**Mengatur pembelajaran mesin dengan alur**

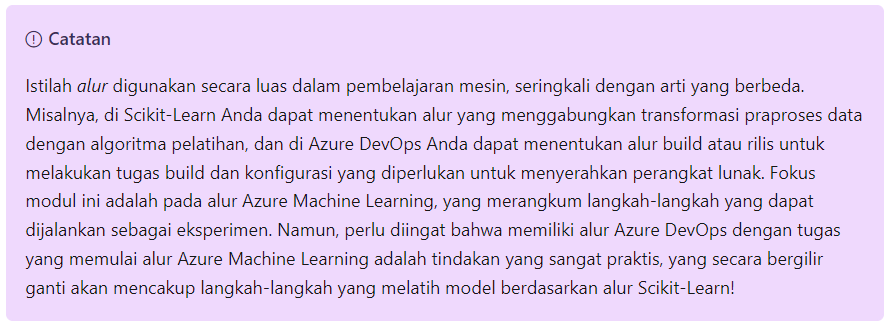
Mengatur pelatihan pembelajaran mesin dengan alur adalah elemen penting DevOps untuk pembelajaran mesin. Dalam modul ini, Anda akan mempelajari cara membuat, menerbitkan, dan menjalankan alur untuk melatih model di Azure Machine Learning.

**Tujuan pembelajaran**

* Membuat langkah-langkah alur
* Meneruskan data antar langkah
* Menerbitkan dan menjalankan alur
* Menjadwalkan alur

1. **Pengantar**

Dalam Azure Machine Learning, Anda menjalankan beban kerja sebagai eksperimen yang memanfaatkan aset data dan sumber daya komputasi. Dalam proses ilmu data perusahaan, Anda biasanya ingin memisahkan proses keseluruhan menjadi tugas-tugas individual, dan mengatur tugas-tugas ini sebagai alur langkah-langkah yang terhubung. Alur adalah kunci untuk menerapkan solusi Operasionalisasi Pembelajaran Mesin (ML Ops) yang efektif di Azure, sehingga Anda akan menjelajahi cara menentukan dan menjalankannya dalam modul ini



.

1. **Tujuan pembelajaran**

Dalam modul ini, Anda akan mempelajari cara:

* Membuat alur Azure Machine Learning.
* Menerbitkan alur Azure Machine Learning.
* Menjadwalkan alur Azure Machine Learning.

1. **Pengenalan alur**

Di Azure Machine Learning, alur adalah alur kerja tugas pembelajaran mesin ketika setiap tugas diimplementasikan sebagai langkah.

Langkah-langkah dapat diatur secara berurutan atau paralel, sehingga memungkinkan Anda untuk membangun logika alur yang mutahir untuk mengatur operasi pembelajaran mesin. Setiap langkah dapat dijalankan pada target komputasi tertentu, sehingga memungkinkan untuk menggabungkan berbagai jenis pemrosesan yang diperlukan untuk mencapai tujuan keseluruhan.

Alur dapat dieksekusi sebagai proses dengan menjalankan alur sebagai percobaan. Setiap langkah dalam alur yang dijalankan pada target komputasi yang dialokasikan sebagai bagian dari keseluruhan eksperimen yang dijalankan.

Anda dapat menerbitkan alur sebagai titik akhir REST, memungkinkan aplikasi klien untuk memulai eksekusi alur. Anda juga dapat menentukan jadwal untuk alur, dan menjalankannya secara otomatis pada interval berkala.

1. **Langkah-langkah alur**

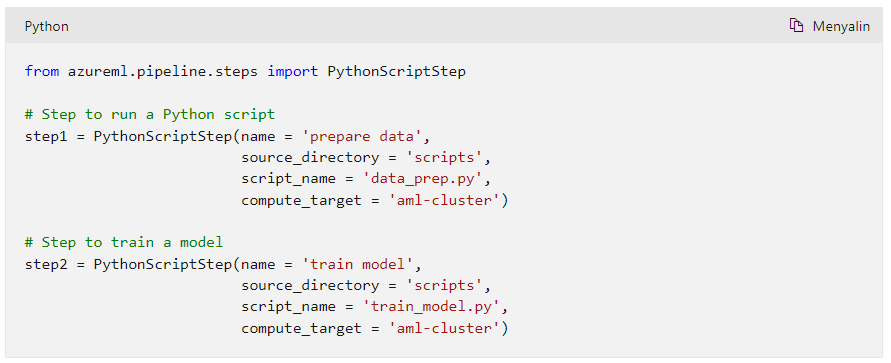
Alur Azure Machine Learning terdiri dari satu atau beberapa langkah yang melakukan tugas. Ada banyak jenis langkah yang didukung oleh alur Azure Machine Learning, masing-masing dengan tujuan khusus dan opsi konfigurasinya sendiri.

Jenis langkah umum dalam alur Azure Machine Learning meliputi:

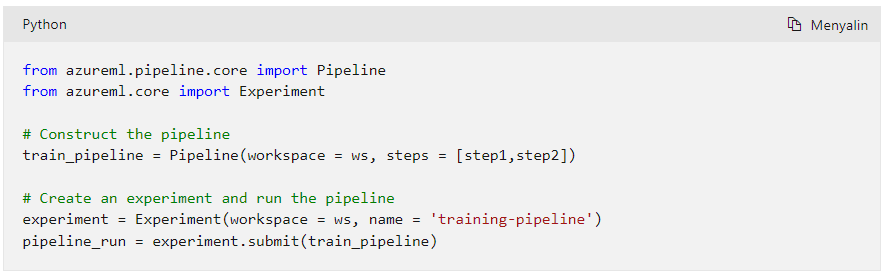
* **PythonScriptStep**: Menjalankan skrip Python tertentu.
* **DataTransferStep**: Menggunakan Azure Data Factory untuk menyalin data antar penyimpanan data.
* **DatabricksStep**: Menjalankan buku catatan, skrip, atau JAR pada kluster databricks.
* **AdlaStep**: Menjalankan pekerjaan U-SQL di Azure Data Lake Analytics.
* **ParallelRunStep** - Menjalankan skrip Python sebagai tugas terdistribusi pada beberapa node komputasi.

**Catatan**: Untuk daftar lengkap terkait jenis langkah yang didukung, lihat dokumentasi paket azure.pipeline.steps. Link : <https://learn.microsoft.com/id-id/python/api/azureml-pipeline-steps/azureml.pipeline.steps?view=azure-ml-py>

Untuk membuat alur, Anda harus terlebih dahulu menentukan setiap langkah lalu membuat alur yang menyertakan langkah-langkahnya. Konfigurasi spesifik dari setiap langkah tergantung pada jenis langkah. Misalnya kode berikut mendefinisikan dua langkah PythonScriptStep untuk menyiapkan data, lalu melatih model.



Setelah menentukan langkah-langkahnya, Anda dapat menetapkannya ke alur, dan menjalankannya sebagai eksperimen:



1. **Meneruskan data antar langkah alur**

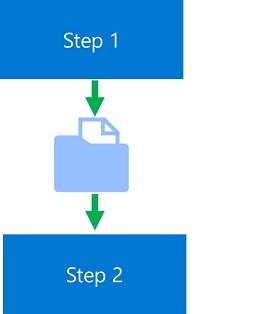
Seringkali, alur mencakup setidaknya satu langkah yang tergantung pada output dari langkah sebelumnya. Misalnya, Anda dapat menggunakan langkah yang menjalankan skrip python untuk melakukan praproses beberapa data, yang kemudian harus digunakan dalam langkah berikutnya untuk melatih model.

1. **Objek OutputFileDatasetConfig**

Objek OutputFileDatasetConfig adalah himpunan data khusus yang:

* Mereferensikan lokasi di penyimpanan data untuk penyimpanan data interim.
* Membuat dependensi data di antara langkah-langkah alur.

Anda dapat melihat objek **OutputFileDatasetConfig** sebagai penyimpanan perantara untuk data yang harus diteruskan dari tiap langkah ke langkah berikutnya.



1. **Input dan Output Langkah OutputFileDatasetConfig**

Untuk menggunakan objek **OutputFileDatasetConfig** untuk meneruskan data antar langkah, Anda harus:

1. Menentukan objek bernama **OutputFileDatasetConfig** yang mereferensikan lokasi di penyimpanan data. Jika tidak ada penyimpanan data eksplisit yang ditentukan, penyimpanan data default akan digunakan.
2. Meneruskan objek **OutputFileDatasetConfig** sebagai argumen skrip dalam langkah-langkah yang menjalankan skrip.
3. Sertakan kode dalam skrip tersebut untuk menulis ke argumen **OutputFileDatasetConfig** sebagai output atau membacanya sebagai input.

Misalnya, kode berikut mendefinisikan objek **OutputFileDatasetConfig** untuk data praproses yang harus dilewatkan di antara langkah-langkah.



Dalam skrip itu sendiri, Anda dapat memperoleh referensi ke objek **OutputFileDatasetConfig** dari argumen skrip, dan menggunakannya seperti folder lokal.



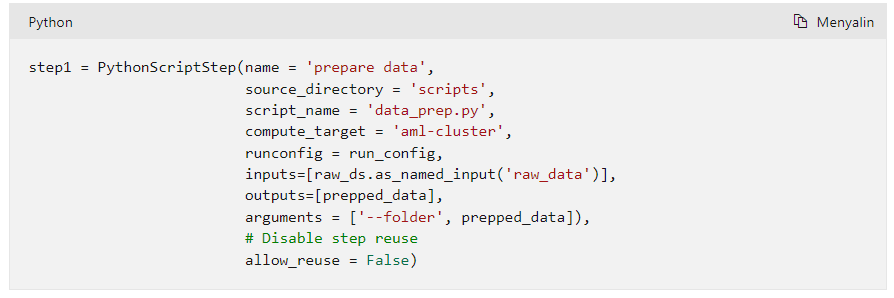
**4. Mempergunakan langkah-langkah alur kembali**

Alur dengan beberapa langkah jangka panjang dapat memakan waktu yang banyak untuk diselesaikan. Azure Machine Learning menyertakan beberapa fitur penembolokan dan penggunaan fitur kembali untuk mengurangi waktu ini.

1. **Mengelola penggunaan output langkah kembali**

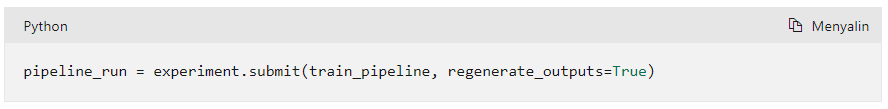
Secara default, output langkah dari alur yang dijalankan sebelumnya digunakan kembali tanpa menjalankan kembali langkah yang disediakan skrip, direktori sumber, dan parameter lain sebab langkah tersebut tidak berubah. Langkah penggunaan kembali dapat mengurangi waktu yang diperlukan untuk menjalankan alur, tetapi dapat menyebabkan hasil yang kedaluwarsa ketika perubahan pada sumber data hilir belum diperhitungkan.

Untuk mengontrol penggunaan kembali untuk langkah individu, Anda dapat mengatur parameter allow\_reuse dalam konfigurasi langkah, seperti ini:



1. **Memaksa semua langkah untuk menjalankan**

Ketika Anda memiliki beberapa langkah, Anda dapat memaksa semuanya untuk berjalan terlepas dari konfigurasi penggunaan kembali individu dengan mengatur parameter regenerate\_outputs saat mengirimkan eksperimen alur:



**5. Menerbitkan alur**

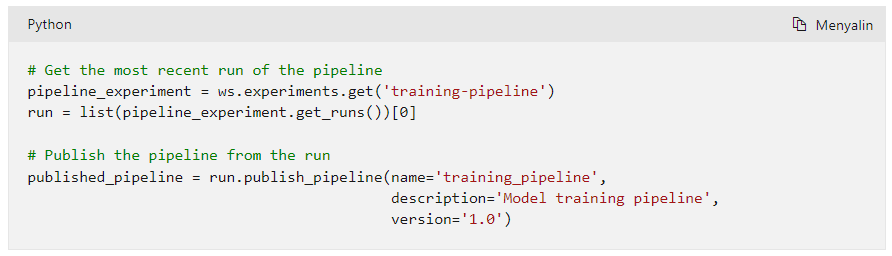
Setelah Anda membuat alur, Anda dapat menerbitkannya untuk membuat titik akhir REST tempat alur dapat dijalankan sesuai permintaan.

1. **Menerbitkan alur**

Untuk menerbitkan alur, Anda dapat memanggil metode **penerbitannya**, seperti yang ditunjukkan di sini:



Atau, Anda dapat memanggil metode **penerbitan** pada eksekusi alur yang berhasil:



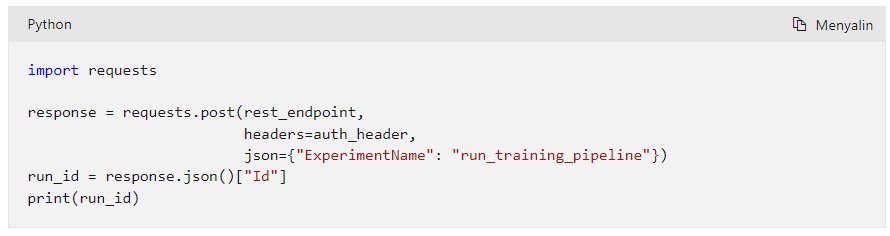
Setelah alur diterbitkan, Anda dapat melihatnya di Studio Pembelajaran Mesin Azure. Anda juga dapat menentukan URI titik akhir seperti ini:



1. **Menggunakan alur yang diterbitkan**

Untuk memulai titik akhir yang diterbitkan, Anda membuat permintaan HTTP ke titik akhir REST-nya, meneruskan header otorisasi dengan token untuk prinsipal layanan dengan izin untuk menjalankan alur, dan muatan JSON yang menentukan nama eksperimen. Alur dijalankan secara asinkron, sehingga respons dari panggilan REST yang berhasil menyertakan ID eksekusi. Anda dapat menggunakan ini untuk melacak eksekusi di Studio Pembelajaran Mesin Azure.

Misalnya, kode Python berikut membuat permintaan REST untuk menjalankan alur dan menampilkan ID eksekusi yang dikembalikan.



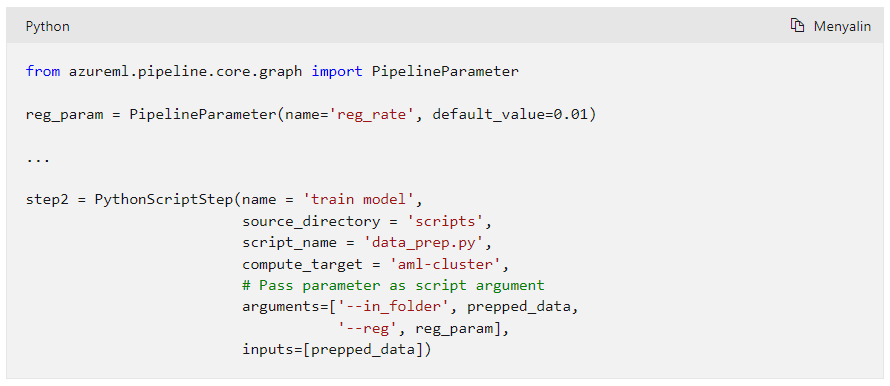
1. **Gunakan parameter pipeline**

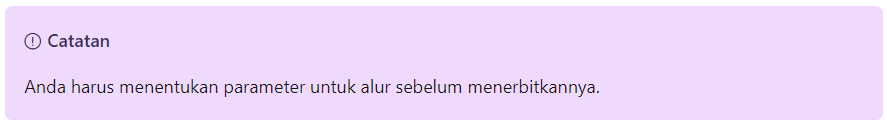
Anda dapat meningkatkan fleksibilitas alur dengan menentukan parameter.

1. **Menentukan parameter untuk alur**

Untuk menentukan parameter untuk alur, buat objek **PipelineParameter** untuk setiap parameter, dan tentukan setiap parameter dalam setidaknya satu langkah.

Misalnya, Anda dapat menggunakan kode berikut untuk menyertakan parameter untuk tingkat yang teratur dalam skrip yang digunakan oleh penghitung:





1. **Menjalankan alur dengan parameter**

Setelah Anda menerbitkan alur berparameter, Anda dapat meneruskan nilai parameter dalam muatan JSON untuk antarmuka REST:



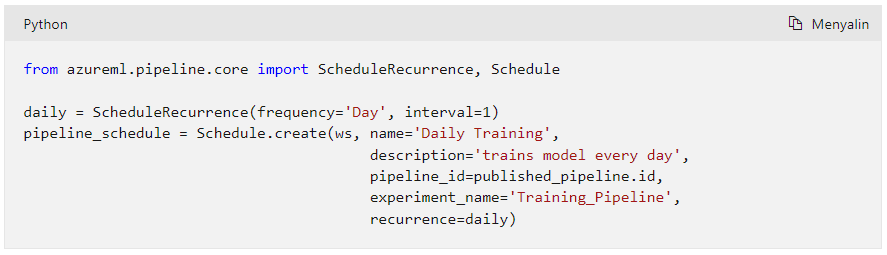
1. **Menjadwalkan alur**

Setelah Anda menerbitkan alur, Anda dapat memulainya sesuai permintaan melalui titik akhir REST-nya, atau Anda dapat menjalankan alur secara otomatis berdasarkan jadwal berkala atau merespons pembaruan data.

1. **Menjadwalkan alur untuk interval berkala**

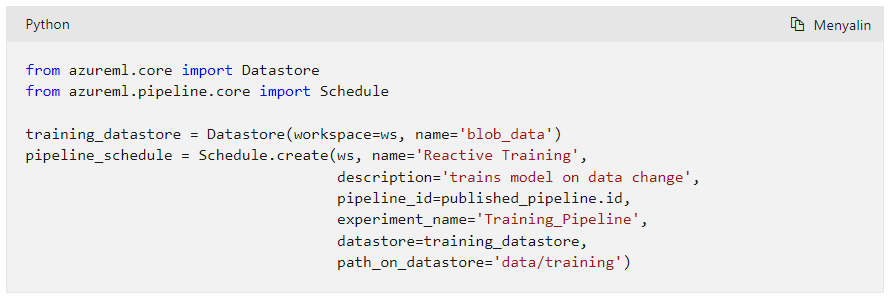
Untuk menjadwalkan alur yang akan dijalankan pada interval berkala, Anda harus menentukan **ScheduleRecurrence** yang menentukan frekuensi eksekusi, dan menggunakannya untuk membuat Jadwal.

Misalnya, kode berikut menjadwalkan eksekusi alur yang diterbitkan setiap hari.



1. **Memicu eksekusi alur pada perubahan data**

Untuk menjadwalkan alur untuk berjalan setiap kali data berubah, Anda harus membuat **Jadwal** yang memantau jalur tertentu di penyimpanan data, seperti ini:



1. **Latihan - Membuat alur**

Sekarang adalah kesempatan Anda untuk membuat dan menjalankan alur Azure Machine Learning.

Dalam latihan ini, Anda akan:

* Membuat alur Azure Machine Learning.
* Menerbitkan alur sebagai layanan REST.
* Menjadwalkan alur.

1. **Instruksi**

Ikuti petunjuk ini untuk menyelesaikan latihan.

1. Jika Anda belum memiliki langganan Azure, daftar untuk uji coba gratis di https://azure.microsoft.com.
2. Menampilkan repo latihan di <https://aka.ms/mslearn-dp100.> \ <https://microsoftlearning.github.io/mslearn-dp100/>
3. Jika Anda belum melakukannya, selesaikan latihan **Membuat ruang kerja Azure Machine Learning** untuk memprovisikan ruang kerja Azure Machine Learning, membuat instans komputasi, dan mengkloning file yang diperlukan.
4. Selesaikan latihan **Membuat alur.**
5. **Ringkasan**

Dalam modul ini, Anda belajar cara:

* Membuat alur Azure Machine Learning.
* Menerbitkan alur Azure Machine Learning.
* Menjadwalkan alur Azure Machine Learning.