

# PERTUKARAN/PENGGANTIAN PAGE PADA MEMORI

## 1. Penggantian/Pertukaran Page

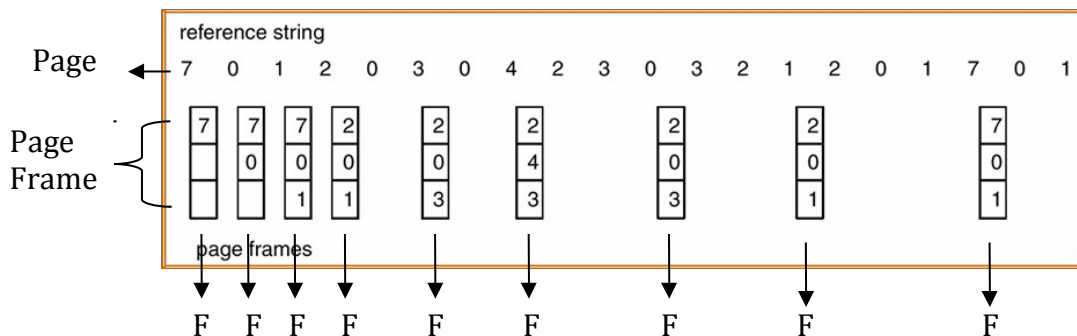
### a. Algoritma penggantian page acak (Random)

- ✓ Setiap terjadi page fault, penggantian page dipilih secara acak.
- ✓ Tidak memakai informasi apapun untuk menentukan page yang akan diganti.
- ✓ Semua page di memori utama dianggap memiliki bobot yang sama.
- ✓ Dapat memilih sembarang page termasuk page yang sedang diacu.

Setiap terjadi page fault, penggantian page dipilih secara acak. Tidak memakai informasi apapun untuk menentukan page yang akan diganti. Semua page di memori utama dianggap memiliki bobot yang sama. Dapat memilih sembarang page termasuk page yang sedang diacu.

### b. Algoritma penggantian page optimal

- ✓ Memilih page yang berpeluang dipakai kembali di masa datang yang paling kecil. (Memprediksi/melihat page berikutnya yang tidak dipakai)
- ✓ Strategi ini menghasilkan jumlah page fault sedikit tapi tidak mungkin diterapkan.



Penjelasan :

- Algoritma dengan 7 page sebagai string pengacuan (page 5 dan 6 dinyatakan invalid atau memiliki present/absent bit 0)
- Memiliki 3 page frame dan page fault.
- Page 7 diacu/ditempatkan di frame : Fault (F)
- Page 0 diacu/ditempatkan di frame : Fault (F)
- Page 1 diacu/ditempatkan di frame : Fault (F)
- Page 2 diacu/ditempatkan di frame dengan mengganti page 7 : (F)

- Page 7 diganti karena page berikutnya adalah page 0 dan page 3, dan page 7 hanya sedikit (1x) pada antrian page berikutnya berikutnya.
  - Page 0 diacu/ditempatkan di frame tanpa mengganti page.
  - Page 3 diacu/ditempatkan di frame dengan mengganti page 1 : (F)
  - Page 0 diacu/ditempatkan di frame tanpa mengganti page.
  - Page 4 diacu/ditempatkan di frame dengan mengganti page 0 : (F)
- Page 0 diganti karena page berikutnya adalah page 2 dan page 3.
  - Page 2 diacu/ditempatkan di frame tanpa mengganti page.
  - Page 3 diacu/ditempatkan di frame tanpa mengganti page.
  - Page 0 diacu/ditempatkan di frame dengan mengganti page 4 : (F)
- Page 4 diganti karena page berikutnya adalah page 3 dan page 2, bahkan page 4 sudah tidak dipakai kembali.
  - Page 3 dan page 2 diacu/ditempatkan di frame tanpa mengganti page.
  - Page 1 diacu/ditempatkan di frame dengan mengganti page 3 : (F)
  - dan seterusnya... hingga urutan page selesai

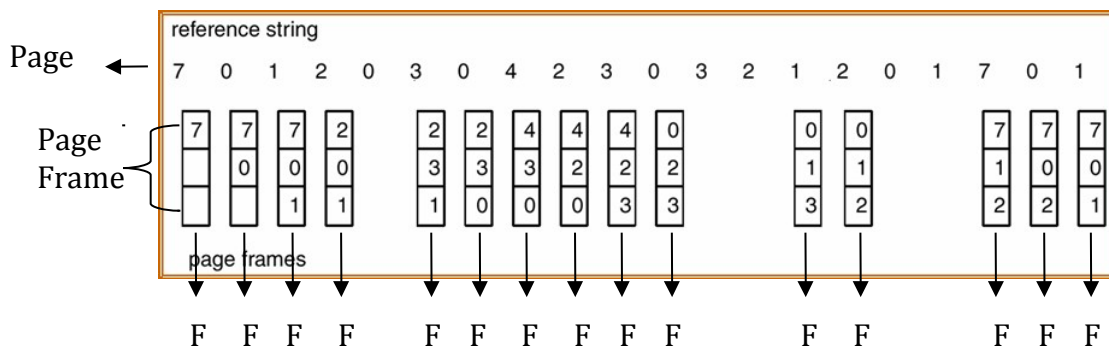
c. Algoritma penggantian page NRU

Page diberi dua bit mencatat status page :

- Bit R (referenced) menyatakan page sedang diacu  
Bit R = 0 , page sedang diacu  
Bit R = 1, page tidak sedang diacu
- Bit M (Modified) menyatakan page telah dimodifikasi  
Bit M = 0 , page belum dimodifikasi  
Bit M = 1, page telah dimodifikasi

d. Algoritma penggantian page FIFO

Bila terjadi page fault, page elemen terdepan diganti dan page baru ditambahkan di bagian belakang senarai.



Penjelasan :

- Page 7 diacu/ditempatkan di frame : Fault (F)
- Page 0 diacu/ditempatkan di frame : Fault (F)
- Page 1 diacu/ditempatkan di frame : Fault (F)
- Page 2 diacu/ditempatkan di frame dengan mengganti page terdepan dalam senarai yaitu page 7 : Fault (F)
- Page 0 diacu/ditempatkan di frame tanpa mengganti page.
- Page 3 diacu/ditempatkan di frame dengan mengganti page terdepan setelah page 7 dalam senarai yaitu page 0 : Fault (F)

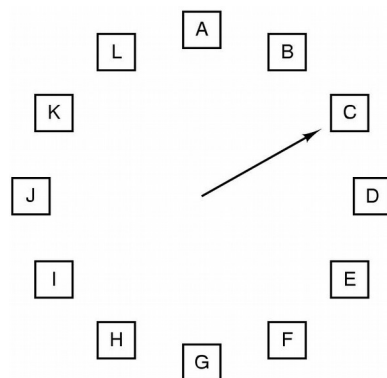
- Page 0 diacu/ditempatkan di frame dengan mengganti page terdepan setelah page 0 dalam senarai yaitu page 1 : Fault (F)
- Page 4 diacu/ditempatkan di frame dengan mengganti page terdepan setelah page 1 dalam senarai yaitu page 2 : Fault (F)
- Page 2 diacu/ditempatkan di frame dengan mengganti page terdepan setelah page 2 dalam senarai yaitu page 3 : Fault (F)
- Page 0 setelah page 2 tidak diperiksa karena tidak fault
- Page 3 diacu/ditempatkan di frame dengan mengganti page terdepan setelah page 3 dalam senarai yaitu page 0 : Fault (F)
- Page 0 diacu/ditempatkan di frame dengan mengganti page terdepan setelah page 0 dalam senarai yaitu page 4 : Fault (F)
- Page 3 dan page 2 diacu/ditempatkan di frame tanpa mengganti page.
- Page 1 diacu/ditempatkan di frame dengan mengganti page terdepan setelah page 4 dalam senarai yaitu page 2 : Fault (F)
- Page 2 diacu/ditempatkan di frame dengan mengganti page terdepan setelah page 2 dalam senarai yaitu page 3 : Fault (F)
- dan seterusnya... hingga senarai selesai

e. Algoritma penggantian page modifikasi FIFO

- 1) Algoritma penggantian page kesempatan kedua
- 2) Algoritma penggantian clock page

Pengacuan		2	3	2	1	5	2	4	5	3	2	5	2
	>	2	2	2	>2	2	2	>2	>2	>	>2	>2	>2
					*	*	*	*	*	2	*	*	*
		>	3	3	3	5	5	5	5*	5	5	5*	5*
Fault		F	F		F	F		F		F			

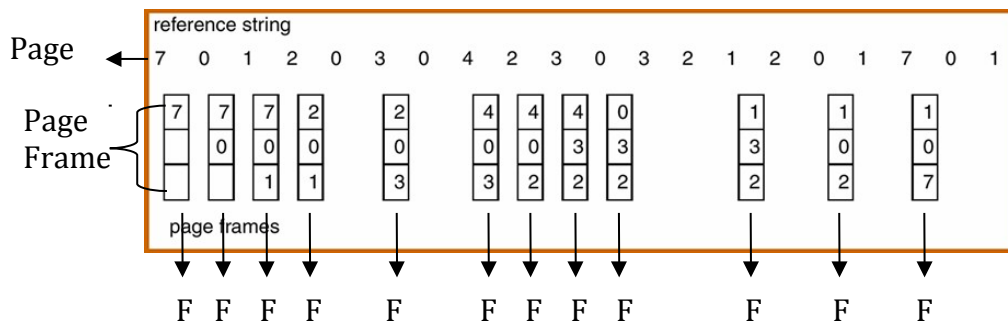
(\* diacu, > ditunjuk pointer)



When a page fault occurs, the page the hand is pointing to is inspected. The action taken depends on the R bit:  
 R = 0: Evict the page  
 R = 1: Clear R and advance hand

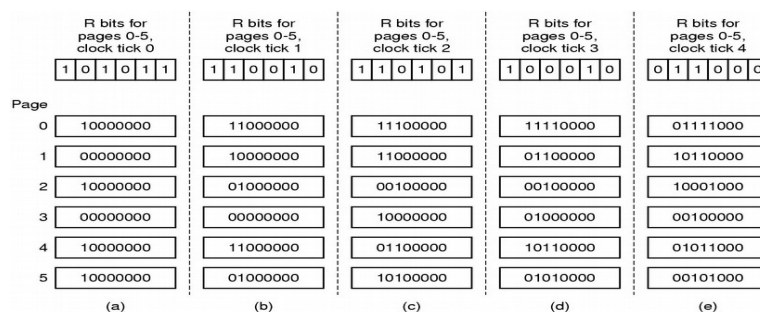
f. Algoritma penggantian page LRU

Beberapa instruksi terakhir dari page, kemungkinan masih dipakai. Jika terjadi page fault, maka algoritma mengganti page yang paling lama tidak digunakan.

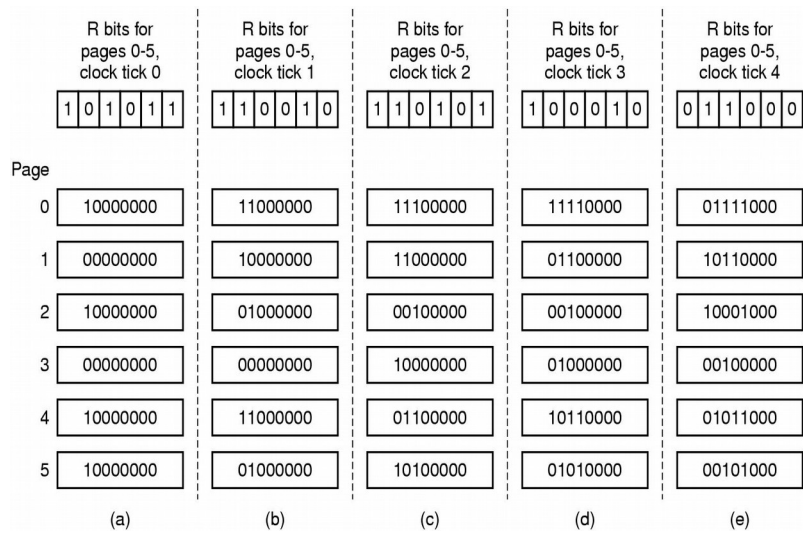


Penjelasan :

- Page 7 diacu/ditempatkan di frame : Fault (F)
- Page 0 diacu/ditempatkan di frame : Fault (F)
- Page 1 diacu/ditempatkan di frame : Fault (F)
- Page 2 diacu/ditempatkan di frame dengan mengganti page terdepan dalam senarai yaitu page 7 : Fault (F)
- Page 0 diacu/ditempatkan di frame tanpa mengganti page.
- Page 3 diacu/ditempatkan di frame dengan mengganti page yang tidak digunakan pada instruksi sebelumnya yaitu page 1 : Fault (F)
- Page 0 dan 2 merupakan instruksi sebelumnya yang masih dipakai.
- Page 0 diacu/ditempatkan di frame tanpa mengganti page.
- Page 4 diacu/ditempatkan di frame dengan mengganti page yang tidak digunakan pada instruksi sebelumnya yaitu page 2 : Fault (F)
- Page 0 dan 3 merupakan instruksi sebelumnya yang masih dipakai.
- Page 2 diacu/ditempatkan di frame dengan mengganti page yang tidak digunakan pada instruksi sebelumnya yaitu page 3 : Fault (F)
- Page 0 dan 4 merupakan instruksi sebelumnya yang masih dipakai.
- Page 3 diacu/ditempatkan di frame dengan mengganti page 0: Fault (F)
- Page 2 dan 4 merupakan instruksi sebelumnya yang masih dipakai.
- Page 0 diacu/ditempatkan di frame dengan mengganti page 4: Fault (F)
- Page 2 dan 3 merupakan instruksi sebelumnya yang masih dipakai
- Page 3 dan page 2 diacu/ditempatkan di frame tanpa mengganti page.
- Page 1 diacu/ditempatkan di frame dengan mengganti page 0: Fault (F)
- Page 2 dan 3 merupakan instruksi sebelumnya yang masih dipakai
- dan seterusnya... hingga senarai selesai



- The aging algorithm simulates LRU in software
- Note 6 pages for 5 clock ticks, (a) – (e)



- The aging algorithm simulates LRU in software
- Note 6 pages for 5 clock ticks, (a) – (e)

### Review Page Replacement Algorithm

Algorithm	Comment
1. Optimal	Tidak dapat diimplementasi tetapi berguna sebagai Benchmark
2. NRU	Tidak optimal
3. FIFO	Sering mengganti Page yang penting
4. Second Chance	Modifikasi dari FIFO
5. Clock	Realistis
6. LRU	Bagus tetapi sulit untuk diimplementasi
7. Aging	Efisien dalam mengimplementasi LRU
8. Working Set	Mahal untuk diimplementasi
9. WSClock	Cukup Efisien

## KUIS :

1. Diberikan suatu untai senarai informasi seperti pada tabel berikut di bawah ini berdasarkan penggunaan Algoritma LRU (Least Recently Used) :

Tabel:

Senarai	1	2	0	3	0	4	1	3	0	3	2	1	2	0	1	5	0	1	0	5
Pageframe1	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
Pageframe2	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
Pageframe3	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
Fault	?	?	?	?	dan seterusnya?															

Tentukan:

- a. Nilai yang akan dimiliki oleh setiap page frame ?
- b. Tentukan jumlah page fault yang terjadi ?