

BUKU AJAR
MATA KULIAH ARSITEKTUR KOMPUTER
BAB 12
ORGANISASI MEMORI KOMPUTER

Buku Ajar Arsitektur Komputer 2026
Berbasis Rencana Pembelajaran Semester (RPS)
IF221101 | 3 SKS

IDENTITAS BAB

Komponen	Keterangan
Mata Kuliah	Arsitektur Komputer
Kode	IF221101
Bobot	3 SKS
Bab	Bab 12
Judul	Organisasi Memori Komputer
Fokus Pembelajaran	Memahami organisasi, hierarki, karakteristik, dan mekanisme pengelolaan memori dalam sistem komputer
Level Pembelajaran	Pemahaman konsep, analisis hierarki, perhitungan kapasitas dan alamat, serta evaluasi kinerja memori

CAPAIAN PEMBELAJARAN BAB

Setelah mempelajari bab ini, mahasiswa diharapkan mampu:

1. Menjelaskan fungsi memori dalam sistem komputer.
2. Membedakan karakteristik utama berbagai jenis memori.
3. Menjelaskan konsep hierarki memori.
4. Menganalisis hubungan kapasitas, kecepatan, biaya, dan kedekatan dengan CPU.
5. Membedakan register, cache, memori utama, dan penyimpanan sekunder.
6. Menjelaskan prinsip locality sebagai dasar efektivitas cache.
7. Menjelaskan konsep cache hit, cache miss, hit ratio, dan miss penalty.
8. Membedakan pendekatan pemetaan cache secara konseptual.
9. Menjelaskan konsep memori virtual dan translasi alamat.
10. Menghitung kebutuhan bit alamat dan kapasitas memori sederhana.
11. Menganalisis alur akses data dari CPU ke berbagai tingkat memori.

PETA KONSEP

ALUR KONSEP BAB 12

CPU → Register → Cache → Memori Utama → Penyimpanan Sekunder → Hierarki Memori → Locality → Cache Hit/Miss → Mapping → Replacement → Write Policy → Virtual Memory → Address Translation → Page dan Frame.

12.1 Pendahuluan

CPU membutuhkan instruksi dan data untuk menjalankan program. Informasi tersebut tidak semuanya disimpan pada satu jenis media. Sistem komputer menggunakan organisasi bertingkat agar dapat menyeimbangkan kebutuhan kecepatan, kapasitas, dan biaya.

Memori yang sangat dekat dengan CPU umumnya memiliki akses lebih cepat tetapi kapasitas lebih terbatas. Sebaliknya, media dengan kapasitas besar cenderung memiliki waktu akses lebih tinggi. Hierarki memori dirancang untuk memanfaatkan kelebihan setiap tingkat tersebut.

PERTANYAAN PEMANTIK

Mengapa komputer tidak hanya menggunakan satu jenis memori yang sangat cepat? Mengapa cache diperlukan? Bagaimana program yang ukurannya besar dapat dijalankan ketika kapasitas memori fisik terbatas?

12.2 Fungsi Memori dalam Sistem Komputer

Memori berfungsi menyimpan instruksi, data, hasil sementara, dan informasi yang diperlukan sistem. Dalam proses eksekusi program, CPU melakukan operasi pembacaan dan penulisan terhadap lokasi penyimpanan sesuai aturan arsitektur.

Informasi	Contoh Peran
Instruksi	Program yang akan dieksekusi CPU
Data input	Nilai yang akan diproses
Data sementara	Hasil antara selama perhitungan
Hasil	Nilai yang akan disimpan atau dikirim ke komponen lain
Informasi sistem	State dan struktur yang dibutuhkan sistem operasi atau perangkat lunak

12.3 Karakteristik Memori

Memori dapat dianalisis berdasarkan sejumlah karakteristik. Perbandingan tidak cukup hanya melihat kapasitas, karena waktu akses, biaya, metode akses, dan volatilitas juga menentukan perannya dalam sistem.

Karakteristik	Makna
Kapasitas	Jumlah informasi yang dapat disimpan
Waktu akses	Waktu yang diperlukan untuk memperoleh data
Bandwidth	Jumlah data yang dapat dipindahkan dalam interval tertentu
Biaya per bit	Biaya relatif terhadap kapasitas
Volatilitas	Apakah data tetap tersedia ketika daya dihentikan
Metode akses	Cara lokasi data dipilih dan diakses
Kedekatan dengan CPU	Posisi relatif dalam jalur akses data

12.4 Hierarki Memori

Hierarki memori mengatur beberapa tingkat penyimpanan berdasarkan karakteristik yang berbeda. Secara konseptual, tingkat yang lebih dekat dengan CPU memiliki kapasitas lebih kecil dan kecepatan lebih tinggi, sedangkan tingkat yang lebih jauh menawarkan kapasitas lebih besar dengan akses yang relatif lebih lambat.

Tingkat	Karakteristik Umum
Register	Sangat dekat dengan unit eksekusi dan berkapasitas terbatas
Cache	Menyimpan salinan data yang sering atau kemungkinan segera diakses
Memori Utama	Menyimpan program dan data aktif
Penyimpanan Sekunder	Kapasitas besar untuk penyimpanan jangka lebih panjang

PRINSIP UTAMA

Hierarki memori berusaha memberikan kesan bahwa sistem memiliki memori yang relatif cepat, besar, dan ekonomis melalui kombinasi beberapa tingkat penyimpanan.

12.5 Register sebagai Tingkat Memori Terdekat

Register berada di dalam atau sangat dekat dengan organisasi internal CPU dan digunakan untuk menyimpan operand, alamat, instruksi, atau state. Karena kapasitasnya terbatas, register tidak digunakan untuk menyimpan seluruh program.

Bab 10 dan BAB 11 telah membahas peran PC, IR, register umum, serta register yang digunakan untuk akses memori. Dalam konteks hierarki, register merupakan tingkat penyimpanan yang paling dekat dengan proses eksekusi instruksi.

12.6 Cache Memory

Cache adalah memori berkapasitas relatif kecil yang digunakan untuk menyimpan salinan blok data dari tingkat memori yang lebih besar. Tujuannya adalah mengurangi rata-rata waktu akses ketika data yang dibutuhkan tersedia di cache.

Cache dapat ditempatkan dalam beberapa tingkat, seperti L1, L2, dan L3, tetapi organisasi, ukuran, dan lokasi detailnya bergantung pada rancangan prosesor.

Konsep	Makna
Cache Hit	Data yang diminta ditemukan pada cache
Cache Miss	Data yang diminta tidak ditemukan pada cache
Hit Ratio	Proporsi akses yang menghasilkan hit
Miss Rate	Proporsi akses yang menghasilkan miss
Miss Penalty	Tambahan waktu atau biaya ketika miss harus dilayani dari tingkat lebih bawah

12.7 Prinsip Locality

Efektivitas cache didasarkan pada kecenderungan program untuk mengakses data dan instruksi dengan pola tertentu. Dua konsep penting adalah temporal locality dan spatial locality.

Jenis Locality	Penjelasan
Temporal Locality	Data atau instruksi yang baru diakses memiliki kemungkinan diakses kembali dalam waktu relatif dekat
Spatial Locality	Data atau instruksi yang lokasinya berdekatan dengan akses saat ini memiliki kemungkinan segera diakses

Karena adanya locality, cache tidak hanya menyimpan satu unit data tunggal, tetapi dapat bekerja menggunakan blok atau cache line yang membawa sejumlah data berdekatan.

12.8 Struktur Dasar Cache

Secara konseptual, sebuah cache menyimpan data dan informasi yang membantu menentukan asal blok tersebut dari memori utama. Informasi seperti tag digunakan untuk membedakan blok yang memiliki posisi tertentu dalam struktur cache.

Komponen Konseptual	Fungsi
Data Block/Cache Line	Menyimpan salinan data
Tag	Mengidentifikasi bagian alamat asal blok
Valid Information	Menunjukkan apakah entri berisi data yang valid
Status Metadata	Dapat digunakan untuk kebijakan penggantian atau penulisan

12.9 Pemetaan Cache

Pemetaan cache menentukan lokasi yang mungkin digunakan sebuah blok memori utama ketika ditempatkan ke cache. Tiga pendekatan konseptual yang umum dibahas adalah direct mapping, fully associative, dan set-associative.

12.9.1 Direct Mapping

Setiap blok memori utama dipetakan ke satu lokasi cache tertentu berdasarkan aturan indeks. Pendekatan ini relatif sederhana, tetapi beberapa blok dapat bersaing pada lokasi yang sama.

12.9.2 Fully Associative

Sebuah blok secara konseptual dapat ditempatkan pada lokasi cache yang tersedia. Fleksibilitas ini memerlukan mekanisme pencarian atau perbandingan yang lebih luas.

12.9.3 Set-Associative

Cache dibagi menjadi sejumlah set. Sebuah blok dipetakan ke satu set tertentu, tetapi dapat ditempatkan pada salah satu line dalam set tersebut.

Metode	Fleksibilitas	Konflik Lokasi
Direct Mapping	Lebih terbatas	Dapat terjadi ketika blok bersaing pada lokasi yang sama
Fully Associative	Paling fleksibel secara konseptual	Konflik pemetaan sangat berkurang
Set-Associative	Kompromi antara keduanya	Bersaing di dalam set yang sama

12.10 Kebijakan Penggantian Cache

Ketika cache atau sebuah set tidak memiliki ruang yang tersedia, sistem memerlukan kebijakan untuk menentukan blok yang akan digantikan. Contoh pendekatan konseptual mencakup:

- LRU atau pendekatan yang mendekati prinsip least recently used.
- FIFO berdasarkan urutan kedatangan.
- Random replacement.
- Kebijakan lain yang disesuaikan dengan rancangan perangkat keras.

Implementasi nyata dapat menggunakan pendekatan perkiraan karena pelacakan urutan penggunaan secara tepat dapat meningkatkan kompleksitas perangkat keras.

12.11 Kebijakan Penulisan Cache

Ketika CPU mengubah data yang berada pada cache, sistem perlu menentukan kapan perubahan tersebut diteruskan ke tingkat memori berikutnya.

Kebijakan	Gagasan Utama
Write-Through	Perubahan diteruskan ke tingkat memori berikutnya sesuai mekanisme penulisan
Write-Back	Perubahan dapat ditahan di cache dan diteruskan ketika blok perlu dipindahkan atau sesuai kebijakan
Write Allocate	Pada kondisi tertentu, blok yang mengalami write miss dapat dimuat terlebih dahulu
No-Write Allocate	Write miss dapat diteruskan tanpa harus menempatkan blok di cache

Kombinasi kebijakan bergantung pada tujuan kinerja dan konsistensi rancangan.

12.12 Kinerja Cache dan Average Memory Access Time

Salah satu model sederhana untuk menganalisis kinerja cache menggunakan Average Memory Access Time atau AMAT.

MODEL KONSEPTUAL

$$\text{AMAT} = \text{Hit Time} + (\text{Miss Rate} \times \text{Miss Penalty})$$

Model ini membantu menunjukkan bahwa penurunan miss rate atau miss penalty dapat mengurangi rata-rata waktu akses. Dalam sistem dengan beberapa tingkat cache, analisis dapat diperluas sesuai struktur hierarkinya.

CONTOH LATIHAN

Jika hit time = 2 satuan waktu, miss rate = 0,05, dan miss penalty = 40 satuan waktu, maka $\text{AMAT} = 2 + (0,05 \times 40) = 4$ satuan waktu.

12.13 Memori Utama

Memori utama menyimpan program dan data yang sedang aktif digunakan sistem. CPU dapat mengakses memori utama melalui mekanisme alamat dan transfer data sesuai organisasi komputer.

Pada tingkat konsep, memori dapat dipandang sebagai kumpulan lokasi yang memiliki alamat. Setiap lokasi atau unit pengalamatan menyimpan informasi sesuai ukuran dan organisasi sistem.

12.14 Organisasi Alamat Memori

Sebuah alamat digunakan untuk memilih lokasi memori. Jumlah bit alamat menentukan jumlah kombinasi alamat yang secara teoritis dapat dinyatakan.

PRINSIP DASAR

Jika tersedia n bit alamat dan setiap alamat menunjuk satu unit yang sama, jumlah lokasi yang dapat dinyatakan adalah 2^n .

Bit Alamat	Jumlah Kombinasi Alamat
8	256
10	1.024
16	65.536
20	1.048.576
32	4.294.967.296

Kapasitas total juga bergantung pada ukuran unit yang dapat dialamatkan. Jika setiap alamat menunjuk satu byte, kapasitas berbeda dibandingkan jika satu alamat menunjuk satu word.

12.15 RAM dan ROM

Dalam pengenalan organisasi memori, RAM dan ROM sering dibedakan berdasarkan fungsi umum dan karakteristik penyimpanannya.

Jenis	Karakteristik Konseptual
-------	--------------------------

RAM	Digunakan sebagai memori baca/tulis untuk data dan program aktif; umumnya bersifat volatile
ROM	Digunakan untuk menyimpan informasi yang tidak dimodifikasi seperti memori kerja biasa; karakteristik implementasi dapat bervariasi

Berbagai teknologi dan varian memori memiliki karakteristik tersendiri. Pembahasan detail implementasi fisik dapat berbeda dari model konseptual organisasi memori pada bab ini.

12.16 DRAM dan SRAM: Pengantar Konseptual

SRAM dan DRAM merupakan dua konsep teknologi memori yang sering dibandingkan. Secara umum, SRAM dikaitkan dengan kebutuhan refresh yang berbeda dari DRAM dan sering digunakan pada konteks yang membutuhkan akses cepat seperti cache. DRAM banyak digunakan sebagai dasar memori utama dengan pertimbangan kapasitas dan biaya.

Aspek Konseptual	SRAM	DRAM
Penggunaan umum	Sering digunakan pada cache	Sering digunakan pada memori utama
Refresh	Tidak memerlukan refresh seperti DRAM	Memerlukan mekanisme refresh
Kepadatan	Relatif lebih rendah	Relatif lebih tinggi
Biaya per bit	Relatif lebih tinggi	Relatif lebih rendah

12.17 Penyimpanan Sekunder

Penyimpanan sekunder menyediakan kapasitas lebih besar dan digunakan untuk penyimpanan data dalam jangka lebih panjang. Dalam hierarki, media ini berada lebih jauh dari jalur eksekusi langsung CPU dibandingkan register, cache, dan memori utama.

Sistem operasi dan perangkat lunak mengelola perpindahan data antara penyimpanan sekunder dan memori utama sesuai kebutuhan.

12.18 Memori Virtual

Memori virtual memungkinkan program menggunakan ruang alamat yang dikelola secara terpisah dari organisasi fisik memori utama. Sistem melakukan translasi dari alamat virtual atau logis ke alamat fisik melalui mekanisme yang ditentukan oleh arsitektur dan sistem operasi.

Konsep ini memberikan abstraksi ruang alamat yang membantu pengelolaan program dan isolasi antar proses.

12.19 Paging dan Frame

Pada paging, ruang alamat dibagi menjadi unit dengan ukuran tetap yang secara konseptual disebut page. Memori fisik dibagi menjadi unit berukuran sesuai yang disebut frame. Translasi alamat menentukan frame fisik yang berkaitan dengan page tertentu.

Komponen	Peran
Virtual Address	Alamat yang digunakan dalam ruang alamat proses
Page Number	Bagian alamat yang mengidentifikasi page
Offset	Posisi di dalam page
Frame	Unit lokasi fisik yang menjadi tempat page ditempatkan
Page Table	Struktur yang membantu translasi page ke frame

KONSEP

Jika ukuran page adalah 2^k byte, maka bagian offset membutuhkan k bit. Sisa bagian alamat dapat digunakan untuk mengidentifikasi page sesuai ukuran alamat yang digunakan.

12.20 Page Fault

Jika page yang dibutuhkan tidak berada di memori fisik, sistem dapat mengalami page fault. Penanganannya melibatkan mekanisme untuk memperoleh page yang diperlukan dari tingkat penyimpanan yang lebih bawah dan memperbarui informasi pemetaan.

Page fault memiliki biaya yang jauh lebih besar dibandingkan cache miss biasa karena melibatkan tingkat penyimpanan dan mekanisme sistem yang berbeda.

12.21 Translation Lookaside Buffer (TLB)

TLB merupakan struktur yang digunakan untuk membantu mempercepat translasi alamat dengan menyimpan informasi translasi yang baru atau sering digunakan. Secara konseptual, TLB menerapkan prinsip penyimpanan informasi translasi dekat dengan jalur akses CPU.

Ketika translasi ditemukan di TLB, proses dapat menghindari pencarian penuh melalui struktur pemetaan yang lebih besar. Ketika tidak ditemukan, diperlukan mekanisme translasi lanjutan.

12.22 Studi Kasus 1: Menentukan Kapasitas Memori

Sebuah sistem memiliki 12 bit alamat dan setiap alamat menunjuk 1 byte. Jumlah lokasi adalah $2^{12} = 4.096$ lokasi, sehingga kapasitas teoritisnya adalah 4.096 byte.

Jika ukuran unit yang dialamatkan berbeda, perhitungan kapasitas harus disesuaikan dengan ukuran setiap unit.

12.23 Studi Kasus 2: Analisis Cache Hit dan Miss

Diberikan 1.000 akses memori dan 920 di antaranya ditemukan pada cache. Hit ratio adalah $920/1.000 = 0,92$, sedangkan miss rate adalah 0,08.

Mahasiswa diminta menganalisis bagaimana perubahan miss penalty memengaruhi AMAT dengan menggunakan model yang telah dipelajari.

12.24 Studi Kasus 3: Translasi Paging Sederhana

Misalkan ukuran page adalah 1.024 byte atau 2^{10} byte. Oleh karena itu, offset memerlukan 10 bit. Sebuah alamat virtual kemudian dapat dipisahkan menjadi bagian page number dan offset. Jika page number dipetakan ke frame tertentu, alamat fisik dibentuk dengan menggabungkan frame number dan offset.

AKTIVITAS ANALISIS

Gunakan ukuran alamat dan page yang diberikan dosen untuk melakukan pemisahan page number dan offset serta membangun alamat fisik secara konseptual.

12.25 Aktivitas Pembelajaran Mahasiswa

Aktivitas 1: Hierarki Memori

Buat diagram hubungan register, cache, memori utama, dan penyimpanan sekunder.

Aktivitas 2: Karakteristik

Bandingkan kapasitas, kecepatan, dan biaya relatif pada setiap tingkat hierarki.

Aktivitas 3: Locality

Identifikasi contoh temporal locality dan spatial locality dalam sebuah loop program.

Aktivitas 4: Cache Mapping

Simulasikan penempatan blok menggunakan direct mapping dan set-associative mapping sederhana.

Aktivitas 5: AMAT

Hitung Average Memory Access Time berdasarkan beberapa skenario hit time, miss rate, dan miss penalty.

Aktivitas 6: Alamat Memori

Tentukan jumlah lokasi dan kapasitas berdasarkan jumlah bit alamat dan ukuran unit.

Aktivitas 7: Paging

Pisahkan alamat virtual menjadi page number dan offset menggunakan ukuran page yang diberikan.

12.26 Rangkuman

- Memori menyimpan instruksi, data, hasil sementara, dan informasi yang diperlukan sistem.
- Hierarki memori menggabungkan beberapa tingkat penyimpanan untuk menyeimbangkan kecepatan, kapasitas, dan biaya.
- Register merupakan tingkat penyimpanan yang sangat dekat dengan CPU dan berkapasitas terbatas.
- Cache menyimpan salinan data atau instruksi untuk mengurangi rata-rata waktu akses.
- Temporal locality dan spatial locality menjadi dasar efektivitas cache.
- Cache hit terjadi ketika data ditemukan pada cache, sedangkan cache miss terjadi ketika data harus diperoleh dari tingkat lain.
- Direct mapping, fully associative, dan set-associative merupakan pendekatan konseptual pemetaan cache.
- Kebijakan replacement menentukan blok yang diganti ketika ruang cache terbatas.

- Write-through dan write-back merupakan contoh kebijakan penulisan cache.
- AMAT dapat digunakan sebagai model sederhana untuk menganalisis rata-rata waktu akses memori.
- Jumlah bit alamat menentukan jumlah kombinasi alamat yang dapat direpresentasikan.
- RAM, ROM, SRAM, dan DRAM memiliki karakteristik serta peran yang berbeda.
- Memori virtual menyediakan abstraksi ruang alamat dan memerlukan translasi ke alamat fisik.
- Paging membagi ruang alamat menjadi page dan memori fisik menjadi frame.
- TLB membantu mempercepat proses translasi alamat.

12.27 Glosarium Bab

Istilah	Pengertian
AMAT	Model Average Memory Access Time.
Cache	Memori kecil dan cepat yang menyimpan salinan data dari tingkat memori lain.
Cache Hit	Kondisi ketika data ditemukan pada cache.
Cache Miss	Kondisi ketika data tidak ditemukan pada cache.
Direct Mapping	Pemetaan blok ke satu lokasi cache tertentu.
DRAM	Teknologi memori yang umum digunakan pada memori utama dan memerlukan refresh.
Effective Capacity	Kapasitas yang dapat dianalisis berdasarkan organisasi unit dan alamat.
Frame	Unit memori fisik yang menjadi tempat page pada sistem paging.
Hit Ratio	Proporsi akses yang berhasil menemukan data di cache.
Locality	Kecenderungan pola akses data atau instruksi.
Page	Unit ruang alamat virtual berukuran tetap dalam paging.
Page Fault	Kondisi ketika page yang diperlukan belum tersedia di memori fisik.
SRAM	Teknologi memori yang sering digunakan pada cache.
TLB	Struktur untuk membantu mempercepat translasi alamat.
Virtual Memory	Mekanisme abstraksi ruang alamat yang dipetakan ke memori fisik.

12.28 Soal Latihan

A. Soal Pemahaman

12. Jelaskan mengapa sistem komputer menggunakan hierarki memori.
13. Bandingkan register, cache, memori utama, dan penyimpanan sekunder.
14. Jelaskan temporal locality dan spatial locality.

15. Apa perbedaan cache hit dan cache miss?
16. Bandingkan direct mapping, fully associative, dan set-associative.
17. Jelaskan fungsi kebijakan replacement.
18. Bandingkan write-through dan write-back.
19. Jelaskan konsep memori virtual.
20. Jelaskan hubungan antara page dan frame.
21. Apa fungsi TLB dalam translasi alamat?

B. Soal Perhitungan dan Analisis

22. Sebuah memori memiliki 14 bit alamat dan setiap alamat menunjuk 1 byte. Tentukan jumlah lokasi dan kapasitas total.
23. Sebuah sistem memiliki 20 bit alamat dengan unit pengalamatan 2 byte. Tentukan kapasitas berdasarkan model tersebut.
24. Jika hit time 1 ns, miss rate 0,04, dan miss penalty 50 ns, hitung AMAT menggunakan model sederhana.
25. Dari 50.000 akses, terdapat 47.500 cache hit. Tentukan hit ratio dan miss rate.
26. Jika ukuran page 4.096 byte, berapa bit offset yang diperlukan?
27. Sebuah alamat virtual 32-bit menggunakan page berukuran 4 KB. Tentukan jumlah bit untuk offset dan bagian page number.
28. Analisis dampak jika miss rate turun tetapi miss penalty meningkat secara signifikan.

C. Soal HOTS dan Studi Kasus

29. Rancang hierarki memori konseptual untuk sistem yang membutuhkan respons cepat tetapi juga kapasitas penyimpanan besar. Jelaskan trade-off desainnya.
30. Analisis pola akses sebuah program yang melakukan traversal array secara berurutan dan jelaskan bentuk locality yang dominan.
31. Bandingkan dampak penggunaan direct mapping dan set-associative terhadap kemungkinan conflict miss.
32. Sebuah aplikasi mengalami banyak page fault. Jelaskan beberapa kemungkinan penyebab berdasarkan konsep memori virtual dan analisis konseptual langkah perbaikannya.
33. Buat simulasi sederhana alur akses CPU ketika terjadi cache hit, cache miss, TLB hit, dan page fault.

12.29 Proyek Mini

Judul proyek: Memory Hierarchy and Cache Simulator.

Mahasiswa membuat aplikasi, program, spreadsheet interaktif, atau simulator sederhana yang menggambarkan perilaku hierarki memori.

- Menyediakan model register, cache, memori utama, dan penyimpanan sekunder.
- Mendukung simulasi hit dan miss.
- Mendukung minimal satu model direct mapping dan satu model set-associative.
- Menghitung hit ratio dan miss rate.
- Menghitung AMAT menggunakan parameter yang dapat diubah.
- Menampilkan blok atau alamat yang sedang diakses.
- Menyediakan simulasi sederhana replacement policy.
- Menyediakan minimal lima skenario pengujian dan analisis hasil.

OUTPUT YANG DIHARAPKAN

Dokumentasi rancangan, diagram hierarki, tabel simulasi akses, statistik hit/miss, hasil perhitungan AMAT,

serta analisis trade-off kinerja.

12.30 Daftar Pustaka

- Patterson, D. A., & Hennessy, J. L. Computer Organization and Design: The Hardware/Software Interface. Morgan Kaufmann.
- Stallings, W. Computer Organization and Architecture: Designing for Performance. Pearson.
- Hennessy, J. L., & Patterson, D. A. Computer Architecture: A Quantitative Approach. Morgan Kaufmann.
- Mano, M. M. Computer System Architecture. Pearson.
- Null, L., & Lobur, J. The Essentials of Computer Organization and Architecture. Jones & Bartlett Learning.
- Dokumen Rencana Pembelajaran Semester Mata Kuliah Arsitektur Komputer IF221101, Program Studi Informatika, Fakultas Ilmu Komputer, Universitas Pembangunan Nasional Veteran Jawa Timur.

CATATAN PENYUSUNAN BAB

Bab ini memperluas pembahasan CPU dan instruksi ke arah sistem penyimpanan yang menyediakan instruksi dan data bagi prosesor. Pemahaman mengenai hierarki memori, cache, locality, pengalamatan, dan memori virtual menjadi dasar penting untuk menganalisis kinerja komputer secara menyeluruh.