**Bekerja dengan Data dalam Pembelajaran Mesin Azure**

Data adalah dasar dari pembelajaran mesin. Dalam modul ini, Anda akan mempelajari cara bekerja dengan penyimpanan data dan kumpulan data di Azure Machine Learning, yang memungkinkan Anda membangun solusi pelatihan model berbasis cloud yang skalabel.

1. **Tujuan pembelajaran**

* Buat dan gunakan penyimpanan data di ruang kerja Azure Machine Learning.
* Buat dan gunakan set data di ruang kerja Azure Machine Learning.

1. **Pengantar**

Data adalah elemen mendasar dalam beban kerja pembelajaran mesin apa pun, jadi dalam modul ini, Anda akan mempelajari cara membuat dan mengelola *penyimpanan data dan kumpulan data* di ruang kerja Pembelajaran Mesin Azure, dan cara menggunakan mereka dalam eksperimen pelatihan model.

1. **Tujuan pembelajaran**

Dalam modul ini, Anda akan mempelajari cara:

* Buat dan gunakan penyimpanan data
* Buat dan gunakan penyimpanan data

1. **Pengantar penyimpanan data**

Di Azure Machine Learning, penyimpanan data adalah abstraksi untuk sumber data awan. Mereka merangkum informasi yang diperlukan untuk terhubung ke sumber data. Anda dapat mengakses penyimpanan data secara langsung dalam kode dengan menggunakan Azure Machine Learning SDK, dan menggunakannya untuk mengunggah atau mengunduh data.

1. **Jenis Datastore**

Pembelajaran Mesin Azure mendukung pembuatan penyimpanan data untuk berbagai jenis sumber data Azure, termasuk:

* Azure Storage (blob dan file kontainer)
* Toko Azure Data Lake
* Azure SQL Database
* Sistem file Azure Databricks (DBFS)

**Catatan**: Untuk daftar lengkap penyimpanan data yang didukung, lihat dokumentasi Pembelajaran Mesin Azure : <https://learn.microsoft.com/id-id/azure/machine-learning/concept-data?tabs=uri-file-example%2Ccli-data-create-example#access-data-in-storage>

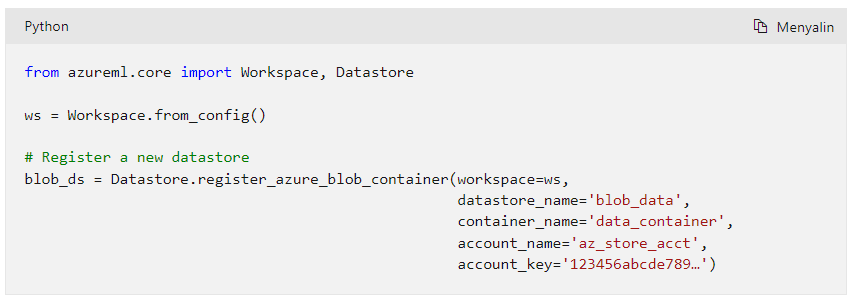
1. **Datastore bawaan**

Setiap ruang kerja memiliki dua penyimpanan data bawaan (wadah gumpalan Azure Storage, dan wadah file Penyimpanan Azure) yang digunakan sebagai penyimpanan sistem oleh Pembelajaran Mesin Azure. Ada juga penyimpanan data ketiga yang ditambahkan ke ruang kerja Anda jika Anda menggunakan kumpulan data terbuka yang disediakan sebagai sampel (misalnya, dengan membuat saluran desainer berdasarkan kumpulan data sampel)

Di sebagian besar proyek pembelajaran mesin, Anda mungkin perlu bekerja dengan sumber data Anda sendiri - baik karena Anda perlu menyimpan volume data yang lebih besar daripada dukungan penyimpanan data bawaan, atau karena Anda perlu mengintegrasikan solusi pembelajaran mesin Anda dengan data dari aplikasi yang ada.

1. **Gunakan penyimpanan data**

Untuk menambahkan penyimpanan data ke ruang kerja Anda, Anda dapat mendaftarkannya menggunakan antarmuka grafis di studio Pembelajaran Mesin Azure, atau Anda dapat menggunakan SDK Pembelajaran Mesin Azure. Misalnya, kode berikut mendaftarkan wadah gumpalan Azure Storage sebagai penyimpanan data bernama **blob\_data.**



1. **Mengelola penyimpanan data**

Anda dapat melihat dan mengelola penyimpanan data di Azure Machine Learning Studio, atau Anda dapat menggunakan Azure Machine Learning SDK. Misalnya, kode berikut mencantumkan nama setiap penyimpanan data di ruang kerja.



Anda bisa mendapatkan referensi ke penyimpanan data apa pun dengan menggunakan metode **Datastore.get()** seperti yang ditunjukkan di sini:



Ruang kerja selalu menyertakan penyimpanan data default (awalnya, ini adalah penyimpanan data **workspaceblobstore** bawaan), yang dapat Anda ambil dengan menggunakan metode **get\_default\_datastore()** dari objek **Ruang Kerja**, seperti ini:



1. **Pertimbangan untuk penyimpanan data**

Saat merencanakan untuk penyimpanan data, pertimbangkan panduan berikut:

* Saat menggunakan penyimpanan gumpalan Azure, penyimpanan tingkat premium dapat memberikan peningkatan kinerja I/O untuk kumpulan data besar. Namun, opsi ini akan meningkatkan biaya dan dapat membatasi opsi replikasi untuk redundansi data.
* Saat bekerja dengan file data, meskipun format CSV sangat umum, format Parket umumnya menghasilkan kinerja yang lebih baik.
* Anda dapat mengakses penyimpanan data apa pun berdasarkan nama, tetapi Anda mungkin ingin mempertimbangkan untuk mengubah penyimpanan data default (yang awalnya merupakan penyimpanan data **workspaceblobstore** bawaan).

Untuk mengubah penyimpanan data default, gunakan metode **set\_default\_datastore()**:



1. **Pengantar kumpulan data**

Dataset adalah objek data terpaket berversi yang dapat digunakan dengan mudah dalam eksperimen dan pipeline. Kumpulan data adalah cara yang disarankan untuk bekerja dengan data, dan merupakan mekanisme utama untuk kemampuan Pembelajaran Mesin Azure tingkat lanjut seperti pelabelan data dan pemantauan penyimpangan data.

1. **Jenis kumpulan data**

Kumpulan data biasanya didasarkan pada file di penyimpanan data, meskipun mereka juga dapat didasarkan pada URL dan sumber lainnya. Anda dapat membuat jenis kumpulan data berikut:

* Tabular: Data dibaca dari kumpulan data sebagai tabel. Anda harus menggunakan jenis kumpulan data ini bila data Anda terstruktur secara konsisten dan Anda ingin bekerja dengannya dalam struktur data tabular umum, seperti kerangka data Pandas.
* File: Kumpulan data menyajikan daftar jalur file yang dapat dibaca seolah-olah dari sistem file. Gunakan jenis kumpulan data ini saat data Anda tidak terstruktur, atau saat Anda perlu memproses data pada tingkat file (misalnya, untuk melatih jaringan saraf convolutional dari kumpulan file gambar).

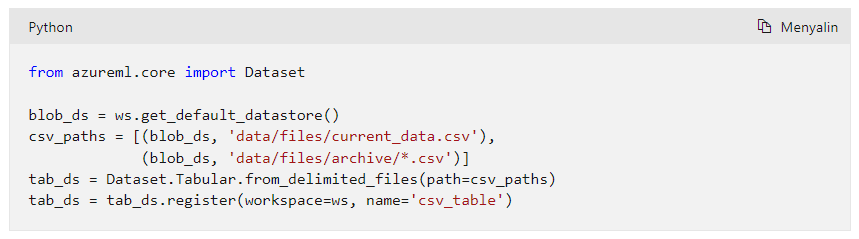
1. **Membuat dan mendaftarkan kumpulan data**

Anda dapat menggunakan antarmuka visual di studio Azure Machine Learning atau Azure Machine Learning SDK untuk membuat kumpulan data dari file individual atau beberapa jalur file. Jalur dapat menyertakan karakter pengganti (misalnya, /files/\*.csv) sehingga memungkinkan untuk merangkum data dari sejumlah besar file dalam satu kumpulan data.

Setelah Anda membuat kumpulan data, Anda dapat mendaftarkannya di ruang kerja agar tersedia untuk digunakan dalam eksperimen dan alur pemrosesan data nanti.

1. **Membuat dan mendaftarkan kumpulan data tabular**

Untuk membuat kumpulan data tabular menggunakan SDK, gunakan metode **from\_delimited\_files** dari kelas **Dataset.Tabular,** seperti ini:



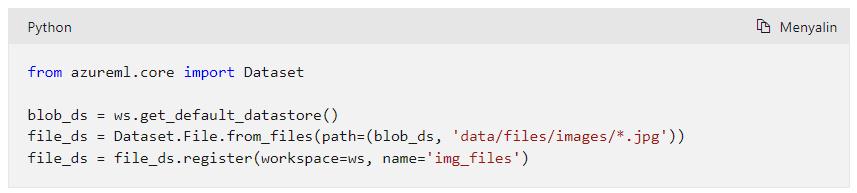
Kumpulan data dalam contoh ini mencakup data dari dua jalur file dalam penyimpanan data default:

* File **current\_data.csv** dalam folder **data/files.**
* Semua file .csv di folder **data/files/archive/.**

Setelah membuat kumpulan data, kode mendaftarkannya di ruang kerja dengan nama **csv\_table**.

1. **Membuat dan mendaftarkan kumpulan data file**

Untuk membuat kumpulan data file menggunakan SDK, gunakan metode **from\_files** dari kelas **Dataset.File**, seperti ini:



Kumpulan data dalam contoh ini mencakup semua file .jpg di jalur **data/file/gambar** dalam penyimpanan data default:

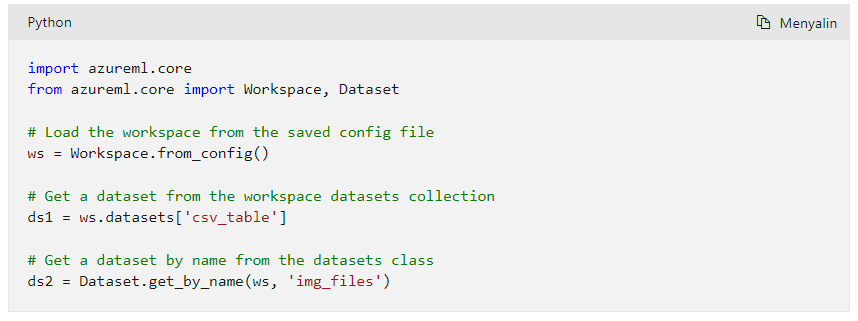
Setelah membuat kumpulan data, kode mendaftarkannya di ruang kerja dengan nama **img\_files.**

1. **Mengambil kumpulan data yang terdaftar**

Setelah mendaftarkan kumpulan data, Anda dapat mengambilnya dengan menggunakan salah satu teknik berikut:

* Atribut kamus **dataset** dari objek **Workspace**.
* Metode **get\_by\_name** atau **get\_by\_id** dari kelas **Dataset**.

Kedua teknik ini ditunjukkan dalam kode berikut:



1. **Versi kumpulan data**

Kumpulan data dapat berversi, memungkinkan Anda melacak versi historis kumpulan data yang digunakan dalam eksperimen, dan mereproduksi eksperimen tersebut dengan data dalam status yang sama.

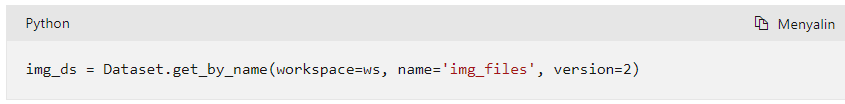


Anda dapat membuat versi baru dari kumpulan data dengan mendaftarkannya dengan nama yang sama seperti kumpulan data yang terdaftar sebelumnya dan menentukan properti **create\_new\_version:**

Dalam contoh ini, file .png di folder **gambar** telah ditambahkan ke definisi contoh kumpulan data **img\_paths** yang digunakan dalam topik sebelumnya.

1. **Mengambil versi kumpulan data tertentu**

Anda dapat mengambil versi tertentu dari kumpulan data dengan menetapkan parameter version dalam metode **get\_by\_name** kelas **Dataset**.

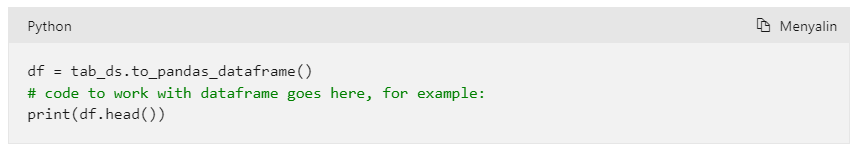


1. **Menggunakan himpunan data**

Kumpulan data adalah cara utama untuk meneruskan data ke eksperimen yang melatih model.

1. **Bekerja dengan kumpulan data tabular**

Anda dapat membaca data langsung dari kumpulan data tabular dengan mengubahnya menjadi kerangka data Pandas atau Spark:



1. **Berikan kumpulan data tabular ke skrip eksperimen**

Saat Anda perlu mengakses kumpulan data dalam skrip eksperimen, Anda harus meneruskan kumpulan data ke skrip. Ada dua cara yang bisa Anda lakukan.

* **Gunakan argumen skrip untuk kumpulan data tabular**

Anda dapat meneruskan kumpulan data tabular sebagai argumen skrip. Saat Anda mengambil pendekatan ini, argumen yang diterima oleh skrip adalah ID unik untuk kumpulan data di ruang kerja Anda. Dalam skrip, Anda kemudian bisa mendapatkan ruang kerja dari konteks run dan menggunakannya untuk mengambil dataset dengan ID-nya.

**ScriptRunConfig:**



**Skrip:**



* **Gunakan input bernama untuk dataset tabular**

Atau, Anda dapat meneruskan kumpulan data tabular sebagai masukan bernama. Dalam pendekatan ini, Anda menggunakan metode **as\_named\_input** dari kumpulan data untuk menentukan nama kumpulan data. Kemudian dalam skrip, Anda dapat mengambil kumpulan data berdasarkan nama dari kumpulan **input\_datasets** konteks yang dijalankan tanpa perlu mengambilnya dari ruang kerja. Perhatikan bahwa jika Anda menggunakan pendekatan ini, Anda masih perlu menyertakan argumen skrip untuk dataset, meskipun Anda tidak benar-benar menggunakannya untuk mengambil dataset.

**ScriptRunConfig:**



**Skrip:**



1. **Bekerja dengan kumpulan data file**

Saat bekerja dengan kumpulan data file, Anda dapat menggunakan metode **to\_path()** untuk mengembalikan daftar jalur file yang dienkapsulasi oleh kumpulan data:



**Berikan kumpulan data file ke skrip eksperimen**

Sama seperti kumpulan data Tabular, ada dua cara Anda dapat meneruskan kumpulan data file ke skrip. Namun, ada beberapa perbedaan utama dalam cara dataset dilewatkan.

* **Gunakan argumen skrip untuk kumpulan data file**

Anda dapat meneruskan kumpulan data file sebagai argumen skrip. Tidak seperti kumpulan data tabular, Anda harus menentukan mode untuk argumen kumpulan data file, yang dapat berupa **as\_download** atau **as\_mount**. Ini menyediakan titik akses yang dapat digunakan skrip untuk membaca file dalam kumpulan data. Biasanya, Anda harus menggunakan **as\_download**, yang menyalin file ke lokasi sementara pada komputasi tempat skrip dijalankan. Namun, jika Anda bekerja dengan sejumlah besar data yang mungkin tidak memiliki cukup ruang penyimpanan pada komputasi eksperimen, gunakan **as\_mount** untuk mengalirkan file langsung dari sumbernya.

**ScriptRunConfig:**



**Skrip:**



* **Gunakan input bernama untuk kumpulan data file**

Anda juga dapat meneruskan kumpulan data file sebagai masukan bernama. Dalam pendekatan ini, Anda menggunakan metode **as\_named\_input** dari kumpulan data untuk menentukan nama sebelum menentukan mode akses. Kemudian dalam skrip, Anda dapat mengambil kumpulan data berdasarkan nama dari kumpulan **input\_datasets** konteks yang dijalankan dan membaca file dari sana. Seperti halnya kumpulan data tabular, jika Anda menggunakan input bernama, Anda masih perlu menyertakan argumen skrip untuk kumpulan data, meskipun Anda tidak benar-benar menggunakannya untuk mengambil kumpulan data.

**ScriptRunConfig:**

env = Environment('my\_env')

packages = CondaDependencies.create(conda\_packages=['pip'],

pip\_packages=['azureml-defaults',

'azureml-dataprep[pandas]'])

env.python.conda\_dependencies = packages

script\_config = ScriptRunConfig(source\_directory='my\_dir',

script='script.py',

arguments=['--ds', file\_ds.as\_named\_input('my\_ds').as\_download()],

environment=env)

**Skrip:**



1. **Latihan - Bekerja dengan data**

Sekarang adalah kesempatan Anda untuk bekerja dengan data di Azure Machine Learning.

Dalam latihan ini, Anda akan:

* Unggah data ke penyimpanan data
* Membuat himpunan data
* Gunakan kumpulan data untuk melatih model

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.>

Link : <https://microsoftlearning.github.io/mslearn-dp100/>

1. 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.
2. Selesaikan latihan **Bekerja dengan data.**
3. **Ringkasan**

Dalam modul ini, Anda belajar cara:

* Membuat dan menggunakan penyimpanan data.
* Membuat dan menggunakan kumpulan data.

Untuk detail selengkapnya tentang bekerja dengan data di Azure Machine Learning, lihat Akses data di Azure Machine Learning. : <https://learn.microsoft.com/id-id/azure/machine-learning/concept-data?tabs=uri-file-example%2Ccli-data-create-example>