

BAHAN AJAR PERKULIAHAN

SISTEM DIGITAL (TIF-1106)

Modul Minggu 2: Kode Digital (BCD 8421, Kode Gray, ASCII), Deteksi Error Paritas, & Aritmatika Penjumlahan BCD

Tim Dosen Teknik Informatika
Program Studi S1 Teknik Informatika — Fakultas Teknik

Semester Ganjil 2026/2027

IDENTITAS MATA KULIAH & CAPAIAN PEMBELAJARAN (SUB-CPMK 1)

Mata Kuliah	: Sistem Digital	Kode / Bobot	: TIF-1106 / 3 SKS
Semester	: Ganjil (Semester 1)	Target Pembelajaran	: Mahasiswa S1 Teknik Informatika

Sub-CPMK 1 (Lanjutan): Mahasiswa mampu mengonversi data desimal dan karakter ke dalam kode **BCD 8421**, **Kode Gray**, dan **ASCII**, menganalisis serta mengoreksi operasi penjumlahan BCD ber-overflow, serta merancang skema pembangkitan dan pemeriksaan bit paritas (**Even/Odd Parity**) untuk deteksi kesalahan transmisi data.

TUGAS MEMBACA WAJIB SEBELUM KELAS (*WEEKLY READING*)

Mahasiswa dipersyaratkan membaca bab-bab relevan pada 3 buku teks terbuka ([buku_referensi.pdf](#)):

1. **[Ref 1] Kuphaldt (2024):** *Lessons In Electric Circuits Vol. IV – Digital*, **Bab 3** (*Binary Codes & Data Representation*).
2. **[Ref 2] Coulston (2020):** *Digital Design: A Datapath and Control*, **Bab 1** (*Data Encodings & Error Detection*).
3. **[Ref 3] Coulston Labs (2020):** *Digital Design Laboratory Manual*, **Lab 2** (*Code Converters & Display Decoders*).

1. Pendahuluan & Klasifikasi Kode Digital

Dalam arsitektur komputer modern, sistem biner murni (0 dan 1) adalah representasi alami bagi prosesor ALU. Namun, antarmuka manusia-mesin (**Human-Machine Interface**) memerlukan representasi desimal, karakter teks, dan posisi fisis sensor mekanis. Untuk kebutuhan tersebut, dirancanglah berbagai sistem **Pengodean Digital** (**Digital Codes**).

Kategori Kode	Prinsip Dasar	Contoh Standar	Bidang Aplikasi Utama
Weighted Code	Posisi bit memiliki bobot nilai fisis matematis ($2^n, \dots$).	BCD 8421, BCD 2421, Excess-3	Kalkulator, Display 7-Segment
Unweighted Code	Posisi bit tidak memiliki bobot kuantitatif langsung.	Kode Gray (*Reflected Binary*)	Sensor Sudut Poros (*Shaft Encoder*)
Alphanumeric Code	Mengodekan teks, simbol, angka, dan perintah kontrol.	ASCII 7-Bit, Extended ASCII, UTF-8	Komunikasi Serial, Keyboard, File Teks
Error-Control Code	Menambahkan bit redundansi untuk deteksi/koreksi error.	Bit Paritas (Even/Odd), Checksum, CRC	Transmisi Jaringan, Bus Memori RAM

2. Kode BCD (Binary Coded Decimal 8421)

Kode **BCD 8421** (**B**inary **C**oded **D**ecimal*) adalah skema pengodean di mana **setiap digit desimal tunggal (0 s.d. 9)** direpresentasikan secara langsung menggunakan **4-bit biner** pembobotan posisi $8 - 4 - 2 - 1$.

2.1 Tabel Pemetaan BCD 8421 & Keadaan Terlarang (*Invalid Codes*)

Karena 4 bit biner mampu membentuk $2^4 = 16$ kombinasi unik, sementara digit desimal hanya ada 10 angka (0 – 9), maka terdapat **6 kombinasi bit yang tidak valid** (*Invalid/Illegal BCD Codes*).

Desimal	BCD ($b_3b_2b_1b_0$)	Desimal	BCD ($b_3b_2b_1b_0$)	Kombinasi 4-Bit	Status BCD
0	0000	5	0101	1010 ₂ (10 ₁₀)	ILLEGAL
1	0001	6	0110	1011 ₂ (11 ₁₀)	ILLEGAL
2	0010	7	0111	1100 ₂ (12 ₁₀)	ILLEGAL
3	0011	8	1000	1101 ₂ (13 ₁₀)	ILLEGAL
4	0100	9	1001	1110 ₂ (14 ₁₀), 1111 ₂ (15 ₁₀)	ILLEGAL

Contoh Konversi: Desimal → BCD 8421

Ubahlah bilangan desimal (853)₁₀ ke dalam kode BCD 8421 dan bandingkan dengan biner murni:

- Digit Ratusan (8₁₀) → **1000_{BCD}**
- Digit Puluhan (5₁₀) → **0101_{BCD}**
- Digit Satuan (3₁₀) → **0011_{BCD}**

Hasil BCD: (853)₁₀ = **(1000 0101 0011)_{BCD}** (membutuhkan 12 bit).
Sebagai perbandingan, dalam biner murni (853)₁₀ = **(1101010101)₂** (hanya butuh 10 bit).

3. Aritmatika Penjumlahan BCD & Algoritma Koreksi +6

Ketika dua bilangan BCD dijumlahkan per digit 4-bit menggunakan sirkuit *Binary Full Adder*, dapat timbul hasil yang melampaui batas representasi desimal valid ($> 9_{10}$) atau menghasilkan *carry-out*. Karena selisih kapasitas heksadesimal (16) dan desimal (10) adalah 6, maka sirkuit **wajib menambahkan nilai $+6_{10}$ (0110_2)**.

ATURAN EMAS KOREKSI PENJUMLAHAN BCD

Koreksi penjumlahan BCD sebesar $+6_{10}$ (0110_2)** wajib ditambahkan pada hasil penjumlahan 4-bit jika dan hanya jika:

1. Hasil penjumlahan 4-bit menghasilkan nilai $> 9_{10}$ (pola 1010_2 s.d. 1111_2), **ATAU**
2. Penjumlahan 4-bit menghasilkan *Carry-Out* ($K = 1$) ke digit berikutnya.

3.1 Persamaan Logika Detektor Koreksi (C_{korek})

Misalkan hasil keluaran penjumlah biner 4-bit adalah Z_3, Z_2, Z_1, Z_0 dengan carry-out K . Sinyal pemicu koreksi C_{korek} dinyatakan dengan persamaan Aljabar Boolean:

$$C_{korek} = K + Z_3Z_2 + Z_3Z_1 \quad (1)$$

Penjelasan: - $Z_3Z_2 = 1$ aktif untuk angka 12_{10} (1100_2), 13_{10} (1101_2), 14_{10} (1110_2), 15_{10} (1111_2).
- $Z_3Z_1 = 1$ aktif untuk angka 10_{10} (1010_2), 11_{10} (1011_2). - $K = 1$ aktif jika penjumlahan menghasilkan carry biner > 15 .

Contoh Soal Penjumlahan BCD: $(48)_{10} + (37)_{10} = (85)_{10}$

Langkah 1: Penjumlahan Digit Satuan **Langkah 2: Penjumlahan Digit Puluhan**
($8 + 7$) an ($4 + 3 + \text{Carry}$)

$$\begin{array}{rcl} 1000 & (8_{BCD}) & \\ +0111 & (7_{BCD}) & \\ \hline 1111 & (= 15_{10} > 9 \rightarrow \text{Illegal}) & \\ +\mathbf{0110} & (\text{Koreksi } +6) & \\ \hline \mathbf{1} \quad \mathbf{0101} & (\text{Carry } K = 1, \text{ Hasil} = \mathbf{5_{BCD}}) & \end{array}$$

$$\begin{array}{rcl} 0100 & (4_{BCD}) & \\ +0011 & (3_{BCD}) & \\ \hline 0111 & (= 7_{10}) & \\ +0001 & (\text{Carry dari Satuan}) & \\ \hline \mathbf{1000} & (= \mathbf{8_{BCD}} \leq 9 \rightarrow \mathbf{VALID!}) & \end{array}$$

Hasil Akhir: Penggabungan digit puluhan dan satuan menghasilkan $(\mathbf{1000 \ 0101})_{BCD} = \mathbf{85_{10}}$ ✓

3.2 Pengurangan BCD (Metode Komplemen 9)

Sirkuit aritmatika ALU umumnya dirancang berbasis *adder*. Untuk melakukan pengurangan ($A - B$) dalam BCD, digunakan sistem **Komplemen 9 (*9's Complement*)**. Komplemen 9 dari digit D adalah $(9 - D)$.

Prosedur Pengurangan BCD $A - B$

1. Ubah pengurang (B) menjadi Komplemen 9-nya.
2. Jumlahkan $A + (\text{Komplemen 9 dari } B)$ dengan aturan BCD biasa (koreksi +6 jika perlu).
3. Jika ada *End-Around Carry* (Carry-out ujung), tambahkan kembali nilai 1 ke LSB hasil akhir.

Contoh: Hitung $(8)_{10} - (3)_{10} = (5)_{10}$ **dalam BCD**

- Komplemen 9 dari 3 adalah 6 (0110_{BCD}).
- Jumlahkan $8 + 6 \implies 1000 + 0110 = \mathbf{1110_2}$ (14_{10} , Illegal $\implies$ Butuh Koreksi $+6$).
- $1110 + 0110 = \mathbf{1\ 0100_2}$. Ada Carry 1 di ujung (*End-Around*).
- Tambahkan Carry tersebut ke hasil: $0100 + 0001 = \mathbf{0101} \implies (5_{BCD})$. ✓

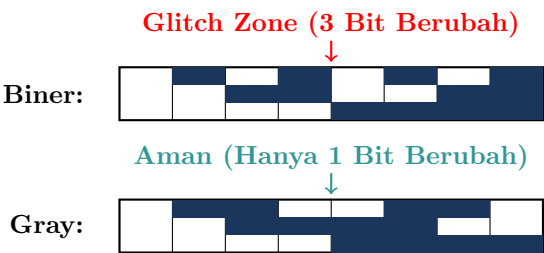
4. Kode Gray (*Reflected Binary Code*)

Kode Gray adalah kode tak berbobot (*unweighted*) yang memiliki karakteristik unik: ****antara dua nilai kuantitas berturutan hanya ada tepat 1 bit yang mengalami perubahan keadaan ($0 \leftrightarrow 1$)**** (*Unit Distance Property*).

Desimal	Biner ($B_3B_2B_1B_0$)	Kode Gray ($G_3G_2G_1G_0$)	Jumlah Bit Berubah (Gray vs Biner)
0	0000	0000	Mulai (0)
1	0001	0001	1 bit berubah
2	0010	0011	1 bit berubah
			(Biner 2 bit berubah)
3	0011	0010	1 bit berubah
4	0100	0110	1 bit berubah
			(Biner 3 bit berubah serentak!)
5	0101	0111	1 bit berubah
6	0110	0101	1 bit berubah
7	0111	0100	1 bit berubah

4.1 Keunggulan Fisis pada Sensor Mekanis (*Shaft Encoder*)

Pada transisi antar-state di dunia nyata (seperti putaran disk sensor optik), ketidaksempurnaan mekanis menyebabkan transisi bit biner tidak terjadi serentak sempurna. Jika transisi dari 3 ke 4 dalam biner murni ($011 \rightarrow 100$) mengalami jeda sedikit, pembacaan sementara bisa menghasilkan 000 (nilai 0), yang disebut ***glitch***. Pada Gray Code ($010 \rightarrow 110$), karena hanya 1 bit yang berubah, posisi transisi hanya akan terbaca sebagai nilai sebelum atau sesudahnya secara solid tanpa ***glitch*** loncatan ekstrim.



4.2 Algoritma Konversi Biner ke Gray dan Gray ke Biner

Konversi Biner → Gray

1. Bit MSB Gray sama dengan MSB Biner:

$$G_{n-1} = B_{n-1}$$

2. Bit berikutnya adalah hasil XOR bit biner bersebelahan:

$$G_i = B_{i+1} \oplus B_i \quad (i < n - 1)$$

Konversi Gray → Biner

1. Bit MSB Biner sama dengan MSB Gray:

$$B_{n-1} = G_{n-1}$$

2. Bit berikutnya adalah hasil XOR bit biner sebelumnya dengan bit Gray saat ini:

$$B_i = B_{i+1} \oplus G_i \quad (i < n - 1)$$

5. Kode Alfanumerik ASCII & Deteksi Error Bit Paritas

Standar **ASCII** (*American Standard Code for Information Interchange*) menggunakan format 7-bit (128 karakter unik). Untuk mendeteksi kesalahan transmisi data akibat gangguan derau elektromagnetik (*noise*), ditambahkan 1 buah bit redundansi bernama **Bit Paritas** (*Parity Bit*).

Tipe Paritas	Definisi Nilai Paritas (P)	Formulasi Logika	Kondisi Dinyatakan Error
Even Parity (Genap)	Total bit '1' (termasuk P) harus berjumlah GENAP .	$P = b_6 \oplus b_5 \oplus \dots \oplus b_0$	Total bit 1 di penerima ganjil
Odd Parity (Ganjil)	Total bit '1' (termasuk P) harus berjumlah GANJIL .	$P = \overline{b_6 \oplus b_5 \oplus \dots \oplus b_0}$	Total bit 1 di penerima genap

Contoh Pembangkitan Even Parity untuk Kata "DATA"

- Karakter 'D' (1000100_2): Ada 2 bit 1 (sudah genap) $\Rightarrow P = 0 \Rightarrow$ Byte: $01000100_2 = 44_{16}$
- Karakter 'A' (1000001_2): Ada 2 bit 1 (sudah genap) $\Rightarrow P = 0 \Rightarrow$ Byte: $01000001_2 = 41_{16}$
- Karakter 'T' (1010100_2): Ada 3 bit 1 (ganjil) $\Rightarrow P = 1 \Rightarrow$ Byte: $11010100_2 = D4_{16}$

Keterbatasan Bit Paritas Sederhana & Solusi Industri Modern

Sistem paritas sederhana (1 bit per byte) memiliki kelemahan kritis: **hanya mampu mendeteksi jumlah error ganjil (1, 3, 5 bit)**. Jika terjadi 2 bit yang membalik (*flip*) secara bersamaan akibat gangguan derau yang kuat, **parity bit** tidak akan berubah, sehingga kesalahan lolos dari deteksi.

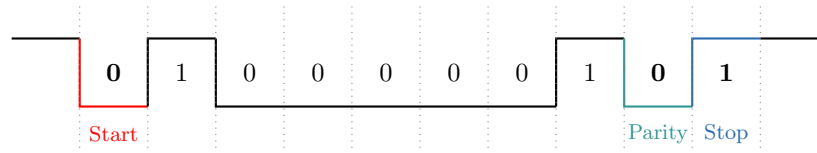
Oleh karena itu, industri modern menggunakan metode yang jauh lebih andal:

- Hamming Code:** Digunakan pada memori server ECC (*Error Correcting Code*) RAM, mampu mengoreksi 1 bit error dan mendeteksi 2 bit error.
- CRC (*Cyclic Redundancy Check*):** Menggunakan polinomial pembagian untuk mendeteksi error pada jaringan Ethernet, WiFi, dan media penyimpanan (Hard disk / SSD).

5.1 Aplikasi Nyata: Transmisi Serial Asinkron (UART / RS-232)

Dalam komunikasi mikrokontroler (misalnya mengirim data dari sensor ke PC), karakter ASCII beserta bit paritasnya ditransmisikan secara serial lewat satu kabel, bit-demi-bit. Data dibungkus dalam sebuah **Serial Frame** yang berisi **Start Bit**, **Data Bits** (Lsb dikirim duluan), **Parity Bit**, dan **Stop Bit**.

Contoh Diagram Pewaktuan (*Timing Diagram*) Transmisi Karakter ‘A’ (Even Parity)



6. Latihan Soal Komprehensif (Bloom C1 – C6)

1. [C1] Sebutkan 6 kombinasi bit terlarang pada sistem pengodean BCD 8421!
2. [C3] Konversikan biner $(10110110)_2$ ke dalam Kode Gray, dan konversikan Kode Gray $(1101101)_{Gray}$ ke biner!
3. [C4] Hitung penjumlahan BCD $(69)_{10} + (58)_{10}$ lengkap dengan langkah koreksi +6 pada masing-masing kelompok digit!
4. [C6] Rancang skema diagram logika sirkuit detektor paritas genap 8-bit menggunakan jaringan gerbang XOR di Logisim Evolution!