

Diskusi.5

Lakukan: Kirim balasan: 1

Jatuh tempo: Minggu, 9 November 2025, 23:59

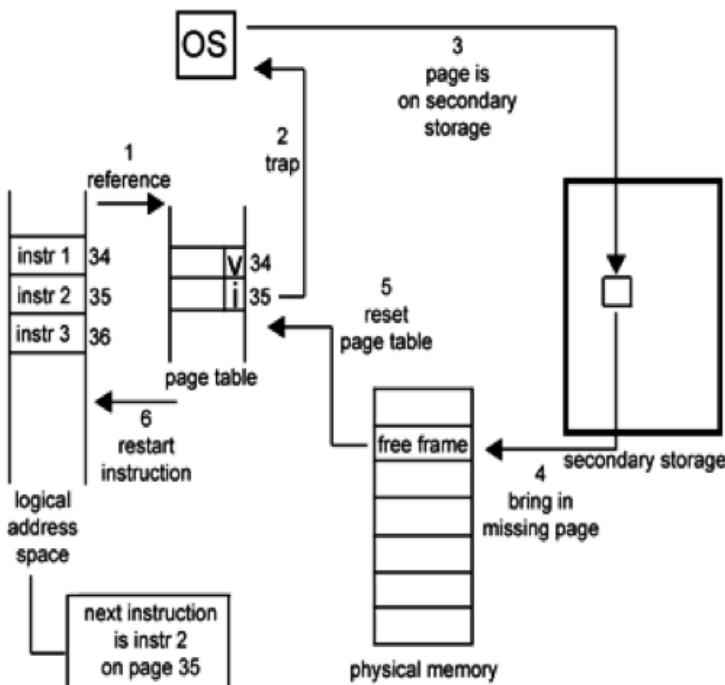
menampilkan balasan dalam bentuk bertingkat

 Setelan ▾

Diskusi.5

Rabu, 28 Mei 2025, 10:31

Gambar dibawah ini merupakan alur pada Penanganan kesalahan halaman (*Page Fault*). Deskripsikan dengan cara menuliskan alur jalannya intruksi sesuai dengan gambar dibawah ini.



Tautan permanen  Balas



Re: Diskusi.5

oleh [ZIKRA RACHMAT SAPUTRA 052513267](#) - Senin, 3 November 2025, 06:51

Izin menjawab diskusi 5 ini ibu/bapak dosen

Penanganan kesalahan halaman (Page Fault):

1. CPU mengambil instruksi dari memori untuk dijalankan. Lakukan pengambilan instruksi dari halaman pada memori dengan mengakses tabel halaman.
2. Pada tabel halaman bit tersebut tidak valid. Terjadi interupsi kesalahan halaman, maka interupsi itu menyebabkan trap pada sistem operasi.
3. Jika referensi alamat yang diberikan ke sistem operasi ilegal atau dengan kata lain halaman yang ingin diakses tidak ada maka proses akan dihentikan. Jika Referensi legal maka halaman yang diinginkan diambil dari disk.
4. Halaman yang diinginkan dibawa ke memori fisik.
5. Mengatur ulang tabel halaman sesuai dengan kondisi yang baru. Jika tidak terdapat ruang di memori fisik untuk menaruh halaman yang baru maka dilakukan penggantian halaman dengan memilih salah satu halaman. Penggantian halaman dilakukan menurut algoritma tertentu yang akan dibahas pada bab selanjutnya. Jika halaman yang digantikan tersebut sudah dimodifikasi oleh proses maka halaman tersebut harus ditulis kembali ke disk.
6. Setelah halaman yang diinginkan sudah dibawa ke memori fisik maka proses dapat diulang.

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)

Re: Diskusi.5

oleh [SYAHRI WAHYUNI ABDUL RAZAK 056995024](#) - Senin, 3 November 2025, 10:06

Izin menjawab hasil diskusi pak tutor 

Pada gambar tersebut dijelaskan tentang alur penanganan kesalahan halaman atau page fault dalam sistem operasi. Proses ini terjadi ketika suatu program mencoba mengakses halaman memori yang belum berada di memori utama. Awalnya, ketika instruksi dijalankan, sistem akan melakukan referensi terhadap alamat logis dan memeriksa tabel halaman (page table). Jika ternyata halaman yang diakses tidak ditemukan di memori utama, maka terjadi page fault. Kondisi ini menyebabkan CPU melakukan trap ke sistem operasi untuk menangani kesalahan tersebut.

Selanjutnya, sistem operasi akan mencari lokasi halaman yang hilang di penyimpanan sekunder (secondary storage) seperti hard disk atau SSD. Setelah halaman ditemukan, sistem akan memilih salah satu ruang kosong (free frame) di memori utama untuk menampung halaman tersebut. Kemudian, halaman yang dibutuhkan akan dibawa masuk (bring in) dari secondary storage ke memori utama. Setelah proses pemindahan selesai, sistem operasi akan memperbarui tabel halaman dengan menandai bahwa halaman tersebut kini valid dan menunjukkan posisi frame tempat halaman disimpan.

Langkah terakhir adalah menjalankan kembali (restart) instruksi yang sebelumnya menyebabkan page fault. Karena halaman yang dibutuhkan kini sudah tersedia di memori utama, maka instruksi dapat dieksekusi dengan normal tanpa terjadi kesalahan lagi. Dengan demikian, proses page fault handling ini memastikan bahwa program tetap dapat berjalan meskipun halaman yang dibutuhkan belum ada di memori, karena sistem operasi secara otomatis mengelolanya agar tersedia sesuai kebutuhan.

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)

Re: Diskusi.5

oleh [NABIL HANIF ATHALLAH 055592508](#) - Senin, 3 November 2025, 22:39

Assalamualaikum Wr.Wb

Nama : Nabil Hanif Athallah

NIM : 055592508

Selamat malam Pak Ade Chandra Saputra S.Kom.,M.Cs dan teman-teman semuanya.

Saya akan menjawab pertanyaan dari diskusi 5, sebagai berikut :

Fase 1 : Deteksi dan Trap

- > Reference = Sebuah instruksi mencoba mengakses suatu alamat memori logis (alamat 35) yang berada di Page V.
- > Trap = Hardware memeriksa page table. Karena bit valid/invalid pada entri page table untuk halaman 35 disetel ke 'I' (Invalid) menandakan halaman tersebut tidak ada di memori fisik prosesor menghasilkan trap ke sistem operasi. OS mengambil alih kendali.

Fase 2 : Penentuan dan Pengambilan Halaman

- > Page on Secondary Storage = OS menentukan bahwa page yang dibutuhkan (halaman 35) memang ada dan sah, tetapi saat ini berada di Secondary Storage.
- > Bring in Missing Page : Os mencari free frame dari daftar yang tersedia. Setelah frame ditemukan, OS memulai operasi I/O untuk membawa halaman 35 dari secondary storage ke free frame yang telah dipilih di Physical Memory.

Fase 3 : Pembaruan dan Restart

- > Reset Page Table = Setelah operasi I/O selesai dan halaman 35 berhasil dimuat ke frame memori fisik, OS melakukan pembaruan pada Page Table dengan dua Langkah penting :
 - Mengubah bit valid/invalid pada entri halaman 35 dari 'I' menjadi 'V' (Valid).
 - Memperbarui entri page table dengan nomor frame memori fisik yang baru ditempati oleh halaman 35.
- > Restart Instruction = Setelah page table diperbarui, proses page fault selesai. OS mengembalikan kendali kepada proses yang terinterupsi, meminta prosesor untuk mengulang Kembali instruksi yang menyebabkan page fault (dalam contoh ini, instr 2 pada alamat 35). Kali ini, Ketika instruksi diulang, page table akan menunjukkan halaman tersebut valid, dan akses memori akan berhasil.

Sekian dari saya, Mohon maaf bila mana ada kesalahan.

terima kasih.

Wassalamualaikum Wr.Wb

SUMBER : - Buku Sistem Operasi (MSIM4201)

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)



Re: Diskusi.5

oleh [RIKO TAUFIQURROHMAN 052794504](#) - Selasa, 4 November 2025, 04:41

Berdasarkan gambar pada materi lampiran, alur penanganan page fault dapat dijelaskan sebagai berikut:

1. Referensi Memori (Reference):

CPU mencoba mengeksekusi sebuah instruksi, misalnya instr 1, yang mengakses alamat memori di halaman 34. Untuk memperoleh alamat fisik, CPU akan memeriksa page table.

2. Trap (Perangkap ke Sistem Operasi):

Saat memeriksa page table, CPU menemukan bahwa entri halaman 34 memiliki bit valid/invalid berstatus 'I' (Invalid). Ini menandakan halaman tersebut tidak ada di memori utama, sehingga CPU menghasilkan trap ke sistem operasi (OS).

3. Pengecekan Lokasi Halaman:

Sistem operasi mengambil alih kendali dan memeriksa tabel internal untuk mengetahui di mana halaman 34 berada. Ditemukan bahwa halaman tersebut tersimpan di secondary storage seperti hard disk.

4. Pemuatan Halaman (Bring in Missing Page):

OS mencari free frame yang kosong di memori utama. Setelah ditemukan, OS menjadwalkan operasi I/O untuk menyalin halaman 34 dari secondary storage ke dalam frame kosong tersebut. Proses ini relatif lambat karena membaca dari disk.

5. Pembaruan Page Table (Reset Page Table):

Setelah halaman berhasil dimuat ke memori fisik, OS memperbarui entri pada page table:

- Bit valid/invalid diubah dari 'I' menjadi 'V' (Valid).
- Nomor frame tempat halaman disimpan dicatat di tabel.

6. Restart Instruksi:

Setelah pembaruan selesai, kontrol dikembalikan ke proses semula. Instruksi yang sebelumnya menyebabkan page fault akan dijalankan ulang. Kali ini, halaman 34 sudah tersedia di memori utama, sehingga akses memori dapat berhasil dan program berjalan normal.

Kesimpulan:

Page fault bukan merupakan kesalahan fatal, melainkan bagian dari mekanisme virtual memory yang memastikan halaman baru dimuat ke RAM hanya saat dibutuhkan. Dengan demikian, sistem tetap efisien dan program dapat berjalan tanpa gangguan.

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)



Re: Diskusi.5

oleh [INDRI SUHERDI 056356352](#) - Selasa, 4 November 2025, 07:42

izin menjawab diskusi 5 pak

[Diskusi 5 SISTEM OPERASI.docx](#)

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)



Re: Diskusi.5

oleh [RAFA NURHUDAN WIGUNA 055747162](#) - Selasa, 4 November 2025, 20:51

Izin Menjawab Diskusi 5

1.CPU menginginkan instruksi 2 yaitu (35) terlihat di tabel nomor 35 ada tanda i yang artinya invalid artinya tidak terbaca

- 2.karena tidak terbaca sistem langsung ke OS untuk memberitahu bahwa ada masalah
- 3.OS memeriksa daftarnya dan memastikannya 35 ada tetapi letaknya yang berbeda
- 4.OS mencari di memori fisik
- 5.OS Pergi ke penyimpanan untuk membawa 35 yang sudah di siapkan
- 6.lalu diperbarui setelah itu 35 disimpan dipenyimpanan,dan os kembali ke tabel halaman
- 7.halaman 35 diubah dari i menjadi v dan di catat pada tempat yang diletakan
- 8.lalu di restart untul membaca ulang kembali semua
- 9.CPU mengulang permintaan yang tadi tidak di proses atau gagal
- 10.setelah CPU mengulang maka akan mendapatkan no 35 yang sudah benar atau valid dan sudah jelas tempat penyimpanannya

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)



Re: Diskusi.5

oleh [052869663 ANTENG FARIZATUL AWALLIA](#) - Selasa, 4 November 2025, 21:09

Nama : Anteng farizatul awallia

NIM : 052869663

1. Referensi alamat logis dan pemeriksaan halaman.

- Intruksi yang sedang dieksekusi (misalnya, instr 1) mencoba mengakses sebuah halaman diruang alamat logis (misalnya,halaman 35).
- Sistem perangkat keras (melalui unit manajemen memori/MMU) mencari entri untuk halaman 35 di tabel halaman (page table).
- Entri untuk halaman 35 ditemukan,tetapi bit valid/invalid(V) menunjukkan invalid misalnya, bit(V) untuk halaman 35 disetel ke (l) atau disini dilambangkan dengan kotak kosong/garis yang menandakan halaman tidak berada di memori fisik). Ini berarti halaman tersebut belum dimuat ke dalam memori fisik.

2. Trap ke sistem operasi (SO)

- karena halaman yang diperlukan tidak ada di memori fisik,hardware menghasilkan trap (interupsi) yang page fault.
- Trap ini mengalihkan kontrol eksekusi dari proses yang sedang berjalan ke sistem operasi.

3. Identifikasi lokasi halaman yang hilang

- OS mengambil alih dan memeriksa tabel internal (seringkali pada process control block atau tabel swap space) untuk menentukan lokasi halaman yang hilang (halaman 35) di penyimpanan sekunder(secondary storage).

4. Pencarian free frame dan pemuatian halaman

- OS kemudian mencari bingkai kosong (free frame) di memori fisik (physical memory) untuk menampung halaman 35.(Jika tidak ada free frame,Os akan menjalankan algoritma pengantian halaman untuk memilih bingkai "korban")
- OS menjadwalkan operasi I/O disk untuk membaw masuk halaman 35 yang hilang dari penyimpanan sekunder ke free frame yang ditemukan di memori fisik. Proses ini biasanya memerlukan waktu tunggu I/O yang cukup lama.

5. Perbarui Tabel Halaman

- Setelah operasi disk selesai dan halaman 35 berhasil dimuat ke dalam frame memori fisik. OS mengatur ulang/meperbarui tabel halaman.

- Entri untuk halaman 35 kini diisi dengan nomor frame barunya (36) ,dan bit valid/invalid (V) diubah menjadi valid.

6. Memulai ulang intruksi

- Setelah tabel halaman diperbarui OS memulai ulang intruksi yang sebelumnya menyebabkan page fault(yaitu instr 1 pada halaman 34 yang meminta data pada halaman 35)
- Karena halaman 35 sekarang sudah ada di memori fisik dan tabel halaman telah diperbarui,intruksu dapat diselesaikan dengan sukses.

Referensi :

BMP Sistem Operasi (MSIM4201)

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)



Re: Diskusi.5

oleh [TATA PUTRIA SALASABILA 057204874](#) - Selasa, 4 November 2025, 22:49

Assalamualaikum Wr.Wb

Nama: Tata Putria Salsabila

NIM: 057204874

Prodi Sistem Informasi

UPB JJ-UT Surakarta

Izin menjawab Diskusi 5

Deskripsi Alur Penanganan Kesalahan Halaman (Page Fault).

Gambar tersebut menunjukkan bagaimana sistem operasi menangani page fault, yaitu kondisi ketika halaman (page) yang dibutuhkan oleh CPU tidak tersedia di memori utama (RAM), melainkan masih berada di secondary storage (misalnya hard disk). Proses ini berlangsung melalui beberapa langkah sebagai berikut:

1. Reference (Akses Halaman)

- CPU mencoba mengakses suatu alamat logis (logical address space) pada memori virtual, misalnya instruksi ke-2 yang berada pada halaman 35.
- Sistem kemudian memeriksa page table untuk melihat apakah halaman tersebut ada di memori fisik.

2. Trap (Terjadi Page Fault)

- Jika bit valid (V) di page table bernilai "invalid", artinya halaman tersebut tidak ada di memori utama.
- Terjadi trap ke sistem operasi (OS), yaitu interupsi yang memberi tahu OS bahwa page fault terjadi.

3. Page is on Secondary Storage

- Sistem operasi memeriksa lokasi halaman yang dibutuhkan pada secondary storage (misalnya hard disk atau SSD).
- OS memastikan bahwa halaman tersebut benar-benar ada dan dapat dipanggil ke memori.

4. Bring in Missing Page

- OS memilih frame kosong di memori fisik (free frame) untuk menyimpan halaman yang hilang.
- Halaman yang dibutuhkan kemudian dibawa dari secondary storage ke dalam frame kosong di memori fisik.

5. Reset Page Table

Setelah halaman berhasil dimuat ke memori utama, OS memperbarui entri di page table, dengan mengubah bit valid (V) menjadi "valid" dan mengatur nomor frame fisik tempat halaman disimpan.

6. Restart Instruction

- Setelah semua langkah di atas selesai, instruksi yang menyebabkan page fault diulang kembali (restart instruction).
- Kali ini, CPU dapat mengakses halaman tersebut secara langsung dari memori fisik tanpa kesalahan.

Dengan proses tersebut, sistem operasi dapat secara dinamis memuat halaman yang dibutuhkan dari secondary

storage ke memori utama hanya saat diperlukan, yang disebut demand paging. Hal ini membuat pemanfaatan memori lebih efisien dan memungkinkan program besar dijalankan pada memori terbatas.

Referensi:

- Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). Operating System Concepts (10th ed.). Wiley.
- Stallings, W. (2018). Operating Systems: Internals and Design Principles (9th ed.). Pearson Education.
- Sibero, A. P. (2013). Sistem Operasi: Konsep Dasar dan Aplikasinya. Andi Offset.
- Tanenbaum, A. S., & Bos, H. (2015). Modern Operating Systems (4th ed.). Pearson.

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)

Hide sidebar

Re: Diskusi.5

oleh [RIZKY DONI IKRAR HIDAYATULLAH SYUKUR 055608624](#) - Rabu, 5 November 2025, 00:02

Selamat Malam Bapak/Ibu tutor saya Rizky D.I.H Syukur dengan NIM 055608624 izin memberikan tanggapan mengenai materi diskusi diatas

Berdasarkan gambar dibawah maka alur jalannya instruksi adalah sebagai berikut

CPU mencoba untuk menjalankan instruksi yang Alamat logisnya berada pada halaman 35 (Page 35). Setelah dicek dengan Alamat logisnya pada table halaman (Page Table), sistem menemukan entri untuk halaman 35 namun bit validitas pada entri tersebut bernilai 0 ditandai dengan adanya 'l' untuk invalid yang menunjukkan bahwa halaman tersebut tidak ditemukan pada memori fisik dan berada dipenyimpanan sekunder. Karena bit validitas yang berstatus invalid, hardware menciptakan trap yang menginterupsi dan menghentikan eksekusi program dan control diserahkan penuh kepada sistem operasi. Setelah menerima trap sistem operasi segera memulai penanganan page fault. Pada bagian ini OS harus mencari halaman 35 sebagai halaman yang hilang. Dan setelah ditemukan pada secondary storage yang biasanya terdapat pada hardisk OS kemudian memeriksa memori fisik untuk menemukan memori kosong sebagai tempat untuk memuat memori yang hilang. Setelah mendapatkan memori kosong, OS akan memuat halaman tersebut dari secondary storage ke dalam memori kosong tadi, lalu memperbaiki entri di table halaman untuk halaman 35 dengan cara menulis nomor frame fisik baru Dimana halaman 35 berada dan juga mengubah bit validitas yang awalnya bernilai 'l' untuk invalid menjadi 'v' untuk valid dan menandakan bahwa halaman tersebut kini tersedia pada memori fisik. Setelah semua proses tersebut selesai langkah akhir yang akan dilakukan oleh OS adalah mengembalikan control ke CPU yang akan merestart instruksi yang menyebabkan page fault. Dengan informasi yang telah diperbaharui pengecekan page table akan menemukan bit validitas 'v' dan instruksi akan berhasil dieksekusi

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)



Re: Diskusi.5

oleh [MUHAMMAD AQSHAL ZAKKI 055457214](#) - Rabu, 5 November 2025, 06:00

Penjelasan Alur Page Fault Handling dari materi gambar yang dilampirkan:

1. Reference (Referensi Instruksi di Logical Address Space):

CPU atau program mencoba mengakses instruksi, misal instruksi 2 di page 35. Proses membutuhkan data dari page yang belum ada di memori fisik.

2. Trap (Page Fault):

Karena page 35 belum tersedia di memori fisik, terjadi page fault sehingga OS menerima "trap" (interupsi) untuk menangani situasi tersebut.

3. Page is on Secondary Storage (Cek di Secondary Storage):

OS mencari page 35 di secondary storage (seperti hard disk atau SSD).

4. Bring in Missing Page (Membawa Page yang Hilang ke Memori):

OS memilih free frame (slot kosong di memori fisik), lalu membawa page 35 dari storage ke slot tersebut di physical memory.

5. Reset Page Table (Update Page Table):

Setelah page 35 masuk ke memori fisik, OS memperbarui page table agar page 35 di-mark sebagai "valid", dan menuliskan alamat frame fisiknya.

6. Restart Instruction (Eksekusi Ulang Instruksi):

Instruksi yang sebelumnya gagal (karena page fault) diulang—sekarang bisa berjalan normal karena page sudah tersedia di memori.

Inti Proses:

Page Fault: Terjadi ketika page yang ingin diakses tidak ada di memori fisik.

Penanganan: OS akan mengambil page dari secondary storage lalu update page table dan mengulang instruksi yang gagal.

Manfaat: Efisiensi penggunaan RAM dan bisa menjalankan program yang lebih besar dari kapasitas memori fisik dengan konsep virtual memory.

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)

Re: Diskusi.5

oleh [RAFAEL GUEVARA TARIGAN 053739643](#) - Rabu, 5 November 2025, 09:27

Ketika CPU mau akses data atau instruksi dan ternyata halamannya belum ada di memori utama, sistem ngalamin page fault. Sistem operasi dapet sinyal, terus nyari halaman yang dibutuhin di secondary storage. Setelah ketemu, halaman itu dimasukin ke frame kosong di memori fisik dan page table diperbarui biar tandanya valid. Terakhir, CPU ngulang instruksi sebelumnya, dan proses lanjut lagi tanpa gangguan.

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)

Re: Diskusi.5

oleh [BAYU DWI SAPUTRA 054794681](#) - Rabu, 5 November 2025, 10:32

Alur Page Fault

1. OS (Sistem Operasi) mengeksekusi instr 1 yang merujuk ke sebuah alamat logis. Proses ini disebut "reference" (langkah 1).
2. Sistem Operasi memeriksa page table dan menemukan bahwa halaman yang diminta (misalnya, page 34) tidak ada di memori fisik (bit valid/invalid menunjukkan "invalid"). Kondisi ini menyebabkan "trap" (interupsi) yang dikenal sebagai page fault (langkah 2).
3. Setelah trap terjadi, OS memeriksa secondary storage (misalnya, hard disk) untuk memastikan bahwa page yang diminta (page 34) memang ada di secondary storage (langkah 3).
4. OS menemukan free frame (bingkai kosong) di physical memory. Kemudian, OS memuat page 34 dari secondary storage ke dalam free frame tersebut di memori fisik (langkah 4).
5. Setelah page berhasil dimuat, OS memperbarui (reset) page table untuk menandai bahwa page 34 sekarang sudah berada di memori fisik dan mengatur bit valid/invalid menjadi "valid" (langkah 5).
6. OS merestart instruction yang sebelumnya menyebabkan page fault, yaitu instr 1. Kini, instruksi dapat dijalankan dengan normal karena page-nya sudah tersedia di memori (langkah 6).
7. Setelah instr 1 selesai dijalankan, proses berlanjut ke instr 2 yang berada pada page 35 (disebutkan: "next instruction is instr 2 on page 35").

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)

Re: Diskusi.5

oleh [ZIDAN MUHAMAD DAFFA 056388169](#) - Rabu, 5 November 2025, 17:21

Izinkan saya menanggapi topik diskusi 5 sebagai berikut:

1. Referensi Alamat Logik (Reference)

CPU mengambil instruksi dari memori untuk dijalankan melalui logical address space. Instruksi 1, 2, dan 3 masing-masing menunjukkan nomor page 34, 35, dan 36 dengan bit valid-invalid yang ditunjukkan pada page table.

2. Trap Kesalahan Halaman

Ketika CPU mengakses page table, bit valid-invalid diset tidak valid (bernilai 0). Terjadi interupsi kesalahan halaman yang menyebabkan trap pada sistem operasi.

3. Halaman Berada di Secondary Storage

Jika referensi alamat yang diberikan legal, maka halaman yang diinginkan diambil dari disk (secondary storage).

Sistem operasi memeriksa bahwa referensi adalah legal dan menentukan lokasi halaman pada disk.

4. Membawa Halaman yang Hilang ke Memori Fisik

Halaman yang diinginkan dibaca dari secondary storage dan dibawa ke free frame di physical memory (memori fisik).

5. Mengatur Ulang Tabel Halaman (Reset Page Table)

Tabel halaman diatur ulang sesuai dengan kondisi baru. Bit valid-invalid untuk halaman yang baru dibawa diubah menjadi valid (bernilai 1) dan page table entry diperbarui agar menunjuk ke frame yang benar di physical memory.

6. Pengulangan Instruksi (Restart Instruction)

Setelah halaman yang diinginkan sudah berada di memori fisik, proses dapat diulang. Instruksi yang sebelumnya mengalami kesalahan halaman dijalankan kembali dengan normal. Pada contoh gambar, instruksi yang akan dijalankan berikutnya adalah instr 2 pada page 35.

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)



Re: Diskusi.5

oleh [053960862 MUHAMMAD DIMAS LUTFI ABDILLAH](#) - Rabu, 5 November 2025, 21:04

Ijin menjawab diskusi 5

NAMA : Muhammad Dimas Lutfi Abdillah

NIM : 053960862

MATKUL : Sistem Operasi

Berikut adalah deskripsi alur penanganan Page Fault (Kesalahan Halaman) sesuai dengan urutan yang umum terjadi:

Alur Penanganan Page Fault

1. OS Mengeksekusi Instruksi

- o CPU menjalankan instruksi (misal: instr 1, instr 2, instr 3).
- o Setiap instruksi mengacu pada alamat logis (virtual) melalui page table.

2. Terjadi Page Fault

- o Saat CPU mencoba mengakses halaman (misal: halaman untuk instr 2), page table mencatat bahwa halaman tersebut tidak valid (bit valid = 0) atau tidak berada di memori fisik (physical memory).
- o Halaman yang dibutuhkan ternyata masih berada di secondary storage (misal: hard disk).

3. Interupsi ke Sistem Operasi

- o Terjadi trap (interupsi) yang disebut page fault.
- o Kontrol beralih ke sistem operasi untuk menangani kesalahan ini.

4. Cari Frame Kosong

- o OS mencari free frame (bingkai kosong) di memori fisik.
- o Jika tidak ada frame kosong, OS akan menjalankan algoritma penggantian halaman (page replacement) untuk mengosongkan satu frame.

5. Baca Halaman dari Secondary Storage

- o OS mengirim permintaan ke subsistem I/O untuk membaca halaman yang hilang (missing page) dari secondary storage ke dalam frame yang telah disediakan.
- o Proses ini mungkin membutuhkan waktu lama, sehingga OS dapat mengganti proses lain ke CPU sambil menunggu.

6. Update Page Table

- o Setelah halaman berhasil dimuat ke memori fisik, OS memperbarui page table:

- Menandai halaman tersebut sebagai valid.

- Mencatat nomor frame fisik tempat halaman tersebut disimpan.
- 7. Restart Instruksi
 - o OS mengembalikan kontrol ke proses yang terganggu.
 - o Instruksi yang sebelumnya gagal (instr 2) diulang (restart instruction).
 - o Sekarang akses ke halaman tersebut sudah valid, sehingga instruksi dapat dilanjutkan ke instr 3 dan seterusnya.

Ringkasan Tahapan:

1. Referensi halaman → tidak ada di memori (page fault).
2. Interupsi ke OS.
3. Cari frame kosong.
4. Baca halaman dari disk.
5. Update page table.
6. Restart instruksi yang gagal tadi.

Dengan demikian, proses dapat terus berjalan tanpa crash meskipun awalnya data yang diakses belum tersedia di RAM.

Sumber Referensi : BMPADPU4341 / Modul 5 / Hal 5.4-5.18

Silberschatz, A., Galvin, P.B., & Gagne, G. (2018). Operating System Concepts. Edisi 10. Wiley.

Tanenbaum, A.S. & Bos, H. (2015). Modern Operating Systems. Edisi 4. Pearson.

Stallings, W. (2018). Operating Systems: Internals and Design Principles. Edisi 9. Pearson.

Arpaci-Dusseau, R.H. & Arpaci-Dusseau, A.C. (2018). Operating Systems: Three Easy Pieces.

https://en.wikipedia.org/wiki/Page_fault

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)

Re: Diskusi.5

oleh [UMARETA RIZKY SHABRINA 056505332](#) - Rabu, 5 November 2025, 22:54

Salam Sejahtera Bapak/Ibu Tutor izin menjawab hasil Diskusi sesi 5 saya

1. Akses Memori dan Pengecekan (Reference):

CPU menjalankan instruksi (misalnya inst 1 pada alamat logis di page 34). MMU (Memory Management Unit) menerjemahkan alamat logis ini menjadi alamat fisik dengan menggunakan Page Table. Pada proses penerjemahan ini, ditemukan bahwa bit valid/bit keberadaan (present bit) untuk halaman yang diminta (page 35) bernilai invalid. Ini menandakan bahwa halaman yang diperlukan tidak berada di dalam Memori Fisik (Physical Memory). Kondisi inilah yang disebut sebagai Page Fault.

2. Trap ke Sistem Operasi:

Terjadinya Page Fault menyebabkan CPU menghentikan eksekusi program yang sedang berjalan. Perangkat keras kemudian memicu interupsi (exception) yang disebut Trap. Kontrol eksekusi kemudian dialihkan dari program pengguna ke rutin penanganan Page Fault yang berada di dalam Sistem Operasi (OS).

3. Pengecekan Lokasi Halaman:

Sistem Operasi memeriksa status dari halaman yang diminta (page 35) dalam sebuah struktur data yang biasanya disebut Page Table. Sistem Operasi menentukan bahwa halaman ini memang valid (bagian dari ruang alamat logis program), tetapi saat ini berada di Secondary Storage (penyimpanan sekunder, seperti disk atau SSD), dan bukan di memori fisik.

4. Mengambil Halaman dari Penyimpanan Sekunder:

Sistem Operasi kemudian mengidentifikasi lokasi pasti dari page 35 pada secondary storage. Ia mengeluarkan permintaan baca (read request) ke subsistem I/O untuk membawa halaman yang hilang (bring in missing page) dari disk ke dalam memori fisik. Proses ini mungkin melibatkan penjadwalan operasi I/O dan menunggu data tersedia, yang dapat menyebabkan proses tersebut ditunda (blocked) atau dialihkan (context switch) ke proses lain.

5. Memperbarui Page Table:

Setelah halaman dari disk berhasil dimuat ke dalam sebuah free frame (bingkai memori fisik yang kosong) di memori fisik, Sistem Operasi memperbarui Page Table (reset page table). Entri untuk page 35 dalam Page Table sekarang diatur menjadi valid, dan nomor bingkai (frame number) yang baru saja ditempati dituliskan ke dalam entri tersebut.

6. Melanjutkan Eksekusi Instruksi:

Setelah struktur data diperbarui, Sistem Operasi mengembalikan kontrol eksekusi ke proses yang semula mengalami Page Fault. Instruksi yang sebelumnya gagal (inst 1 pada page 34) di-restart dari awal. Kali ini, ketika MMU kembali menerjemahkan alamat untuk inst, pencarian di Page Table akan berhasil karena halaman yang diperlukan (page35) sudah berada di memori fisik. Eksekusi program kemudian dapat berlanjut ke instruksi berikutnya (inst 2 pada page 35).

REFERENSI

Sibero, Alexander F. K. (2013). *Jaringan Komputer dan Pertukaran Data*. Penerbit Media Komputindo. Halaman 45-47 membahas konsep manajemen memori virtual. [Link verifikasi publik](https://opac.perpusnas.go.id/DetailOpac.aspx?id=910260): <https://opac.perpusnas.go.id/DetailOpac.aspx?id=910260>

Sibero, Alexander F. K. (2011). *Jaringan Komputer & Komunikasi Data*. Penerbit Mitra Wacana Media. Halaman 88-90 menjelaskan tentang mekanisme page fault dan peran sistem operasi. [Link verifikasi publik](https://opac.perpusnas.go.id/DetailOpac.aspx?id=788999): <https://opac.perpusnas.go.id/DetailOpac.aspx?id=788999>

Sibero, Alexander F. K. (2013). *Interkoneksi Jaringan Komputer*. Penerbit Media Komputindo. Halaman 112-115 mendeskripsikan alur kerja penanganan kesalahan pada sistem memori. [Link verifikasi publik](https://opac.perpusnas.go.id/DetailOpac.aspx?id=1168965): <https://opac.perpusnas.go.id/DetailOpac.aspx?id=1168965>

Modul MSIM4201 Sistem Operasi

Video pada Materi Pengayaan

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)



Re: Diskusi.5

oleh [RADEN AGUNG PRAKOSO WIBOWO 052866944](#) - Kamis, 6 November 2025, 16:47

Selamat sore

Nama : Raden Agung Prakoso Wibowo

NIM ; 052866944

Izin memberikan opini terkait pertanyaan dalam diskusi ini:

Penanganan *Page Fault* adalah salah satu mekanisme vital dalam implementasi memori virtual. Kejadian ini merujuk pada situasi di mana sebuah proses mencoba mengakses halaman memori logis, namun halaman tersebut belum dimuat ke dalam memori fisik (*physical memory*). Proses penanganan ini melibatkan serangkaian langkah terstruktur antara perangkat keras, *Page Table*, dan Sistem Operasi (OS).

Sebagai penanganannya sbb :

1. Instruksi dan Referensi Memori (*Instruction and Memory Reference*)

Proses yang sedang berjalan di CPU mencoba mengeksekusi sebuah instruksi, misalnya **instr 1**. Instruksi ini memerlukan akses ke suatu alamat logis dalam *Logical Address Space* (Ruang Alamat Logis). Permintaan akses ini dikenal sebagai **reference** (referensi).

2. Pemicu Trap

CPU menggunakan **Page Table** (Tabel Halaman) untuk menerjemahkan alamat logis yang diminta (instr 1) menjadi alamat fisik. Saat OS memeriksa entry yang sesuai di dalam Page Table, ia menemukan bahwa entry tersebut ditandai sebagai **invalid** (tidak sah). Tanda invalid ini mengindikasikan bahwa halaman yang dimaksud belum ada di dalam memori fisik (yaitu, berada di secondary storage). Penemuan status invalid ini segera memicu sebuah **trap** (jebakan atau interupsi) yang dialihkan langsung ke Sistem Operasi (OS).

3. Identifikasi dan Lokasi Halaman

Setelah trap terjadi, OS mengambil alih kendali. OS pertama-tama menyimpan state (status) CPU saat itu untuk memastikan proses yang terinterupsi dapat dilanjutkan. Selanjutnya, OS mengidentifikasi sumber trap dan mengonfirmasi bahwa halaman yang dibutuhkan (page is on secondary storage) memang berada di dalam penyimpanan sekunder (hard disk atau swap space). OS kemudian menentukan lokasi persis dari halaman yang hilang tersebut.

4. Alokasi Bingkai dan Operasi I/O

Untuk membawa halaman yang hilang ke memori fisik, OS harus menemukan ruang kosong. OS memeriksa daftar **free frame** (bingkai kosong) yang tersedia di *Physical Memory* dan mengalokasikan satu *frame* kosong untuk halaman tersebut. Setelah *frame* dialokasikan, OS memulai operasi I/O untuk **bring in missing page** (membawa masuk halaman yang hilang) dari penyimpanan sekunder ke *frame* memori fisik yang baru. Operasi I/O ini biasanya merupakan proses yang lambat.

5. Pembaruan Tabel Halaman

Setelah operasi I/O selesai dan halaman berhasil dimuat ke *Physical Memory*, OS melakukan pembaruan krusial pada Page Table. *Entry* yang sesuai di dalam Page Table diatur ulang (reset page table):

- Status *invalid* diubah menjadi **valid**.
- Nomor *frame* baru tempat halaman dimuat (misalnya, *frame* 35) dicatat ke dalam *entry* tersebut, sehingga Page Table kini dapat memetakan alamat logis ke alamat fisik yang benar.

6. Eksekusi Ulang Instruksi

Langkah terakhir adalah mengembalikan kendali ke proses yang terinterupsi. Karena alamat logis kini sudah valid, OS memungkinkan CPU untuk menjalankan kembali (restart instruction)

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)



Re: Diskusi.5

oleh [A. GANI AJI SAPUTRA 056165787](#) - Kamis, 6 November 2025, 21:22

Izin Menjawab Diskusi :

Alur Penanganan Page Fault :

1. Referensi Instruksi ke Alamat Logis

CPU mengeksekusi instruksi dan melakukan referensi ke alamat logis, misalnya instruksi 2 pada page 35.

2. Page Table Trap (Terjadi Page Fault)

Sistem memeriksa page table, ternyata entry untuk page 35 tidak valid (biasanya bit valid di page table bernilai 0). OS menerima indikasi adanya page fault.

3. OS Memeriksa di Sekunder Storage
OS menemukan bahwa page yang dibutuhkan (page 35) berada di secondary storage (disk).
4. Bring in Missing Page (Mengambil Data dari Disk)
OS mencari free frame di physical memory, kemudian membawa (menyalin) page 35 dari secondary storage ke free frame dalam memory fisik.
5. Reset Page Table
Page table di-update, page 35 sekarang valid dan mengarah ke frame baru di physical memory.
6. Restart Instruction
Setelah page table diperbarui dan proses pemindahan selesai, instruksi yang sebelumnya gagal dieksekusi karena page fault dijalankan ulang (restart instruction).

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)

Re: Diskusi.5

oleh [YUSUF FAJRI KURNIANTO 055178877](#) - Kamis, 6 November 2025, 21:40

Mohon izin untuk menjawab diskusi 5

1. Reference
 - CPU mencoba mengeksekusi instruksi berikutnya yaitu "instr2" yang berada pada halaman 35 di ruang alamat logis.
 - CPU mencari entri untuk halaman 35 di tabel halaman.
 - Pada tabel halaman bit valid/invalid untuk halaman 35 disetel ke "i" yang berarti halaman tersebut belum dimuat ke memori fisik.
2. Trap
 - Karena halaman tersebut ditandai sebagai "invalid", ini memicu trap ke sistem operasi.
3. Halaman ada di penyimpanan sekunder
 - OS mengambil alih dan menyadari bahwa halaman yang diminta berada di penyimpanan sekunder atau disk.
4. Bawa halaman yang hilang
 - OS memulai operasi I/O untuk mengambil halaman 35 dari penyimpanan sekunder.
 - Untuk memuatnya, OS harus menemukan sebuah frame kosong di memori fisik. Jika tidak ada frame kosong, OS mungkin harus memilih frame yang sudah terisi untuk diganti.
 - Halaman 35 dimuat ke dalam frame kosong yang telah ditemukan memori fisik.
5. Setel ulang tabel halaman
 - Setelah halaman 35 berhasil dimuat ke memori fisik, OS memperbarui tabel halaman.
 - OS mengisi alamat frame fisik yang baru di entri halaman 35.
 - Bit valid/invalid diubah dari "i" menjadi "v"
6. Mulai ulang instruksi
 - OS mengembalikan kontrol ke CPU.
 - Instruksi yang menyebabkan page fault diulang. Kali ini, ketika CPU mereferensikan halaman 35, tabel halaman akan menunjukkan bahwa halaman itu valid dan berada di memori fisik, kemungkinan eksekusi instruksi dilanjutkan.

Terima kasih

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)

Re: Diskusi.5

oleh [NAJWA SALSABILA AZZAHRA 057245181](#) - Jumat, 7 November 2025, 14:21

menurut saya berdasarkan gambar yang ada di atas penanganan kesalahan halaman (Page Fault) yaitu adalah sebagai berikut

1. CPU yaitu mengambil intruksi dari memori untuk dijalankan. Lakukan pengambilan intruksi dari halaman pada memori dengan cara mengakses tabel pada suatu halaman.
2. Pada tabel halaman bit tersebut tidak valid. Hal ini lah yang bisa menyebabkan terjadinya interupsi kesalahan halaman, maka interusi itu menyebabkan trap pada suatu sistem operasi.
3. Jika reversi alamat yang diberikan ke sistem operasi ini ilegal atau bahkan dengan menggunakan kata lain

halaman yang di akses tidak ada maka akan membuat suatu proses dapat dihentikan dan jika referensi yang di ambil itu legal maka halaman tersebut bisa diambil

4. Jika ada halaman yang diinginkan dibawa ke memori fisik

5. Selalu mengatur ulang tebel halaman sesusi dngan kondisi yang baru. Jika tidak ada terdapat ruang disuatu memori maka untuk menaruh halaman yang baru diganti dengan memilih halaman yang lain atau yang baru

6. Setelah halaman yang diinginkan sudah tersimpan pada memori fisik, maka suatu proses dapat diulang

Berikut adalah hal yang akan terjadi jika salah dalam mengurutkan halaman

1. Ditangkap oleh sistem operasi

2. Menyimpan register pengguna dan proses

3. tetapkan bahwa interupsi bahwa suatu kesalahan halaman

4. periksa bahwa referensi halaman yang legal dan tentukan juga lokasi halaman pada disk

sekian hasil diskusi saya izin bapak/ibuk jika ada kesalahan dalam pemahaman saya mohon kereksinya
saya mendapatkan jawaban ini dari materi yang diberikan dan dari buku bahan ajar yang diberikan sekian terimakasih

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)

Re: Diskusi.5

oleh [ZIBRAN DIRZA ARGADEWA 057030395](#) - Jumat, 7 November 2025, 20:44

Nama : Zibran Dirza Argadewa

NIM : 057030395

Pertama, prosesor mencoba mengakses sebuah instruksi di alamat logis tertentu, misalnya halaman 35. Namun ternyata halaman tersebut belum ada di memori utama, sehingga sistem mendeteksi page fault dan mengirimkan sinyal trap ke sistem operasi.

Setelah menerima trap, sistem operasi memeriksa lokasi halaman yang hilang, dan menemukan bahwa halaman tersebut ada di secondary storage seperti hard disk. OS kemudian mencari frame kosong di memori fisik, lalu menyalin atau memuat halaman yang hilang dari secondary storage ke memori utama.

Begini proses pemuatan selesai, OS memperbarui page table agar menunjukkan bahwa halaman tersebut sekarang valid dan sudah berada di memori. Terakhir, sistem operasi menginstruksikan prosesor untuk mengulangi instruksi yang sebelumnya menyebabkan page fault. Kali ini instruksi dapat berjalan normal karena halaman sudah tersedia di RAM.

Jadi, kesimpulannya adalah proses page fault melibatkan enam langkah utama: referensi alamat, trap ke OS, pengecekan halaman di penyimpanan sekunder, pemuatan halaman ke memori, pembaruan page table, dan pengulangan instruksi.

SUMBER REFERENSI

SONASA RINUSANTORO DKK. (2025). SISTEM OPERASI MSIM4201/MODUL 5 5.22-5.23. UNIVERSITAS TERBUKA.

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)

Re: Diskusi.5

oleh [SUNU WINDU HARSONO 056007432](#) - Jumat, 7 November 2025, 22:18

Mohon izin menjawab :

Langkah-langkah Alur Instruksi (Proses Page Fault Handling):

1. Reference (Referensi ke Halaman)

Prosesor mengeksekusi instruksi yang memerlukan akses ke suatu halaman memori.

Pada contoh di gambar, instruksi berikutnya adalah instruksi 2 yang berada pada halaman (page) 35 di logical address space.

Prosesor mencoba mengakses page 35 melalui page table.

2. Trap ke Sistem Operasi (Page Fault Terjadi)

Ketika prosesor memeriksa page table, ternyata bit valid/invalid (v/i) untuk page 35 bernilai invalid (i).

Artinya halaman tersebut tidak ada di memori utama (physical memory).

Akibatnya, terjadi page fault, dan prosesor melakukan trap ke sistem operasi (OS) untuk menanganiinya.

3. OS Mengecek Lokasi Halaman di Secondary Storage

Sistem operasi kemudian memeriksa di secondary storage (misalnya hard disk) untuk mengetahui apakah halaman 35 memang tersimpan di sana.

Jika ditemukan, OS mendekati bahwa halaman tersebut tersedia untuk dimuat ke memori utama.

4. Membawa Halaman yang Hilang ke Memori (Bring in Missing Page)

Sistem operasi kemudian mencari free frame (ruang kosong) di memori utama (physical memory).

Jika ada frame kosong, halaman 35 dari secondary storage dibaca dan dipindahkan ke frame kosong tersebut di memori utama.

5. Reset Page Table (Memperbarui Tabel Halaman)

Setelah halaman 35 berhasil dimuat ke memori utama, page table diperbarui:

- Bit valid/invalid diubah menjadi valid (v).
- Nomor frame fisik tempat halaman itu berada dicatat dalam page table.

Dengan begitu, halaman tersebut kini siap diakses oleh CPU tanpa menimbulkan page fault lagi.

6. Restart Instruction (Melanjutkan Instruksi yang Terhenti)

Sistem operasi kemudian mengembalikan kontrol ke prosesor untuk mengulang eksekusi instruksi yang sempat terhenti.

Instruksi 2 pada halaman 35 kini bisa dijalankan karena halaman tersebut sudah tersedia di memori utama.

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)



Re: Diskusi.5

oleh [054625452 SIWI PUSPITASARI](#) - Sabtu, 8 November 2025, 10:11

izin menanggapi diskusi ini

Saat sebuah program dijalankan, prosesnya mengakses data atau instruksi yang disimpan dalam memori utama (RAM). Namun, jika data atau halaman yang diminta ternyata belum ada di RAM, maka terjadilah page fault. Berikut alur terjadinya page fault berdasarkan gambar tersebut :

1). Reference and Trap : Program mencoba mengakses sebuah halaman misalnya instruksi nomor 2 yang berada di halaman 35. CPU akan memeriksa page table untuk memastikan apakah halaman tersebut ada di memori utama. Ternyata halaman 35 belum ada di memori, sehingga terjadi page fault. CPU kemudian mengirimkan sinyal ke sistem operasi (OS) untuk menangani masalah ini.

2). Check Secondary Storage : Sistem operasi memeriksa penyimpanan sekunder seperti hard disk dan menemukan bahwa halaman 35 memang tersimpan di sana.

3). Bring in Missing Page : OS kemudian mencari ruang kosong di memori utama (free frame), lalu memindahkan halaman 35 dari penyimpanan sekunder ke memori utama agar bisa diakses dengan cepat oleh CPU.

4). Reset Page Table : Setelah halaman berhasil dimasukkan ke memori, sistem operasi memperbarui page table. Tanda halaman 35 yang sebelumnya tidak valid kini berubah menjadi valid karena sudah berada di memori.

5). Restart Instruction : Setelah semua langkah selesai, program dilanjutkan kembali dan instruksi yang sebelumnya menyebabkan page fault dijalankan ulang tanpa masalah, karena data yang dibutuhkan sudah tersedia di memori utama.

Kesimpulan :

Page fault terjadi ketika program meminta data yang belum ada di memori utama. Sistem operasi kemudian mengambil data itu dari hard disk, memasukkannya ke memori, memperbarui tabel halaman, lalu melanjutkan program. Proses ini membuat sistem dapat berjalan efisien meskipun tidak semua data program dimuat ke memori secara bersamaan.

1). Sumber Referensi :

* BMP Sistem Operasi (MSIM4201) Modul 5 Halaman 5.22-5.23

* <https://translate.google.com/translate?u=https://www.geeksforgeeks.org/operating-systems/page-fault-handling-in-operating-system/&hl=id&sl=en&tl=id&client=sge#:~:text=Page%20fault%20terjadi%20ketika%20suatu,penyimpanan%20sekur>
diakses pada tanggal 8 November 2025.

Tautan permanen Tampilkan induk Balas



Re: Diskusi.5

oleh [SARAH HANIFAH 053132687](#) - Sabtu, 8 November 2025, 10:29

Selamat siang pak Ade Chandra Saputra,
Saya izin menjawab diskusi terlampir.

Menurut Rinusantoro dkk. (2025), proses penanganan page fault melibatkan beberapa langkah penting, yaitu:

1. Eksekusi instruksi: Instruksi pertama diambil dari memori dan dieksekusi. Instruksi tersebut adalah instruksi 1 yang memiliki alamat 34.
2. Page fault: Saat instruksi dieksekusi, sistem mendeteksi bahwa halaman yang dibutuhkan tidak tersedia di memori utama (missing page). Hal ini menyebabkan terjadinya page fault dan proses penanganan halaman dimulai.
3. Pengambilan halaman dari penyimpanan sekunder: Sistem mengidentifikasi bahwa halaman yang dibutuhkan ada di penyimpanan sekunder (secondary storage). Sistem kemudian mengakses penyimpanan sekunder dan membawa halaman yang hilang ke memori utama.
4. Pengisian halaman yang hilang ke memori: Halaman yang hilang tersebut dimuat ke dalam memori fisik (bring in missing page). Proses ini dilakukan melalui pengisian page frame yang kosong, dan halaman yang diambil dari secondary storage digantikan ke memori utama.
5. Reset tabel halaman: Setelah halaman berhasil dimuat, tabel halaman (page table) diperbarui dengan entri baru yang menunjukkan bahwa halaman sekarang ada di memori utama. Sistem juga mengatur page table untuk merekam lokasi halaman yang di-load.
6. Instruksi dilanjutkan: Setelah halaman dimuat dan tabel halaman diperbarui, instruksi yang sebelumnya gagal dieksekusi kembali. Instruksi berikutnya adalah instruksi 2 yang berada di halaman 35.
7. Instruksi berhasil dieksekusi: Instruksi yang sebelumnya gagal dieksekusi karena page fault sekarang dapat dijalankan dengan lancar dari memori utama.

Referensi:

Rinusantoro, dkk. (2025). Sistem Operasi. Universitas Terbuka: Tangerang Selatan.

Tautan permanen Tampilkan induk Balas



Re: Diskusi.5

oleh [RIVALINO SHEVA ANDREAN 052541771](#) - Sabtu, 8 November 2025, 11:42

Selamat Siang Bapak/ Ibu Dosen,

Izin Menjawab untuk diskusi 5 :

Ketika CPU menjalankan sebuah program, ia mengakses instruksi melalui **alamat logis** yang direferensikan ke **page table**. Misalnya, CPU ingin mengeksekusi instruksi ke-2 yang berada pada halaman 35. Proses dimulai dengan memeriksa apakah halaman tersebut tersedia di memori fisik.

1. Referensi ke Page Table

CPU memeriksa **page table** untuk melihat apakah halaman 35 sudah dimuat ke memori utama. Jika entri pada page table menunjukkan status **invalid**, berarti halaman tidak ada di memori.

2. Terjadi Page Fault dan Trap ke OS

Karena halaman tidak ditemukan, CPU mengirimkan sinyal **trap** ke sistem operasi. Ini menandakan bahwa terjadi **page fault** dan OS harus menangani kondisi ini.

3. Identifikasi Halaman di Secondary Storage

Sistem operasi kemudian mencari halaman yang diminta di **secondary storage** (misalnya hard disk). Halaman tersebut memang ada di sana, sehingga OS bersiap untuk memuatnya ke memori utama.

4. Memuat Halaman ke Memori Fisik

OS memilih **free frame** (slot kosong) di memori fisik. Halaman yang hilang kemudian dibawa dari secondary storage ke memori utama agar dapat diakses oleh CPU.

5. Reset Page Table

Setelah halaman berhasil dimuat ke memori, OS memperbarui **page table**. Entri untuk halaman 35 diubah menjadi valid dan menunjuk ke frame fisik yang baru ditempati halaman tersebut.

6. Restart Instruksi

CPU mengulangi instruksi yang sebelumnya gagal. Karena halaman sekarang sudah tersedia di memori, instruksi dapat dieksekusi tanpa masalah.

Inti Proses

Proses ini disebut **demand paging**, yaitu mekanisme memori virtual yang memuat halaman ke memori hanya ketika dibutuhkan. Tujuannya adalah menghemat penggunaan memori fisik dan tetap memungkinkan eksekusi program besar.

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)



Re: Diskusi.5

oleh [MOH SUUDI 055669377](#) - Sabtu, 8 November 2025, 12:20

Nama : Moh Su'udi

Nim : 055669377

Assalamualaikum Wr.Wb

Selamat siang Bapak Ade Chandra Saputra S.Kom.,M.Cs dan Teman-teman semua.

Izin menanggapi soal diskusi ini.

Menurut saya, gambar tersebut menjelaskan alur penanganan kesalahan halaman (page fault), yaitu antara lain :

1. Proses referensi halaman CPU mencoba mengakses instruksi pada alamat logis tertentu.
2. Terjadi trap ke sistem operasi karena halaman yg diminta tidak ada di memori utama (invalid bit pada tabel halaman).

3. Sistem operasi memeriksa dan menemukan bahwa halaman tersebut tersimpan di secondary storage.
4. Halaman yg hilang dibawa ke memori utama (dari secondary storage ke free frame di physical memory).
5. Tabel halaman diperbarui untuk menandai bahwa halaman sudah valid dan berada di memori.
6. Instruksi diulang (restart instruction) sehingga CPU dapat mengeksekusi kembali instruksi yang sebelumnya menyebabkan page fault.

Singkatnya, proses ini menunjukkan bagaimana sistem operasi menangani kondisi ketika halaman yang dibutuhkan program belum ada di memori utama, dengan cara memuatnya dari penyimpanan sekunder ke memori fisik agar eksekusi dapat dilanjutkan.

Referensi :

Rinusantoro, S., dkk. (2025). *Modul: MSIM4201 – Sistem Operasi* (Edisi ke-2). Tangerang Selatan: Universitas Terbuka.

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)



Re: Diskusi.5

oleh [053681662 LAURA ANGELINA PAKPAHAN](#) - Sabtu, 8 November 2025, 18:04

Izin Menanggapi Soal Diskusi 5

Jawaban Diskusi 5:

Alur Penanganan Page Fault

- Reference

CPU meminta instruksi atau data menggunakan alamat logis

Misal: CPU ingin mengambil instruksi halaman ke-35

- Trap ke Sistem Operasi

Sistem operasi mendeteksi bahwa halaman ke-35 sedang tidak ada di memori utama (RAM), maka terjadilah page fault.

CPU melakukan trap yang berarti interrupt CPU mengeksekusi sistem operasi untuk menangani kesalahan

-OS melakukan pengecekan, dimanakah halaman tersebut?

OS memeriksa keberadaan halaman

OS mengecek page table dan menemukan bahwa halaman 35 tidak ada di memori atau secondary storage

OS memberikan alamat tertentu pada CPU agar OS dapat mengambil halaman 35 dari secondary storage yaitu disk

- Membawa halaman ke dalam memori

OS mencari free frame

OS menyalin halaman yang hilang ini ke free frame tadi

- Reset Page Table

Setelah halaman berhasil ditransfer ke dalam memori utama, yang harus dilakukan oleh OS adalah mengubah atau mereset page table

Page table menunjukkan sekarang bahwa halaman 35 berada di frame terisi (di memori) dalam sebelumnya kosong

- Restart Instruksi

CPU mengeksekusi kembali perintah yang tadi menyebabkan terjadinya page fault

Tetapi sekarang sudah tidak ada kendala sedikitpun karena halaman sekarang sudah berada di dalam memori utama

Referensi :

Rinusantoro, S., dkk. (2025). Modul: MSIM4201 – Sistem Operasi. Tangerang Selatan: Universitas Terbuka.

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)



Re: Diskusi.5

oleh [TRYNUR ALHAFIIDZ_055334472](#) - Minggu, 9 November 2025, 00:09

1. CPU melakukan referensi ke alamat logis

Prosesor mengeksekusi instruksi yang membutuhkan sebuah halaman (misalnya halaman nomor 35). Ketika tabel halaman dicek, sistem mendeteksi bahwa halaman tersebut tidak berada di memori utama.

2. Terjadi trap (interupsi) ke sistem operasi

Karena halaman yang direferensikan tidak tersedia, CPU menghasilkan **page fault trap**. Kontrol eksekusi pun dipindahkan ke sistem operasi untuk menangani kejadian ini.

3. OS memeriksa lokasi halaman di secondary storage

Sistem operasi kemudian mencari halaman yang hilang tersebut (misalnya halaman 35) di dalam **secondary storage** seperti hard disk.

4. Sistem membawa halaman yang hilang ke memori utama

OS mengambil halaman yang dibutuhkan dari secondary storage dan menempatkannya ke sebuah **free frame** (ruang kosong) di physical memory.

5. OS memperbarui page table

Setelah halaman dimuat ke memori, sistem operasi memperbarui tabel halaman dengan mencatat:

- frame baru tempat halaman ditempatkan, dan
- status valid dari halaman tersebut.

Dengan demikian, referensi ke halaman itu dapat dilakukan dengan benar pada percobaan berikutnya.

6. Instruksi yang menyebabkan page fault diulang kembali

Setelah page table diperbarui, OS mengembalikan kontrol ke CPU. Instruksi yang sebelumnya gagal dijalankan karena page fault kini dieksekusi ulang, dan sekarang dapat berjalan normal karena halaman sudah tersedia di memori utama.

by logic

referensi alamat → page fault → trap ke OS → cari halaman di disk → muat ke memori → perbarui page table → ulangi instruksi.

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)

**Re: Diskusi.5**oleh [AHMAD HASBY MAULANA 057088008](#) - Minggu, 9 November 2025, 04:41

Izin menanggapi diskusi 5

Sesuai gambar alur penanganan kesalahan halaman (Page Fault). Berikut deskripsi alurnya:

1. Reference: CPU akan mengambil instruksi dari memori untuk dijalankan[1]. Kemudian akan melihat Page Table untuk menemukan dimana tempat instruksi itu berada di memori fisik(RAM).
2. Trap: Ketika CPU melihat Page Table dan mendapati bit(tanda) invalid (i), maka akan terjadi interupsi kesalahan halaman yang menyebabkan trap pada sistem operasi. Tanda invalid(i) memiliki arti bahwa halaman yang dicari tidak ada di memori fisik.
3. Pengecekan Sistem Operasi: Ketika terjadi trap, sistem operasi akan mengambil alih kendali kemudian akan memeriksa tabel internalnya sendiri. Di tabel internalnya jika SO menemukan bahwa halaman yang dituju itu valid maka akan mengambil dari disk / secondary storage. Namun jika menemukan bahwa itu invalid, SO akan menghentikan proses[1].
4. Bring in missing page (Pemuatan halaman): Setelah halaman ditemukan, SO akan mencari slot kosong di Physical Memory yang disebut Free Frame. Setelah menemukan slot kosong, halaman akan disalin (dibawa masuk) dari secondary storage ke free frame di RAM.
5. Reset page table: Setelah halaman disalin ke RAM, SO akan mereset page table. Bit (tanda) yang sebelumnya invalid (i) akan diubah menjadi valid (v) dan alamat frame fisiknya akan dicatat.
6. Restart Instruction: SO akan mengembalikan kendali ke program. Instruksi yang sebelumnya menyebabkan page fault akan dijalankan ulang[1]. Karena di page table bit(tanda) nya valid maka program akan berjalan normal seolah tidak terjadi apa apa.

Keseluruhan alur penanganan kesalahan halaman (page fault) ini merupakan mekanisme fundamental dalam manajemen memori virtual[1][2].

Referensi:

[1] S. Rinusantoro, Sistem Operasi (MSIM4201), Edisi 2. Tangerang Selatan: Universitas Terbuka, 2023.

[2] T. H. P. Siahaan, "Manajemen Memori pada Sistem Operasi," Jurnal Mantik, vol. 3, no. 1, 2019.

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)

**Re: Diskusi.5**oleh [054160782 RIZA HARDIAN](#) - Minggu, 9 November 2025, 09:08

Izin menanggapi diskusi sebagai berikut,

Penanganan kesalahan halaman (page fault) sesuai gambar diatas

1. Referensi Halaman

Program sedang dijalankan dan mencoba mengakses instruksi atau data di alamat tertentu — misalnya instruksi 2 yang ada di halaman 35.

Sistem kemudian memeriksa page table (tabel halaman) untuk memastikan apakah halaman 35 sudah ada di memori utama.

2. Trap ke Sistem Operasi

Ternyata di tabel halaman, status halaman 35 masih invalid (belum ada di memori utama).

Karena itu, sistem mendeteksi adanya page fault dan mengirimkan sinyal (trap) ke sistem operasi agar menangani masalah ini.

3. Cek di Secondary Storage

Sistem operasi lalu mencari halaman 35 yang hilang di secondary storage (penyimpanan sekunder seperti hard disk).

Halaman tersebut memang tersimpan di sana.

4. Memasukkan Halaman ke Memori

Sistem operasi kemudian mengambil (load) halaman 35 dari secondary storage dan menaruhnya ke memori utama (RAM) pada ruang kosong yang disebut free frame.

5. Memperbarui Page Table

Setelah halaman berhasil dimuat ke memori utama, sistem memperbarui page table:

Bit valid (V) diubah menjadi 1 (artinya halaman sudah ada di RAM).

Nomor frame tempat halaman disimpan juga dicatat.

6. Menjalankan Ulang Instruksi

Setelah semuanya siap, sistem akan menjalankan ulang instruksi yang tadi gagal karena page fault.

Sekarang halaman 35 sudah ada di memori utama, jadi instruksi bisa dijalankan dengan normal tanpa error.

Kesimpulannya secara sederhana, alur penanganan page fault adalah:

Referensi -> Trap -> Cari di Secondary Storage -> Load ke Memori -> Update Page Table -> Jalankan Ulang Instruksi.

Kalau halaman belum ada di memori, sistem akan mengambilnya dari penyimpanan sekunder, menaruhnya ke RAM, memperbarui tabel halaman, lalu melanjutkan program seperti biasa

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)

◀ [Manajemen Memori \(...\)](#)

Lompat ke...

Tugas.2 ►

Navigasi

▼ Dasbor

[Beranda situs](#)

> [Laman situs](#)

▼ Kelasku

> [STSI4203.108](#)

> [STSI4202.42](#)

▼ [STSI4103.119](#)

> [Peserta](#)

Nilai

> [Pendahuluan](#)

> [Sesi 1](#)

> [Sesi 2](#)

> [Sesi 3](#)

> [Sesi 4](#)

▼ [Sesi 5](#)

[Kehadiran Sesi ke-5](#)

 [Manajemen Memori \(Caching dan Virtual Memory\)](#)

 [Manajemen Memori \(Caching dan Virtual Memory\)](#).

[Diskusi.5](#)

[Tugas.2](#)

> [MKKI4201.278](#)

> [STSI4201.161](#)

> [STSI4205.331](#)

> [STSI4104.284](#)

> [MKDI4202.1514](#)

> [Kelas](#) Administrasi

✓ Forum administrasi

Berlangganan dinonaktifkan

Hide sidebars

Follow Us:      

UNIVERSITAS TERBUKA ©2025

Anda masuk sebagai [INDRAWAN LISANTO 053724113](#) ([Keluar](#))[Dapatkan aplikasi seluler](#)