulfikri, M., & Joko. 2013. Pengaruh Metode Pembelajaran Aktif Dengan Strategi Pembelajaran Reconnecting. *Jurnal Pendidikan Teknik Elektro, Vol. 2 Nomor. 2*, 515-522. Diambil kembali dari <http://ejournal.unesa.ac.id/index.php/jurnal-pendidikan-teknik-elektro/article/view/2215/1359>
- Mustaji. 2013. *Teknologi Pendidikan Universitas Negeri Surabaya*. Diambil kembali dari <http://pasca.tp.ac.id/>: <http://pasca.tp.ac.id/site/desain-pembelajaran-dengan-menggunakan-model-pembelajaran-kolaborasi-untuk-meningkatkan-kemampuan-berkolaborasi>
- Angera, A. A., & Ariyus, D. 2010. *Sistem Operasi*. Yogyakarta: Penerbit Andi.
- Romeo. 2003. *Testing Dan Implementasi Sistem. Edisi Pertama*. Stikom Surabaya. Surabaya: Tidak dipublikasikan.
- Santoso, T. S., & Ismayati, E. 2013. Pengembangan Media Pembelajaran Berbasis Computer Based Intruction (CBI). *Jurnal Pendidikan Teknik Elektro, Vol. 2 Nomor 2* (Media Pembelajaran), 458-464.

Jurnal 1 - Achmad Arrosyidi - Implementasi Program Simulasi Page Replacement

ORIGINALITY REPORT

10%	%	%	10%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Universitas Muria Kudus Student Paper	5%
2	Submitted to Universitas Dian Nuswantoro Student Paper	1%
3	Submitted to Universitas Pendidikan Indonesia Student Paper	1%
4	Submitted to Sriwijaya University Student Paper	1%
5	Submitted to Program Pascasarjana Universitas Negeri Yogyakarta Student Paper	1%
6	Submitted to STIKOM Surabaya Student Paper	<1%
7	Submitted to Universitas Trunojoyo Student Paper	<1%
8	Submitted to KYUNG HEE UNIVERSITY Student Paper	<1%
9	Submitted to Universitas Brawijaya Student Paper	

<1%

Exclude quotes On

Exclude matches Off

Exclude bibliography On