

BUKU AJAR
MATA KULIAH ARSITEKTUR KOMPUTER
BAB 9
REPRESENTASI DATA DAN SISTEM
BILANGAN
DALAM KOMPUTER

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

IDENTITAS BAB

Komponen	Keterangan
Mata Kuliah	Arsitektur Komputer
Kode	IF221101
Bobot	3 SKS
Bab	Bab 9
Judul	Representasi Data dan Sistem Bilangan dalam Komputer
Fokus Pembelajaran	Memahami bagaimana data numerik dan nonnumerik direpresentasikan, disimpan, dan diproses dalam sistem komputer
Level Pembelajaran	Pemahaman konsep, konversi sistem bilangan, analisis representasi data, dan evaluasi keterbatasan representasi digital

CAPAIAN PEMBELAJARAN BAB

Setelah mempelajari bab ini, mahasiswa diharapkan mampu:

1. Menjelaskan peran sistem bilangan dalam representasi data komputer.
2. Membedakan sistem bilangan biner, oktal, desimal, dan heksadesimal.
3. Mengonversi bilangan antar basis dengan prosedur yang tepat.
4. Menjelaskan representasi integer tak bertanda dan bertanda.
5. Menerapkan representasi sign-magnitude, ones complement, dan twos complement.
6. Menjelaskan konsep overflow dalam operasi bilangan digital.
7. Membedakan fixed-point dan floating-point.
8. Menjelaskan komponen dasar representasi floating-point.
9. Menjelaskan representasi karakter menggunakan ASCII dan Unicode.
10. Menganalisis dampak keterbatasan jumlah bit terhadap rentang dan ketelitian representasi data.

PETA KONSEP

ALUR KONSEP BAB 9

Data Digital → Bit dan Byte → Sistem Bilangan → Biner / Oktal / Desimal / Heksadesimal → Konversi → Integer Unsigned → Integer Signed → Sign-Magnitude → Ones Complement → Twos Complement → Overflow → Fixed-Point → Floating-Point → ASCII → Unicode → Representasi Data dalam Memori.

9.1 Pendahuluan

Komputer menyimpan dan memproses berbagai bentuk informasi, seperti angka, teks, gambar, instruksi, alamat, dan simbol. Meskipun bentuk informasi tersebut tampak berbeda bagi pengguna, pada tingkat digital semuanya direpresentasikan melalui pola bit.

Pemahaman representasi data sangat penting dalam arsitektur komputer karena jumlah bit yang tersedia menentukan rentang nilai, ketelitian, cara penanganan bilangan negatif, serta kapasitas representasi informasi. Sebuah nilai desimal yang mudah dipahami manusia harus diterjemahkan ke bentuk yang dapat diproses oleh perangkat digital.

PERTANYAAN PEMANTIK

Mengapa komputer menggunakan biner? Bagaimana bilangan negatif direpresentasikan? Mengapa 0,1 dalam perhitungan komputer tidak selalu dapat disimpan secara tepat? Bagaimana huruf, angka, dan simbol dapat diubah menjadi pola bit?

9.2 Data, Bit, Byte, dan Unit Representasi

Bit merupakan unit informasi digital paling kecil yang secara konseptual memiliki dua keadaan, yaitu 0 dan 1. Sekelompok bit membentuk unit yang lebih besar. Dalam penggunaan umum, delapan bit membentuk satu byte.

Unit	Hubungan Konseptual
Bit	Satu digit biner
Nibble	4 bit
Byte	8 bit
Word	Ukuran unit data yang bergantung pada arsitektur
Kilobyte dan seterusnya	Satuan kapasitas yang digunakan untuk menggambarkan penyimpanan

Ukuran word dapat berbeda antar arsitektur. Oleh karena itu, istilah word perlu dipahami dalam konteks sistem yang sedang dibahas.

9.3 Sistem Bilangan Posisional

Sistem bilangan menggunakan basis atau radix. Nilai suatu digit tidak hanya ditentukan oleh simbol digit tersebut, tetapi juga oleh posisinya. Pada bilangan berbasis b , posisi memiliki bobot pangkat b .

PRINSIP UMUM

Untuk bilangan $d_n d_{n-1} \dots d_0$ berbasis b , nilai dapat dihitung sebagai jumlah setiap digit dikalikan dengan b berpangkat posisi digit tersebut.

Sistem yang paling sering digunakan dalam pembahasan komputer adalah biner (basis 2), oktal (basis 8), desimal (basis 10), dan heksadesimal (basis 16).

9.4 Sistem Bilangan Biner

Sistem biner hanya menggunakan digit 0 dan 1. Setiap posisi dari kanan ke kiri memiliki bobot 2^0 , 2^1 , 2^2 , dan seterusnya.

Contoh: $101101_2 = 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 45_{10}$.

Posisi	5	4	3	2	1	0
Bobot	32	16	8	4	2	1
Digit 101101	1	0	1	1	0	1

9.5 Sistem Bilangan Oktal

Sistem oktal menggunakan digit 0 sampai 7. Karena 8 merupakan 2^3 , setiap digit oktal dapat dipetakan secara langsung ke tiga bit biner.

Oktal	Biner
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Contoh: 725_8 dapat diubah ke biner dengan memetakan setiap digit: $7 \rightarrow 111$, $2 \rightarrow 010$, $5 \rightarrow 101$ sehingga diperoleh 111010101_2 .

9.6 Sistem Bilangan Heksadesimal

Heksadesimal menggunakan enam belas simbol: 0 sampai 9 dan A sampai F. Karena 16 merupakan 2^4 , setiap digit heksadesimal dapat direpresentasikan oleh empat bit.

Heksadesimal	Desimal	Biner
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001

A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

Contoh: $2F_{16} = 0010\ 1111_2 = 47_{10}$. Heksadesimal banyak digunakan sebagai notasi ringkas untuk pola bit yang panjang.

9.7 Konversi Antar Sistem Bilangan

9.7.1 Desimal ke Biner

Bilangan bulat desimal dapat dikonversi ke biner melalui pembagian berulang dengan basis 2 dan membaca sisa pembagian dari bawah ke atas.

CONTOH

45_{10} : $45 \div 2$ sisa 1; $22 \div 2$ sisa 0; $11 \div 2$ sisa 1; $5 \div 2$ sisa 1; $2 \div 2$ sisa 0; $1 \div 2$ sisa 1. Hasilnya 101101_2 .

9.7.2 Biner ke Desimal

Konversi dilakukan dengan menjumlahkan nilai setiap bit sesuai bobot posisinya.

9.7.3 Biner ke Oktal

Kelompokkan bit dari kanan menjadi kelompok tiga bit. Setiap kelompok dikonversi ke satu digit oktal.

9.7.4 Biner ke Heksadesimal

Kelompokkan bit dari kanan menjadi kelompok empat bit. Setiap kelompok dikonversi ke satu digit heksadesimal.

CATATAN

Oktal dan heksadesimal berguna karena memberikan representasi yang lebih ringkas terhadap bilangan biner tanpa mengubah pola bit yang mendasarinya.

9.8 Bilangan Pecahan dalam Berbagai Basis

Bilangan pecahan memiliki bagian di sebelah kanan titik radix. Dalam basis 2, posisi setelah titik memiliki bobot 2^{-1} , 2^{-2} , 2^{-3} , dan seterusnya.

Contoh: $0.101_2 = 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} = 0,5 + 0 + 0,125 = 0,625_{10}$.

Untuk mengubah pecahan desimal ke biner, bagian pecahan dapat dikalikan berulang dengan 2. Bagian integer dari setiap hasil digunakan sebagai digit berikutnya.

PERHATIAN

Tidak semua pecahan desimal memiliki representasi biner yang berhingga. Kondisi ini menjadi salah satu alasan munculnya isu ketelitian pada representasi floating-point.

9.9 Integer Tak Bertanda (Unsigned Integer)

Unsigned integer menggunakan seluruh bit untuk merepresentasikan nilai nonnegatif. Dengan n bit, rentang nilainya adalah 0 sampai $2^n - 1$.

Jumlah Bit	Rentang Unsigned
4	0 sampai 15
8	0 sampai 255
16	0 sampai 65.535
32	0 sampai 4.294.967.295

Jumlah bit yang lebih besar memberikan rentang nilai yang lebih luas, tetapi membutuhkan kapasitas penyimpanan lebih besar.

9.10 Integer Bertanda (Signed Integer)

Sistem komputer juga perlu merepresentasikan bilangan negatif. Beberapa pendekatan historis dan konseptual yang penting adalah sign-magnitude, ones complement, dan twos complement.

9.10.1 Sign-Magnitude

Pada representasi sign-magnitude, satu bit digunakan sebagai tanda dan bit lainnya menyatakan besar nilai. Secara konseptual, 0 menunjukkan tanda positif dan 1 menunjukkan tanda negatif.

Kelemahan pendekatan ini adalah adanya dua representasi untuk nol, yaitu +0 dan -0.

9.10.2 Ones Complement

Ones complement diperoleh dengan membalik setiap bit dari representasi positif. Pendekatan ini juga memiliki dua representasi untuk nol.

CONTOH KONSEPTUAL

Untuk pola 4-bit, representasi +5 adalah 0101. Ones complement dari +5 adalah 1010 yang merepresentasikan -5 dalam sistem tersebut.

9.10.3 Twos Complement

Twos complement diperoleh dengan membalik seluruh bit kemudian menambahkan 1. Pendekatan ini sangat penting dalam pembahasan sistem komputer modern karena memungkinkan operasi penjumlahan dan pengurangan ditangani dengan mekanisme biner yang konsisten.

CONTOH

Untuk 4 bit: +5 = 0101. Balik bit → 1010. Tambahkan 1 → 1011. Dengan demikian, 1011 merepresentasikan

–5 dalam twos complement 4-bit.

Jumlah Bit	Rentang Twos Complement
4	–8 sampai 7
8	–128 sampai 127
16	–32.768 sampai 32.767
32	–2.147.483.648 sampai 2.147.483.647

9.11 Operasi Aritmetika dan Overflow

Operasi penjumlahan dalam sistem digital dilakukan pada jumlah bit yang terbatas. Jika hasil berada di luar rentang representasi, terjadi kondisi overflow.

CONTOH 4-BIT TWOS COMPLEMENT

7 direpresentasikan sebagai 0111. Jika ditambahkan 1, pola 4-bit menjadi 1000. Pola tersebut tidak dapat diinterpretasikan sebagai +8 pada rentang twos complement 4-bit karena +8 berada di luar rentang –8 sampai 7.

Overflow perlu dibedakan dari carry biasa. Interpretasi hasil bergantung pada apakah bilangan diperlakukan sebagai unsigned atau signed.

9.12 Fixed-Point

Pada fixed-point, posisi titik radix dianggap tetap. Sebagian bit digunakan untuk bagian integer dan sebagian lainnya untuk bagian pecahan sesuai format yang telah ditentukan.

Pendekatan fixed-point dapat memberikan representasi yang sederhana dan dapat diprediksi, tetapi rentang serta ketelitian ditentukan langsung oleh pembagian bit yang tersedia.

Aspek	Fixed-Point
Posisi titik radix	Tetap
Rentang	Bergantung pada jumlah bit integer
Ketelitian	Bergantung pada jumlah bit pecahan
Kelebihan	Model representasi relatif sederhana

9.13 Floating-Point

Floating-point digunakan untuk merepresentasikan nilai dengan rentang yang lebih luas. Secara konseptual, sebuah nilai direpresentasikan menggunakan tiga komponen utama: sign, exponent, dan significand atau fraction.

Komponen	Fungsi
Sign	Menunjukkan tanda nilai
Exponent	Menentukan skala atau orde nilai
Significand/Fraction	Menyimpan bagian signifikan dari nilai

Model floating-point memungkinkan representasi nilai yang sangat besar maupun sangat kecil, tetapi tidak menjamin bahwa semua bilangan desimal dapat direpresentasikan secara tepat.

9.13.1 Normalisasi

Normalisasi menuliskan nilai biner dalam bentuk yang mempertahankan digit signifikan dan menggunakan exponent untuk menggeser posisi titik radix. Konsep ini membantu menyediakan rentang representasi yang lebih luas.

9.13.2 Ketelitian Floating-Point

Jumlah bit pada significand membatasi jumlah digit biner yang dapat disimpan. Karena itu, hasil operasi floating-point dapat mengalami pembulatan.

STUDI REFLEKSI

Jika sebuah nilai tidak dapat direpresentasikan secara tepat dalam jumlah bit yang tersedia, sistem menyimpan pendekatan terhadap nilai tersebut. Akumulasi operasi dapat menghasilkan perbedaan kecil antara hasil matematis ideal dan hasil representasi komputer.

9.14 Representasi Karakter

Komputer tidak menyimpan huruf sebagai bentuk visual. Setiap karakter dikaitkan dengan nilai kode tertentu. Kode tersebut kemudian direpresentasikan dalam bentuk bit.

9.14.1 ASCII

ASCII merupakan salah satu sistem pengkodean karakter yang secara historis digunakan untuk huruf, angka, tanda baca, dan karakter kendali. Dalam representasi dasar, ASCII berkaitan dengan kumpulan karakter yang lebih terbatas dibandingkan kebutuhan bahasa dan simbol modern.

9.14.2 Unicode

Unicode menyediakan sistem pengkodean yang dirancang untuk mendukung kumpulan karakter dari berbagai bahasa dan simbol. Unicode memperluas kemampuan representasi teks dibandingkan pendekatan karakter yang lebih terbatas.

Konsep	Peran
Character	Entitas abstrak, misalnya huruf atau simbol
Code Point	Nilai identifikasi dalam ruang karakter
Encoding	Cara nilai direpresentasikan menjadi satuan penyimpanan
UTF-8	Salah satu bentuk encoding Unicode dengan panjang representasi yang bervariasi

9.15 Representasi Data Lain dalam Sistem Komputer

Prinsip representasi bit tidak hanya digunakan untuk integer dan karakter. Gambar dapat direpresentasikan sebagai kumpulan nilai piksel, suara sebagai sampel digital, dan instruksi sebagai pola bit yang ditafsirkan oleh prosesor sesuai arsitekturnya.

Jenis Data	Representasi Konseptual
Integer	Pola bit dengan format tertentu
Pecahan	Fixed-point atau floating-point
Karakter	Kode karakter dan encoding
Gambar	Nilai piksel dan komponen warna
Audio	Sampel digital
Instruksi	Kode operasi dan field sesuai format instruksi

9.16 Endianness: Pengantar Representasi Multi-Byte

Ketika sebuah nilai menggunakan lebih dari satu byte, urutan penyimpanan byte dalam memori dapat menjadi bagian dari rancangan arsitektur. Konsep umum yang dikenal adalah little-endian dan big-endian.

Pendekatan	Gagasan Umum
Big-endian	Byte yang paling signifikan ditempatkan terlebih dahulu pada urutan alamat yang lebih rendah
Little-endian	Byte yang paling rendah signifikan ditempatkan terlebih dahulu pada urutan alamat yang lebih rendah

Endianness tidak mengubah nilai abstrak, tetapi memengaruhi cara urutan byte ditafsirkan ketika data multi-byte dibaca atau dipertukarkan antar sistem.

9.17 Representasi Data dalam Memori

Memori menyimpan data sebagai pola bit dan byte. Makna suatu pola bergantung pada bagaimana pola tersebut ditafsirkan. Pola yang sama dapat memiliki interpretasi berbeda jika diperlakukan sebagai unsigned integer, signed integer, karakter, atau bagian dari format data lain.

CONTOH KONSEPTUAL

Pola 01000001 dapat ditafsirkan sebagai nilai numerik 65 dalam unsigned 8-bit atau sebagai karakter A dalam sistem pengkodean karakter tertentu.

Karena itu, representasi data dan konteks interpretasi merupakan dua aspek yang tidak dapat dipisahkan.

9.18 Studi Kasus 1: Sensor Suhu Digital

Sebuah sensor menghasilkan nilai pengukuran yang memiliki bagian integer dan pecahan. Perancang sistem harus menentukan apakah nilai akan disimpan menggunakan fixed-point atau floating-point.

Pertimbangan	Pertanyaan Analisis
Rentang	Seberapa besar dan kecil nilai yang harus direpresentasikan?
Ketelitian	Seberapa banyak digit pecahan yang diperlukan?
Kapasitas	Berapa bit yang tersedia?
Kompleksitas	Apakah representasi sederhana sudah mencukupi?

Mahasiswa diminta membandingkan konsekuensi penggunaan format yang berbeda berdasarkan kebutuhan sistem.

9.19 Studi Kasus 2: Overflow pada Counter

Sebuah counter unsigned 8-bit memiliki rentang 0 sampai 255. Ketika berada pada nilai maksimum dan ditambah 1, pola bit kembali ke 00000000 jika hanya delapan bit yang dipertahankan.

Kasus ini menunjukkan bahwa keterbatasan jumlah bit menghasilkan batas rentang representasi dan perilaku wrap-around dalam konteks tertentu.

9.20 Studi Kasus 3: Teks Multibahasa

Sebuah aplikasi perlu menyimpan teks yang mencakup alfabet Latin, karakter Asia, simbol matematika, dan emoji. Sistem pengkodean yang sangat terbatas tidak cukup untuk mendeskripsikan seluruh karakter tersebut.

Mahasiswa menganalisis alasan perlunya sistem representasi karakter yang lebih luas dan membedakan konsep karakter, code point, dan encoding.

9.21 Aktivitas Pembelajaran Mahasiswa

Aktivitas 1: Konversi Basis

Konversikan 93_{10} ke biner, oktal, dan heksadesimal.

Aktivitas 2: Analisis Bit

Tentukan nilai desimal dari 11010110_2 .

Aktivitas 3: Signed Integer

Representasikan -23 dalam twos complement 8-bit.

Aktivitas 4: Overflow

Analisis apakah operasi signed 8-bit $120 + 20$ dapat direpresentasikan tanpa overflow.

Aktivitas 5: Pecahan

Konversikan $0,625_{10}$ ke biner.

Aktivitas 6: Fixed vs Floating

Bandingkan dua pendekatan untuk menyimpan data harga dengan kebutuhan dua angka di belakang koma.

Aktivitas 7: Karakter

Analisis perbedaan antara karakter, code point, dan encoding menggunakan contoh teks multibahasa.

9.22 Rangkuman

- Komputer merepresentasikan berbagai bentuk data melalui pola bit.
- Sistem bilangan biner merupakan dasar representasi digital, sedangkan oktal dan heksadesimal memberikan notasi yang lebih ringkas.
- Konversi antar basis dilakukan dengan mempertimbangkan bobot posisi dan hubungan antar basis.
- Unsigned integer menggunakan seluruh bit untuk nilai nonnegatif.
- Signed integer dapat direpresentasikan melalui beberapa pendekatan, termasuk twos complement.
- Jumlah bit menentukan rentang nilai yang dapat direpresentasikan.
- Overflow terjadi ketika hasil berada di luar rentang format yang tersedia.
- Fixed-point menggunakan posisi titik radix yang tetap.
- Floating-point menggunakan sign, exponent, dan significand untuk menyediakan rentang representasi yang lebih luas.
- Keterbatasan jumlah bit dapat menyebabkan pembulatan dan ketelitian terbatas.
- Karakter direpresentasikan melalui kode dan encoding seperti ASCII dan Unicode.
- Pola bit memperoleh makna berdasarkan format dan konteks interpretasinya.
- Endianness berkaitan dengan urutan penyimpanan byte untuk data multi-byte.

9.23 Glosarium Bab

Istilah	Pengertian
ASCII	Sistem pengkodean karakter historis untuk karakter dasar dan karakter kendali.
Basis/Radix	Jumlah simbol dasar dalam suatu sistem bilangan.
Bit	Unit informasi digital dengan dua keadaan konseptual.
Byte	Sekelompok delapan bit dalam penggunaan umum.
Code Point	Nilai identifikasi karakter dalam sistem Unicode.
Endianness	Konvensi urutan penyimpanan byte pada data multi-byte.
Exponent	Komponen floating-point yang menentukan skala nilai.
Fixed-Point	Representasi numerik dengan posisi titik radix tetap.
Floating-Point	Representasi numerik dengan komponen sign, exponent, dan significand/fraction.
Heksadesimal	Sistem bilangan basis 16.

Ones Complement	Representasi negatif melalui pembalikan bit dari representasi positif.
Overflow	Kondisi ketika hasil berada di luar rentang representasi format.
Sign-Magnitude	Representasi signed yang memisahkan tanda dan besar nilai.
Twos Complement	Representasi negatif melalui pembalikan bit dan penambahan satu.
Unicode	Sistem representasi karakter yang mendukung banyak bahasa dan simbol.
Unsigned Integer	Representasi integer tanpa tanda negatif.

9.24 Soal Latihan

A. Soal Pemahaman

11. Jelaskan mengapa sistem komputer menggunakan representasi biner.
12. Bedakan sistem bilangan biner, oktal, desimal, dan heksadesimal.
13. Jelaskan hubungan antara jumlah bit dan rentang unsigned integer.
14. Bandingkan sign-magnitude, ones complement, dan twos complement.
15. Mengapa twos complement penting dalam representasi integer bertanda?
16. Jelaskan perbedaan overflow dan carry.
17. Bandingkan fixed-point dan floating-point.
18. Jelaskan fungsi sign, exponent, dan significand.
19. Jelaskan perbedaan ASCII dan Unicode.
20. Jelaskan mengapa pola bit yang sama dapat memiliki makna berbeda.

B. Soal Konversi dan Perhitungan

21. Konversikan 156_{10} ke biner dan heksadesimal.
22. Konversikan 10110111_2 ke desimal, oktal, dan heksadesimal.
23. Konversikan $7D_{16}$ ke desimal dan biner.
24. Representasikan -37 dalam twos complement 8-bit.
25. Tentukan rentang nilai signed twos complement 12-bit.
26. Analisis hasil $100 + 50$ pada signed integer 8-bit dan tentukan apakah terjadi overflow.
27. Konversikan $0,375_{10}$ ke biner.
28. Jika sebuah nilai unsigned menggunakan 10 bit, tentukan nilai maksimum yang dapat direpresentasikan.

C. Soal HOTS dan Studi Kasus

29. Sebuah sistem akuisisi data harus menyimpan nilai suhu dari -50 sampai 150 dengan ketelitian $0,01$.
Bandingkan secara konseptual penggunaan fixed-point dan floating-point.
30. Analisis risiko penggunaan jumlah bit yang terlalu kecil pada sistem pencacah transaksi.
31. Rancang format representasi sederhana untuk menyimpan skor yang memiliki rentang 0 sampai 100 dan dua digit pecahan. Jelaskan pertimbangan jumlah bit.

32. Jelaskan bagaimana perbedaan endianness dapat menimbulkan masalah interpretasi ketika data multi-byte dipertukarkan antar sistem.
33. Analisis mengapa hasil perhitungan floating-point perlu diperlakukan secara hati-hati ketika digunakan dalam perbandingan nilai yang sangat presisi.

9.25 Proyek Mini

Judul proyek: Konverter dan Analyzer Representasi Data Digital.

Mahasiswa membuat aplikasi, program, spreadsheet, atau simulator sederhana yang memiliki kemampuan:

- Mengonversi bilangan desimal, biner, oktal, dan heksadesimal.
- Menampilkan representasi integer unsigned dan twos complement.
- Mendeteksi kondisi overflow berdasarkan jumlah bit yang dipilih.
- Membandingkan rentang beberapa ukuran integer.
- Menampilkan proses konversi secara bertahap.
- Menampilkan representasi karakter berdasarkan konsep kode dan encoding.
- Menyediakan minimal lima skenario pengujian dengan analisis hasil.

OUTPUT YANG DIHARAPKAN

Dokumentasi rancangan, hasil konversi, analisis representasi signed dan unsigned, pengujian overflow, serta refleksi mengenai keterbatasan representasi data digital.

9.26 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., & Ciletti, M. D. Digital Design. Pearson.
- Harris, D. M., & Harris, S. L. Digital Design and Computer Architecture. Morgan Kaufmann.
- Englander, I., & Wong, W. The Architecture of Computer Hardware, Systems Software, and Networking. John Wiley & Sons.
- 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 membangun pemahaman bahwa seluruh data pada sistem komputer pada akhirnya direpresentasikan dalam bentuk pola bit. Materi sistem bilangan, integer bertanda, overflow, floating-point, dan pengkodean karakter menjadi landasan penting untuk pembahasan organisasi data, instruksi, memori, dan pemrosesan pada bab-bab berikutnya.