

LABORATORIUM

Menggunakan model IBM
Granite untuk pembuatan
kode dan tugas
pemrograman

Daftar Isi

Pendahuluan	2
Persyaratan perangkat lunak.....	2
Tujuan.....	2
Langkah-langkah laboratorium	2
Perkiraan durasi untuk menyelesaikan	2
Skenario	3
Informasi latar belakang.....	3
Tantangan	3
Solusi.....	3
Langkah 1: Membuat akun GitHub	4
Ikhtisar.....	4
Instruksi	4
Langkah 2: Membuat akun Replicate	9
Ikhtisar.....	9
Instruksi	9
Langkah 3: Mendaftar ke Google Colab.....	15
Ikhtisar.....	15
Instruksi	15
Langkah 4: Memuat Jupyter notebook dan menginisialisasi model	20
Ikhtisar.....	20
Instruksi	20
Langkah 5: Membuat kode menggunakan model IBM Granite.....	27
Ikhtisar.....	27
Instruksi	27
Kesimpulan	31

Pengantar

Di lab ini, kamu akan menggunakan model IBM Granite untuk menghasilkan kode Python untuk skenario tertentu. Kamu akan menerapkan pengetahuanmu tentang teknik prompting untuk mendefinisikan dan menjalankan prompt guna menghasilkan kode menggunakan IBM Granite.

Persyaratan perangkat lunak

Untuk menyelesaikan lab ini, kamu memerlukan akses ke akun Replicate, yang memungkinkan kamu menggunakan model AI untuk melakukan tugas. Kamu juga memerlukan token API dari akun Replicate kamu. Token API seperti kunci digital yang memungkinkan lab terhubung dengan aman ke akun Replicate kamu. Token ini akan ditambahkan dengan aman ke lingkungan Google Colab kamu sehingga lab dapat berjalan dengan benar.

Meskipun kamu tidak perlu mengetahui Python untuk mengikutinya, keakraban dengannya dapat membantu kamu lebih memahami kode yang dibuat selama lab.

Tujuan

Setelah menyelesaikan lab ini, kamu seharusnya dapat:

- Menggunakan model IBM Granite untuk pembuatan kode dan tugas pemrograman

Langkah-langkah laboratorium

Lab ini mengharuskan kamu untuk menyelesaikan langkah-langkah berikut:

- Langkah 1: Membuat akun GitHub
- Langkah 2: Membuat akun Replicate
- Langkah 3: Mendaftar ke Google Colab
- Langkah 4: Memuat Jupyter notebook dan menginisialisasi model
- Langkah 5: Membuat kode menggunakan model IBM Granite

Perkiraan durasi untuk menyelesaikan

30 menit

Menggunakan model IBM Granite untuk pembuatan kode dan tugas pemrograman

IBM SkillsBuild

Skenario

Informasi latar belakang

Reader's Verse adalah toko buku lokal yang berencana membangun kehadiran online dengan membuat situs web yang memungkinkan para pembacanya untuk mencari katalog dan memeriksa ketersediaan sebelum mengunjungi toko fisik. Kamu adalah desainer web yang ditugaskan untuk proyek ini, dan Reader's Verse adalah klien kamu.

Tantangan

Klien mengharapkan kamu untuk menyelesaikan proyek dengan cepat dan meminta prototipe yang berfungsi. Kamu sudah menangani banyak proyek dan memiliki waktu terbatas untuk mengerjakan permintaan ini. Kamu memiliki pemahaman dasar tentang pemrograman dan pengodean tetapi tidak memiliki keahlian mendalam yang dibutuhkan untuk buat situs web dari perspektif pemrograman.

Solusi

Kamu mempertimbangkan untuk menggunakan model IBM Granite, yang dirancang untuk pembuatan kode, penjelasan kode, dan tugas pengeditan kode. Model-model ini dilatih secara ekstensif dengan kode yang ditulis dalam 116 bahasa pemrograman dan digunakan untuk berbagai pembuatan kode dan aktivitas terkait.

Menggunakan model IBM Granite untuk pembuatan kode dan tugas pemrograman

IBM SkillsBuild

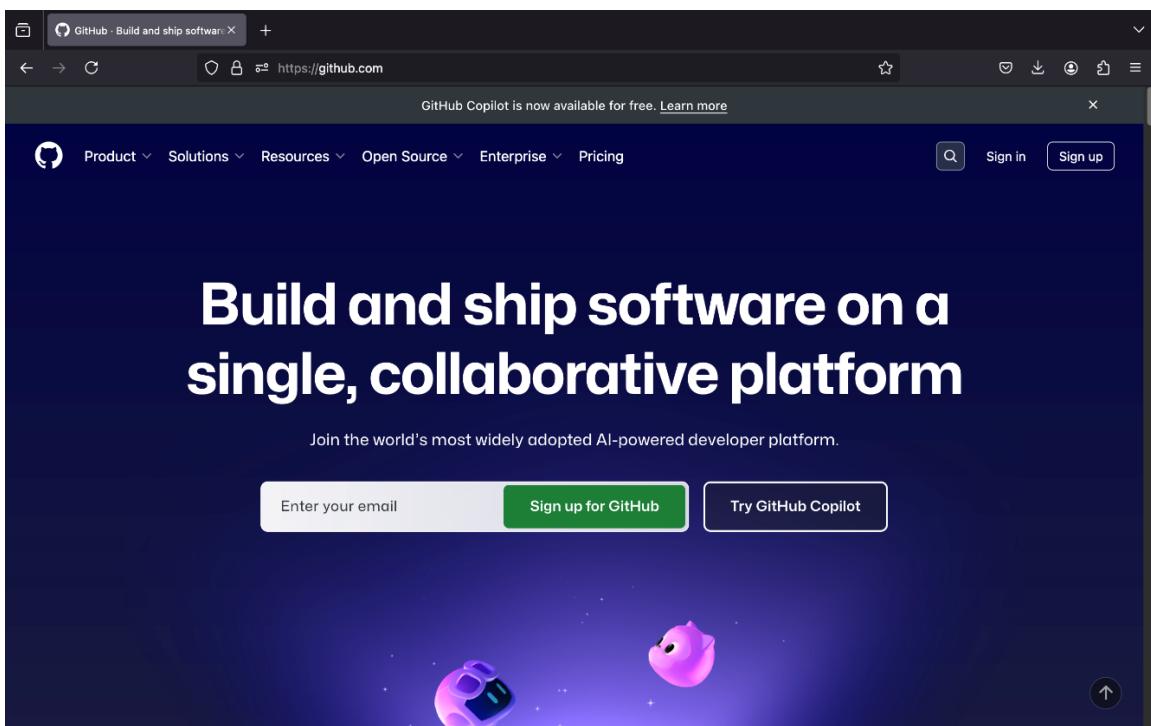
Langkah 1: Membuat akun GitHub

Ikhtisar

Pada langkah ini, kamu akan menyiapkan akun GitHub untuk mendaftar akun Replicate. GitHub adalah platform yang membantu pengembang menyimpan, mengelola, dan berbagi kode, sekaligus mendukung kolaborasi melalui alat seperti kontrol versi, pelacakan bug, dan manajemen tugas. Pengaturan ini memastikan kamu memiliki akses ke lingkungan cloud Replicate yang diperlukan untuk menyelesaikan lab secara efisien.

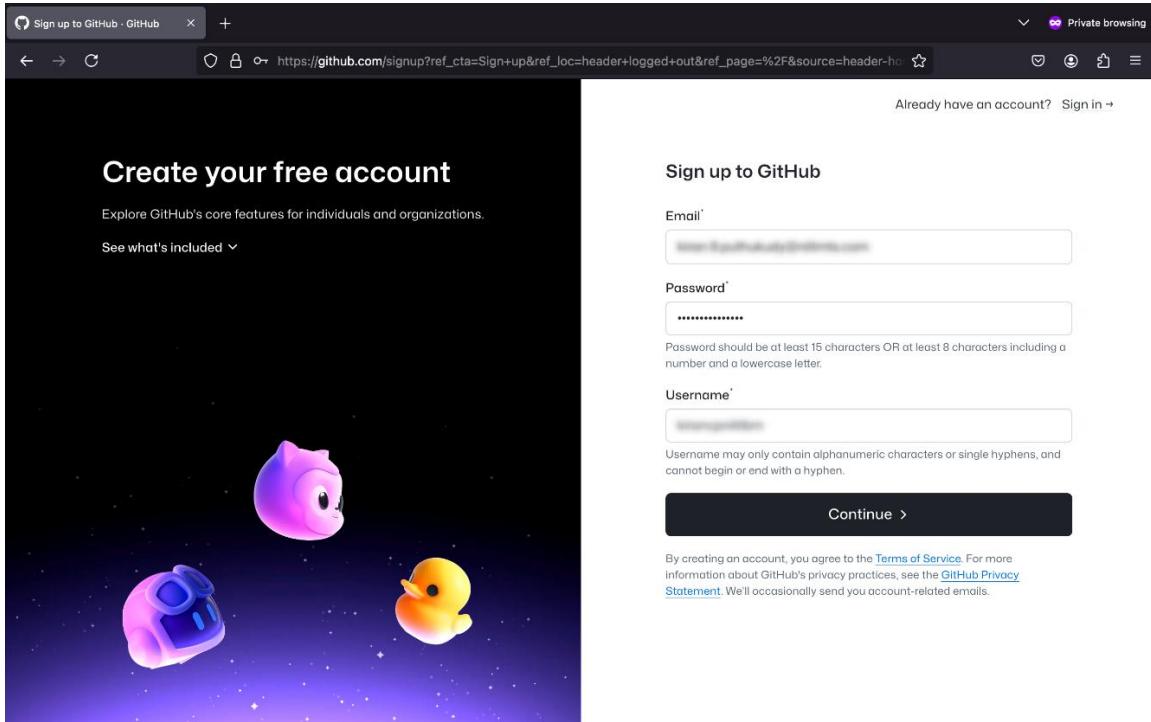
Instruksi

1. Buka situs web [GitHub](https://github.com) untuk membuat akun GitHub, lalu pilih **Sign up (Daftar)**.

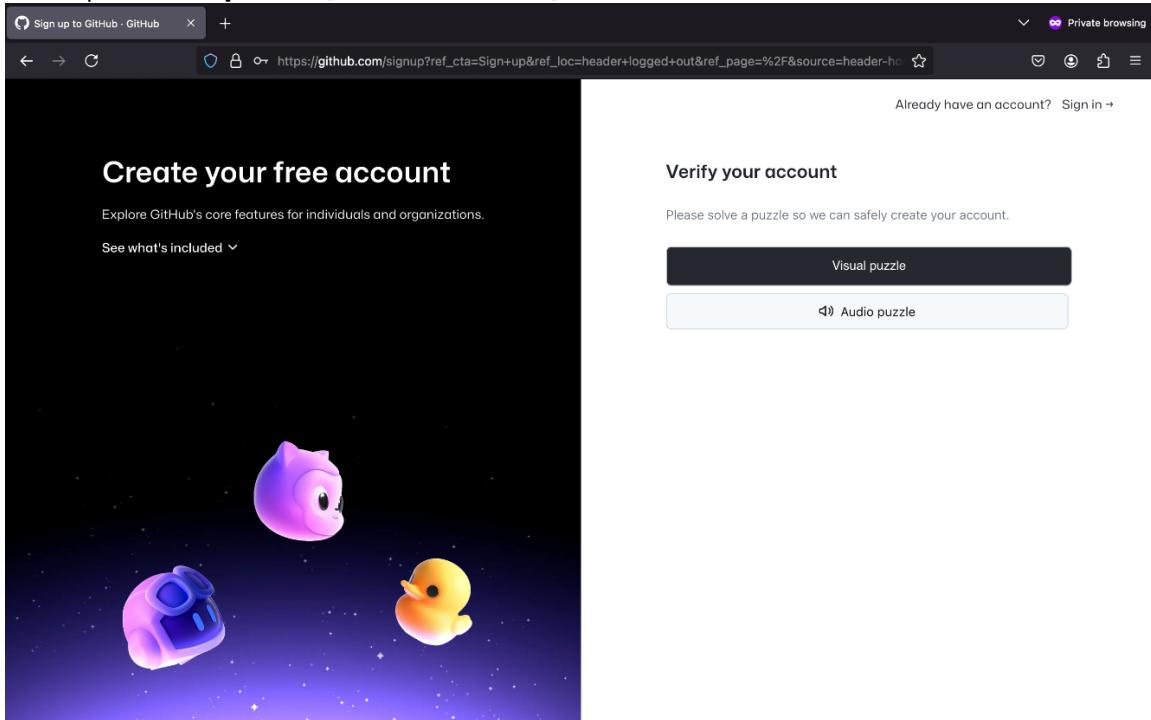


Menggunakan model IBM Granite untuk pembuatan kode dan tugas pemrograman

- Ketik detail kamu di bidang **Email (Email)**, **Password (Kata Sandi)**, dan **Username (Nama Pengguna)**. Lalu, pilih **Continue (Lanjutkan)**.



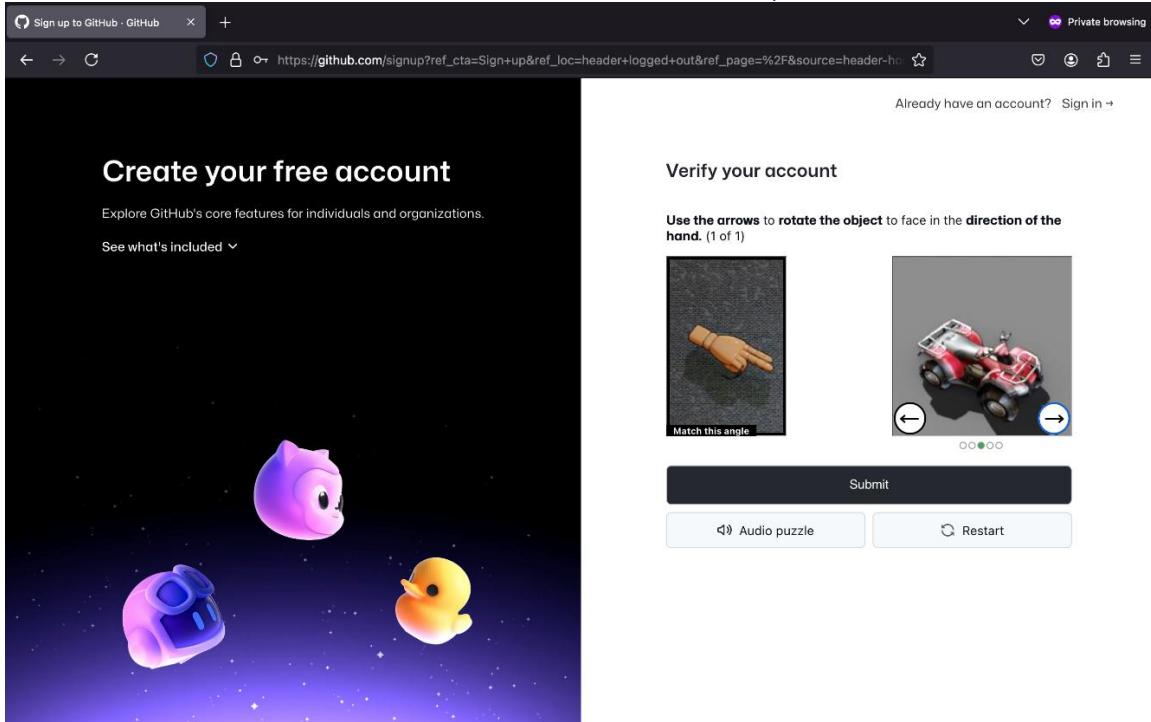
- Pilih opsi **Visual puzzle (Teka-teki visual)** untuk memverifikasi akun kamu.



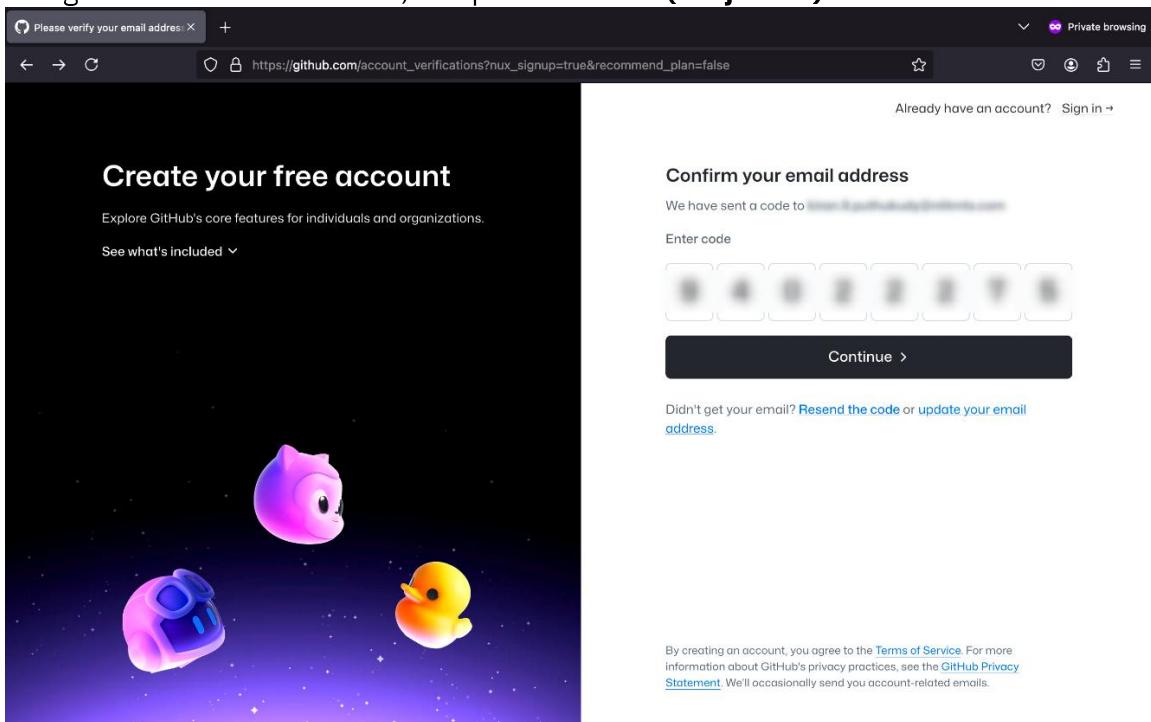
Menggunakan model IBM Granite untuk pembuatan kode dan tugas pemrograman

IBM SkillsBuild

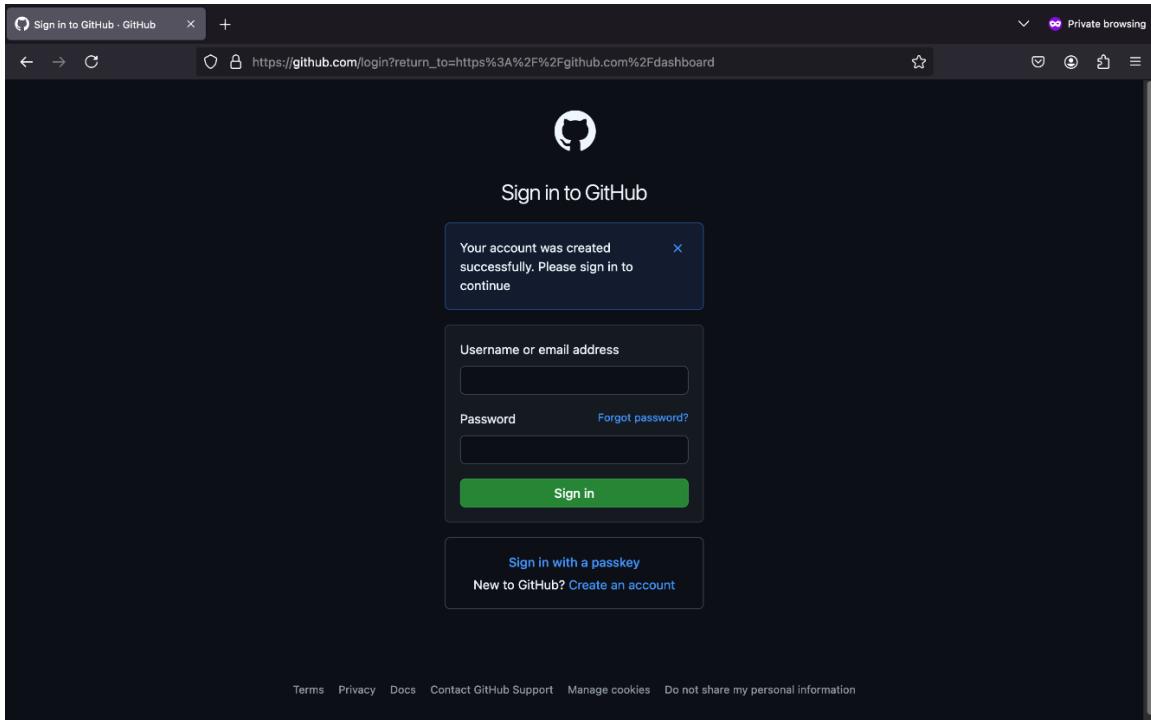
4. Selesaikan teka-teki untuk memverifikasi akun kamu, lalu pilih **Submit (Kirim)**.



5. Ketik kode konfirmasi pada bidang **Enter code (Masukkan kode)** untuk mengonfirmasi ID email kamu, lalu pilih **Continue (Lanjutkan)**.



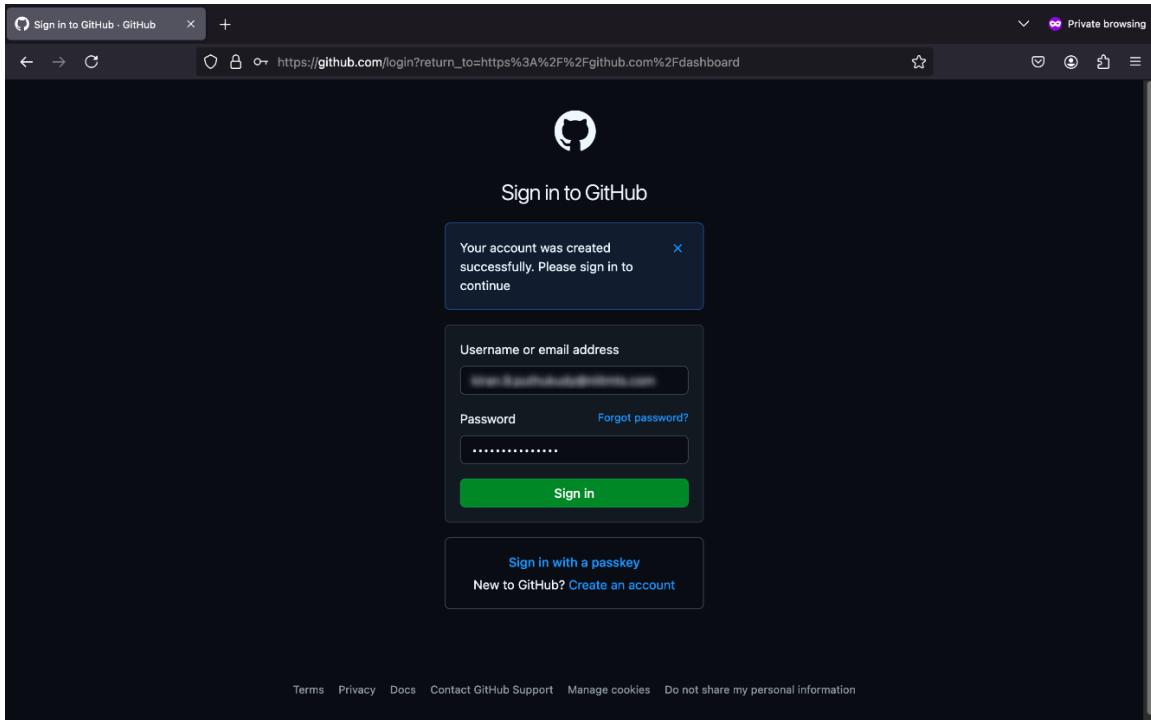
- Kamu akan melihat pesan konfirmasi setelah berhasil menyelesaikan akun GitHub.



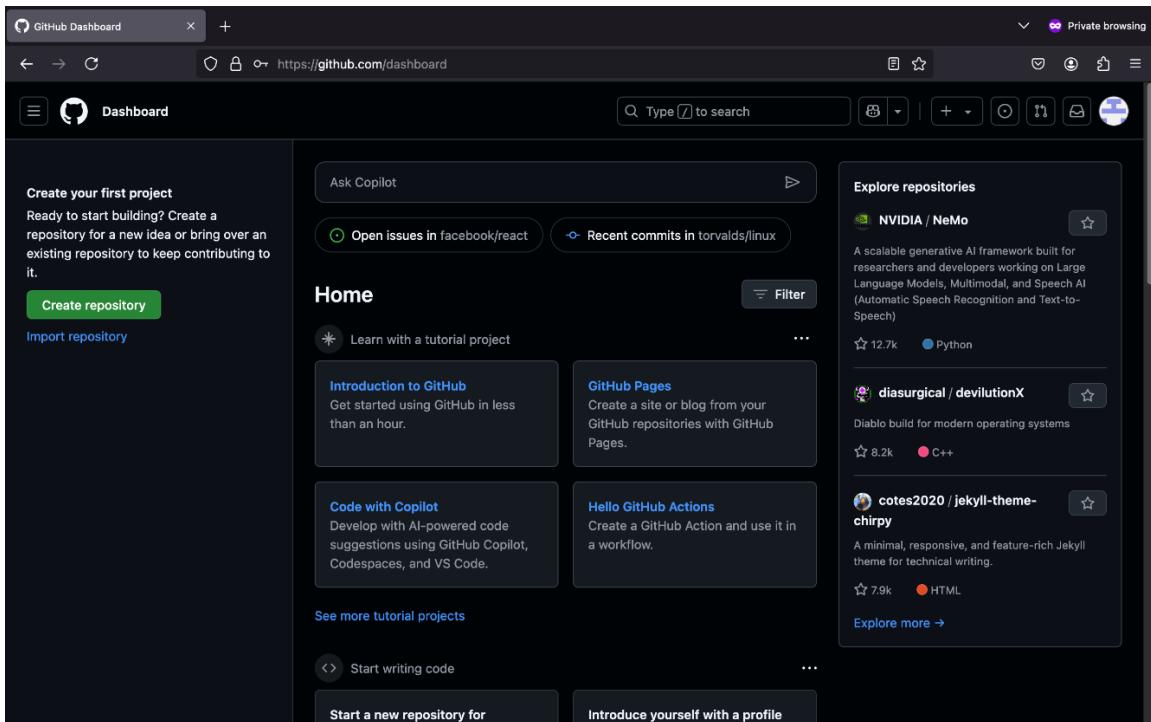
- Ketik kredensial kamu pada bidang **Username or email address (Nama Pengguna atau alamat email)** dan **Password (Kata Sandi)**, lalu pilih **Sign in (Masuk)**.

Menggunakan model IBM Granite untuk pembuatan kode dan tugas pemrograman

IBM SkillsBuild



- Dasbor GitHub ditampilkan untuk menunjukkan bahwa kamu telah berhasil masuk ke akun GitHub.



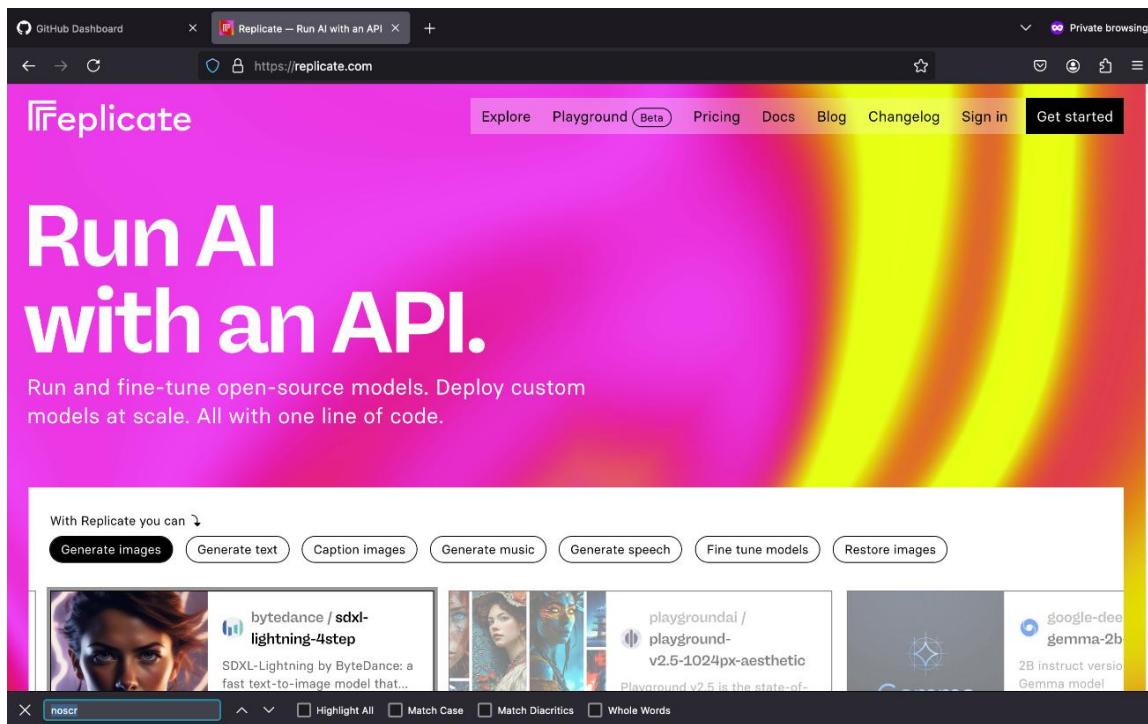
Langkah 2: Membuat akun Replicate

Ikhtisar

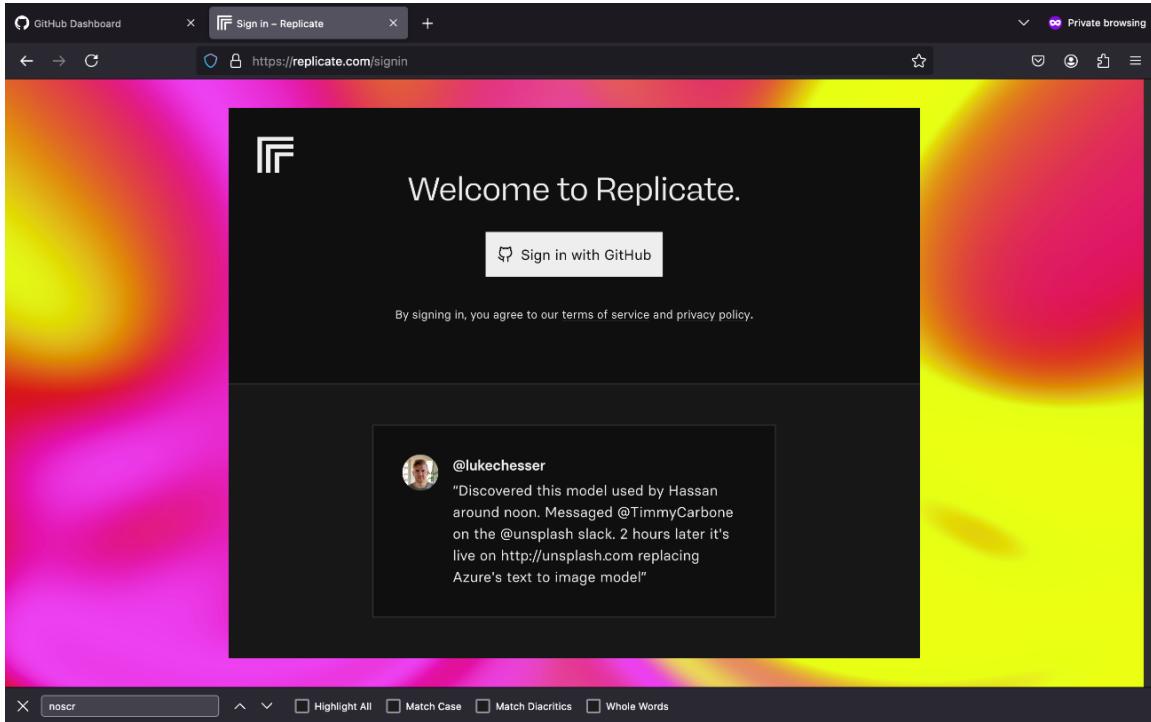
Pada langkah ini, kamu akan menggunakan akun GitHub untuk mendaftar akun Replicate. Replicate adalah platform berbasis cloud yang memungkinkan kamu menggunakan model AI tanpa memerlukan perangkat keras kompleks. Sebagai bagian dari langkah ini, kamu akan membuat token Replicate. Token seperti kunci digital aman yang memungkinkan lab terhubung ke akun Replicate dan mengakses alat yang diperlukan untuk menjalankan lab di Google Colab.

Instruksi

1. Buka situs web [Replicate](https://replicate.com), lalu pilih **Get started (Mulai)**.



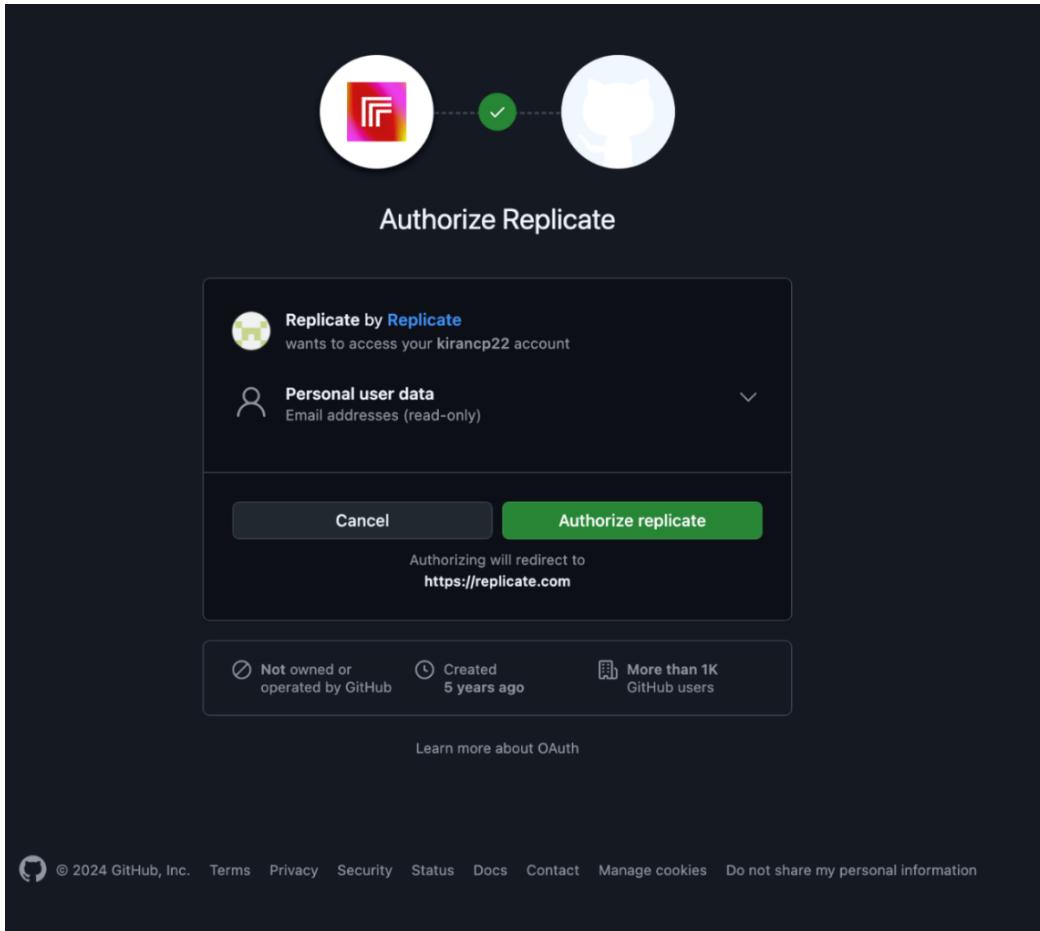
2. Pilih **Sign in with GitHub (Masuk dengan GitHub)** pada halaman Welcome to Replicate (Selamat Datang di Replicate).



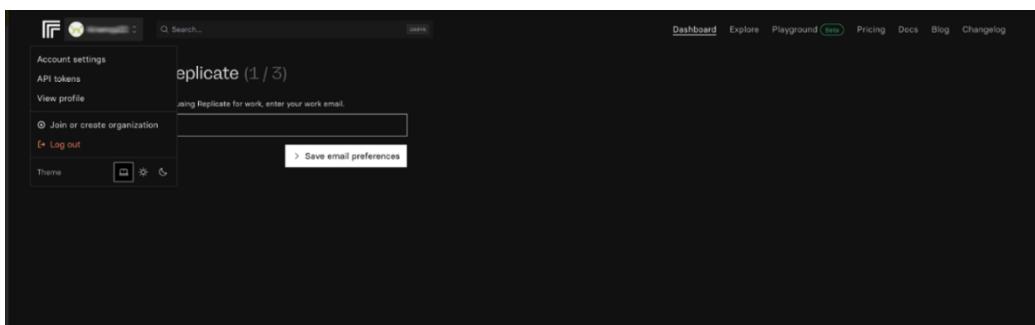
Menggunakan model IBM Granite untuk pembuatan kode dan tugas pemrograman

IBM SkillsBuild

- Pilih **Authorize replicate (Otorisasi replicate)** untuk melanjutkan.



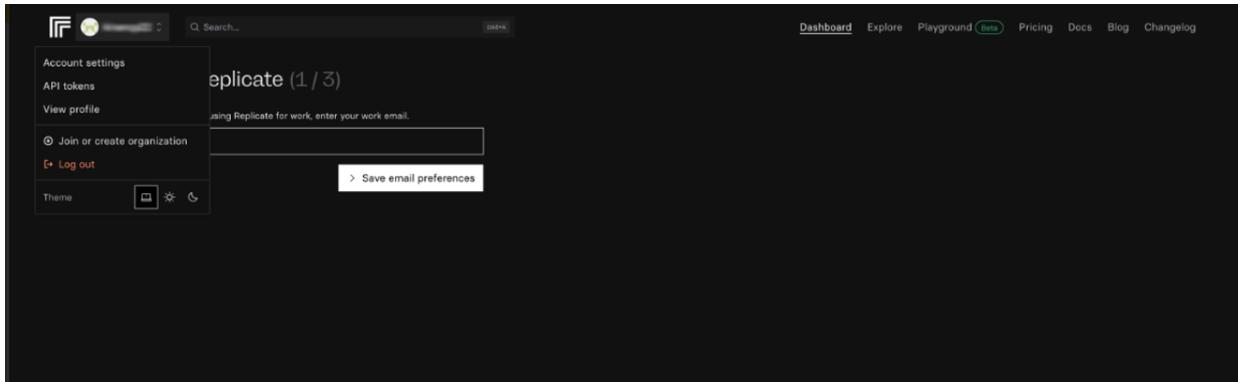
- Untuk membuat token Replicate, pilih ikon **Account setting (Pengaturan akun)** pada bilah navigasi.



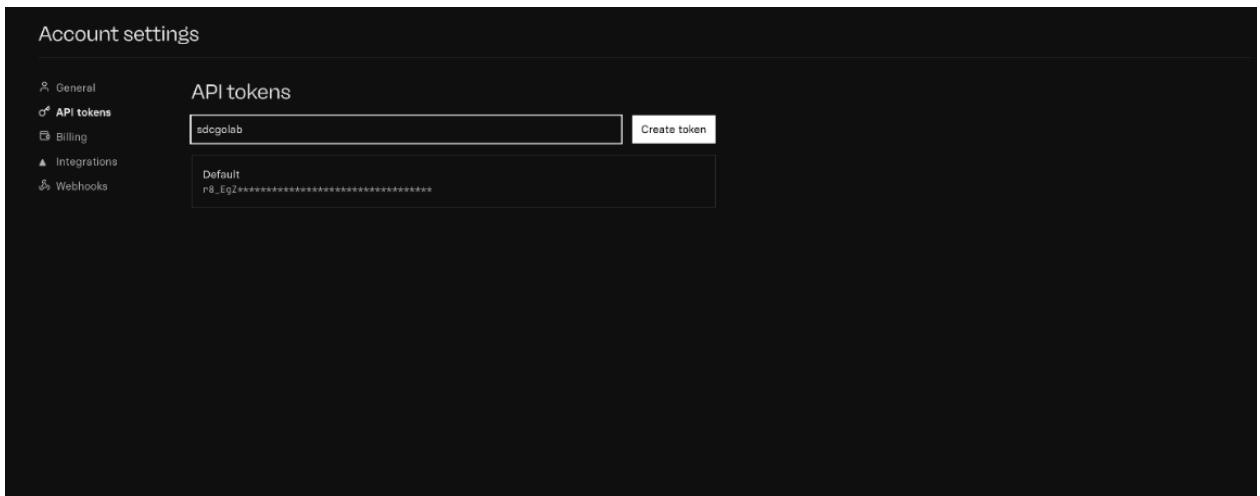
Menggunakan model IBM Granite untuk pembuatan kode dan tugas pemrograman

IBM SkillsBuild

- Pilih **API tokens (token API)** pada menu “Account setting (Pengaturan akun)”.

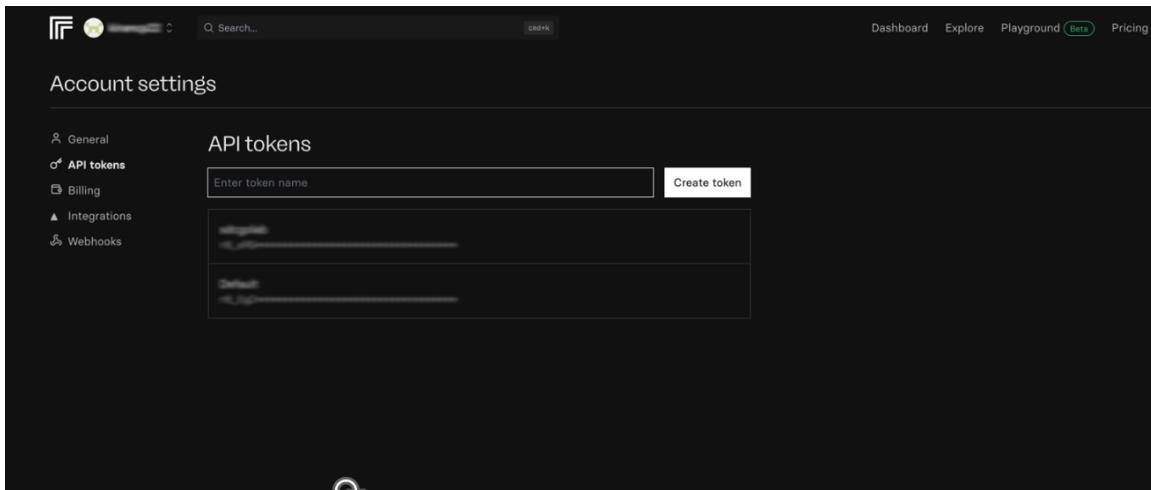


- Ketik nama untuk token di bidang **API tokens (token API)**, lalu pilih **Create token (Buat token)**.



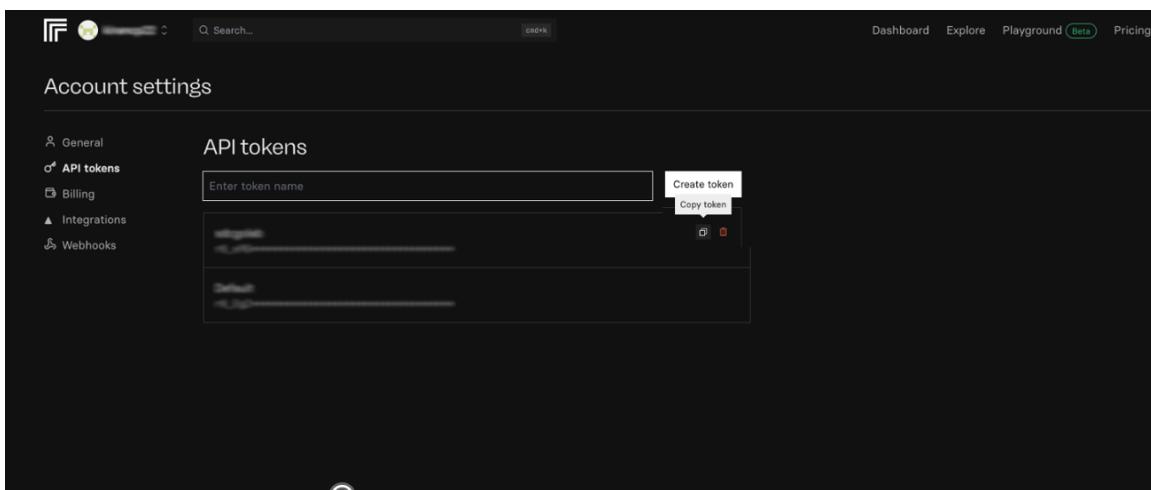
Menggunakan model IBM Granite untuk pembuatan kode dan tugas pemrograman

7. Token API kamu akan ditampilkan di layar setelah bidang **API tokens (token API)**.



8. Pilih ikon **Copy token (Salin token)** untuk menyalin token API Replicate.

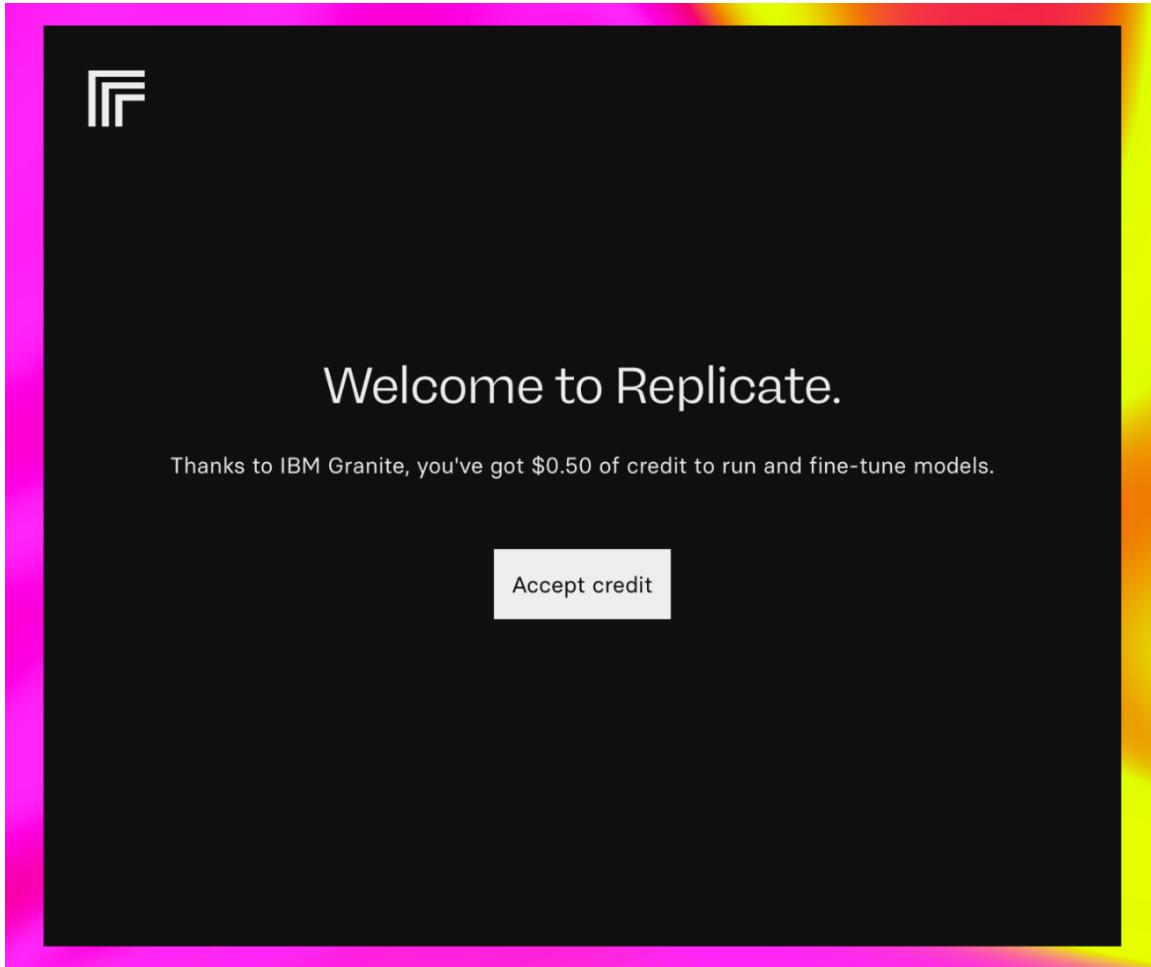
Catatan: Simpan token Replicate karena kamu akan memerlukan token Replicate API untuk menjalankan instans Google Colab di lab ini.



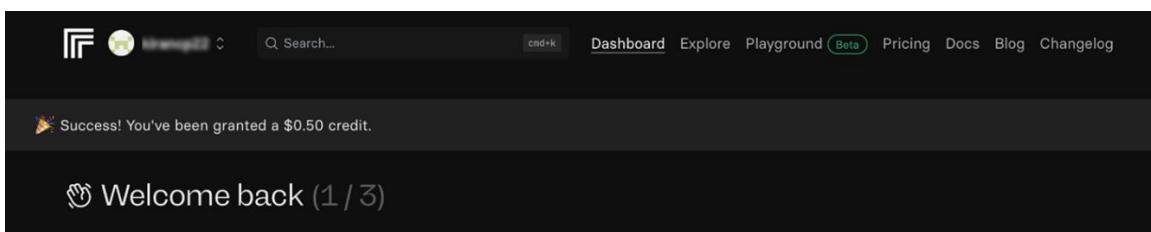
Menggunakan model IBM Granite untuk pembuatan kode dan tugas pemrograman

IBM SkillsBuild

- Buka tautan [Undangan Replicate](#) untuk mengklaim 50 sen kredit Replicate, yang akan kamu gunakan untuk menjalankan lab. Pilih **Accept kredit (Terima kredit)** untuk mengklaim kredit.



- Kamu akan melihat pesan konfirmasi yang mengindikasikan bahwa kredit 50 sen telah ditambahkan ke akun Replicate kamu sebagai berikut.



Menggunakan model IBM Granite untuk pembuatan kode dan tugas pemrograman

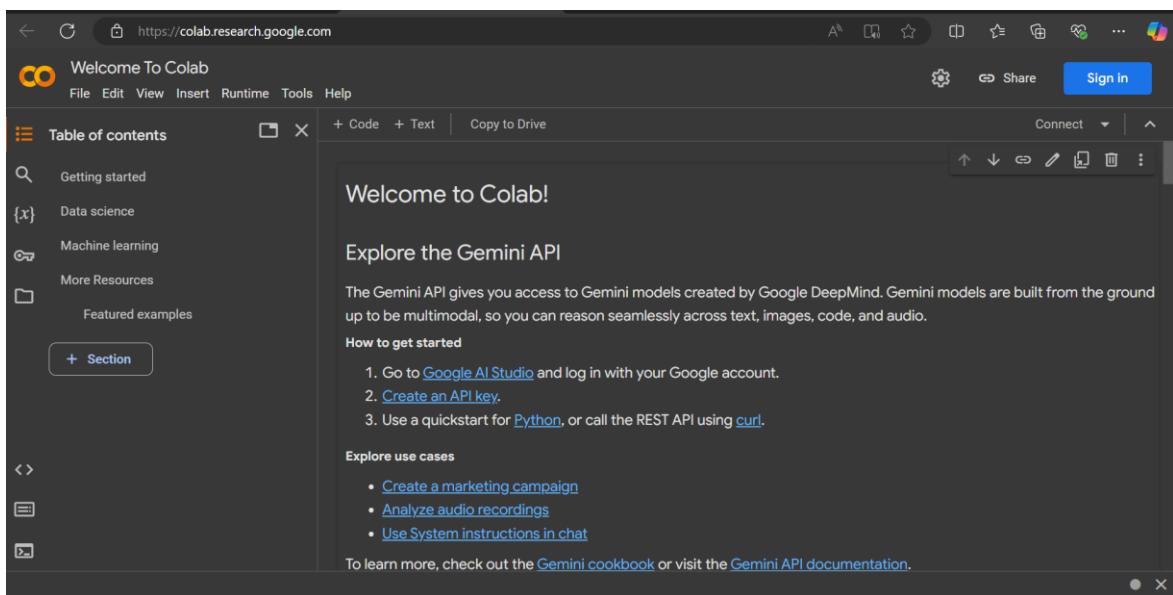
Langkah 3: Mendaftar ke Google Colab

Ikhtisar

Pada langkah ini, kamu akan menyiapkan akun Google Colab. Google Colab adalah platform cloud gratis yang memungkinkan kamu menjalankan kode di notebook, yang biasanya digunakan untuk pembelajaran mesin, ilmu data, dan tugas-tugas AI. Akun ini akan memungkinkan kamu untuk menginstal dan menggunakan alat yang diperlukan untuk menyelesaikan lab.

Instruksi

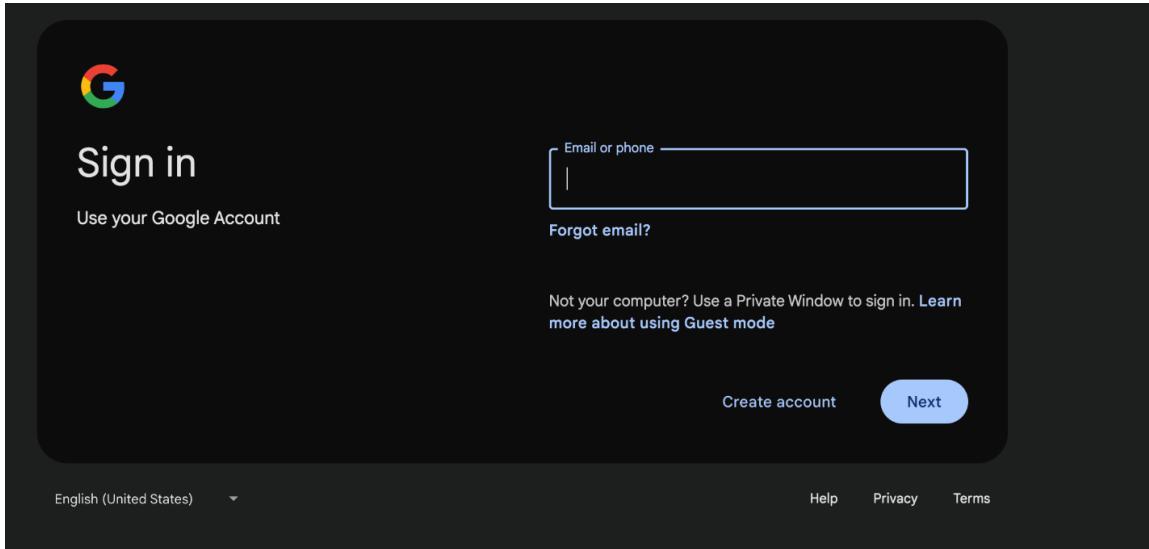
1. Buka situs web [Google Colab](https://colab.research.google.com) untuk mendaftar akun Google Colab, lalu pilih **Sign in (Masuk)**.



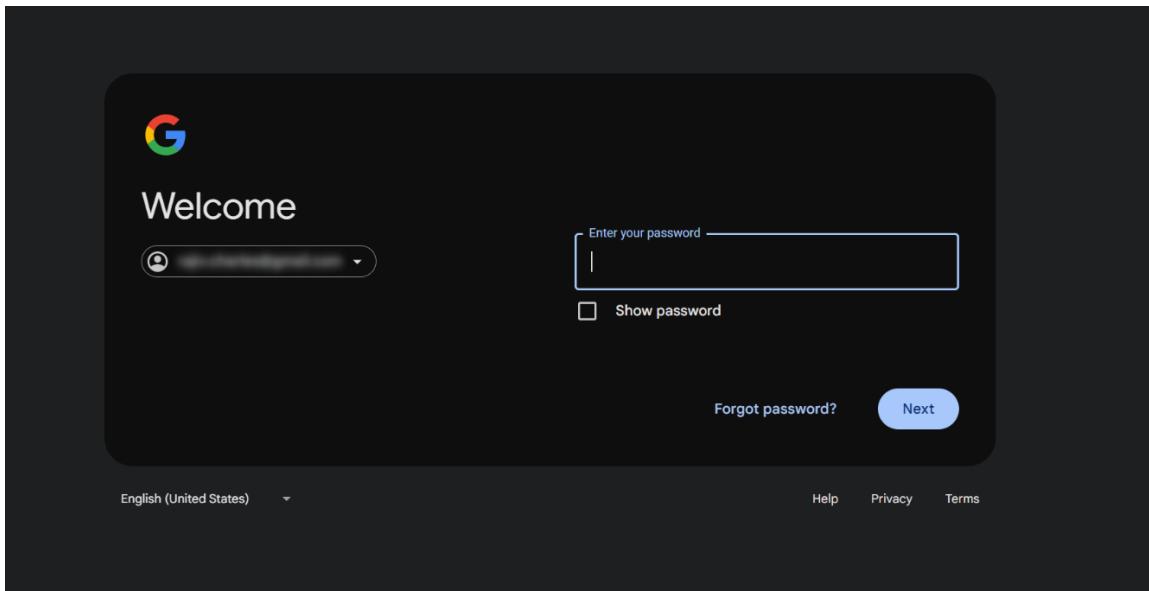
Menggunakan model IBM Granite untuk pembuatan kode dan tugas pemrograman

IBM SkillsBuild

- Ketik email Google atau nomor telepon kamu di bidang **Email or phone (Email atau telepon)**, lalu pilih **Next (Berikutnya)**.

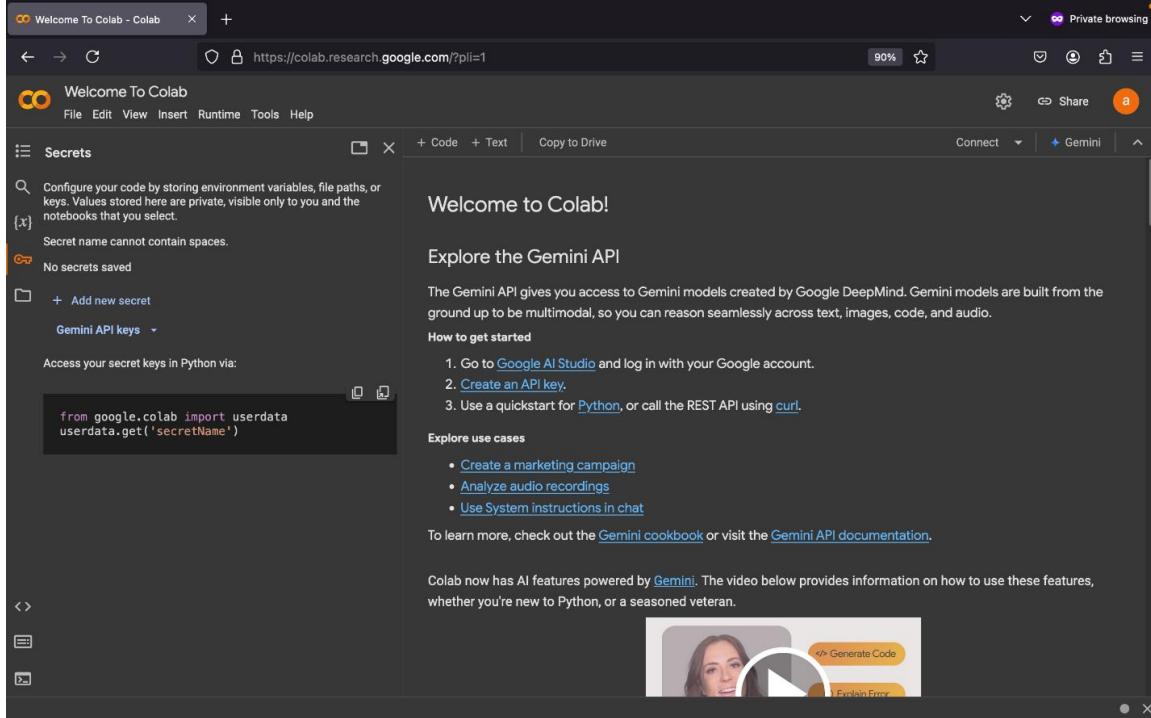


- Ketik kata sandi kamu di bidang **Enter your password (Masukkan kata sandi)**, lalu pilih **Next (Berikutnya)**.

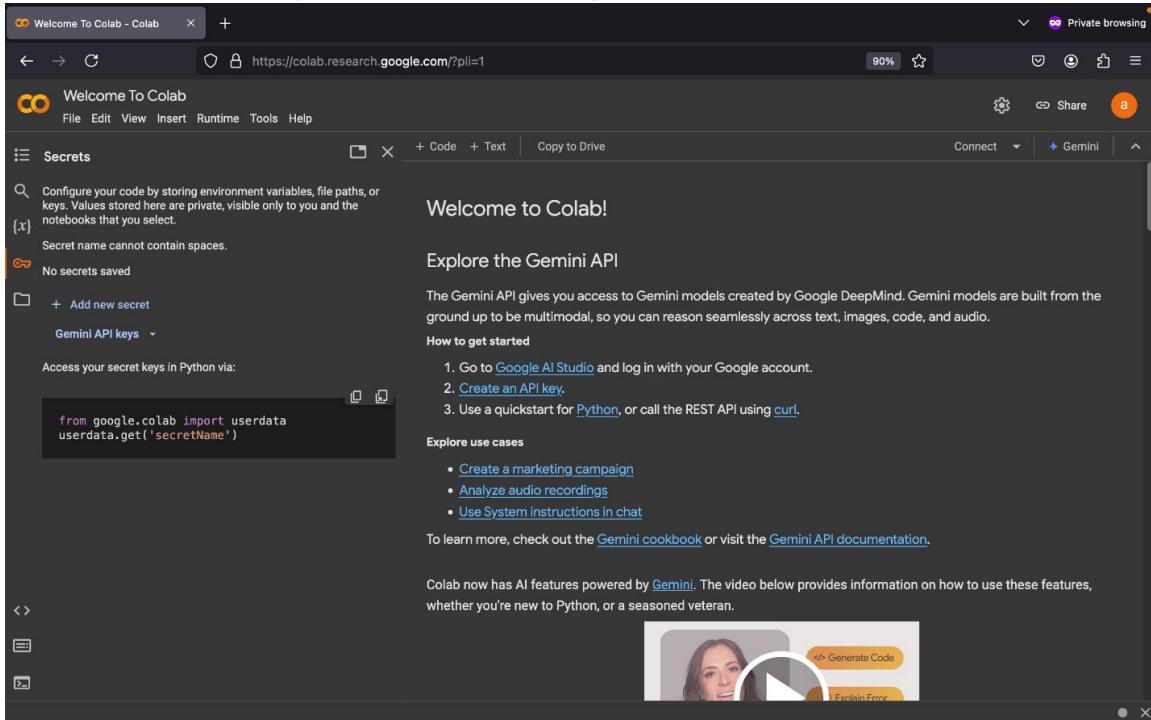


Menggunakan model IBM Granite untuk pembuatan kode dan tugas pemrograman

- Selanjutnya, pilih ikon **Kunci** pada menu **bilah sisi** di beranda Welcome to Colab (Selamat Datang di Colab) untuk menyimpan token API Replicate kamu dalam Google Colab secret (rahasia Google Colab).



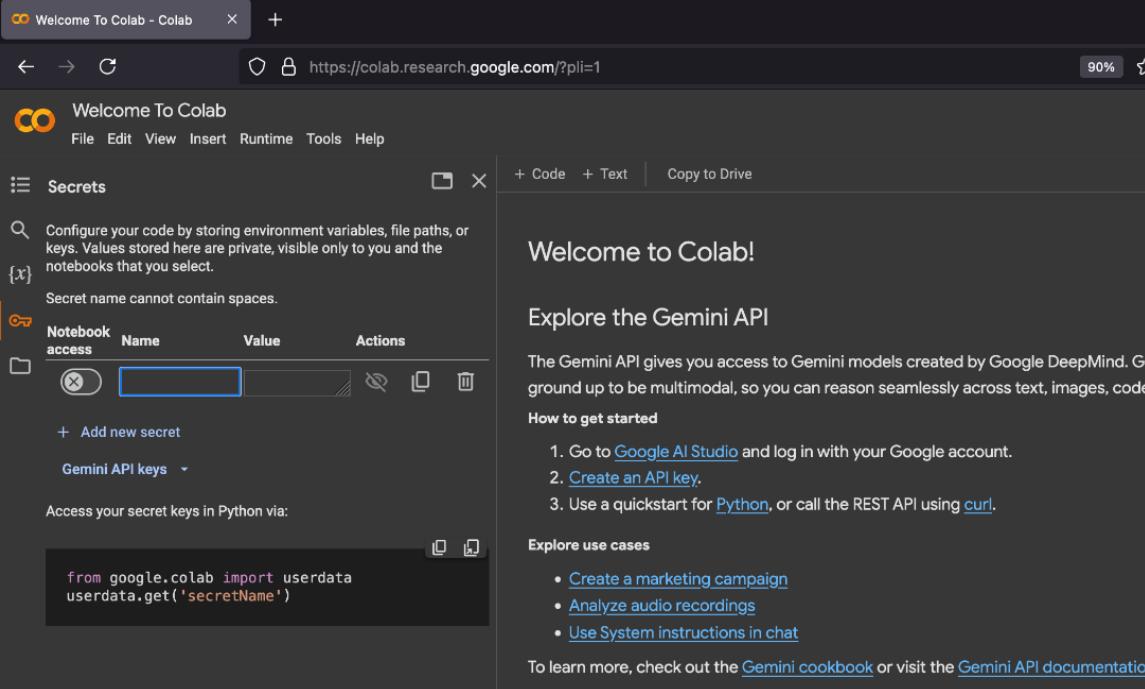
- Pilih **Add new secret** (Tambah rahasia baru).



Menggunakan model IBM Granite untuk pembuatan kode dan tugas pemrograman

IBM SkillsBuild

6. Ketik **REPLICATE_API_TOKEN** di bidang **Name (Nama)**.

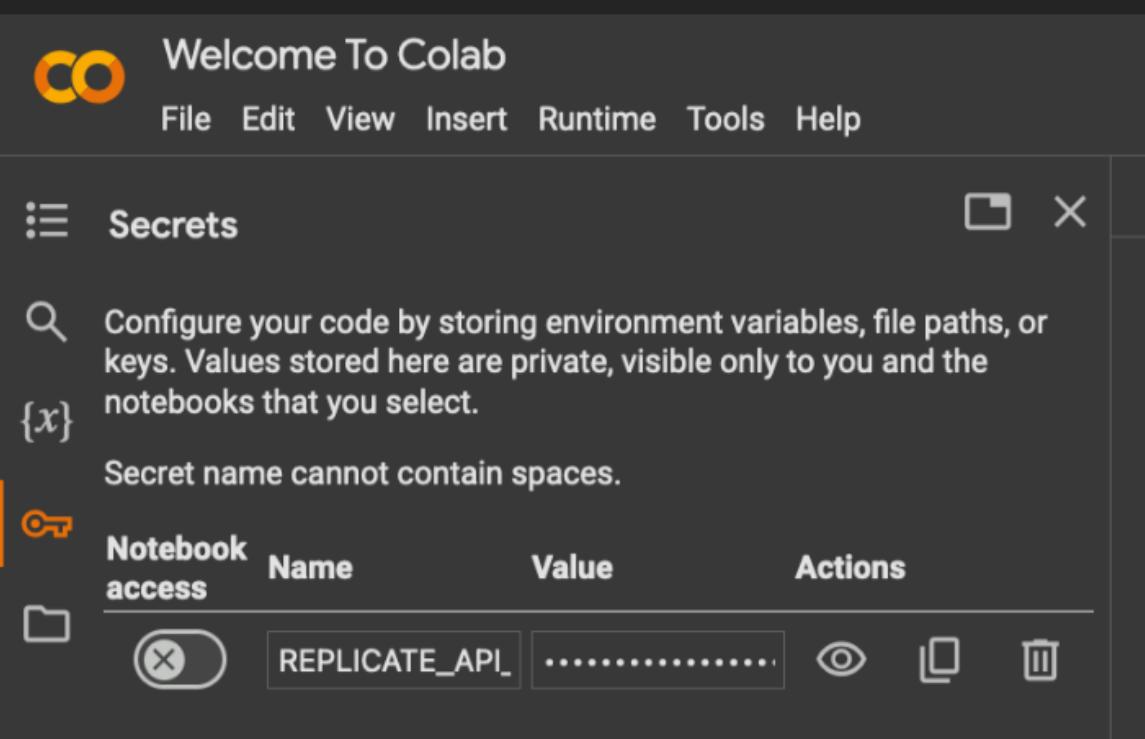


The screenshot shows the Google Colab interface with the title "Welcome To Colab". On the left, there's a sidebar titled "Secrets" which contains a search bar and a note about configuring code by storing environment variables or keys. Below this is a table with columns "Notebook access", "Name", "Value", and "Actions". A single row is present with a delete icon in the Actions column. At the bottom of the sidebar, there's Python code for retrieving a secret key:

```
from google.colab import userdata
userdata.get('secretName')
```

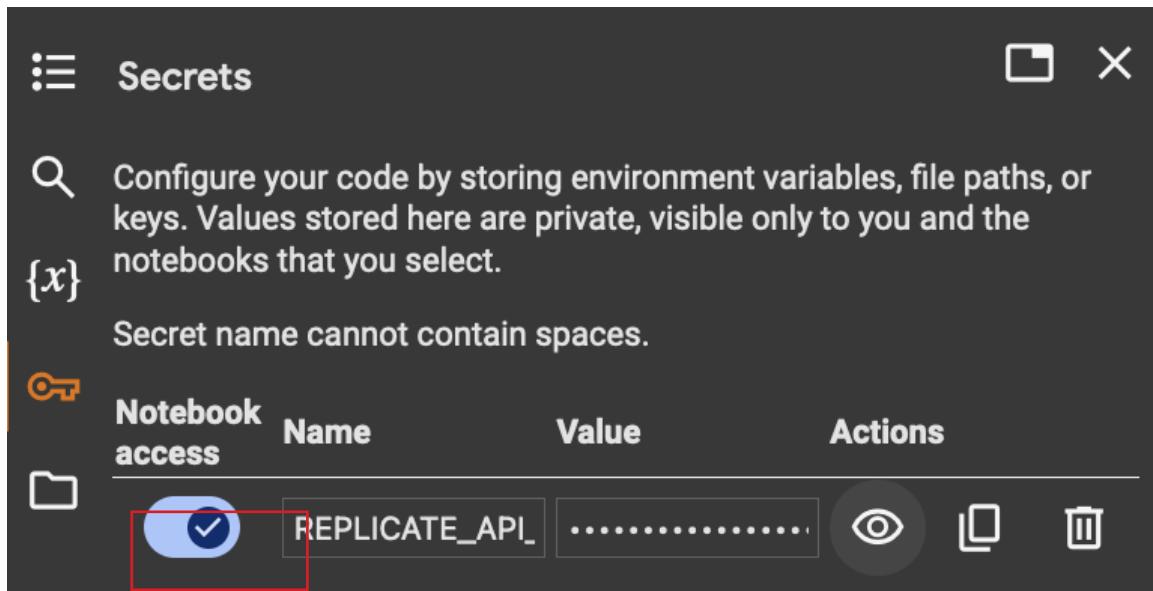
The main content area features a "Welcome to Colab!" message and a "Explore the Gemini API" section. It includes a "How to get started" list with three steps: 1. Go to [Google AI Studio](#) and log in with your Google account. 2. [Create an API key](#). 3. Use a quickstart for [Python](#), or call the REST API using [curl](#). There's also a "Explore use cases" section with links to "Create a marketing campaign", "Analyze audio recordings", and "Use System instructions in chat".

7. Tempelkan token API Replicate ke dalam bidang **Value (Nilai)**.



This screenshot shows the "Secrets" section in Google Colab with one row added. The table has columns "Notebook access", "Name", "Value", and "Actions". The "Name" column contains "REPLICATE_API_". The "Value" column contains a series of dots (...). The "Actions" column contains icons for edit, copy, and delete.

- Pilih **toggle** untuk mengaktifkan **Notebook access (Akses Notebook)**. Selanjutnya, pilih **X (Tutup)** untuk keluar dari konfigurasi.



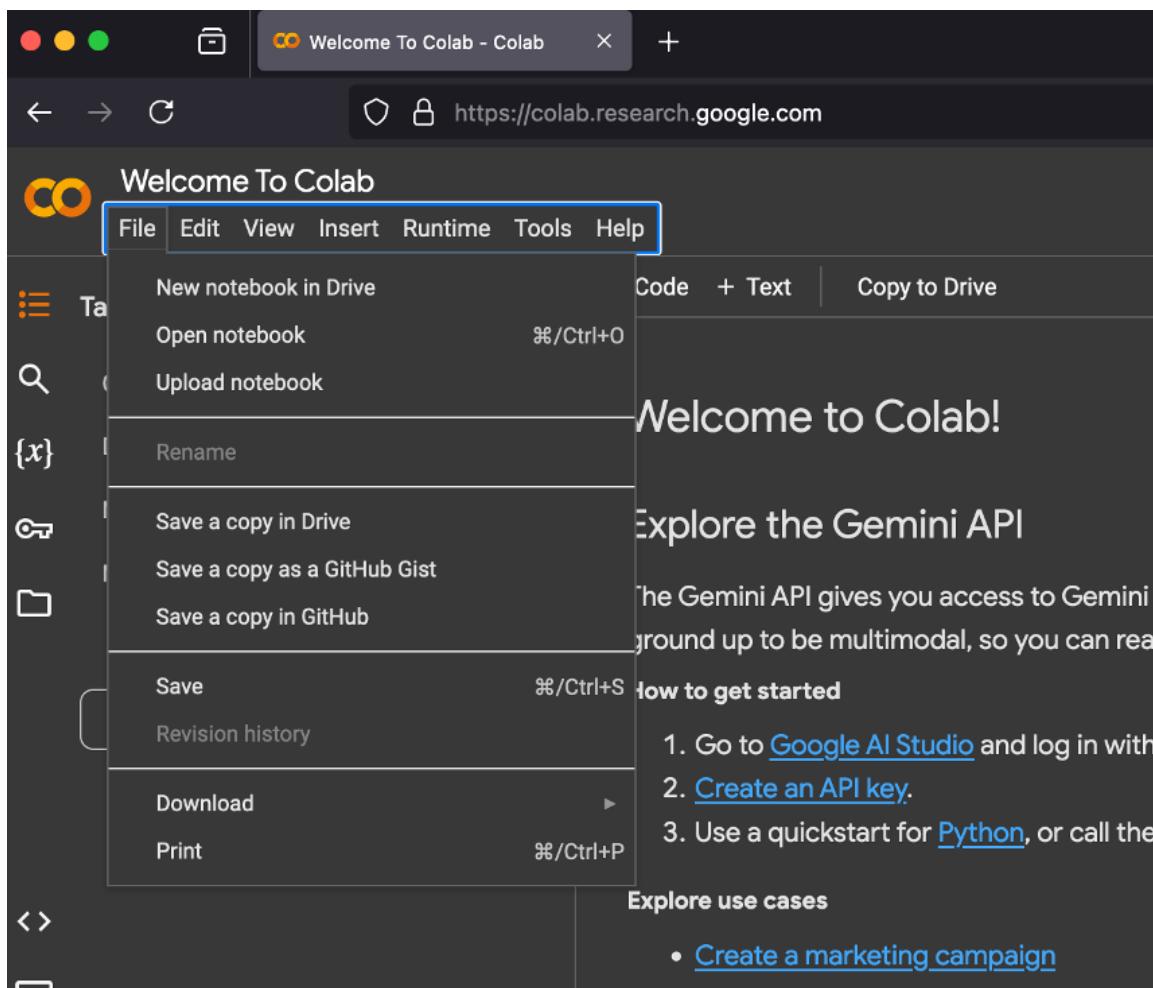
Langkah 4: Memuat Jupyter notebook dan menginisialisasi model

Ikhtisar

Pada langkah ini, kamu akan memuat Jupyter notebook yang berisi kode yang diperlukan untuk melakukan lab ini. Jupyter notebook adalah dokumen yang dapat dibagikan yang menggabungkan kode komputer, deskripsi bahasa sederhana, data, dan visualisasi. Kamu akan memuat file Jupyter notebook dari GitHub ke Google Colab dan menginisialisasi model IBM Granite untuk menghasilkan kode untuk skenario yang diberikan.

Instruksi

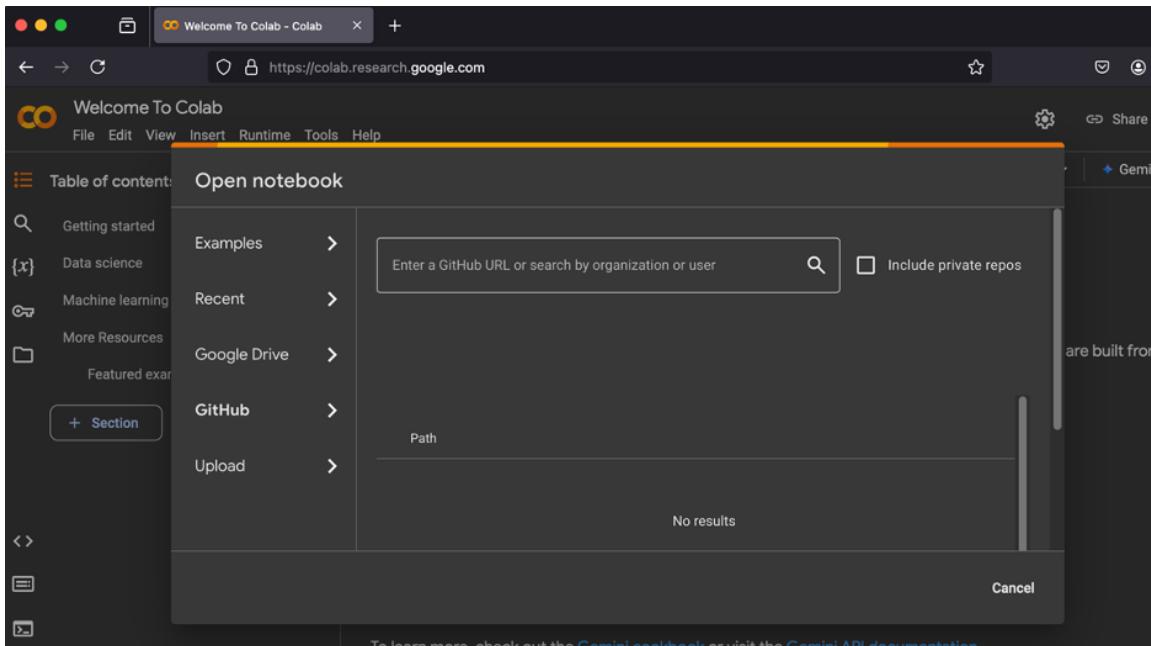
1. Di ruang kerja Google Colab, pada menu **File**, pilih **Open notebook (Buka notebook)**.



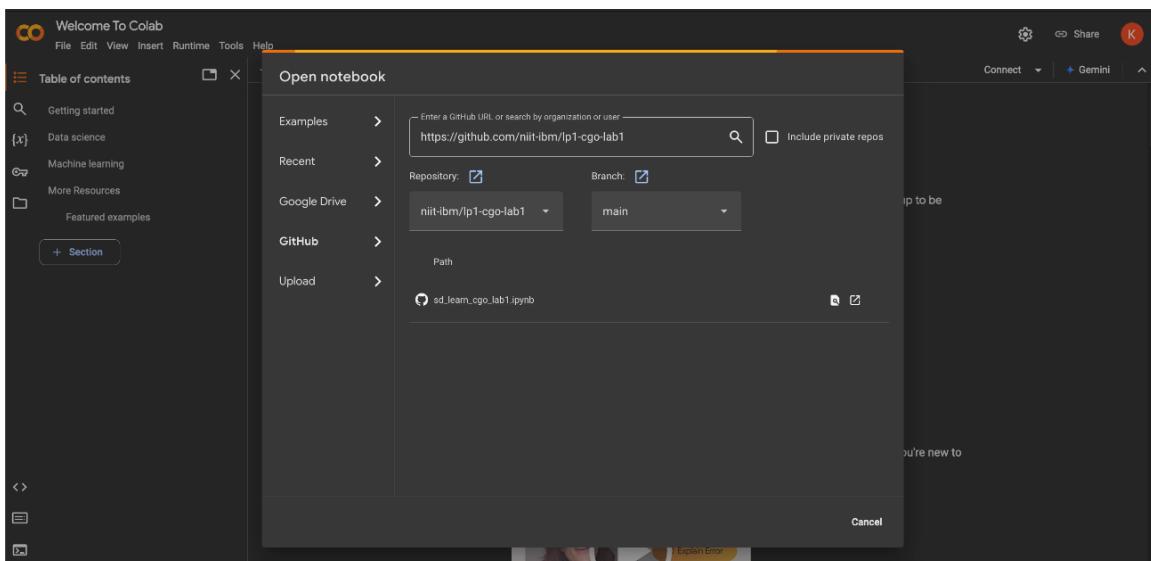
Menggunakan model IBM Granite untuk pembuatan kode dan tugas pemrograman

IBM SkillsBuild

2. Pilih tab GitHub.

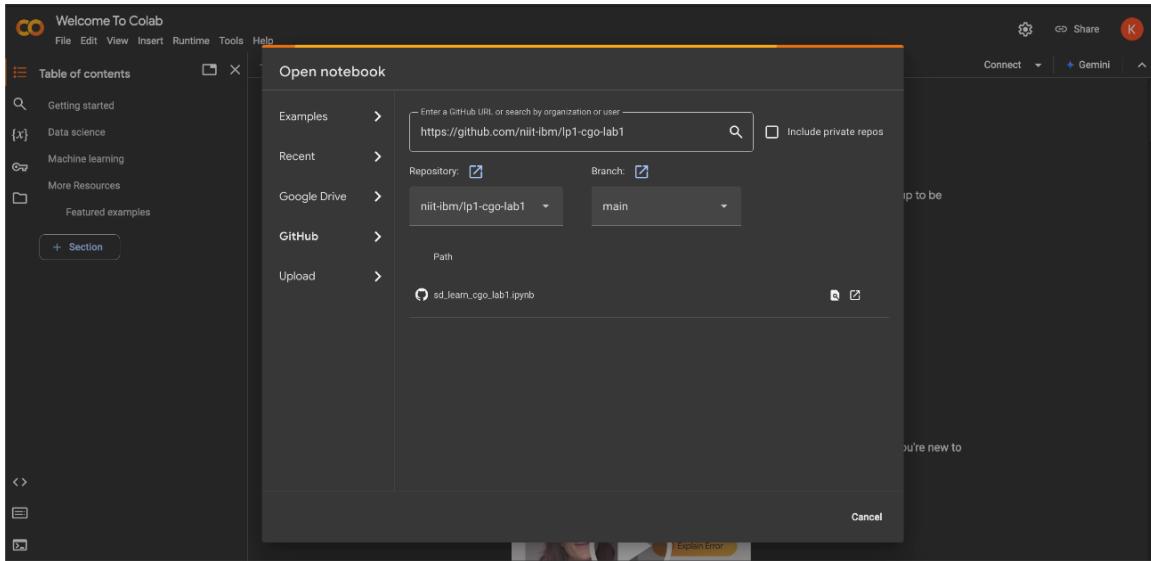


3. Pada bidang **Enter a GitHub URL or search by organization or user (Masukkan URL GitHub atau cari berdasarkan organisasi atau pengguna)**, salin URL berikut ini: <https://github.com/niit-ibm/lp1-cgo-lab1> dan pilih ikon kaca pembesar.

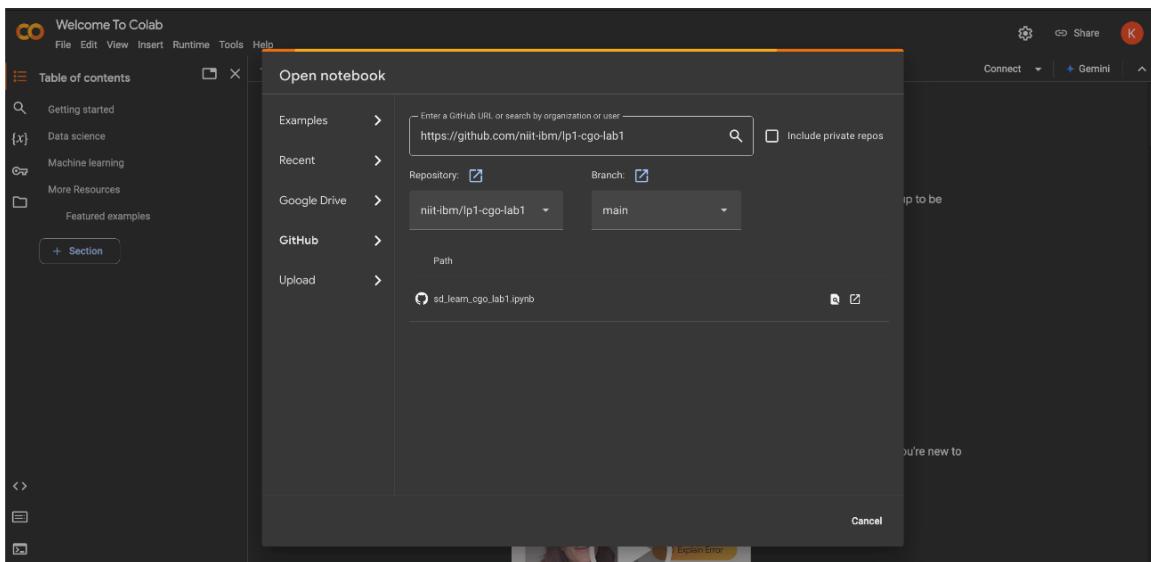


Menggunakan model IBM Granite untuk pembuatan kode dan tugas pemrograman

4. Pilih **main (utama)** di bagian **Branch (Cabang)**.

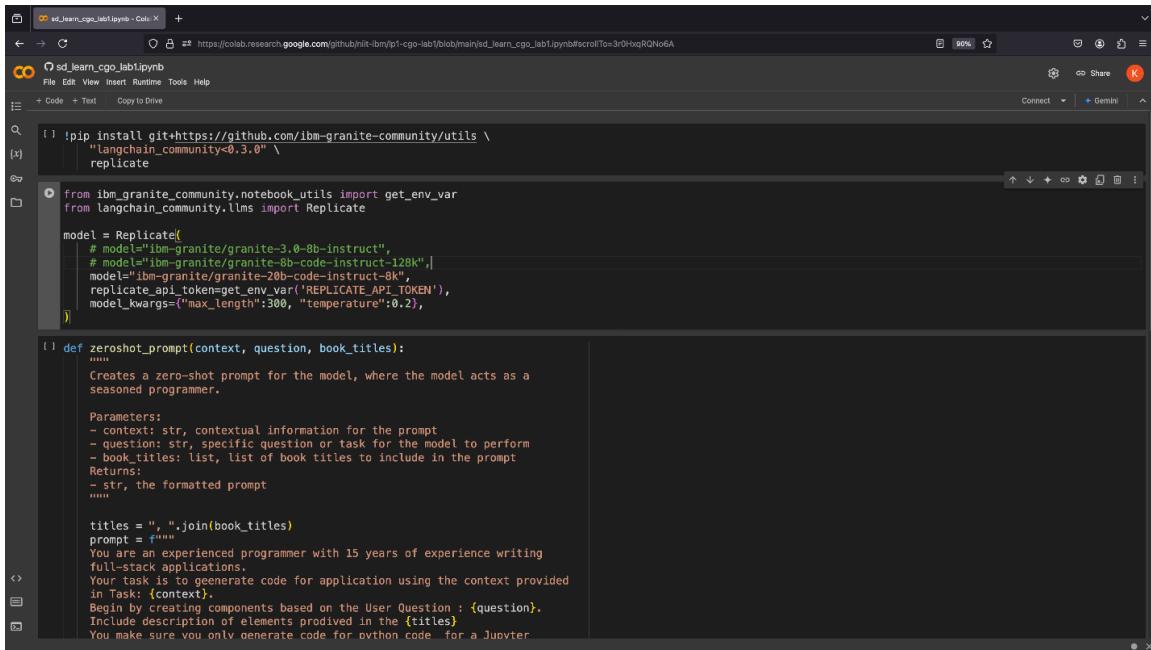


5. Pilih notebook **sd_learn_cgo_lab1.ipynb** di bagian **Path (Jalur)** untuk membuka notebook di Colab.



6. Kamu sekarang akan membuka notebook sd_learn_cgo_lab1 di ruang kerja Colab.

Perhatikan bahwa setiap baris di buku catatan disebut sebagai **sel**. Sel-sel dalam buku catatan tidak diberi nomor. Notebook diatur sedemikian rupa sehingga kamu harus menjalankan kode di setiap sel secara berurutan, dimulai dari sel pertama, untuk menyelesaikan lab ini.



```
!pip install git+https://github.com/ibm-granite-community/utils \
"langchain_community==0.3.0" \
replicate

from ibm_granite_community.notebook_utils import get_env_var
from langchain_community.llms import Replicate

model = Replicate(
    # model="ibm-granite-3.0-8b-instruct",
    # model="ibm-granite/granite-3b-code-instruct-128K",
    model="ibm-granite/granite-20b-code-instruct-8k",
    replicate_api_token=get_env_var('REPLICATE_API_TOKEN'),
    model_kwargs={"max_length":300, "temperature":0.2},
)

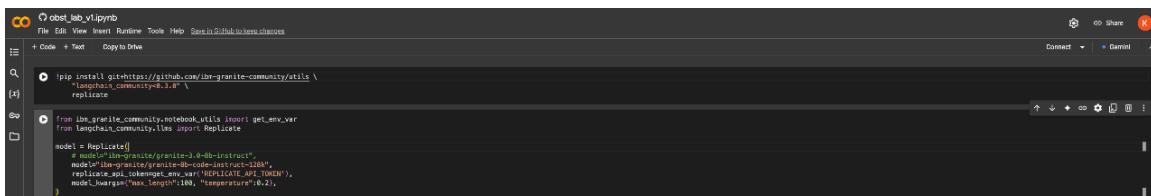
def zeroshot_prompt(context, question, book_titles):
    """
    Creates a zero-shot prompt for the model, where the model acts as a
    seasoned programmer.

    Parameters:
    - context: str, contextual information for the prompt
    - question: str, specific question or task for the model to perform
    - book_titles: list, list of book titles to include in the prompt
    Returns:
    - str, the formatted prompt
    """

    titles = ", ".join(book_titles)
    prompt = f"""
You are an experienced programmer with 15 years of experience writing
full-stack applications,
Your task is to generate code for application using the context provided
in Task: {context}.
Begin by creating components based on the User Question : {question}.
Include description of elements produced in the {titles}
You make sure you only generate code for python code for a Jupyter
"""

    return prompt
```

7. Kamu harus memulai contoh baru di runtime Google Colab untuk menjalankan kode di Jupyter notebook. Pilih **Connect (Hubungkan)** di bilah navigasi Google Colab.



```
!pip install git+https://github.com/ibm-granite-community/utils \
"langchain_community==0.3.0" \
replicate

from ibm_granite_community.notebook_utils import get_env_var
from langchain_community.llms import Replicate

model = Replicate(
    # model="ibm-granite-3.0-8b-instruct",
    # model="ibm-granite/granite-3b-code-instruct-128K",
    model="ibm-granite/granite-20b-code-instruct-8k",
    replicate_api_token=get_env_var('REPLICATE_API_TOKEN'),
    model_kwargs={"max_length":100, "temperature":0.2},
)
```

Menggunakan model IBM Granite untuk pembuatan kode dan tugas pemrograman

- Pilih **Connect to a hosted runtime (Hubungkan ke runtime yang dihosting)** pada menu **Connect (Hubungkan)**.

```

! pip install git+https://github.com/ibm-granite-community/utils \
"langchain_community<0.3.0" \
replicate

```

- Tanda centang hijau menunjukkan kamu telah berhasil terhubung ke runtime yang dihosting.

```

! pip install git+https://github.com/ibm-granite-community/utils \
"langchain_community<0.3.0" \
replicate

```

- Di sel pertama notebook, pilih tombol **Putar** untuk menginstal pustaka yang diperlukan dari komunitas Granite.

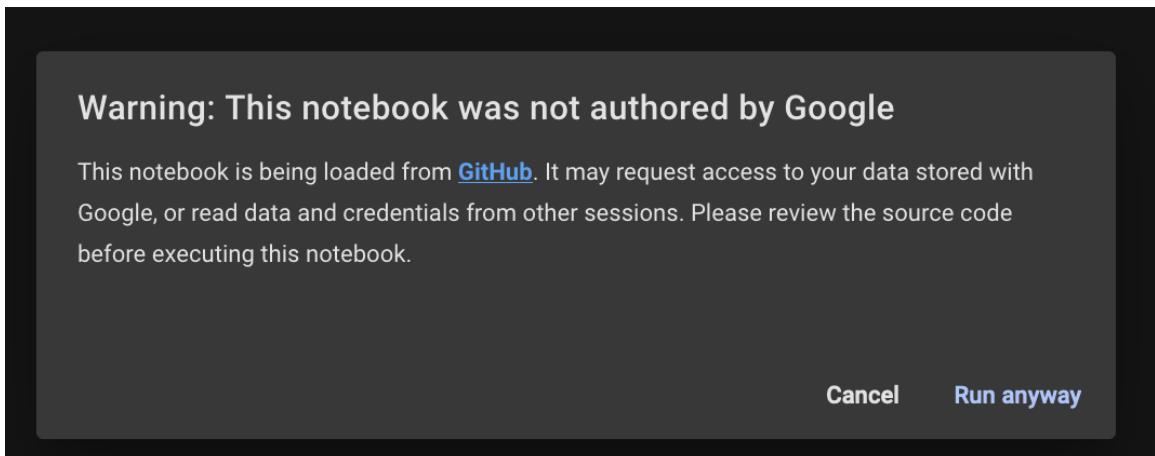
```

!pip install git+https://github.com/ibm-granite-community/utils \
"langchain_community<0.3.0" \
replicate

```

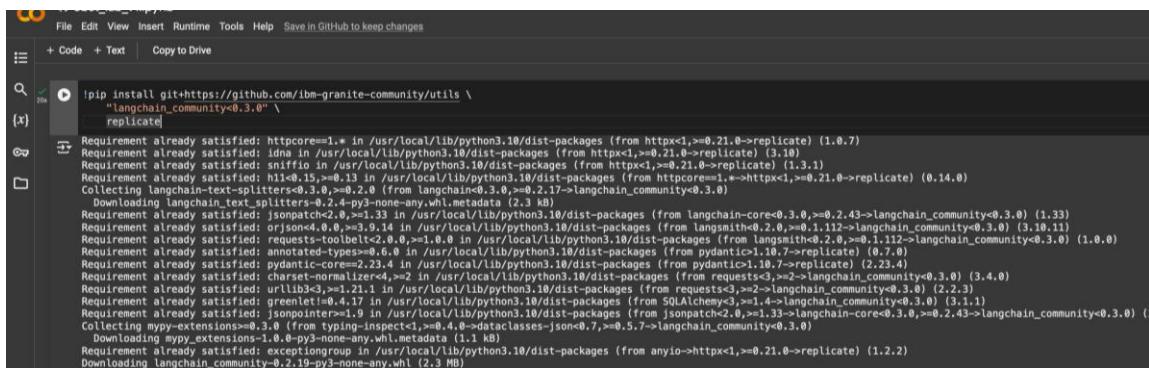
Catatan: Kamu harus menjalankan sel secara berurutan, mulai dari sel pertama dalam notebook.

- Pilih **Run anyway (Jalankan)** untuk melanjutkan pemuatan pustaka yang diperlukan.



Menggunakan model IBM Granite untuk pembuatan kode dan tugas pemrograman

12. Tanda centang hijau muncul di samping tombol **Putar** untuk menunjukkan bahwa pustaka yang diperlukan berhasil diinstal.

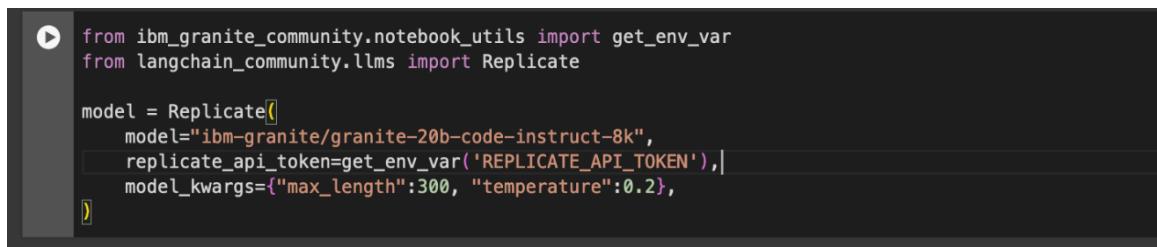


```
File Edit View Insert Runtime Tools Help Save in GitHub to keep changes
+ Code + Text | Copy to Drive

[x] pip install git+https://github.com/ibm-granite-community/utils \
"langchain_community@0.3.0" \
replicate

Requirement already satisfied: httpcore==1.* in /usr/local/lib/python3.10/dist-packages (from httpx<1,>=0.21.0->replicate) (1.0.7)
Requirement already satisfied: idna in /usr/local/lib/python3.10/dist-packages (from httpx<1,>=0.21.0->replicate) (3.10)
Requirement already satisfied: sniffio in /usr/local/lib/python3.10/dist-packages (from httpx<1,>=0.21.0->replicate) (1.3.1)
Requirement already satisfied: h11<0.15,>=0.13 in /usr/local/lib/python3.10/dist-packages (from httpcore==1.*->httpx<1,>=0.21.0->replicate) (0.14.0)
Collecting langchain-text-splitters<0.3.0,>=0.2.0 (from langchain@0.3.0,>=0.2.17->langchain_community@0.3.0)
  Downloading langchain_text_splitters-0.2.4-py3-none-any.whl.metadata (2.3 kB)
Requirement already satisfied: jsonpatch<0.33 in /usr/local/lib/python3.10/dist-packages (from langchain<0.3.0,>=0.2.4->langchain_community@0.3.0) (1.33)
Requirement already satisfied: requests<2.0.0,>=0.1.0 in /usr/local/lib/python3.10/dist-packages (from langchain@0.3.0,>=0.1.112->langchain_community@0.3.0) (3.10.11)
Requirement already satisfied: requests_toolbelt<2.0.0,>=0.8.0 in /usr/local/lib/python3.10/dist-packages (from langchain@0.3.0,>=0.1.112->langchain_community@0.3.0) (1.0.0)
Requirement already satisfied: annotated-types==0.6.0 in /usr/local/lib/python3.10/dist-packages (from pydantic<1.10.7->replicate) (0.7.0)
Requirement already satisfied: pydantic-core==2.23.4 in /usr/local/lib/python3.10/dist-packages (from pydantic<1.10.7->replicate) (2.23.4)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2->langchain_community@0.3.0) (3.4.0)
Requirement already satisfied: urllib3<2.1.1 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2->langchain_community@0.3.0) (2.2.3)
Requirement already satisfied: greenlet==0.4.17 in /usr/local/lib/python3.10/dist-packages (from SQLAlchemy<3,>=1.4->langchain_community@0.3.0) (3.1.1)
Requirement already satisfied: jsonpointer<1.9 in /usr/local/lib/python3.10/dist-packages (from jsonpatch<2.0,>=1.33->langchain<0.3.0,>=0.2.43->langchain_community@0.3.0) (1.0.0)
Collecting mypy-extensions<1.0.0-py3-none-any.whl.metadata (1.1 kB)
Requirement already satisfied: exceptiongroup in /usr/local/lib/python3.10/dist-packages (from anyio->httpx<1,>=0.21.0->replicate) (1.2.2)
  Downloading exceptiongroup-1.0.0-py3-none-any.whl (2.3 MB)
Requirement already satisfied: langchain_community==0.2.19-py3-none-any.whl (2.3 MB)
```

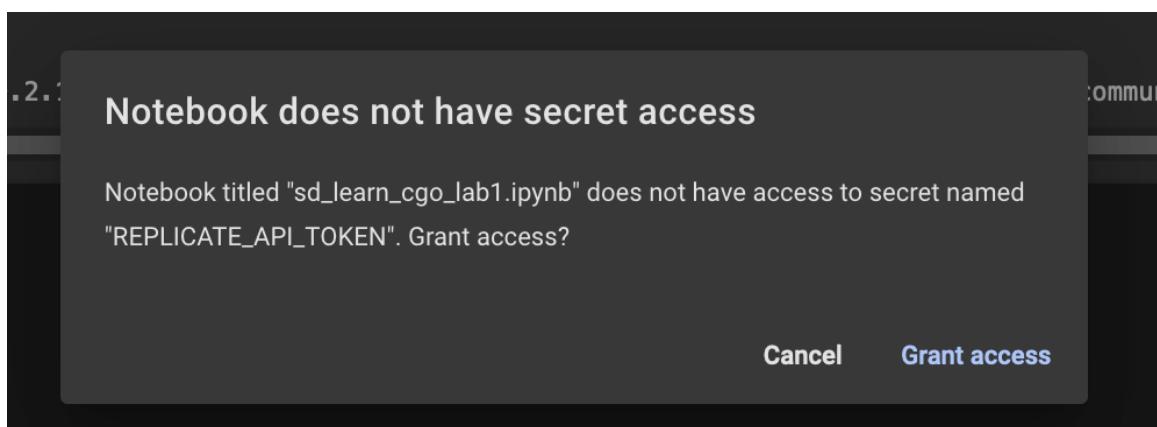
13. Pada sel kedua notebook, pilih tombol **Putar** untuk menginisialisasi model IBM Granite menggunakan Replicate untuk pembuatan kode.



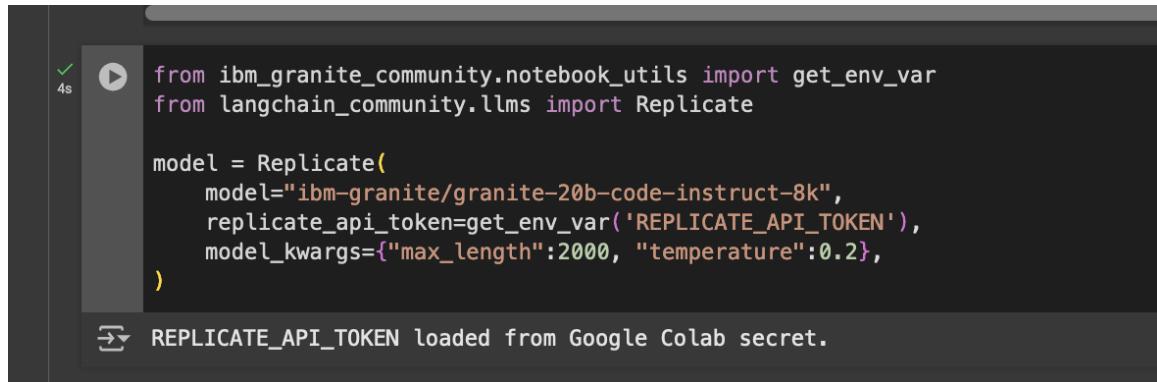
```
from ibm_granite_community.notebook_utils import get_env_var
from langchain_community.llms import Replicate

model = Replicate([
    model="ibm-granite-20b-code-instruct-8k",
    replicate_api_token=get_env_var('REPLICATE_API_TOKEN'),
    model_kwarg={"max_length":300, "temperature":0.2},
```

14. Pilih **Grant access (Berikan akses)** untuk melanjutkan pemuatian model dari Replicate.



15. Pesan berikut ditampilkan di bagian bawah sel : "REPLICATE_API_TOKEN dimuat dari rahasia Google Colab.

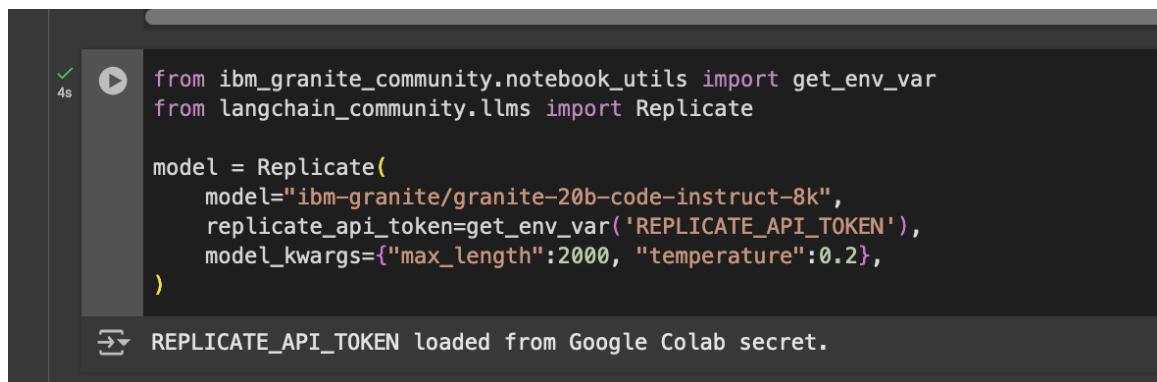


```
4s ✓  from ibm_granite_community.notebook_utils import get_env_var
      from langchain_community.llms import Replicate

      model = Replicate(
          model="ibm-granite/granite-20b-code-instruct-8k",
          replicate_api_token=get_env_var('REPLICATE_API_TOKEN'),
          model_kwargss={"max_length":2000, "temperature":0.2},
      )

→ REPLICATE_API_TOKEN loaded from Google Colab secret.
```

16. Tanda centang hijau di samping tombol **Putar** menunjukkan bahwa model IBM Granite diinisialisasi menggunakan Replicate. Kamu sekarang siap untuk meminta model untuk buat kode untuk skenario yang diberikan.



```
4s ✓  from ibm_granite_community.notebook_utils import get_env_var
      from langchain_community.llms import Replicate

      model = Replicate(
          model="ibm-granite/granite-20b-code-instruct-8k",
          replicate_api_token=get_env_var('REPLICATE_API_TOKEN'),
          model_kwargss={"max_length":2000, "temperature":0.2},
      )

→ REPLICATE_API_TOKEN loaded from Google Colab secret.
```

Langkah 5: Membuat kode menggunakan model IBM Granite

Ikhtisar

Pada langkah ini, kamu akan meminta model IBM Granite untuk menghasilkan kode untuk halaman arahan situs web Reader's Verse. Kamu akan menentukan prompt zero-shot untuk model dan menjalankan prompt untuk membuat komponen antarmuka pengguna (UI) untuk halaman arahan situs web klien.

Instruksi

1. Tentukan dan jalankan fungsi yang membuat prompt zero-shot untuk model tersebut. Prompt zero-shot menjelaskan tugas yang perlu dilakukan model tanpa memberikan contoh untuk memandu output model.

Pada sel ketiga notebook, pilih tombol **Putar** untuk menentukan fungsi **zeroshot_prompt**, yang berisi serangkaian instruksi yang memandu model pada output yang akan dihasilkan. Dalam skenario ini, instruksi menginformasikan model untuk menghasilkan kode untuk toko buku online berdasarkan parameter yang diberikan, yaitu konteks, pertanyaan, dan judul buku.

The screenshot shows a Jupyter Notebook interface with the title 'sd_learn_cgo_lab1.ipynb'. The code cell contains the following Python code:

```
replicate_api_token=get_env_var('REPLICATE_API_TOKEN'),
model_kwargs={"max_length":300, "temperature":0.2},  
)  
  
def zeroshot_prompt(context, question, book_titles):  
    """  
    Creates a zero-shot prompt for the model, where the model acts as a  
    seasoned programmer.  
  
    Parameters:  
    - context: str, contextual information for the prompt  
    - question: str, specific question or task for the model to perform  
    - book_titles: list, list of book titles to include in the prompt  
    Returns:  
    - str, the formatted prompt  
    """  
  
    titles = ", ".join(book_titles)  
    prompt = f"""  
You are an experienced programmer with 15 years of experience writing  
full-stack applications.  
Your task is to generate code for an application using the context provided  
in the task: {context}.  
Begin by creating components based on the user question: {question}.  
Include a description of elements provided in {titles}.  
Ensure that you generate only Python code for a Jupyter Notebook using  
ipywidgets UI components.  
Do not output anything but the code. The output code should not exceed 300  
tokens.  
"""  
    return prompt
```

Catatan: Kamu tidak akan melihat output apa pun ketika kamu jalankan sel ini di notebook.

2. Berikutnya, ciptakan fungsi untuk menghasilkan respons dari model dengan menggunakan prompt zero-shot. Kamu akan mengirim instruksi yang ditentukan dalam prompt zero-shot ke model dan bersiap untuk mendapatkan respons dari model.

Di sel keempat notebook, pilih tombol **Putar** untuk menentukan fungsi `get_answer_using_zeroshot`.

```
▶ def get_answer_using_zeroshot(context, question, book_titles):
    """
    Generates the response from the model based on a zero-shot prompt.

    Parameters:
    - context: str, contextual information for the prompt
    - question: str, specific question for the model to answer
    - book_titles: list, list of book titles to include in the prompt

    Returns:
    - str, the generated result from the model
    """
    prompt = zeroshot_prompt(context, question, book_titles)
    result = model.invoke(prompt)

    return result
```

Catatan: Kamu tidak akan melihat output apa pun ketika kamu jalankan sel ini di notebook.

3. Berikutnya, kamu memberikan nilai yang diperlukan untuk parameter yang ditentukan dalam prompt zero-shot, kemudian memanggil fungsi `get_answer_using_zeroshot` untuk menghasilkan output kode. Pada skenario ini, nilai yang diberikan adalah sebagai berikut:

- **Konteks:** Meminta model untuk mendesain UI dengan tema minimalis
- **Pertanyaan:** Mendefinisikan atribut halaman arahan situs web seperti header dan pesan selamat datang
- **Judul:** Menentukan daftar judul buku yang akan ditampilkan di halaman arahan

Di sel kelima notebook, pilih tombol **Putar** untuk menjalankan fungsi `get_answer_using_zeroshot`. Ini menghasilkan dan menampilkan kode Python untuk UI halaman arahan toko buku online mengikuti kode sel.

```
sd_learn_cgo_lab1.ipynb - Colab X + https://colab.research.google.com/github/ibm-cgo-lab1/notebooks/sd_learn_cgo_lab1.ipynb?scrollTo=95c5a7bd90xR8
```

```
sd_learn_cgo_lab1.ipynb
```

```
File Edit View Insert Runtime Tools Help Save in GitHub to keep changes
```

```
+ Code + Text Copy to Drive
```

```
context = "Design and develop an online bookstore UI components with minimalist theme."
question = "Create the landing page for users visiting my bookstore. The landing page should display a header 'Reader's Online Store', a welcome message 'Welcome to Reader's Verse' along with a catalog of books titles as title. Vertically align the sections."
book_titles = ["The Great Gatsby", "Pride and Prejudice", "The Hobbit", "The Lord of the Rings", "Animal Farm", "Brave New World"]

# Generate and display the UI code for the landing page
result = get_answer_using_zeroshot(context, question, book_titles)
print(f"Generated Code:\n{result}")
```

4. Output yang dihasilkan mencakup kode Python untuk membuat elemen UI seperti halaman arahan dan katalog buku, menggunakan kerangka kerja UI asli Jupyter Notebook . Kode ini akan buat UI halaman arahan toko buku, termasuk hak milik dan katalog judul buku.

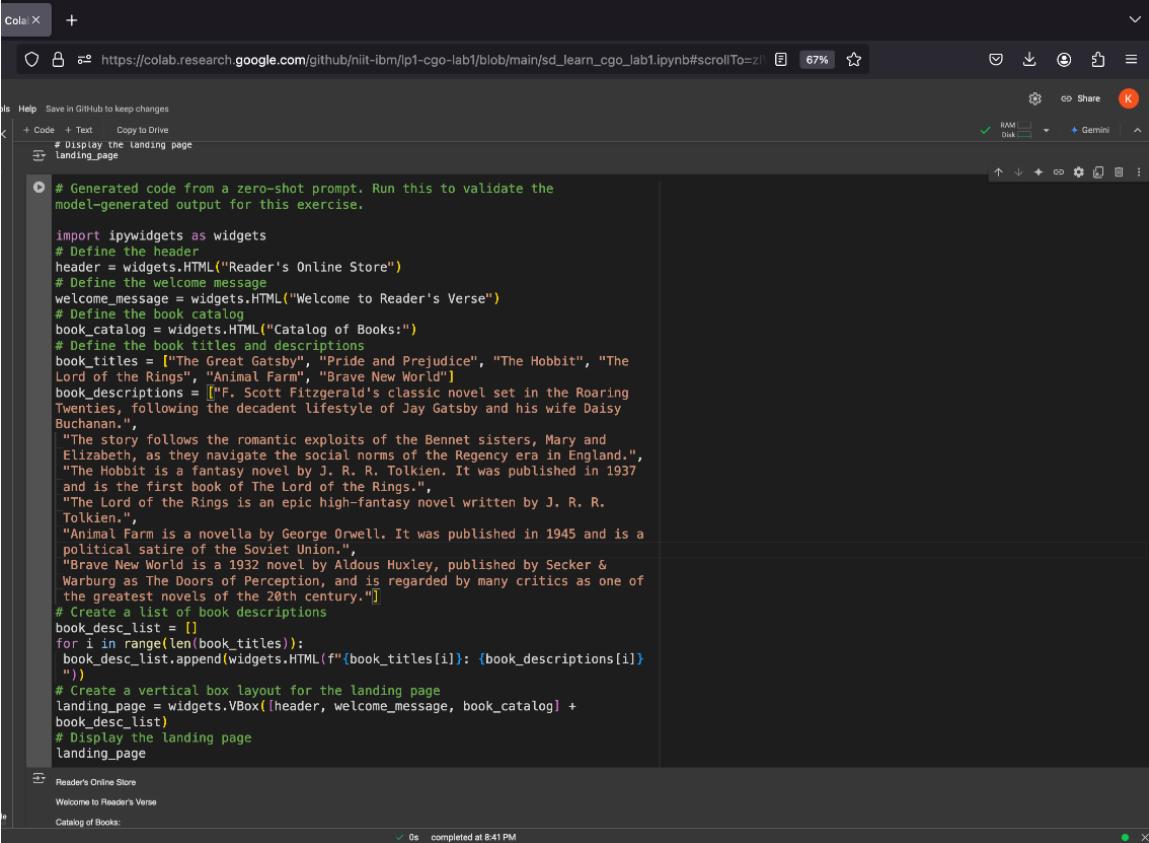
Catatan: Saat kamu membuat kode, mungkin sedikit berbeda dari output kode berikut. Perbedaan kecil ini adalah normal dan tidak akan memengaruhi fungsionalitas atau hasil lab secara keseluruhan.

```
context = "Design and develop an online bookstore UI components with minimalistic theme."
question = "Create the landing page for users visiting my bookstore. The landing page should display a header 'Reader's Online Store', a welcome message 'Welcome to Reader's Verse' along with a catalog of books titled 'Catalog of Books'. The catalog should list the book titles and descriptions. The descriptions should include a short summary of the book. The landing page should be generated using ipywidgets."# Generate and display the UI code for the landing page
result = get_answer_using_zeroshot(context, question, book_titles)
print(f"Generated Code:\n{result}")


Generated Code:
Here's the code for the landing page of the online bookstore using ipywidgets:
```python
import ipywidgets as widgets
Define the header
header = widgets.HTML("Reader's Online Store")
Define the welcome message
welcome_message = widgets.HTML("Welcome to Reader's Verse")
Define the book catalog
book_catalog = widgets.HTML("Catalog of Books")
Define the book titles and descriptions
book_titles = ["The Great Gatsby", "Pride and Prejudice", "The Hobbit", "The Lord of the Rings", "Animal Farm", "Brave New World"]
book_descriptions = [
 "The Great Gatsby is a classic American novel set in the Roaring Twenties, following the decadent lifestyle of Jay Gatsby and his wife Daisy Buchanan.", "The story follows the romantic exploits of the Bennet sisters, Mary and Elizabeth, as they navigate the social norms of the Regency era in England.", "The Hobbit is a fantasy novel by J. R. R. Tolkien. It was published in 1937 and is the first book of The Lord of the Rings.", "The Lord of the Rings is an epic high-fantasy novel written by J. R. R. Tolkien.", "Animal Farm is a political satire by George Orwell. It was published in 1945 and is a political satire of the Soviet Union.", "Brave New World is a 1932 novel by Aldous Huxley, published by Secker & Warburg as The Doors of Perception, and is regarded by many critics as one of the greatest novels of the 20th century."]
Create a list of book descriptions
book_desc_list = []
for i in range(len(book_titles)):
 book_desc_list.append(widgets.HTML(f"{book_titles[i]}; {book_descriptions[i]}"))
Create a vertical box layout for the landing page
landing_page = widgets.VBox([header, welcome_message, book_catalog] + book_desc_list)
Display the landing page
landing_page
```

Menggunakan model IBM Granite untuk pembuatan kode dan tugas pemrograman

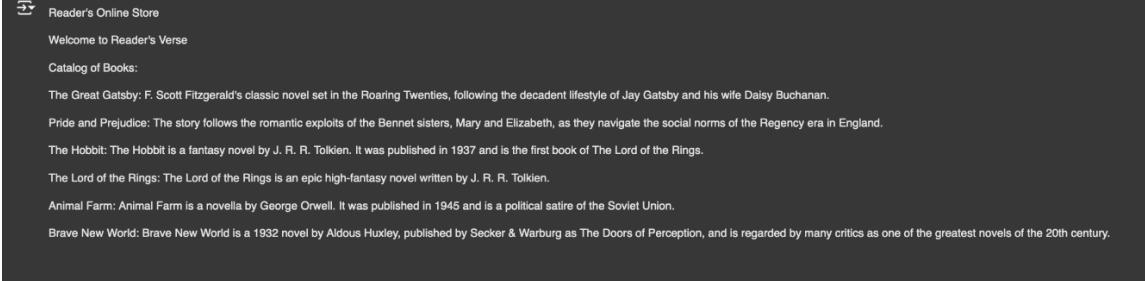
- Untuk memudahkan eksekusi, kode output sudah diisi sebelumnya dalam sel baru pada Jupyter notebook. Pilih tombol **Putar** pada sel keenam pada notebook untuk memvalidasi output kode Python yang dihasilkan oleh model IBM Granite untuk skenario yang diberikan.



```
Generated code from a zero-shot prompt. Run this to validate the
model-generated output for this exercise.

import ipywidgets as widgets
Define the header
header = widgets.HTML("Reader's Online Store")
Define the welcome message
welcome_message = widgets.HTML("Welcome to Reader's Verse")
Define the book catalog
book_catalog = widgets.HTML("Catalog of Books:")
Define the book titles and descriptions
book_titles = ['The Great Gatsby', 'Pride and Prejudice', 'The Hobbit', 'The
Lord of the Rings', 'Animal Farm', 'Brave New World']
book_descriptions = [F. Scott Fitzgerald's classic novel set in the Roaring
Twenties, following the decadent lifestyle of Jay Gatsby and his wife Daisy
Buchanan.,
"The story follows the romantic exploits of the Bennet sisters, Mary and
Elizabeth, as they navigate the social norms of the Regency era in England.",
"The Hobbit is a fantasy novel by J. R. R. Tolkien. It was published in 1937
and is the first book of The Lord of the Rings.",
"The Lord of the Rings is an epic high-fantasy novel written by J. R. R.
Tolkien.",
"Animal Farm is a novella by George Orwell. It was published in 1945 and is a
political satire of the Soviet Union.",
"Brave New World is a 1932 novel by Aldous Huxley, published by Secker &
Warburg as The Doors of Perception, and is regarded by many critics as one of
the greatest novels of the 20th century."]
Create a list of book descriptions
book_desc_list = []
for i in range(len(book_titles)):
 book_desc_list.append(widgets.HTML(f'{book_titles[i]}: {book_descriptions[i]}'))
Create a vertical box layout for the landing page
landing_page = widgets.VBox([header, welcome_message, book_catalog] +
 book_desc_list)
Display the landing page
landing_page
```

- Outputnya akan menampilkan UI halaman arahan toko buku yang mengikuti sel kode, yang menampilkan komponen-komponen berikut:
  - Sebuah judul: "Welcome to Reader's Verse (Selamat datang di Reader's Verse)"
  - Kisi katalog yang menampilkan judul, penulis, dan deskripsi buku



Menggunakan model IBM Granite untuk pembuatan kode dan tugas pemrograman

# IBM SkillsBuild

## Kesimpulan

Kamu telah menggunakan model IBM Granite untuk menghasilkan kode untuk membuat halaman arahan situs web untuk Reader's Verse, toko buku online. Klien senang dengan betapa cepatnya kamu mengembangkan prototipe UI yang berfungsi untuk halaman sambutan situs web. Kamu juga telah berhasil mendefinisikan prompt zero-shot untuk menghasilkan output kode Python dari model IBM Granite dan memvalidasi kode yang dihasilkan AI untuk memastikan bahwa kode tersebut benar-benar memberikan output yang diharapkan.

© Hak Cipta IBM Corporation 2025.

Informasi yang terkandung dalam materi ini disediakan hanya untuk tujuan informasi dan disediakan SEBAGAIMANA ADANYA tanpa jaminan dalam bentuk apa pun, tersurat maupun tersirat. IBM tidak bertanggung jawab atas segala kerusakan apa pun yang timbul dari penggunaan, atau terkait dengan, materi ini. Tidak ada yang terkandung dalam materi ini dimaksudkan untuk, atau akan memiliki efek untuk, menciptakan jaminan atau representasi apa pun dari IBM atau pemasok atau pemberi lisensinya, atau mengubah syarat dan ketentuan dari berlaku perjanjian lisensi yang mengatur penggunaan perangkat lunak IBM. Referensi dalam materi ini terhadap produk, program, atau layanan IBM tidak menyiratkan bahwa produk, program, atau layanan tersebut akan tersedia di semua negara tempat IBM beroperasi. Informasi ini didasarkan pada rencana dan strategi produk IBM saat ini, yang dapat diubah oleh IBM tanpa pemberitahuan. Tanggal rilis produk dan/atau kemampuan yang dirujuk dalam materi ini dapat berubah sewaktu-waktu atas direksi IBM sendiri berdasarkan peluang pasar atau faktor lainnya, dan tidak dimaksudkan untuk menjadi komitmen terhadap ketersediaan produk atau fitur di masa mendatang dengan cara apa pun.

IBM, logo IBM dan ibm.com adalah merek dagang International Business Machines Corp., yang terdaftar di banyak yurisdiksi di seluruh dunia. Nama produk dan layanan lainnya mungkin merupakan merek dagang IBM atau perusahaan lain. Daftar merek dagang IBM saat ini tersedia di

Menggunakan model IBM Granite untuk pembuatan kode dan tugas pemrograman

# IBM SkillsBuild

Web di " informasi Hak hukum dan Merek Dagang"  
di [www.ibm.com/hukum /copytrade.shtml](http://www.ibm.com/hukum/copytrade.shtml).



Please Recycle