

BUKU AJAR

MATA KULIAH ARSITEKTUR KOMPUTER

BAB 3

SISTEM BILANGAN DAN REPRESENTASI DATA

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 3
Judul	Sistem Bilangan dan Representasi Data
Fokus RPS	Penerapan sistem biner, heksadesimal, atau oktal dalam program sederhana
Level Pembelajaran	Pemahaman konsep dan penerapan melalui latihan konversi serta representasi data

CAPAIAN PEMBELAJARAN BAB

Setelah mempelajari bab ini, mahasiswa diharapkan mampu:

1. Menjelaskan konsep sistem bilangan dan basis bilangan.
2. Membedakan sistem bilangan desimal, biner, oktal, dan heksadesimal.
3. Mengonversi bilangan dari satu basis ke basis lainnya.
4. Melakukan operasi dasar pada bilangan biner.
5. Menjelaskan representasi bilangan positif dan negatif dalam sistem komputer.
6. Menjelaskan konsep representasi integer dan floating point.
7. Menjelaskan representasi karakter menggunakan ASCII dan Unicode.
8. Menganalisis bagaimana data direpresentasikan dan diproses dalam sistem komputer.
9. Menerapkan konsep sistem bilangan dalam contoh program atau penyelesaian masalah sederhana.

PETA KONSEP

ALUR KONSEP BAB 3

Data → Sistem Bilangan → Basis Bilangan → Desimal | Biner | Oktal | Heksadesimal → Konversi → Operasi Biner → Representasi Integer → Bilangan Negatif → Floating Point → Representasi Karakter → ASCII dan Unicode → Implementasi dalam Sistem Komputer.

3.1 Pendahuluan

Komputer modern memproses berbagai bentuk informasi, mulai dari angka, teks, gambar, suara, video, hingga instruksi program. Walaupun bentuk informasi tersebut terlihat beragam bagi manusia, pada tingkat perangkat keras informasi harus direpresentasikan dalam bentuk yang dapat disimpan, dipindahkan, dan diproses oleh sistem digital.

Salah satu fondasi utama representasi tersebut adalah sistem bilangan. Manusia sehari-hari menggunakan sistem desimal berbasis 10, sedangkan sistem komputer secara mendasar menggunakan representasi biner berbasis 2.

Sistem oktal dan heksadesimal digunakan sebagai bentuk representasi yang lebih ringkas dan lebih mudah dibaca manusia ketika bekerja dengan data biner.

Pemahaman sistem bilangan sangat penting dalam Arsitektur Komputer karena konsep ini menjadi dasar untuk memahami alamat memori, instruksi mesin, register, data integer, representasi karakter, dan berbagai operasi yang dilakukan oleh perangkat keras.

PERTANYAAN PEMANTIK

Mengapa komputer menggunakan 0 dan 1? Mengapa programmer sering menggunakan bilangan heksadesimal ketika bekerja dengan alamat memori atau data tingkat rendah?

3.2 Konsep Sistem Bilangan

Sistem bilangan adalah cara untuk merepresentasikan nilai menggunakan seperangkat simbol dan aturan posisi. Setiap sistem memiliki basis atau radix. Basis menunjukkan jumlah simbol dasar yang digunakan dalam sistem tersebut.

Sistem	Basis	Simbol
Desimal	10	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Biner	2	0, 1
Oktal	8	0 sampai 7
Heksadesimal	16	0 sampai 9 dan A sampai F

Nilai suatu digit ditentukan oleh posisi dan basis. Secara umum, bilangan dengan digit d dapat dipandang sebagai penjumlahan digit yang dikalikan dengan pangkat basis sesuai posisinya.

3.3 Sistem Bilangan Desimal

Sistem desimal menggunakan basis 10. Bilangan 572 dapat diuraikan menjadi $5 \times 10^2 + 7 \times 10^1 + 2 \times 10^0$. Posisi paling kanan memiliki bobot satuan, sedangkan posisi di sebelah kirinya memiliki bobot puluhan, ratusan, dan seterusnya.

Bilangan	Uraian	Nilai
572_{10}	$5 \times 10^2 + 7 \times 10^1 + 2 \times 10^0$	572
348_{10}	$3 \times 10^2 + 4 \times 10^1 + 8 \times 10^0$	348

3.4 Sistem Bilangan Biner

Sistem biner menggunakan dua simbol, yaitu 0 dan 1. Dua nilai tersebut sesuai dengan konsep keadaan digital yang dapat dibedakan, misalnya rendah dan tinggi, tidak aktif dan aktif, atau dua keadaan logika.

Contoh: $1011_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 8 + 0 + 2 + 1 = 11_{10}$.

Posisi	2^3	2^2	2^1	2^0
--------	-------	-------	-------	-------

Bobot	8	4	2	1
Digit pada 1011_2	1	0	1	1

3.5 Sistem Bilangan Oktal

Sistem oktal menggunakan basis 8 dan simbol 0 sampai 7. Oktal memiliki hubungan langsung dengan biner karena satu digit oktal dapat direpresentasikan oleh tiga bit biner.

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

Contoh: 725_8 dapat dikonversi ke desimal sebagai $7 \times 8^2 + 2 \times 8^1 + 5 \times 8^0 = 448 + 16 + 5 = 469_{10}$.

3.6 Sistem Bilangan Heksadesimal

Sistem heksadesimal menggunakan basis 16. Setelah digit 9, digunakan simbol A, B, C, D, E, dan F untuk mewakili nilai 10 sampai 15. Heksadesimal sangat berguna karena satu digit heksadesimal dapat merepresentasikan empat bit biner.

Hex	Desimal	Biner 4-bit
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: $2F_{16} = 2 \times 16^1 + 15 \times 16^0 = 32 + 15 = 47_{10}$.

3.7 Konversi Antar Sistem Bilangan

3.7.1 Desimal ke Biner

Konversi bilangan bulat desimal ke biner dapat dilakukan dengan pembagian berulang terhadap 2. Sisa pembagian dibaca dari bawah ke atas.

Langkah	Pembagian	Sisa
1	$25 \div 2$	1
2	$12 \div 2$	0
3	$6 \div 2$	0
4	$3 \div 2$	1
5	$1 \div 2$	1

Dengan membaca sisa dari bawah ke atas, diperoleh $25_{10} = 11001_2$.

3.7.2 Biner ke Desimal

Konversi biner ke desimal dilakukan dengan menjumlahkan setiap digit yang dikalikan dengan pangkat 2 sesuai posisinya.

Contoh: $11010_2 = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 16 + 8 + 2 = 26_{10}$.

3.7.3 Biner ke Oktal dan Sebaliknya

Untuk mengonversi biner ke oktal, digit biner dikelompokkan tiga bit dari kanan ke kiri. Setiap kelompok kemudian diganti dengan digit oktal yang ekuivalen.

Contoh: $111\ 010\ 101_2 = 7\ 2\ 5_8$, sehingga $111010101_2 = 725_8$.

3.7.4 Biner ke Heksadesimal dan Sebaliknya

Untuk mengonversi biner ke heksadesimal, digit biner dikelompokkan empat bit dari kanan ke kiri.

Contoh: $1011\ 1110_2 = BE_{16}$.

3.7.5 Konversi Desimal ke Oktal dan Heksadesimal

Metode pembagian berulang juga dapat digunakan. Untuk oktal dilakukan pembagian berulang terhadap 8, sedangkan untuk heksadesimal dilakukan pembagian berulang terhadap 16.

TIPS BELAJAR

Untuk konversi yang melibatkan biner, oktal, dan heksadesimal, pemetaan 3-bit untuk oktal dan 4-bit untuk heksadesimal merupakan cara yang cepat dan sistematis.

3.8 Operasi Aritmetika Biner

3.8.1 Penjumlahan Biner

A	B	Hasil	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Contoh: $1011_2 + 0101_2 = 10000_2$. Penjumlahan dilakukan dari digit paling kanan dengan memperhatikan carry.

3.8.2 Pengurangan Biner

Pengurangan biner mengikuti prinsip yang mirip dengan pengurangan desimal. Ketika digit yang dikurangi lebih kecil daripada digit pengurang, dilakukan peminjaman dari posisi yang lebih tinggi.

Contoh sederhana: $1010_2 - 0011_2 = 0111_2$.

3.8.3 Multiplikasi Biner

Multiplikasi biner memanfaatkan fakta bahwa 1×1 menghasilkan 1, sedangkan kombinasi yang melibatkan 0 menghasilkan 0. Hasil parsial kemudian dijumlahkan sesuai posisi.

Contoh: $101_2 \times 11_2$ menghasilkan 1111_2 , yang ekuivalen dengan $5 \times 3 = 15$ dalam desimal.

3.9 Bit, Byte, Word, dan Unit Data

Bit adalah unit data terkecil yang dapat bernilai 0 atau 1. Sekelompok delapan bit secara umum disebut byte. Dalam konteks arsitektur komputer, word merupakan satuan data yang secara alami diproses oleh arsitektur tertentu.

Satuan	Makna
Bit	Satu digit biner
Nibble	Kelompok 4 bit
Byte	Kelompok 8 bit
Word	Ukuran data alami yang digunakan arsitektur
Kilobyte, Megabyte, Gigabyte	Satuan kapasitas penyimpanan dengan konvensi yang digunakan sistem

3.10 Representasi Bilangan Positif dan Negatif

Komputer harus memiliki cara untuk membedakan nilai positif dan negatif. Beberapa pendekatan historis dan konseptual meliputi sign-magnitude, komplemen satu, dan komplemen dua. Pada sistem komputer modern, komplemen dua merupakan representasi yang sangat penting untuk bilangan integer bertanda.

3.10.1 Sign-Magnitude

Bit tertentu digunakan sebagai tanda, sementara bit lainnya menyatakan magnitudo. Pendekatan ini mudah dipahami, tetapi memiliki dua representasi untuk nol.

3.10.2 Komplemen Satu

Bilangan negatif diperoleh dengan membalik setiap bit dari representasi positifnya. Pendekatan ini juga memiliki keterbatasan dalam representasi nol.

3.10.3 Komplemen Dua

Untuk memperoleh representasi negatif dalam komplemen dua, langkah umum adalah membalik seluruh bit lalu menambahkan 1. Pendekatan ini memungkinkan operasi aritmetika ditangani secara lebih seragam oleh perangkat keras.

Contoh 4-bit	Representasi
+5	0101
-5: balik 0101 → 1010, tambah 1	1011

PERHATIAN

Rentang nilai bergantung pada jumlah bit yang tersedia. Ketika hasil melebihi rentang representasi, dapat terjadi overflow.

3.11 Representasi Integer

Integer adalah bilangan bulat yang direpresentasikan menggunakan jumlah bit tertentu. Integer dapat berupa unsigned atau signed. Pada unsigned integer, seluruh bit digunakan untuk nilai non-negatif. Pada signed integer, salah satu pendekatan yang umum digunakan adalah komplemen dua.

Ukuran	Unsigned	Signed Komplemen Dua
8 bit	0 sampai 255	-128 sampai 127
16 bit	0 sampai 65.535	-32.768 sampai 32.767

Pemilihan ukuran integer merupakan kompromi antara rentang nilai dan kebutuhan memori.

3.12 Representasi Floating Point

Tidak semua nilai dapat direpresentasikan secara praktis sebagai integer. Bilangan pecahan memerlukan pendekatan floating point. Secara konseptual, floating point menggunakan komponen tanda, eksponen, dan significand atau mantissa.

Representasi floating point memberikan kemampuan merepresentasikan rentang nilai yang sangat besar, tetapi dapat memiliki keterbatasan presisi. Oleh karena itu, hasil operasi floating point tidak selalu identik dengan intuisi aritmetika desimal manusia.

CONTOH KONSEPTUAL

Bilangan 6,25 dapat dipahami sebagai bilangan biner dengan bagian pecahan. Representasi aktual dalam memori mengikuti aturan format floating point yang digunakan oleh sistem.

3.13 Representasi Karakter

Komputer tidak hanya menyimpan angka. Huruf, angka, tanda baca, dan simbol juga harus diberi representasi numerik. Sistem pengkodean karakter menyediakan pemetaan antara karakter dengan nilai tertentu.

3.13.1 ASCII

ASCII merupakan salah satu sistem pengkodean karakter yang secara historis banyak digunakan untuk karakter dasar. Sebagai contoh, karakter tertentu seperti huruf kapital, huruf kecil, angka, dan simbol diberi kode numerik.

3.13.2 Unicode

Unicode dikembangkan untuk mendukung representasi karakter dari berbagai bahasa dan sistem tulisan. Konsep ini penting dalam sistem komputer modern karena aplikasi dapat memproses teks multibahasa.

Konsep	Fokus
ASCII	Representasi karakter dasar
Unicode	Representasi karakter yang jauh lebih luas dan multibahasa
UTF-8	Salah satu cara pengodean Unicode yang umum digunakan

3.14 Representasi Data dalam Sistem Komputer

Pada tingkat perangkat keras, berbagai bentuk data harus diterjemahkan ke pola bit. Sebuah angka dapat direpresentasikan sebagai integer atau floating point. Sebuah karakter direpresentasikan melalui pengkodean karakter. Gambar dan suara juga pada akhirnya disimpan dalam bentuk data digital.

Perbedaan interpretasi sangat penting. Pola bit yang sama dapat memiliki makna berbeda tergantung konteks dan aturan yang digunakan oleh perangkat lunak atau perangkat keras.

3.15 Studi Kasus: Membaca Alamat dan Data Heksadesimal

Dalam pemrograman tingkat rendah, debugging, dan dokumentasi sistem, nilai heksadesimal sering digunakan untuk menyederhanakan pembacaan pola bit.

Misalkan sebuah nilai ditulis sebagai 0x3A. Digit 3 merepresentasikan 0011 dan A merepresentasikan 1010. Dengan demikian, 0x3A dapat ditulis sebagai 00111010₂ dan memiliki nilai desimal 58.

ANALISIS

Heksadesimal bukan format yang “lebih dipahami” oleh komputer daripada biner. Heksadesimal terutama membantu manusia membaca dan menulis kelompok bit secara lebih ringkas.

3.16 Contoh Penerapan dalam Program Sederhana

Konsep sistem bilangan dapat diterapkan dalam program sederhana, misalnya melakukan konversi basis, membaca flag menggunakan operasi bit, atau menampilkan nilai dalam format desimal dan heksadesimal.

PSEUDOCODE

```
INPUT n
basis_biner ← konversi(n, 2)
basis_oktal ← konversi(n, 8)
basis_hex ← konversi(n, 16)
TAMPILKAN n, basis_biner, basis_oktal, basis_hex
```

Mahasiswa dapat mengimplementasikan konsep tersebut menggunakan bahasa pemrograman yang dipelajari di program studi, dengan tetap memastikan bahwa hasil konversi diverifikasi melalui perhitungan manual.

3.17 Aktivitas Pembelajaran Mahasiswa

Aktivitas 1: Konversi Bilangan

Lakukan konversi 45_{10} , 127_{10} , dan 255_{10} ke bentuk biner, oktal, dan heksadesimal. Tunjukkan langkah perhitungan.

Aktivitas 2: Analisis Pola Bit

Diberikan nilai heksadesimal 7F dan A3. Ubah ke biner dan desimal. Identifikasi pola 4-bit yang digunakan.

Aktivitas 3: Bilangan Negatif

Representasikan -18 dalam format komplement dua 8-bit. Tunjukkan proses pembentukan representasi.

Aktivitas 4: Karakter dan Encoding

Bandingkan penggunaan ASCII dan Unicode dalam aplikasi yang harus mendukung teks Bahasa Indonesia, Jepang, Arab, dan simbol khusus.

Aktivitas 5: Implementasi Program

Buat program sederhana yang menerima bilangan desimal dan menampilkan representasinya dalam biner, oktal, dan heksadesimal.

3.18 Rangkuman

- Sistem bilangan menjadi dasar representasi dan pengolahan data dalam komputer.
- Sistem desimal menggunakan basis 10, biner basis 2, oktal basis 8, dan heksadesimal basis 16.
- Konversi antar sistem bilangan dapat dilakukan menggunakan metode nilai posisi, pembagian berulang, dan pengelompokan bit.

- Oktal berhubungan dengan kelompok 3 bit, sedangkan heksadesimal berhubungan dengan kelompok 4 bit.
- Operasi aritmetika biner menjadi dasar bagi proses aritmetika pada sistem digital.
- Bilangan negatif dapat direpresentasikan melalui beberapa metode, termasuk komplemen dua.
- Integer memiliki rentang yang bergantung pada jumlah bit dan metode representasinya.
- Floating point digunakan untuk merepresentasikan bilangan dengan bagian pecahan dan rentang nilai yang luas.
- Karakter direpresentasikan menggunakan sistem pengkodean seperti ASCII dan Unicode.
- Berbagai bentuk informasi pada akhirnya direpresentasikan sebagai data digital yang dapat disimpan dan diproses oleh sistem komputer.

3.19 Glosarium Bab

Istilah	Pengertian
ASCII	Sistem pengkodean karakter dasar yang menggunakan nilai numerik untuk merepresentasikan karakter.
Basis/Radix	Jumlah simbol dasar dalam suatu sistem bilangan.
Biner	Sistem bilangan berbasis 2.
Bit	Unit data biner bernilai 0 atau 1.
Byte	Kelompok delapan bit.
Floating Point	Metode representasi bilangan pecahan menggunakan komponen seperti tanda dan eksponen.
Heksadesimal	Sistem bilangan berbasis 16.
Integer	Bilangan bulat yang direpresentasikan menggunakan sejumlah bit.
Komplemen Dua	Metode representasi bilangan bertanda yang umum digunakan sistem komputer.
Oktal	Sistem bilangan berbasis 8.
Overflow	Kondisi ketika hasil tidak dapat direpresentasikan dalam rentang bit yang tersedia.
Unicode	Standar representasi karakter yang mendukung berbagai bahasa dan sistem tulisan.

3.20 Soal Latihan

A. Soal Pemahaman

10. Jelaskan perbedaan antara sistem desimal, biner, oktal, dan heksadesimal.
11. Mengapa sistem komputer menggunakan representasi biner pada tingkat dasar?
12. Konversikan 156_{10} ke biner.
13. Konversikan 11010110_2 ke desimal.
14. Konversikan 745_8 ke biner dan desimal.

15. Konversikan $3C_{16}$ ke biner dan desimal.
16. Jelaskan hubungan antara satu digit oktal dan kelompok bit.
17. Jelaskan hubungan antara satu digit heksadesimal dan kelompok bit.
18. Jelaskan perbedaan ASCII dan Unicode.
19. Apa yang dimaksud dengan overflow?

B. Soal Perhitungan dan Analisis

20. Hitung hasil $101101_2 + 11011_2$.
21. Hitung hasil $111000_2 - 10101_2$.
22. Representasikan -27 menggunakan komplemen dua 8-bit.
23. Tentukan rentang nilai unsigned dan signed komplemen dua untuk 6-bit.
24. Analisis mengapa nilai floating point tertentu dapat mengalami keterbatasan presisi.
25. Ubahlah alamat heksadesimal $0xAF21$ ke bentuk biner dan jelaskan keuntungan penggunaan notasi heksadesimal.

C. Soal HOTS dan Studi Kasus

26. Sebuah sistem embedded hanya memiliki memori terbatas. Analisis pertimbangan pemilihan ukuran integer untuk menyimpan data sensor.
27. Rancang strategi representasi data untuk aplikasi yang menyimpan suhu pecahan, status perangkat, dan teks multibahasa.
28. Sebuah program menghasilkan nilai berbeda antara perhitungan desimal yang diharapkan pengguna dan hasil floating point. Analisis kemungkinan penyebab secara konseptual.
29. Jelaskan bagaimana pemahaman biner dan heksadesimal membantu proses debugging perangkat lunak tingkat rendah.

3.21 Proyek Mini

Judul proyek: Aplikasi Konverter dan Visualisasi Representasi Data.

Mahasiswa membuat aplikasi sederhana yang menerima input bilangan dan menyediakan fungsi:

- Konversi desimal ke biner, oktal, dan heksadesimal.
- Konversi biner, oktal, atau heksadesimal ke desimal.
- Visualisasi posisi bit.
- Penjelasan representasi integer signed dan unsigned.
- Demonstrasi sederhana representasi karakter.
- Validasi hasil melalui minimal tiga contoh perhitungan manual.

3.22 Daftar Pustaka

- Comer, D. (2017). *Essentials of Computer Architecture*. Chapman and Hall/CRC.
- Englander, I., & Wong, W. (2021). *The Architecture of Computer Hardware, Systems Software, and Networking: An Information Technology Approach*. John Wiley & Sons.
- Heuring, V. P., & Murdocca, M. J. (2021). *Principles of Computer Architecture*.
- 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 disusun dengan mempertahankan sistem biner, heksadesimal, dan oktal sebagai inti yang secara eksplisit tercantum dalam aktivitas pembelajaran RPS. Pembahasan mengenai representasi integer, komplemen dua, floating point, ASCII, Unicode, dan penerapan program sederhana merupakan elaborasi dari outline buku ajar untuk memberikan kesinambungan konsep representasi data dalam mata kuliah Arsitektur Komputer.