The provided code implements a **Retrieval-Augmented Generation (RAG)** system using a combination of retrieval and language model techniques to summarize PDF documents in Arabic. Here’s a breakdown of the approach and type of RAG used:

**Approach:**

1. **Retrieval and Embedding Generation**:
   * The documents are uploaded (in PDF format) and split into chunks using RecursiveCharacterTextSplitter. These chunks are then embedded using OllamaEmbeddings (paraphrase-multilingual:latest), creating document embeddings.
   * The embeddings are stored in the **FAISS vector store** for efficient similarity-based retrieval.
2. **Retrieval-Augmented Generation**:
   * **Dense Retrieval**: The code uses **FAISS**, which performs dense retrieval by searching for the most relevant documents (or chunks) based on vector similarity.
   * A **retriever** object is created from the vector store to retrieve the most relevant documents (based on the user's query or context).
   * These retrieved documents are then passed to a generative language model (Ollama), which processes the retrieved documents and generates a summarized response in Arabic.
3. **Generative Model**:
   * The language model Ollama is used to generate the summary. This is a **generative model** that outputs a natural language response based on the retrieved documents and the customized summarization prompt provided in Arabic.
4. **Document Chain**:
   * The document\_chain (created via create\_stuff\_documents\_chain) is used to combine the documents retrieved from the vector database and process them for the final generation of the summarized content.

**Type of RAG:**

* **RAG-Sequence**: This code follows the **RAG-Sequence** approach where documents are retrieved first (via the FAISS retriever) and then passed to the language model (Ollama) for summarization. The generation happens **after** retrieval, where the retrieved documents are processed by the LLM in sequence.

**Key Components:**

* **Dense Retrieval**: The use of FAISS for vector-based document retrieval suggests that the system is using **Dense Retrieval** where embeddings are leveraged to find semantically relevant chunks of documents.
* **Retriever-Generator Pipeline**: The retrieved documents are passed to the language model for generating the final response (summary). This is a classic **retriever-generator** pipeline where retrieval enhances the generative capabilities of the model.

**Techniques Used:**

* **Text Splitting**: The RecursiveCharacterTextSplitter ensures that documents are split into manageable chunks (with overlap for context preservation), which enhances retrieval precision.
* **Embeddings-Based Retrieval**: The use of OllamaEmbeddings and FAISS for generating embeddings and retrieving documents based on vector similarity.
* **Prompt-Based Generation**: A custom prompt in Arabic is used for the LLM to generate a well-structured and concise summary in the target language.

**Summary:**

* **RAG Type**: **RAG-Sequence**.
* **Retrieval Method**: **Dense Retrieval** (using FAISS and embeddings).
* **Technique**: **Retriever-Generator Pipeline**, with chunking, embedding-based retrieval, and prompt-based generation.