

```
import React, { useState, useEffect, useCallback } from 'react';
import { initializeApp } from 'firebase/app';
import { getAuth, signInAnonymously, signInWithCustomToken, onAuthStateChanged } from
'firebase/auth';
import { getFirestore, collection } from 'firebase/firestore';

// --- KONFIGURASI DAN UTILITAS FIREBASE/API ---

// Variabel global dari lingkungan Canvas (WAJIB DIGUNAKAN)
const appId = typeof __app_id !== 'undefined' ? __app_id : 'default-app-id';
const firebaseConfig = typeof __firebase_config !== 'undefined' ?
JSON.parse(__firebase_config) : null;
const initialAuthToken = typeof __initial_auth_token !== 'undefined' ? __initial_auth_token :
null;
const AI_MODEL_NAME = "gemini-2.5-flash-preview-09-2025";
const API_KEY = ""; // Kunci API akan disediakan oleh runtime Canvas

// Utilitas untuk animasi mengetik
const typeResponse = (text, setter) => {
    let i = 0;
    setter("");
    const typing = setInterval(() => {
        if (i < text.length) {
            // Memastikan karakter yang tidak diinginkan dibersihkan sebelum ditampilkan
            const char = text.charAt(i);
            const cleanedChar = char.replace(/[^#\\[]>/g, " ");
            if (cleanedChar.length > 0) {
                setter(prev => prev + cleanedChar);
            } else if (char.trim() === "") {
                setter(prev => prev + char);
            }
            i++;
        } else {
            clearInterval(typing);
        }
    }, 10); // Kecepatan mengetik
};

// Fungsi Keamanan (Update 2)
const isDangerous = (query) => {
    const dangerousKeywords = [
        'membuat bom', 'merakit senjata', 'narkotika', 'peretasan', 'cyber attack',
        'bunuh diri', 'self-harm', 'kekerasan seksual', 'bahan peledak', 'terror'
    ];
    // Normalisasi query untuk deteksi yang lebih baik
    const normalizedQuery = query.toLowerCase().replace(/\w\s/g, " ");
    return dangerousKeywords.some(keyword => normalizedQuery.includes(keyword));
};
```

```
// --- KOMPONEN DROPDOWN KUSTOM ---  
  
const Dropdown = ({ label, options, value, onChange }) => {  
  const [isOpen, setIsOpen] = useState(false);  
  
  return (  
    <div className="relative w-full mb-4">  
      <label className="block text-sm font-medium text-gray-700 mb-1">{label}</label>  
      <button  
        type="button"  
        className="w-full bg-white border border-gray-300 rounded-lg shadow-sm px-4  
        py-2 text-left text-gray-700 hover:bg-gray-50 focus:outline-none focus:ring-2  
        focus:ring-green-500 transition duration-150 ease-in-out flex justify-between items-center"  
        onClick={() => setIsOpen(!isOpen)}  
      >  
        {value || `Pilih ${label.toLowerCase()}`}  
        <span className={`transition-transform duration-300 ${isOpen ? 'rotate-180' :  
          'rotate-0'}`}>#9660;</span>  
      </button>  
      {isOpen && (  
        <div className="absolute z-10 w-full mt-1 bg-white rounded-lg shadow-xl  
        max-h-60 overflow-y-auto border border-green-500">  
          {options.map((option) => (  
            <div  
              key={option}  
              className="px-4 py-2 text-sm text-gray-700 hover:bg-green-100  
              cursor-pointer transition duration-150"  
              onClick={() => {  
                onChange(option);  
                setIsOpen(false);  
              }}  
            >  
              {option}  
            </div>  
          ))}  
        </div>  
      )}  
    </div>  
  );  
};
```

// --- KOMPONEN MODAL (Update 3) ---

```
const UpcomingFeatureModal = ({ isVisible, onClose }) => {  
  if (!isVisible) return null;  
  
  return (
```

```

<div className="fixed inset-0 bg-black bg-opacity-50 z-40 flex items-center justify-center p-4">
  <div className="bg-white rounded-xl shadow-2xl p-6 max-w-sm w-full border-t-4 border-green-500">
    <h3 className="text-xl font-bold text-gray-800 mb-4">Fitur Segera Hadir</h3>
    <p className="text-gray-700 mb-6">
      Fitur ini masih belum siap untuk dipakai mohon tunggu sampai update yang akan datang.
    </p>
    <button
      onClick={onClose}
      className="w-full bg-green-600 hover:bg-green-700 text-white font-bold py-2 rounded-lg transition duration-300"
    >
      TUTUP
    </button>
  </div>
</div>
);
};

// --- KOMPONEN FORM UTAMA (NASKAH) ---

```

```

const ScriptForm = ({ generateContent }) => {
  const [platform, setPlatform] = useState("");
  const [age, setAge] = useState("");
  const [theme, setTheme] = useState("");
  const [description, setDescription] = useState("");

  const platforms = ['YouTube', 'TikTok', 'Facebook'];
  const ages = ['Anak-anak', 'Remaja', 'Dewasa', 'Orang Tua'];
  const themes = [
    'Naskah Animasi', 'Naskah Film Pendek', 'Naskah Dokumenter',
    'Naskah Tutorial / Edukasi', 'Naskah Drama / Komedi', 'Naskah Presentasi / Company Profile'
  ];
}

const handleSubmit = (e) => {
  e.preventDefault();
  if (!description || !platform || !age || !theme) {
    alert("Harap lengkapi semua kolom untuk membuat naskah.");
    return;
  }

  const systemPrompt = `Anda adalah penulis naskah video profesional. Buatkan naskah video (termasuk intro, isi, dan outro) yang menarik dan disesuaikan untuk platform ${platform}. Gaya naskah harus ${theme} dan ditujukan untuk audiens ${age}.`;

```

```

const userQuery = `Ide dan deskripsi produk/video singkatnya adalah: ${description}.`;

generateContent(userQuery, systemPrompt);
};

return (
  <form onSubmit={handleSubmit} className="p-4 space-y-4">
    <label className="block text-sm font-medium text-gray-700">Ide & Deskripsi
    Singkat</label>
    <textarea
      className="w-full p-3 border border-gray-300 rounded-lg focus:ring-green-500
      focus:border-green-500 transition duration-150"
      rows="3"
      placeholder="Contoh: Aplikasi belajar bahasa Inggris interaktif untuk anak SD."
      value={description}
      onChange={(e) => setDescription(e.target.value)}
      required
    ></textarea>

    <Dropdown label="Pilih Platform" options={platforms} value={platform}
    onChange={setPlatform} />
    <Dropdown label="Pilih Usia Audiens" options={ages} value={age}
    onChange={setAge} />
    <Dropdown label="Pilih Tema Naskah" options={themes} value={theme}
    onChange={setTheme} />

    <button
      type="submit"
      className="w-full bg-green-600 hover:bg-green-700 text-white font-bold py-3
      rounded-lg shadow-md transition duration-300 ease-in-out transform hover:scale-[1.01]"
    >
      BUAT NASKAH
    </button>
  </form>
);
};

// --- KOMPONEN FORM UTAMA (DESKRIPSI PRODUK - Update 1) ---

```

```

const ProductDescForm = ({ generateContent }) => {
  const [page, setPage] = useState(1);
  const [productName, setProductName] = useState("");
  const [productTheme, setProductTheme] = useState("");
  const [productGoal, setProductGoal] = useState("");
  const [fullDescription, setFullDescription] = useState("");
  const [nickname, setNickname] = useState("");
  const [age, setAge] = useState("");

```

```

// Tema produk yang diperbarui (Update 1)
const themes = [
  'Makanan & Minuman', 'Fashion & Aksesoris', 'Elektronik & Gadget',
  'Perawatan Tubuh & Kecantikan', 'Rumah Tangga & Dapur',
  'Mainan, Hobi & Koleksi', 'Olahraga & Outdoor', 'Kesehatan & Medis Ringan',
  'Perlengkapan Bayi & Anak', 'Otomotif & Aksesoris Kendaraan'
];
const goals = ['Untuk Deskripsi Produk (Website/Toko)', 'Untuk Iklan (Iklan Digital/Media Sosial)'];
const ages = ['Anak-anak', 'Remaja', 'Dewasa', 'Orang Tua'];

const handleSubmit = (e) => {
  e.preventDefault();
  if (!productName || !productTheme || !productGoal || !fullDescription || !age) {
    alert("Harap lengkapi semua kolom untuk membuat deskripsi produk.");
    return;
  }

  const systemPrompt = `Anda adalah spesialis pemasaran dan copywriter ulung.
Buatkan deskripsi produk/iklan yang sangat persuasif dan menarik untuk produk bernama
"${productName}", yang bertema ${productTheme} dan ditujukan untuk audiens ${age}.`;

  const userQuery = `
    Nama Produk: ${productName} (${nickname ? `Julukan: ${nickname}` : 'Tanpa
Julukan'})
    Tujuan: ${productGoal}
    Deskripsi lengkap produk (WAJIB LENGKAP): ${fullDescription}
  `;

  generateContent(userQuery, systemPrompt);
};

const isPage1Valid = productName && productTheme && productGoal;

if (page === 1) {
  return (
    <div className="p-4 space-y-4">
      <input
        type="text"
        className="w-full p-3 border border-gray-300 rounded-lg focus:ring-green-500
focus:border-green-500 transition duration-150"
        placeholder="Masukkan Nama Produk Anda"
        value={productName}
        onChange={(e) => setProductName(e.target.value)}
        required
      />
      <Dropdown label="Pilih Tema Produk" options={themes} value={productTheme}
        onChange={setProductTheme} />
    </div>
  );
}

const isPage2Valid = age && fullDescription;

```

```
<Dropdown label="Pilih Tujuan Produk" options={goals} value={productGoal} onChange={setProductGoal} />

<div className="flex justify-end pt-4">
  <button
    type="button"
    onClick={() => isPage1Valid ? setPage(2) : alert("Harap lengkapi semua isian di Halaman 1.")}
    className={`${px-6 py-2 text-sm font-medium rounded-lg shadow-md transition duration-300 ${isPage1Valid ? 'bg-green-500 hover:bg-green-600 text-white' : 'bg-gray-300 text-gray-500 cursor-not-allowed'}`}>
    >
      NEXT
    </button>
  </div>
</div>
);

}

return (
  <form onSubmit={handleSubmit} className="p-4 space-y-4">
    <label className="block text-sm font-medium text-gray-700">Deskripsi Produk (Wajib Lengkap)</label>
    <textarea
      className="w-full p-3 border border-gray-300 rounded-lg focus:ring-green-500 focus:border-green-500 transition duration-150"
      rows="4"
      placeholder="Jelaskan fitur, manfaat, dan keunggulan produk Anda secara rinci."
      value={fullDescription}
      onChange={(e) => setFullDescription(e.target.value)}
      required
    ></textarea>

    <label className="block text-sm font-medium text-gray-700">Julukan Produk (Opsional)</label>
    <input
      type="text"
      className="w-full p-3 border border-gray-300 rounded-lg focus:ring-green-500 focus:border-green-500 transition duration-150"
      placeholder="Contoh: Si Pintar Bahasa, Komik Pembawa Tawa"
      value={nickname}
      onChange={(e) => setNickname(e.target.value)}
    />

    <Dropdown label="Pilih Usia Audiens" options={ages} value={age} onChange={setAge} />

    <div className="flex justify-between items-center pt-4">
```

```

        <button
            type="button"
            onClick={() => setPage(1)}
            className="px-4 py-2 text-sm font-medium bg-gray-200 hover:bg-gray-300
rounded-lg transition duration-300"
        >
            &larr; Kembali
        </button>
        <button
            type="submit"
            className="bg-green-600 hover:bg-green-700 text-white font-bold py-3 px-6
rounded-lg shadow-md transition duration-300 ease-in-out transform hover:scale-[1.01]"
        >
            BUAT DESKRIPSI
        </button>
    </div>
</form>
);
};

// --- KOMPONEN FORM UTAMA (PERINTAH AI) ---


```

```

const CommandForm = ({ generateContent }) => {
    const [command, setCommand] = useState("");
    const [age, setAge] = useState("");
    const [selectedAi, setSelectedAi] = useState('Gemini 3');
    const [goal, setGoal] = useState("");
    const [theme, setTheme] = useState("");

    const ages = ['Anak-anak', 'Remaja', 'Dewasa', 'Orang Tua'];
    const ais = [
        'Gemini 3', 'chat cbt 5.2', 'cloud 4.5', 'meta android 2.24.24.78', 'meta iPhone 2.24.24.83'
    ];
    const goals = [
        'Membantu Belajar', 'Meningkatkan Produktivitas', 'Hiburan Interaktif',
        'Kreativitas & Ide', 'Asisten Cerdas'
    ];
    const themes = [
        'Teknologi & Masa Depan', 'Fantasi & Dunia Imajinasi', 'Sci-Fi & AI',
        'Pendidikan Santai', 'Game & Petualangan', 'Misteri & Detektif',
        'Horror Ringan', 'Anak-anak & Edukatif', 'Kehidupan Sehari-hari',
        'Perang & Strategi'
    ];

    const handleSubmit = (e) => {
        e.preventDefault();
        if (!command || !age || !goal || !theme) {
            alert("Harap lengkapi semua kolom untuk mengirim perintah.");
        }
    };
};


```

```

        return;
    }

    const systemPrompt = `Anda adalah AI dengan persona ${goal}. Bertindaklah dengan
gaya dan tema ${theme}. Sesuaikan bahasa dan tingkat kerumitan jawaban Anda untuk
audiens ${age}. Anda akan menggunakan model AI ${selectedAi} untuk menjalankan tugas
ini.`;

    const userQuery = `Lakukan perintah berikut: ${command}`;

    generateContent(userQuery, systemPrompt);
};

return (
    <form onSubmit={handleSubmit} className="p-4 space-y-4">
        <label className="block text-sm font-medium text-gray-700">Isi Perintah Ai</label>
        <textarea
            className="w-full p-3 border border-gray-300 rounded-lg focus:ring-green-500
focus:border-green-500 transition duration-150"
            rows="3"
            placeholder="Contoh: Tulis puisi tentang kota yang hilang di bawah laut."
            value={command}
            onChange={(e) => setCommand(e.target.value)}
            required
        ></textarea>

        <Dropdown label="Pilih Usia Audiens" options={ages} value={age}>
        onChange={setAge} />
        <Dropdown label="Pilih Ai" options={ais} value={selectedAi}>
        onChange={setSelectedAi} />
        <Dropdown label="Pilih Tujuan Ai" options={goals} value={goal}>
        onChange={setGoal} />
        <Dropdown label="Pilih Tema Respon" options={themes} value={theme}>
        onChange={setTheme} />

        <button
            type="submit"
            className="w-full bg-green-600 hover:bg-green-700 text-white font-bold py-3
rounded-lg shadow-md transition duration-300 ease-in-out transform hover:scale-[1.01]"
        >
            BUAT PERINTAH
        </button>
    </form>
);
};

// --- KOMPONEN FORM UTAMA (BUAT BUKU - Update 4) ---

```

```

const BookForm = ({ generateContent, bookContent, setIsBookViewing }) => {
  const [description, setDescription] = useState("");
  const [age, setAge] = useState("");
  const [theme, setTheme] = useState("");

  const ages = ['Anak-anak', 'Remaja', 'Dewasa']; // Tanpa Orang Tua
  const themes = [
    'Buku Fiksi', 'Buku Nonfiksi', 'Buku Pelajaran', 'Novel', 'Komik / Manga',
    'Ensiklopedia', 'Buku Anak', 'Buku Religi', 'Modul / Workbook', 'Majalah / Jurnal'
  ];

  const handleSubmit = (e) => {
    e.preventDefault();
    if (!description || !age || !theme) {
      alert("Harap lengkapi semua kolom untuk membuat buku.");
      return;
    }

    const systemPrompt = `Anda adalah penulis buku profesional. Buatkan naskah buku singkat yang menarik berdasarkan ide berikut. Pastikan output Anda berupa teks naratif panjang dengan jeda baris alami untuk keperluan pemformatan. Tema buku: ${theme}, ditujukan untuk audiens ${age}. Berikan respon yang panjang dan mengandung banyak baris kalimat.`;

    const userQuery = `Ide atau deskripsi singkat tentang buku: ${description}.`;

    // Parameter ketiga 'true' menandakan ini adalah permintaan buku
    generateContent(userQuery, systemPrompt, true);
  };

  return (
    <form onSubmit={handleSubmit} className="p-4 space-y-4">
      <label className="block text-sm font-medium text-gray-700">Ide atau Deskripsi Singkat Buku</label>
      <textarea
        className="w-full p-3 border border-gray-300 rounded-lg focus:ring-green-500 focus:border-green-500 transition duration-150"
        rows="3"
        placeholder="Contoh: Petualangan seekor kucing luar angkasa mencari galaksi baru."
        value={description}
        onChange={(e) => setDescription(e.target.value)}
        required
      ></textarea>

      <Dropdown label="Pilih Usia" options={ages} value={age} onChange={setAge} />
      <Dropdown label="Pilih Tema Buku" options={themes} value={theme} onChange={setTheme} />
    </form>
  );
}

```

```

<button
  type="submit"
  className="w-full bg-green-600 hover:bg-green-700 text-white font-bold py-3
rounded-lg shadow-md transition duration-300 ease-in-out transform hover:scale-[1.01]
mb-4"
>
  BUAT BUKU
</button>

{bookContent && (
  <button
    type="button"
    onClick={() => setIsBookViewing(true)}
    className="w-full bg-blue-600 hover:bg-blue-700 text-white font-bold py-3
rounded-lg shadow-md transition duration-300 ease-in-out transform hover:scale-[1.01]"
>
  AKSES BUKU
</button>
)
</form>
);
};

// --- KOMPONEN BOOK VIEWER (Update 4 - Desain Diperbarui) ---

```

```

const BookViewer = ({ bookContent, setIsBookViewing }) => {
  const [pageIndex, setPageIndex] = useState(0);
  const [pages, setPages] = useState([]);
  const [touchStart, setTouchStart] = useState(null);
  const LINES_PER_PAGE = 4;

  // Logika Pagination (4 baris kalimat per halaman)
  useEffect(() => {
    if (!bookContent) return;

    // Pisahkan konten menjadi baris
    // Filter baris kosong untuk memastikan hitungan 4 baris lebih akurat
    const allLines = bookContent.split("\n").filter(line => line.trim() !== "");

    // Gabungkan 4 baris menjadi satu halaman
    let tempPages = [];
    for (let i = 0; i < allLines.length; i += LINES_PER_PAGE) {
      tempPages.push(allLines.slice(i, i + LINES_PER_PAGE).join("\n"));
    }

    setPages(tempPages);
  }, [bookContent]);
}

// --- KOMPONEN BOOK VIEWER (Update 4 - Desain Diperbarui) ---

```

```
    setPageIndex(0);
}, [bookContent]);

// Navigasi
const nextPage = () => setPageIndex(prev => Math.min(prev + 1, pages.length - 1));
const prevPage = () => setPageIndex(prev => Math.max(prev - 1, 0));

// Handler Swipe
const handleTouchStart = (e) => {
    setTouchStart(e.touches[0].clientX);
};

const handleTouchEnd = (e) => {
    if (touchStart === null) return;
    const touchEnd = e.changedTouches[0].clientX;
    const deltaX = touchStart - touchEnd;
    const threshold = 50;

    if (deltaX > threshold) {
        nextPage(); // Geser ke kiri -> Halaman Selanjutnya
    } else if (deltaX < -threshold) {
        prevPage(); // Geser ke kanan -> Halaman Sebelumnya
    }
    setTouchStart(null);
};

if (pages.length === 0) {
    return <div className="p-10 text-center text-gray-500">Buku kosong.</div>;
}

return (
<div
    className="fixed inset-0 bg-gray-900 bg-opacity-95 z-50 flex flex-col
transition-opacity duration-300 ease-in-out font-serif"
    onT
```