

The left side of the slide features three distinct abstract geometric patterns. The top pattern is a square divided into four quadrants, each containing a different circular or semi-circular design in teal and white. The middle pattern is a large semi-circle divided into two halves, each with a different circular design. The bottom pattern is a square divided into four quadrants, each containing a different geometric design, including concentric circles and a triangular grid. The text 'DAY 4' is positioned to the right of the top pattern.

**DAY 4**

---

# AI PROMPTING

---

**BY ARIO TUA PURBA**



# Timeline

- 1 APA ITU PROMPT ENGINEERING.
- 2 TEKNIK-TEKNIK DALAM PROMPTING.
- 3 TEKNIK PROMPTING DI PROGRAMMING.
- 4 AI ETHICS.



# APA ITU PROMPT ENGINEERING

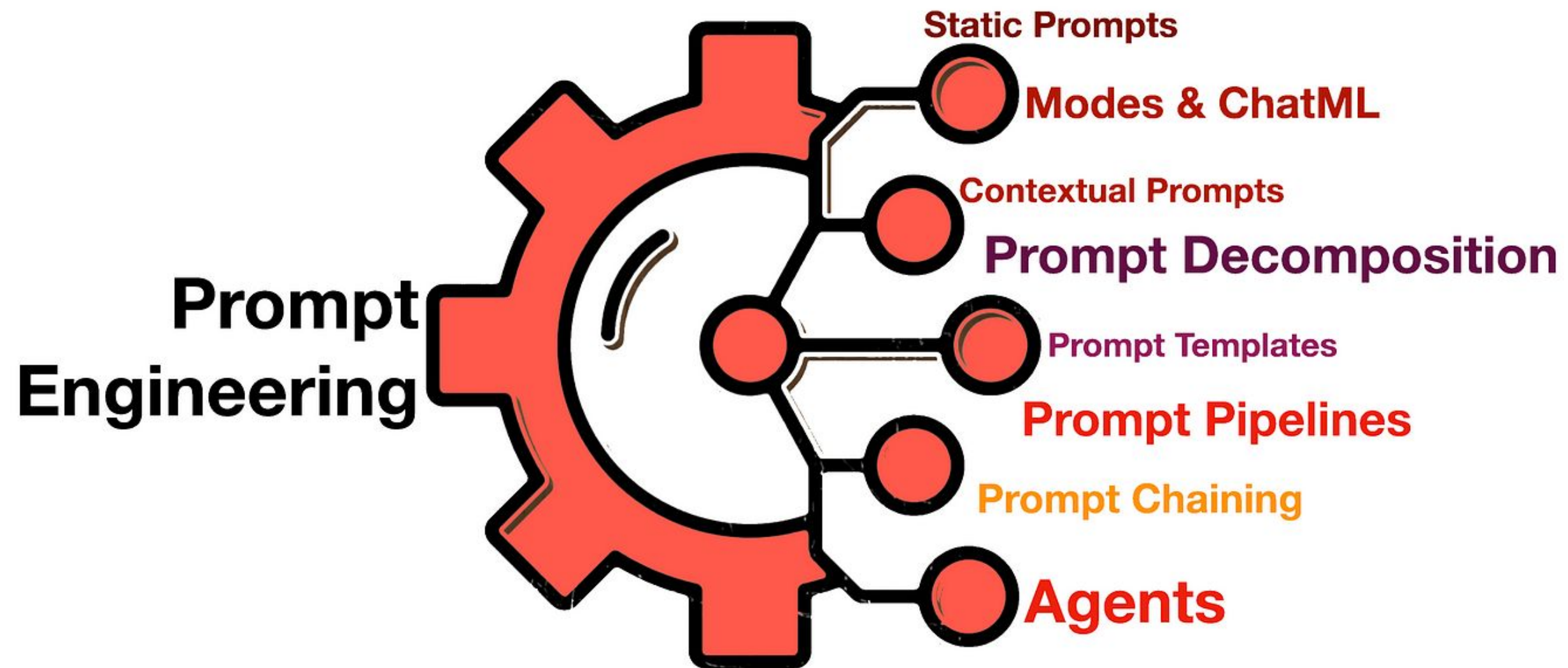


# APA ITU PROMPT ENGINEERING.

Prompt engineering adalah proses membuat/merancang dan mengoptimalkan prompt untuk meningkatkan kinerja model kecerdasan buatan generatif (Generative AI). Proses ini melibatkan pemberian masukan spesifik ke tools seperti ChatGPT, Midjourney, DALL-E atau Gemini, untuk mengarahkan kecerdasan buatan (AI) agar menghasilkan output yang lebih akurat dan relevan secara kontekstual.

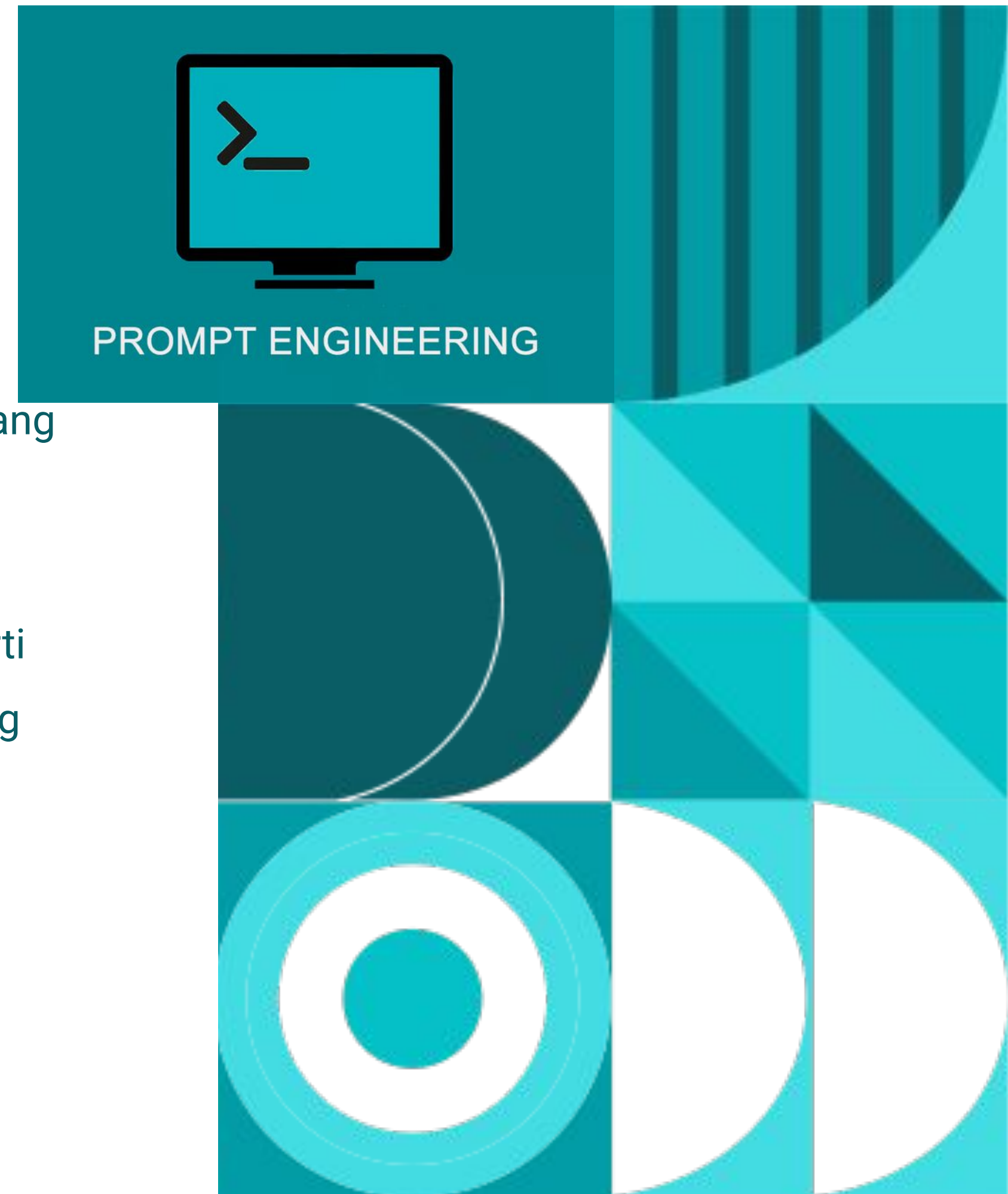






# MENGAPA PROMPT ENGINEERING PENTING?

- Ia menjembatani kesenjangan antara question/prompt yang terlalu umum/samar dengan hasil yang lebih spesifik dan dapat diberikan oleh Generative AI
- Ini dapat membantu mengurangi kesalahan/errors, seperti menghasilkan konten yang tidak relevan atau respon yang tidak benar/salah.
- Ini akan memastikan bahwa AI dapat menangani tugas-tugas seperti creative writing, image generation, or even code development dengan minimal proses yang dibutuhkan.





# Lalu Apa itu Prompt?

Sebuah prompt adalah masukan atau instruksi yang diberikan kepada AI Model untuk membuat suatu respon/keluaran. Prompts bisa simple (pertanyaan sederhana) atau kompleks (instruksi detail dengan konteks, tone, style, dan spesifik format).

Kualitas dari AI's response tergantung bagaimana clear/jelasnya, detail, dan terstruktur kita membuat prompt.

# CONTOH PENERAPAN DARI PROMPT ENGINEERING

**Pembuatan  
Teks**

**Pembuatan  
Gambar**

**Pembuatan  
Code**





# TEKNIK-TEKNIK DALAM PROMPTING



# Prompt Engineering Techniques



Zero-shot  
prompting



Few-shot prompting  
or in-context  
learning



Chain-of-thought  
prompting



Tree-of-thought  
prompting



Iterative  
refinement



Feedback  
loops



Prompt  
chaining



Role-playing



Maieutic  
prompting



Complexity-based  
prompting

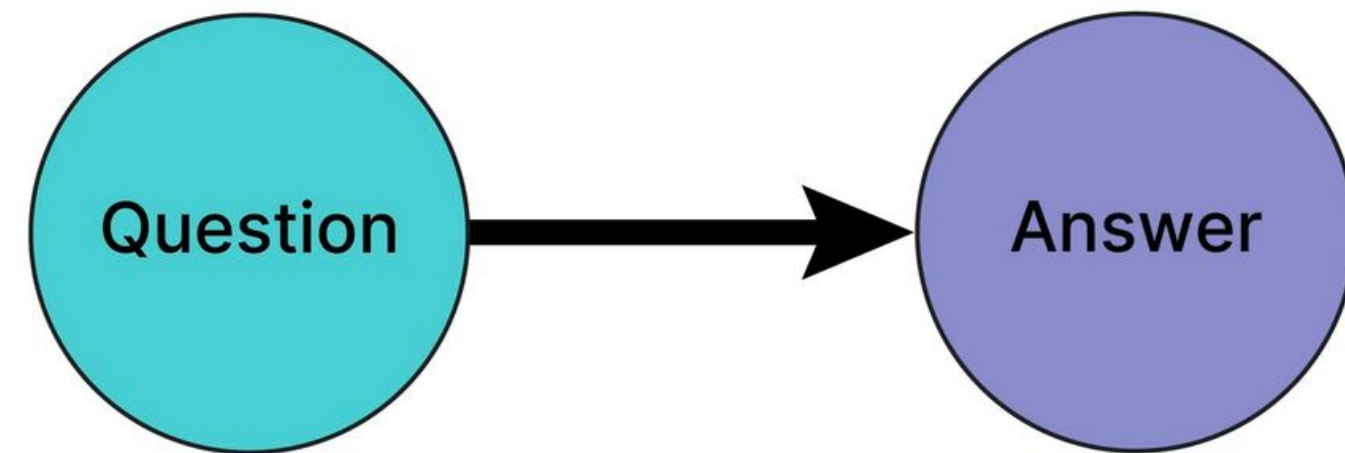
# ZERO SHOT PROMPTING

Memberikan instruksi  
langsung tanpa contoh  
sebelumnya.

Contoh:

"Jelaskan apa itu fotosintesis dalam 2 kalimat."

Kegunaan: Cepat dan sederhana, tetapi mungkin kurang akurat untuk tugas kompleks.



**Zero-shot**



# FEW-SHOT PROMPTING

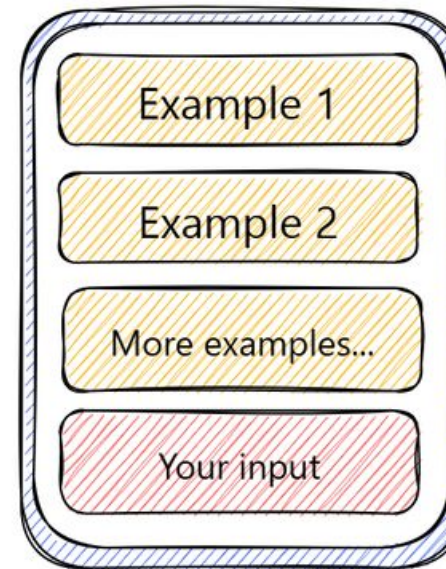
Memberikan beberapa  
contoh (demonstrations)  
sebelum instruksi utama.

Contoh:

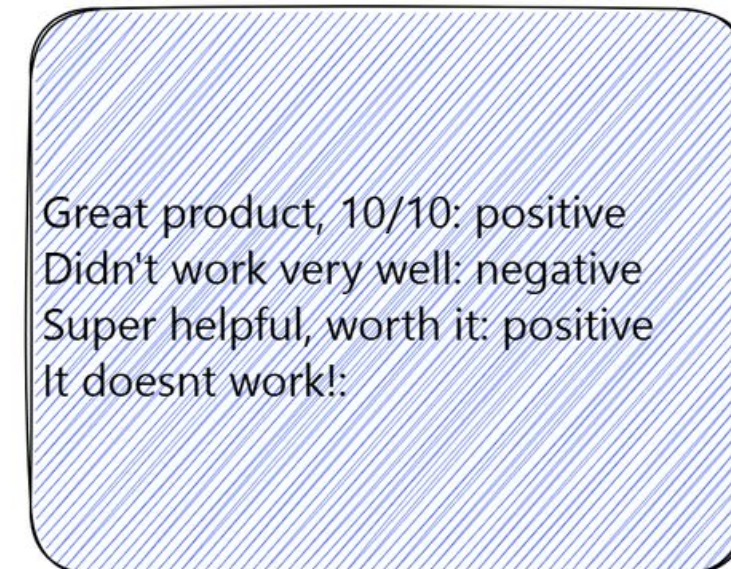
"Contoh sinonim dari 'bahagia': senang, gembira. Sekarang berikan sinonim untuk 'sedih'."

Kegunaan: Membantu model memahami pola yang diinginkan.

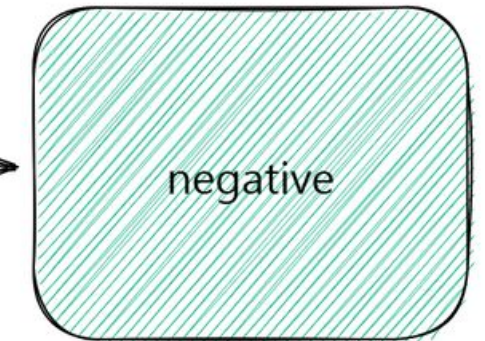
A Few Shot Prompt



Example



Model Output





# Zero-Shot vs. Few-Shot: Key Comparisons

Comparison Factor	Zero-Shot Prompting	Few-Shot Prompting
Complexity of Task	Best for simpler, general tasks.	Essential for complex or specialized tasks.
Accuracy Requirements	Prioritizes speed over precision. Suitable for cases where responses can be verified by users.	Prioritizes precision and relevance. Ideal for applications requiring high accuracy.
Resource Availability	Minimal resources required, making it cost-effective and efficient for low-compute environments.	Higher resource consumption due to processing examples, suited for applications with available resources.
Deployment Needs	Ideal for production-ready applications that require quick response times and can handle large request volumes.	Best for specialized deployments where accuracy outweighs speed, such as personalized or experimental applications.

# Chain-of-Thought (CoT) Prompting

**Meminta model untuk menjelaskan langkah-langkah logisnya.**

Contoh:

"Jika harga 3 apel adalah Rp15.000, berapa harga 5 apel? Jelaskan langkah perhitunganmu."

Kegunaan: Meningkatkan akurasi untuk soal matematika atau penalaran kompleks.



## Standard Prompting

### Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

### Model Output

A: The answer is 27. ❌

## Chain-of-Thought Prompting

### Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls.  $5 + 6 = 11$ . The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

### Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had  $23 - 20 = 3$ . They bought 6 more apples, so they have  $3 + 6 = 9$ . The answer is 9. ✅



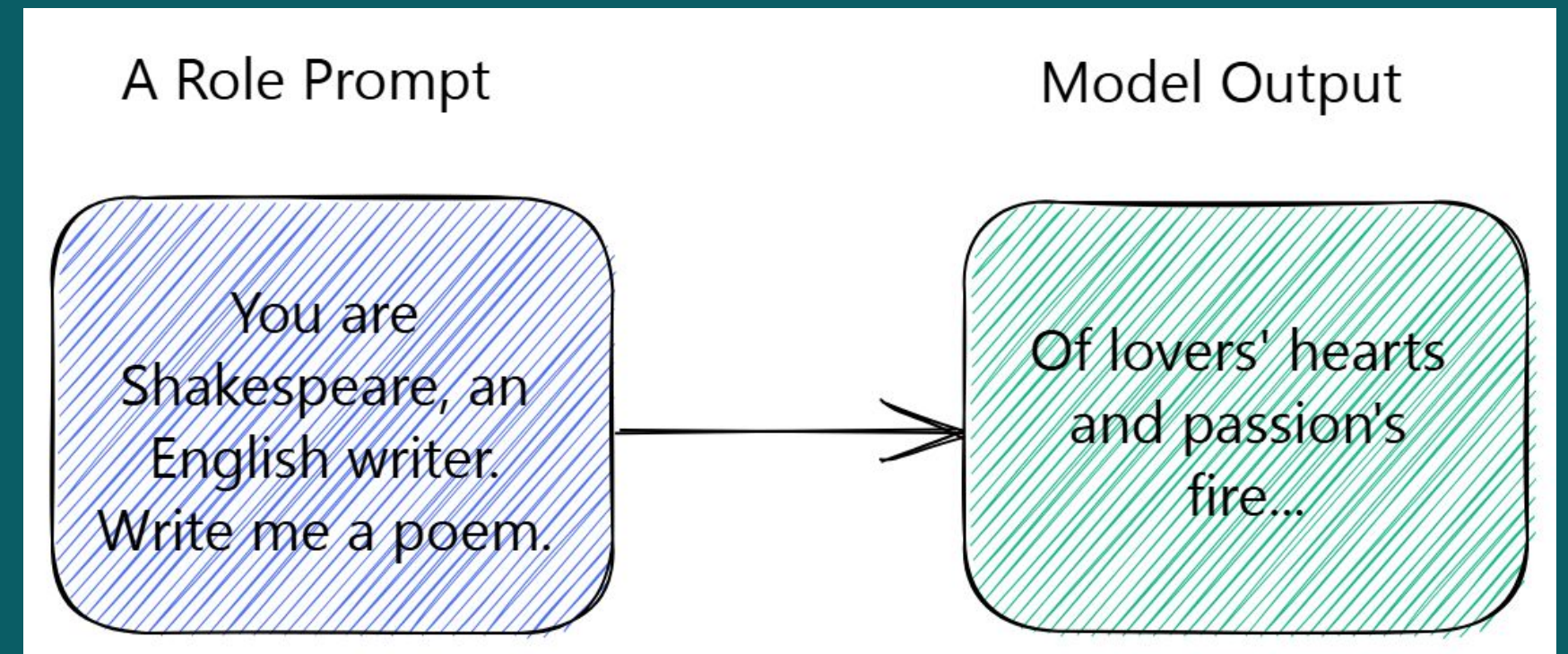
# ROLE PROMPTING

**Menentukan peran (role) yang harus diambil oleh AI.**

Contoh:

"Kamu adalah seorang dokter. Jelaskan cara mencegah flu dengan bahasa yang mudah dipahami."

Kegunaan: Membuat output lebih relevan dan sesuai perspektif tertentu





# TEMPLATE PROMPTING

Menggunakan struktur tetap untuk konsistensi.

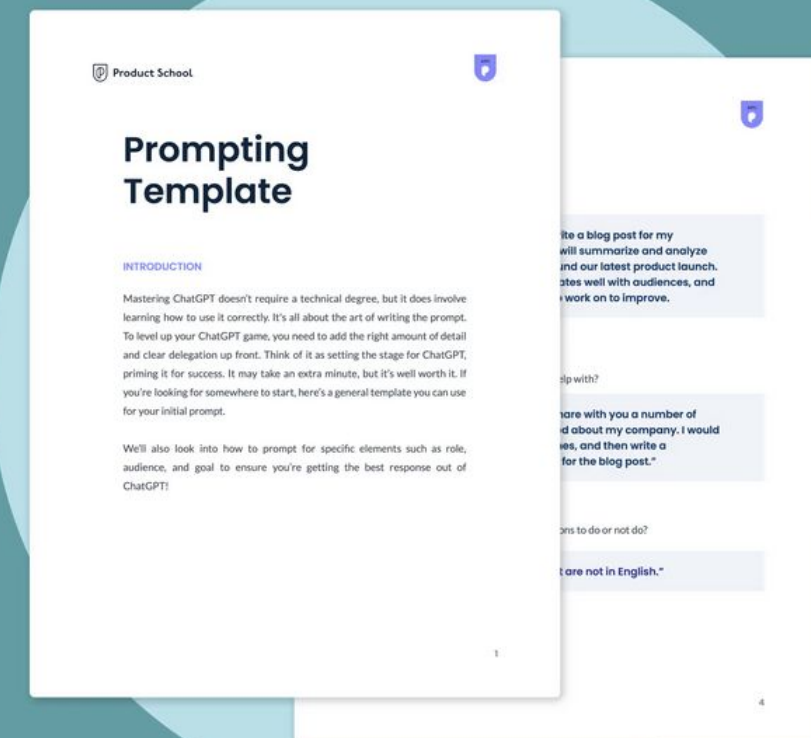
Contoh:

\*"Buat [jenis konten] tentang [topik] dengan format:

1. Pendahuluan
2. 3 Poin Utama
3. Kesimpulan

"\*

Kegunaan: Berguna untuk generating konten terstruktur (artikel, laporan, dll.).



# Iterative Refinement

**Memperbaiki prompt secara bertahap berdasarkan output sebelumnya.**

Contoh:

Prompt Awal: "Tulis puisi tentang hujan."

Revisi: "Tulis puisi pendek tentang hujan dengan suasana sedih dan menggunakan metafora alam."

Kegunaan: Mendapatkan hasil yang lebih spesifik.

# NEGATIVE PROMPTING

**Menyebutkan apa yang tidak diinginkan dalam output.**

Contoh:

"Jelaskan blockchain tanpa menggunakan istilah teknis seperti 'decentralized' atau 'cryptography'."

Kegunaan: Membatasi output yang tidak relevan.





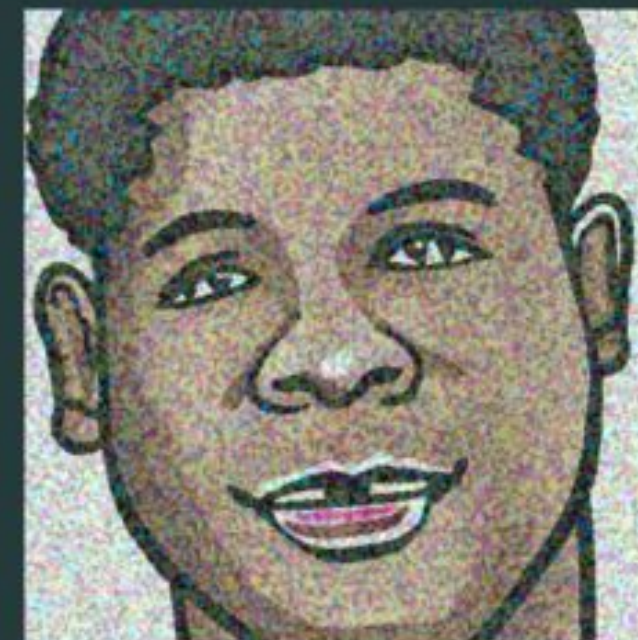
Conditional  
sampling



“Unconditional”  
sampling with  
negative prompt



Denoising



-

=

Repeat





# MULTI-TURN PROMPTING

**Memecah pertanyaan kompleks menjadi beberapa langkah interaktif.**

Contoh:

Pengguna: "Apa ibu kota Indonesia?"

AI: "Jakarta."

Pengguna: "Sebutkan 3 tempat wisata terkenal di sana."

Kegunaan: Berguna untuk percakapan dinamis atau eksplorasi topik.



How many people is the booking for?

Oh are you showing the football that day?

Yes, we have TVs throughout the restaurant where we show live sports.  
How many people will be coming?

# Keyword Highlighting

Menekankan kata kunci penting dalam prompt.

Contoh:

\*"Fokus pada dampak ekonomi dari pandemi COVID-19, bukan aspek kesehatan."\*

Kegunaan: Memandu AI ke fokus tertentu.



**Keywords**  
Keywords are the  
Particular words or  
context of search  
describes the co  
describes or de

# Analogi & Perbandingan

**Meminta model membuat analogi untuk penjelasan yang lebih mudah.**

Contoh:

"Jelaskan cara kerja VPN seperti analogi kehidupan sehari-hari."

Kegunaan: Membuat konsep abstrak lebih mudah dipahami.

# TEKNIK PROMPTING DI PROGRAMMING





# SPESIFIK BAHASA & FRAMEWORK

**Sebutkan bahasa pemrograman, library, atau framework yang digunakan.**

**Manfaat: Menghindari kode generik yang tidak sesuai dengan stack teknologi Anda.**

Contoh:

"Buat fungsi Python menggunakan Pandas untuk membersihkan data CSV dari nilai NULL."



# CONTOH INPUT-OUTPUT

**Berikan contoh input dan output yang diharapkan.**

**Manfaat: Memandu AI memahami kebutuhan konkret.**

Contoh:

`*"Buat fungsi JavaScript yang mengubah string '2023-12-31' menjadi objek Date.`

`Contoh Input: '2023-12-31' → Output: Date object untuk 31 Desember 2023.*"`



# DEBUGGING DENGAN ERROR MESSAGE

**Sertakan pesan error atau perilaku tidak diharapkan.**

**Manfaat: Mempercepat proses debugging.**

Contoh:

\*"Kode Python ini menghasilkan error index out of range'. Perbaiki:

```
'IndexError: list iarr = [1, 2]  
print(arr[3])
```





# CONSTRAINT & OPTIMIZATION

**Tambahkan batasan (kinerja, kompleksitas waktu, dll.).**

**Manfaat: Mendapatkan solusi yang lebih efisien.**

Contoh:

Tulis algoritma sorting di C++ dengan kompleksitas  $O(n \log n)$ , tanpa menggunakan fungsi bawaan.



# STEP-BY-STEP EXPLANATION

**Minta AI menjelaskan kode baris per baris.**

**Manfaat: Membantu pemula memahami logika kode.**

Contoh:

Jelaskan cara kerja kode berikut:

```
def factorial(n):  
    return 1 if n == 0 else n * factorial(n-1)
```



# REFACTORING REQUEST

# Minta AI untuk memperbaiki atau modernisasi kode.

**Manfaat: Meningkatkan readability dan best practices.**

## Contoh:

Refactor kode Javascript ini agar menggunakan arrow function dan destructuring:

```
function getUser(data) {  
    return { name: data.name, age: data.age };  
}
```



# GENERATE TEST CASES

**Minta AI membuat unit test  
untuk kode anda.**

**manfaat : memverifikasi  
keandalan kode.**

Contoh:

Buat 3 Case untuk fungsi berikut menggunakan pytest:

```
def add(a, b):  
    return a + b
```



# CODE TRANSLATION

**Konversi kode antar bahasa pemrograman.**

**Manfaat: Migrasi atau belajar bahasa baru.**

Contoh:

Ubah kode python in ke Go:

```
def greet(name):  
    print(f"Hello, {name}!")
```



# DOKUMENTASI OTOMATIS

**Minta AI membuat Dokumentasi atau komentar.**

**Manfaat: Membutuhkan kolaborasi dan maintenance.**

Contoh:

Tambahkan docstring untuk fungsi python ini:

```
def calculate_area(radius):  
    return 3.14 * radius ** 2
```





# COMPLETE THE CODE

**Berikan konteks partial kode dan minta AI melanjutkan.**

**Manfaat: Mengisi gap saat stuck.**

Contoh:

Lengkapi kode berikut untuk menghitung rata-rata nilai array:

```
function calculateAverage(arr) {  
    // ...  
}
```



# ASK FOR ALTERNATIVES

**Minta solusi alternatif untuk suatu masalah.**

**Manfaat: Eksplorasi berbagai solusi/pendekatan.**

Contoh:

Berikan 3 cara berbeda untuk mengimplementasikan Fibonacci sequence di Java.



# AI ETHICS





# MENGHORMATI PRIVASI

**Meminta AI menghasilkan  
atau mengolah data pribadi  
tanpa izin.**

Contoh Tidak Etis:

“Buat profil psikologis berdasarkan riwayat chat WhatsApp ini: [paste data pribadi].”

Solusi:

- Anonimisasi data sebelum digunakan.
- Hindari meminta informasi sensitif (nomor KTP, rekam medis, dll.)



# KEAMANAN (SECURITY)

**Memanfaatkan AI untuk aktivitas berbahaya (Peretasan, eksploitasi).**

Contoh:

“Ajarkan cara membuat virus komputer”

Solusi:

- Patuhi kebijakan penggunaan AI (Misalnya ChatGPT melarang prompt berbahaya)



# Hak Kekayaan Intelektual

Menghasilkan kode/desain/konten yang melanggar hak cipta.

Contoh:

“Buat kode yang meniru fitur premium Photoshop”

Solusi:

- Gunakan AI untuk belajar atau inspirasi, bukan duplikasi ilegal.



# AKUNTABILITAS

**Mengandalkan AI  
sepenuhnya tanpa  
verifikasi.**

**Solusi:  
Selalu cross-check output AI,  
terutama untuk keputusan kritis.**

Contoh:

“Buat diagnosis medis berdasarkan gejala saya.”



# KESEJAHTERAAN SOSIAL

**Konten berbahaya  
(misinformation, ujaran  
kebencian)**

**Solusi:**

- **Promosikan konten yang memajukan masyarakat.**

Contoh:

“Tulis pesan propaganda politik yang memanipulasi emosi”



---

# THANK YOU

---