

BUKU AJAR
MATA KULIAH ARSITEKTUR KOMPUTER
BAB 11
SET INSTRUKSI DAN MODE
PENGALAMATAN

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 11
Judul	Set Instruksi dan Mode Pengalamatan
Fokus Pembelajaran	Memahami organisasi set instruksi, format instruksi, operand, serta mekanisme CPU dalam menentukan lokasi data yang akan diakses
Level Pembelajaran	Pemahaman konsep, analisis format instruksi, perhitungan alamat efektif, dan simulasi eksekusi instruksi

CAPAIAN PEMBELAJARAN BAB

Setelah mempelajari bab ini, mahasiswa diharapkan mampu:

1. Menjelaskan pengertian Instruction Set Architecture (ISA) sebagai antarmuka konseptual antara perangkat lunak dan perangkat keras.
2. Mengidentifikasi komponen dasar sebuah instruksi mesin.
3. Membedakan opcode, operand, alamat, dan informasi kendali dalam format instruksi.
4. Mengelompokkan instruksi berdasarkan fungsi operasinya.
5. Menjelaskan hubungan antara register, memori, dan operand.
6. Membedakan berbagai mode pengalamatan.
7. Menghitung alamat efektif pada mode pengalamatan tertentu menggunakan model konseptual.
8. Menganalisis dampak mode pengalamatan terhadap fleksibilitas, panjang instruksi, dan kebutuhan akses memori.
9. Menyimulasikan eksekusi instruksi sederhana dengan beberapa mode pengalamatan.
10. Menghubungkan konsep ISA dengan siklus instruksi dan organisasi CPU.

PETA KONSEP

ALUR KONSEP BAB 11

ISA → Set Instruksi → Format Instruksi → Opcode + Operand → Register dan Memori → Kategori Instruksi → Mode Pengalamatan → Alamat Efektif → Operand Access → Execute → Update State.

11.1 Pendahuluan

Pada bab sebelumnya telah dibahas bagaimana CPU mengambil, mendekode, dan mengeksekusi instruksi. Namun, agar proses tersebut dapat dilakukan, CPU harus memiliki aturan mengenai bentuk instruksi, jenis operasi yang tersedia, serta cara menentukan lokasi operand.

Kumpulan aturan tersebut merupakan bagian penting dari Instruction Set Architecture atau ISA. ISA menjadi salah satu tingkat abstraksi yang menjembatani perangkat lunak dengan organisasi perangkat keras. Program pada akhirnya diterjemahkan menjadi instruksi yang sesuai dengan aturan arsitektur prosesor.

PERTANYAAN PEMANTIK

Bagaimana CPU mengetahui operasi apa yang harus dilakukan? Dari mana CPU memperoleh operand? Mengapa satu instruksi dapat menggunakan register, sedangkan instruksi lain menggunakan alamat memori?

11.2 Konsep Instruction Set Architecture (ISA)

Instruction Set Architecture dapat dipahami sebagai spesifikasi konseptual mengenai instruksi yang dapat dipahami oleh sebuah keluarga atau rancangan arsitektur prosesor. ISA menjelaskan aspek yang terlihat oleh perangkat lunak, seperti operasi yang tersedia, register, format instruksi, representasi data, serta cara pengalamatan operand.

Aspek ISA	Contoh Konseptual
Operasi	Aritmetika, logika, perpindahan data, kontrol
Register	Nama atau peran register yang dapat diakses instruksi
Format instruksi	Susunan field dalam instruksi
Operand	Sumber dan tujuan data
Mode pengalamatan	Cara menentukan lokasi operand
Representasi data	Aturan interpretasi integer, karakter, dan format lainnya

Implementasi internal prosesor dapat berubah, tetapi kompatibilitas ISA memungkinkan perangkat lunak pada tingkat tertentu tetap menggunakan aturan instruksi yang sama.

11.3 Set Instruksi

Set instruksi adalah kumpulan operasi yang dapat dijalankan oleh prosesor sesuai arsitekturnya. Sebuah instruksi mesin pada umumnya direpresentasikan sebagai pola bit dan memuat informasi mengenai tindakan yang harus dilakukan.

Kelompok	Contoh Operasi Konseptual
Transfer data	LOAD, STORE, MOVE
Aritmetika	ADD, SUB, INC, DEC
Logika	AND, OR, XOR, NOT
Shift/Rotate	Shift kiri, shift kanan, rotate
Perbandingan	COMPARE atau operasi perbandingan
Kontrol program	JUMP, BRANCH, CALL, RETURN
Input/Output	Operasi yang berhubungan dengan perangkat I/O sesuai arsitektur

11.4 Struktur Dasar Instruksi Mesin

Sebuah instruksi dapat terdiri atas beberapa field. Format yang sebenarnya bergantung pada ISA, tetapi secara konseptual elemen penting dapat mencakup opcode dan informasi operand.

Field	Fungsi
Opcode	Menentukan operasi yang harus dilakukan
Operand Field	Menunjukkan data, register, atau informasi alamat
Addressing Information	Menentukan cara memperoleh operand
Destination Information	Menentukan lokasi hasil pada format tertentu
Immediate Data	Menyimpan nilai langsung pada instruksi bila didukung

CONTOH KONSEPTUAL

Instruksi ADD R1, R2 dapat ditafsirkan sebagai permintaan untuk melakukan operasi penjumlahan menggunakan operand yang berkaitan dengan register R1 dan R2. Format sebenarnya dapat berbeda antar arsitektur.

11.5 Opcode

Opcode atau operation code adalah bagian instruksi yang menunjukkan jenis operasi. Ketika instruksi berada pada Instruction Register, proses decode menggunakan informasi opcode untuk menentukan tindakan yang harus dilakukan oleh datapath dan Control Unit.

Jumlah bit yang tersedia untuk opcode membatasi jumlah kombinasi operasi yang dapat dikodekan dalam suatu format. Namun, rancangan ISA dapat menggunakan lebih dari satu format atau mekanisme pengkodean untuk menyediakan variasi instruksi.

11.6 Operand

Operand adalah data yang digunakan oleh sebuah instruksi. Operand dapat berupa nilai yang berada pada register, memori, bagian instruksi itu sendiri, atau informasi yang diperoleh melalui perhitungan alamat.

Sumber Operand	Karakteristik Konseptual
Register	Akses melalui register yang ditentukan instruksi
Memori	Operand diperoleh dari lokasi memori
Immediate	Nilai disediakan langsung dalam instruksi
Implicit	Operand ditentukan secara tersirat oleh aturan instruksi

Pemilihan sumber operand memengaruhi jumlah akses yang diperlukan dan kompleksitas instruksi.

11.7 Klasifikasi Instruksi Berdasarkan Jumlah Operand

Secara konseptual, instruksi dapat dikelompokkan berdasarkan jumlah operand eksplisit yang ditampilkan dalam bentuk instruksi.

Jenis	Contoh Konseptual	Karakteristik
Tiga alamat	ADD R1, R2, R3	Dua sumber dan satu tujuan dapat dinyatakan secara eksplisit
Dua alamat	ADD R1, R2	Salah satu operand dapat sekaligus menjadi lokasi hasil
Satu alamat	ADD X	Organisasi tertentu dapat menggunakan accumulator implisit
Nol alamat	ADD	Operand dapat diambil dari struktur seperti stack

Tabel tersebut digunakan sebagai model pembelajaran. Arsitektur nyata memiliki format dan aturan yang dapat lebih kompleks.

11.8 Format Instruksi

Format instruksi menentukan bagaimana bit-bit dalam instruksi dibagi menjadi field. Pertimbangan desain mencakup jumlah opcode, jumlah register, ukuran alamat, kebutuhan immediate, dan panjang instruksi.

Model	Karakteristik
Fixed Length	Instruksi memiliki panjang yang sama secara konseptual
Variable Length	Instruksi dapat memiliki panjang berbeda sesuai kebutuhan format
Multiple Formats	ISA menyediakan beberapa format untuk kelompok instruksi berbeda

Rancangan format memengaruhi kepadatan kode, proses decoding, serta fleksibilitas ekspresi operasi.

11.9 Mode Pengalamatan

Mode pengalamatan atau addressing mode adalah aturan yang digunakan CPU untuk menentukan lokasi atau nilai operand. Field alamat pada instruksi tidak selalu secara langsung menunjukkan nilai operand. Dalam beberapa mode, field tersebut digunakan sebagai dasar untuk memperoleh alamat efektif.

KONSEP ALAMAT EFEKTIF

Alamat efektif adalah alamat operasional yang diperoleh setelah aturan mode pengalamatan diterapkan. Pada beberapa mode, tidak diperlukan perhitungan alamat efektif karena operand tersedia secara langsung atau berada pada register.

11.10 Immediate Addressing

Pada immediate addressing, operand berupa nilai yang disimpan langsung sebagai bagian dari instruksi. CPU tidak perlu melakukan akses tambahan ke lokasi memori hanya untuk memperoleh nilai tersebut.

CONTOH KONSEPTUAL

ADD R1, #5 berarti menambahkan nilai langsung 5 ke operand yang ditentukan oleh R1 sesuai aturan ISA.

Keterbatasan utama immediate addressing adalah ukuran nilai dibatasi oleh jumlah bit yang tersedia pada field instruksi.

11.11 Register Addressing

Pada register addressing, field operand menunjukkan register yang menyimpan data. Operand dapat diakses melalui register CPU.

CONTOH KONSEPTUAL

ADD R1, R2 menggunakan operand yang berada pada register R1 dan R2 sesuai format instruksi.

Pendekatan ini memanfaatkan register sebagai penyimpanan sementara berkecepatan tinggi dan mengurangi kebutuhan akses memori untuk operand yang sudah tersedia di CPU.

11.12 Direct Addressing

Pada direct addressing, field alamat dalam instruksi secara konseptual menunjukkan lokasi memori tempat operand berada. CPU menggunakan informasi tersebut untuk mengakses memori.

MODEL

$EA = A$, dengan EA sebagai effective address dan A sebagai field alamat pada instruksi.

Jangkauan alamat yang dapat dinyatakan langsung dipengaruhi oleh ukuran field alamat.

11.13 Indirect Addressing

Pada indirect addressing, field alamat menunjuk ke lokasi yang berisi informasi alamat operand. Dengan demikian, diperlukan langkah tambahan untuk memperoleh alamat efektif.

MODEL

$EA = \text{Memory}[A]$. Operand kemudian diperoleh dari $\text{Memory}[EA]$.

Pendekatan ini meningkatkan fleksibilitas pengalamatan, tetapi secara konseptual dapat membutuhkan akses memori tambahan.

11.14 Register Indirect Addressing

Pada register indirect addressing, sebuah register menyimpan alamat lokasi operand. Isi register tidak diperlakukan langsung sebagai operand, melainkan sebagai penunjuk ke memori.

MODEL

$EA = (R)$, dengan R sebagai register yang menyimpan alamat operand.

Mode ini penting untuk memahami konsep pointer dan akses data melalui alamat yang disimpan dalam register.

11.15 Indexed Addressing

Indexed addressing menggunakan kombinasi alamat dasar dengan nilai indeks. Secara konseptual, alamat efektif dapat diperoleh dengan menambahkan informasi basis dan indeks sesuai aturan format.

MODEL UMUM

$EA = A + (IX)$, dengan A sebagai bagian alamat dan IX sebagai register indeks.

Mode ini berguna untuk mengakses elemen struktur data seperti array.

11.16 Base Addressing

Base addressing menggunakan register basis bersama displacement atau offset dari instruksi. Secara konseptual:

MODEL

$EA = (\text{Base Register}) + \text{displacement}$.

Pendekatan ini memungkinkan sekelompok data diakses relatif terhadap suatu alamat dasar.

11.17 Relative Addressing

Relative addressing menggunakan nilai relatif terhadap Program Counter. Mode ini banyak digunakan secara konseptual untuk menjelaskan percabangan ke lokasi yang berada pada jarak tertentu dari posisi instruksi.

MODEL

$EA = (PC) + \text{displacement}$.

Interpretasi waktu pembaruan PC dan detail perhitungan bergantung pada arsitektur, sehingga simulasi harus selalu mengikuti aturan ISA yang digunakan.

11.18 Auto-Increment dan Auto-Decrement

Pada beberapa rancangan, register yang digunakan sebagai penunjuk dapat diperbarui secara otomatis sebelum atau sesudah akses operand. Mekanisme ini mendukung pola traversal data.

Mode	Gagasan Konseptual
------	--------------------

Auto-Increment	Alamat diperoleh melalui register, kemudian register diperbarui setelah atau sesuai aturan akses
Auto-Decrement	Register diperbarui terlebih dahulu atau sesuai aturan akses, kemudian digunakan sebagai dasar alamat

Urutan tepat pembaruan merupakan bagian dari spesifikasi arsitektur.

11.19 Implied Addressing

Pada implied atau implicit addressing, operand tertentu tidak perlu ditulis secara eksplisit karena telah ditentukan oleh aturan instruksi atau state arsitektur.

Contohnya dapat berupa operasi yang secara konseptual menggunakan accumulator, stack pointer, atau register tertentu sebagai bagian dari definisi instruksi.

11.20 Perbandingan Mode Pengalamatan

Mode	Sumber Informasi	Kebutuhan Konseptual
Immediate	Nilai dalam instruksi	Tidak perlu mengambil operand dari lokasi terpisah
Register	Register	Akses register
Direct	Alamat dalam instruksi	Akses operand ke memori
Indirect	Alamat menunjuk ke alamat lain	Dapat memerlukan akses tambahan
Register Indirect	Register berisi alamat	Akses memori melalui alamat register
Indexed	Alamat + indeks	Perhitungan alamat efektif
Base	Base register + displacement	Perhitungan alamat efektif
Relative	PC + displacement	Perhitungan alamat relatif
Implied	Ditentukan instruksi	Operand tersirat

11.21 Studi Kasus 1: Mengakses Array

Misalkan sebuah array memiliki alamat dasar tertentu dan register indeks menyimpan posisi elemen. Dengan pendekatan indexed addressing, CPU dapat menghitung alamat elemen berdasarkan kombinasi alamat dasar dan indeks.

Mahasiswa diminta membuat tabel yang memuat alamat dasar, nilai indeks, ukuran elemen secara konseptual, dan alamat efektif. Jika ukuran elemen lebih dari satu satuan alamat, faktor skala harus mengikuti aturan arsitektur atau model yang ditetapkan.

11.22 Studi Kasus 2: Pointer dan Register Indirect

Sebuah register menyimpan alamat suatu data. Instruksi menggunakan register indirect untuk mengambil operand dari lokasi tersebut.

State	Nilai Konseptual
R1	500
Memory[500]	37
Operasi	LOAD R2, (R1)
Hasil	R2 memperoleh nilai 37 dalam model konseptual

Kasus ini menunjukkan perbedaan antara nilai alamat yang berada pada register dan data yang tersimpan pada alamat tersebut.

11.23 Studi Kasus 3: Percabangan Relatif

Misalkan PC memiliki nilai tertentu dan instruksi branch menyimpan displacement. Dengan relative addressing, alamat target diperoleh melalui aturan yang menggabungkan nilai PC dengan displacement.

CONTOH LATIHAN

Jika pada model pembelajaran PC yang digunakan untuk perhitungan bernilai 1000 dan displacement bernilai 24, maka $EA = 1024$. Mahasiswa harus memperhatikan bahwa implementasi nyata dapat menentukan titik pembacaan PC yang berbeda.

11.24 Hubungan ISA dengan Siklus Instruksi

ISA menentukan informasi yang harus dipahami CPU ketika melakukan decode. Mode pengalamatan menentukan langkah operand access, sedangkan opcode menentukan operasi execute.

Tahap Siklus	Hubungan dengan ISA
Fetch	Mengambil representasi instruksi
Decode	Menginterpretasikan opcode dan format
Operand Access	Menerapkan aturan mode pengalamatan
Execute	Menjalankan operasi yang ditentukan opcode
Write Back	Menyimpan hasil sesuai aturan instruksi

Dengan demikian, siklus instruksi dan ISA saling berkaitan: ISA menjelaskan aturan operasi, sedangkan organisasi CPU menyediakan mekanisme internal untuk menjalankan aturan tersebut.

11.25 RISC dan CISC: Pengantar Konseptual

Dalam pembahasan organisasi komputer, istilah RISC dan CISC sering digunakan untuk menggambarkan kecenderungan desain set instruksi. RISC secara historis dikaitkan dengan pendekatan set instruksi yang lebih

sederhana dan organisasi yang mendukung eksekusi serta decoding yang lebih teratur. CISC dikaitkan dengan set instruksi yang dapat menyediakan operasi dan format yang lebih beragam.

Perbedaan implementasi prosesor modern tidak selalu dapat disederhanakan hanya melalui dua label tersebut. Pembahasan ini digunakan sebagai pengantar sebelum analisis organisasi instruksi yang lebih spesifik.

Aspek Konseptual	Kecenderungan RISC	Kecenderungan CISC
Instruksi	Lebih sederhana atau teratur	Dapat lebih beragam dan kompleks
Format	Sering dikaitkan dengan keteraturan	Dapat menggunakan variasi panjang/format
Operasi	Sering menekankan operasi sederhana	Dapat menyediakan instruksi yang lebih kaya
Catatan	Implementasi modern dapat menggabungkan berbagai teknik	Tidak semua karakteristik berlaku mutlak

11.26 Aktivitas Pembelajaran Mahasiswa

Aktivitas 1: Identifikasi ISA

Jelaskan komponen ISA yang berkaitan dengan operasi, register, format instruksi, dan mode pengalamatan.

Aktivitas 2: Analisis Instruksi

Pisahkan opcode dan operand dari lima contoh instruksi konseptual.

Aktivitas 3: Alamat Efektif

Hitung EA untuk contoh direct, indirect, indexed, base, dan relative addressing berdasarkan nilai yang diberikan dosen.

Aktivitas 4: Array

Simulasikan akses beberapa elemen array menggunakan indexed addressing.

Aktivitas 5: Pointer

Buat tabel state register dan memori untuk register indirect addressing.

Aktivitas 6: Branch

Simulasikan perubahan alamat target menggunakan PC-relative addressing.

Aktivitas 7: Analisis

Bandingkan kelebihan dan keterbatasan immediate, register, dan indirect addressing.

11.27 Rangkuman

- Instruction Set Architecture menjelaskan aturan konseptual instruksi yang dapat digunakan perangkat lunak untuk berinteraksi dengan arsitektur prosesor.

- Set instruksi mencakup operasi transfer data, aritmetika, logika, kontrol, dan operasi lain sesuai rancangan ISA.
- Instruksi mesin umumnya memuat opcode dan informasi operand.
- Operand dapat berasal dari register, memori, nilai immediate, atau sumber tersirat.
- Format instruksi menentukan pembagian bit ke dalam field yang memiliki fungsi berbeda.
- Mode pengalamatan menentukan cara CPU memperoleh nilai atau lokasi operand.
- Immediate addressing menyediakan nilai langsung di dalam instruksi.
- Register addressing menggunakan register sebagai lokasi operand.
- Direct, indirect, dan register indirect membedakan cara informasi alamat diperoleh.
- Indexed dan base addressing mendukung pengaksesan data relatif terhadap indeks atau alamat dasar.
- Relative addressing menggunakan hubungan dengan Program Counter.
- Pemilihan mode pengalamatan memengaruhi fleksibilitas, panjang instruksi, dan kebutuhan akses.
- Mode pengalamatan diterapkan terutama pada tahap operand access dalam siklus instruksi.
- ISA dan organisasi CPU saling melengkapi dalam menjelaskan apa yang harus dilakukan instruksi dan bagaimana instruksi tersebut dijalankan.

11.28 Glosarium Bab

Istilah	Pengertian
Addressing Mode	Aturan untuk menentukan cara memperoleh operand atau alamat operand.
Base Addressing	Pengalamatan yang menggabungkan register basis dan displacement.
Direct Addressing	Field alamat secara konseptual menunjukkan lokasi operand.
Effective Address	Alamat operasional yang diperoleh setelah aturan pengalamatan diterapkan.
Immediate	Nilai operand tersedia langsung pada instruksi.
Indexed Addressing	Pengalamatan yang menggunakan alamat dasar dan informasi indeks.
Indirect Addressing	Alamat menunjuk ke lokasi yang menyimpan alamat operand.
Instruction Set	Kumpulan operasi yang dapat dipahami dan dijalankan oleh arsitektur prosesor.
ISA	Spesifikasi konseptual antarmuka instruksi antara perangkat lunak dan arsitektur komputer.
Opcode	Bagian instruksi yang menentukan operasi.
Operand	Data atau informasi yang digunakan instruksi.
PC-relative	Pengalamatan relatif terhadap Program Counter.
Register Addressing	Operand ditentukan melalui register.
Register Indirect	Register menyimpan alamat operand di memori.

RISC	Istilah untuk kecenderungan desain dengan instruksi yang lebih sederhana/teratur.
CISC	Istilah untuk kecenderungan desain dengan variasi instruksi yang dapat lebih kaya.

11.29 Soal Latihan

A. Soal Pemahaman

- Jelaskan pengertian Instruction Set Architecture.
- Apa perbedaan antara ISA dan organisasi internal CPU?
- Jelaskan fungsi opcode dalam sebuah instruksi.
- Apa yang dimaksud operand?
- Jelaskan perbedaan instruksi satu alamat, dua alamat, dan tiga alamat secara konseptual.
- Mengapa arsitektur dapat memiliki beberapa format instruksi?
- Jelaskan perbedaan immediate dan register addressing.
- Bandingkan direct, indirect, dan register indirect addressing.
- Jelaskan konsep effective address.
- Bagaimana indexed addressing membantu akses struktur data?

B. Soal Perhitungan dan Analisis Alamat

- Dalam model direct addressing, field A=500. Tentukan EA.
- Dalam model indirect addressing, A=200 dan Memory[200]=750. Tentukan EA.
- Dalam register indirect, R1=640. Tentukan EA.
- Dalam indexed addressing, A=1000 dan IX=36. Tentukan EA.
- Dalam base addressing, Base Register=4000 dan displacement=120. Tentukan EA.
- Dalam model PC-relative, PC=2500 dan displacement=-40. Tentukan EA.
- Buat tabel perubahan register dan memori untuk instruksi konseptual LOAD R2, (R1) jika R1 menunjuk ke lokasi yang berisi nilai 99.
- Analisis jumlah langkah konseptual yang diperlukan untuk memperoleh operand pada immediate, register, direct, dan indirect addressing.

C. Soal HOTS dan Studi Kasus

- Rancang format instruksi konseptual sederhana yang mendukung ADD, LOAD, STORE, dan BRANCH. Jelaskan pembagian field yang Anda pilih.
- Sebuah sistem sering mengakses array. Pilih mode pengalamatan yang sesuai dan jelaskan alasan desainnya.
- Bandingkan dampak penggunaan field alamat yang pendek dan panjang terhadap ukuran instruksi dan jangkauan alamat.
- Analisis trade-off antara register indirect dan indirect addressing dari sudut jumlah akses yang diperlukan.
- Buat simulasi lima instruksi dengan kombinasi register, immediate, dan indexed addressing. Tunjukkan perubahan PC, register, alamat efektif, dan hasil operasi.

11.30 Proyek Mini

Judul proyek: Instruction Set dan Addressing Mode Simulator.

Mahasiswa membuat aplikasi, program, spreadsheet interaktif, atau simulator sederhana untuk memvisualisasikan proses decoding dan pengalamatan operand.

- Menyediakan model register minimal PC, IR, dua general purpose register, dan satu register indeks atau basis.
- Menyediakan model memori sederhana.
- Mendukung minimal empat mode pengalamatan.
- Menghitung dan menampilkan effective address jika diperlukan.
- Mendukung minimal empat jenis instruksi.
- Menampilkan proses decode opcode.
- Menampilkan sumber operand dan lokasi tujuan.
- Menampilkan perubahan register atau memori setelah execute.
- Menyediakan minimal lima skenario pengujian.

OUTPUT YANG DIHARAPKAN

Dokumentasi format instruksi, tabel opcode, simulasi mode pengalamatan, tabel perhitungan alamat efektif, hasil pengujian, dan analisis kelebihan serta keterbatasan setiap mode.

11.31 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.
- Mano, M. M. Computer System Architecture. Pearson.
- Harris, D. M., & Harris, S. L. Digital Design and Computer Architecture. Morgan Kaufmann.
- 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 melanjutkan pembahasan organisasi CPU dengan menempatkan instruksi sebagai penghubung antara program dan mekanisme internal prosesor. Pemahaman mengenai opcode, operand, format instruksi, dan mode pengalamatan menjadi landasan untuk pembahasan lebih lanjut mengenai desain instruksi, organisasi datapath, memori, dan eksekusi program.