Setiap kriteria dapat bernilai 0 sampai 4 *points* (pts). Untuk lulus dari submission ini, Anda **harus mendapatkan 2 points dari setiap kriteria**. Submission akan **ditolak** jika masih terdapat kriteria dengan 0 points.

**Kriteria 1: Melakukan Eksperimen terhadap Dataset Pelatihan**

Kriteria pertama merupakan senjata utama untuk menyelesaikan submission kelas ini. Hal ini sangat berguna sebagai eksplorasi dan eksperimen awal sebelum Anda melakukan otomatisasi pada kriteria berikutnya.

Pada tahap ini, Anda **wajib** menggunakan [**Template Eksperimen MSML**](https://colab.research.google.com/drive/1vSTQWWgGqPGBGHvv8lbeGdoa5N92D_UC?usp=sharing) sebagai panduan awal sebelum membuat file untuk melakukan otomatisasi data preprocessing. Pastikan template tersebut diikuti dengan benar untuk memastikan proses berjalan sesuai standar yang ditetapkan.

Setelah melakukan eksplorasi, Anda telah memiliki panduan utama untuk membuat file yang dapat melakukan preprocessing data secara otomatis. Selanjutnya, silakan konversi langkah-langkah yang ada pada notebook eksperimen untuk membuat file otomatisasi tersebut.

Pada akhirnya agar dapat memenuhi kriteria ini, Anda harus membuat sebuah repository (GitHub dan lokal) dengan struktur seperti ini.

Eksperimen\_SML\_Nama-siswa

├── .workflow (jika menerapkan advance)

├── namadataset\_raw (bisa berupa file atau folder)

├── preprocessing

    └── Eksperimen\_Nama-siswa.ipynb

    └── automate\_Nama-siswa.py (jika menerapkan skilled)

    └── namadataset\_preprocessing (bisa berupa file atau folder)

Berikut adalah penilaian lengkap untuk kriteria 1:

* **Reject (0 pts)**
  + Tidak melakukan seluruh tahapan experimentation yang ada pada template secara manual.
  + Tidak melakukan data loading pada notebook.
  + Tidak melakukan EDA pada notebook.
  + Tidak melakukan preprocessing pada notebook.
* **Basic (2 pts)**
  + Melakukan tahapan experimentation secara manual.
  + Melakukan data loading pada notebook.
  + Melakukan EDA pada notebook.
  + Melakukan preprocessing pada notebook.
* **Skilled (3 pts)**
  + Tahap *basic*terpenuhi.
  + Membuat sebuah file automate\_Nama-siswa.py yang berisikan fungsi untuk melakukan preprocessing secara otomatis sehingga mengembalikan data yang siap dilatih.
    - Pada tahap ini Anda harus melakukan konversi dari proses eksperimen sebelumnya, sehingga tahapannya harus sama tetapi memiliki struktur yang berbeda.
* **Advance (4 pts)**
  + Tahap *skilled* terpenuhi.
  + Membuat sebuah workflow pada GitHub Actions agar dapat melakukan preprocessing setiap kali trigger terpantik.
    - Anda harus membuat sebuah repository dengan nama Eksperimen\_SML\_Nama-siswa berisi seluruh file yang sama dengan rekomendasi struktur folder pada kriteria 1.
    - Pastikan Actions yang dibuat mengembalikan sebuah dataset terbaru yang sudah diproses sedemikian rupa.

**Kriteria 2: Membangun Model Machine Learning**

Setelah selesai melalui tahapan preprocessing, Anda harus melatih model menggunakan dataset yang sudah siap digunakan (bukan raw). Nantinya Anda harus membuat sebuah folder yang berisikan file **modelling.py** beserta *dependencies*nya dengan struktur seperti berikut.

Membangun\_model

├── modelling.py

├── modelling\_tuning.py (jika menerapkan skilled/advanced)

├── namadataset\_preprocessing (bisa berupa file atau folder)

├── screenshoot\_dashboard.jpg

├── screenshoot\_artifak.jpg

├── requirements.txt

├── DagsHub.txt (berisikan tautan DagsHub jika menerapkan advanced)

Sebagai informasi, tahapan ini dapat Anda jalankan pada lokal environment sebagai jembatan penghubung ke kriteria tiga.

Berikut adalah penilaian lengkap untuk kriteria 2:

* **Reject (0 pts)**
  + Tidak membuat model machine learning/deep learning menggunakan MLflow dan menyimpan artefak di MLflow Tracking UI.
  + Tidak menyimpan informasi apa pun pada logging.
* **Basic (2 pts)**
  + Melatih model machine learning (Scikit-Learn) menggunakan MLflow Tracking UI yang disimpan secara lokal tanpa menggunakan hyperparameter tuning.
  + Menggunakan autolog dari MLflow.
  + Mengirimkan screenshot yang valid.
* **Skilled (3 pts)**
  + Melatih model machine learning/deep learning menggunakan MLflow Tracking UI yang disimpan secara lokal dengan menerapkan hyperparameter tuning.
  + Alih-alih menggunakan autolog, Anda diharapkan menggunakan manual logging dengan metriks yang sama dengan autolog.
* **Advance (4 pts)**
  + Melatih model machine learning/deep learning menggunakan MLflow Tracking UI yang disimpan secara online dengan DagsHub.
  + Alih-alih menggunakan autolog, siswa diharapkan menggunakan manual logging dengan metriks yang tidak hanya tercover pada autolog (autolog + minimal 2 nilai tambahan).

**Kriteria 3: Membuat Workflow CI**

Setelah membuat dan memastikan file **modelling.py**berjalan dengan baik, selanjutnya Anda harus membuat workflow CI menggunakan MLflow Project agar dapat melakukan re-training model secara otomatis ketika trigger dipantik.

Silakan Anda buat sebuah project repository baru di GitHub dengan struktur seperti berikut ini.

Workflow-CI

├── .workflow

├── MLProject (folder)

    └── modelling.py

    └── conda.yaml

    └── MLProject

    └── namadataset\_preprocessing (bisa berupa file atau folder)

    └── Tautan ke Docker Hub

    └── (file tambahan jika diperlukan)

Anda dapat menggunakan file **modelling.py, conda.yaml** serta dataset yang sudah siap dilatih dari hasil eksperimen sebelumnya. Pada tahap ini, Anda hanya perlu membuat struktur yang diminta beserta file MLProjectnya saja. Namun, tidak menutup kemungkinan Anda harus menyesuaikan file modelling.py ketika masuk ke tahap ini.

Berikut adalah penilaian lengkap untuk kriteria 3:

* **Reject (0 pts)**
  + Tidak membuat folder MLProject.
  + Tidak membuat workflow CI menggunakan GitHub Actions.
* **Basic (2 pts)**
  + Membuat folder MLProject.
  + Membuat Worflow CI yang dapat membuat model machine learning ketika trigger terpantik.
* **Skilled (3 pts)**
  + Membuat workflow CI dan menyimpan artefak ke suatu repositori (GitHub yang sama atau Google Drive).
* **Advance (4 pts)**
  + Membuat workflow CI dan menyimpan artefak ke suatu repositori (GitHub yang sama atau Google Drive) serta membuat Docker Images ke Docker Hub menggunakan fungsi mlflow build-docker.

**Kriteria 4: Membuat Sistem Monitoring dan Logging**

Monitoring dan Logging merupakan tahapan yang tidak bisa berdiri sendiri karena membutuhkan artefak yang dihasilkan oleh kriteria tiga. Nantinya, Anda hanya akan mengumpulkan tangkapan layar mengenai skill yang diampu dengan struktur seperti berikut ini.

Monitoring dan Logging

├── 1.bukti\_serving

├── 2.prometheus.yml

├── 3.prometheus\_exporter.py

├── 4.bukti monitoring Prometheus (folder)

    └── 1.monitoring\_<metriks>

    └── 2.monitoring\_<metriks>

    └── dst (sesuaikan dengan poin yang diraih)

├── 5.bukti monitoring Grafana (folder)

    └── 1.monitoring\_<metriks>

    └── 2.monitoring\_<metriks>

    └── dst (sesuaikan dengan poin yang diraih)

├── 6.bukti alerting Grafana (folder)

    └── 1.rules\_<metriks>

    └── 2.notifikasi\_<metriks>

    └── 3.rules\_<metriks>

    └── 4.notifikasi\_<metriks>

    └── dst (sesuaikan dengan poin yang diraih)

├── 7.inference.py

├── folder/file tambahan

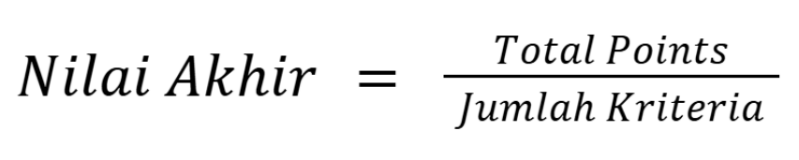
**Penting**, pastikan untuk membuat dashboard dengan nama **username akun Dicoding** sehingga tangkapan layar yang Anda kirimkan akan berisikan kredensial.

Berikut adalah penilaian lengkap untuk kriteria 4:

* Reject (0 pts)
  + Tidak melakukan serving model pada environment local.
  + Tidak melakukan monitoring performa sistem machine learning menggunakan Prometheus
  + Tidak menggunakan Grafana sebagai tools visualisasi dan alerting sistem machine learning
* Basic (2 pts)
  + Melakukan serving model baik itu melalui artefak yang sudah dibuat atau pull Images (jika menerapkan kriteria CI untuk melakukan push ke Docker Hub)
    - Bisa melalui mlflow model serve, mlflow deployments, atau pull images jika memenuhi kriteria 3 advanced.
  + Melakukan monitoring menggunakan Prometheus minimal dengan tiga metriks yang berbeda.
  + Melakukan monitoring menggunakan Grafana dengan metriks yang sama dengan Prometheus.
* Skilled (3 pts)
  + Melakukan monitoring menggunakan Grafana dengan minimal 5 metriks yang berbeda.
  + Membuat satu alerting menggunakan Grafana.
* Advance (4 pts)
  + Melakukan monitoring menggunakan Grafana dengan minimal 10 metriks yang berbeda.
  + Membuat tiga alerting menggunakan Grafana.

**Perhitungan Nilai**

Nilai akhir yang Anda dapatkan diperoleh melalui perhitungan formula berikut.



|  |
| --- |
| **Catatan:**  Perhitungan nilai akhir di atas digunakan apabila setiap kriteria mendapatkan nilai 2 pts atau tidak ada kriteria yang ditolak. |

**Tabel Penilaian**

Adapun untuk penilaian submission dapat dilihat pada tabel berikut.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ketentuan Penilaian** | | | | |
| **Nilai Akhir** | **Nilai Dicoding** | **Nilai Huruf** | **Level of Mastery** | **Makna Nilai** | **Keterangan** |
| <1 | Rejected | E | - | Tidak Lulus | Anda sudah mencoba, tetapi belum memenuhi kompetensi minimal. |
| 1 — <2 | Bintang 2 | D | Below Basic | Kurang | Anda sudah memenuhi semua kompetensi minimal, tetapi terdapat area yang masih bisa ditingkatkan. |
| 2 — <3 | Bintang 3 | C | Basic | Cukup | Anda sudah memenuhi semua kompetensi minimal dari *learning objective*. |
| 3 — <4 | Bintang 4 | B | Skilled | Mahir | Anda sudah memenuhi semua kompetensi dengan baik atau mahir. |
| 4 | Bintang 5 | A | Advanced | Tingkat Lanjut | Anda sudah memenuhi semua kompetensi dengan sangat baik atau tingkat lanjut. |

* Anda disarankan menggunakan environment berikut untuk menunjang submission:
  + Python 3.12.7
    - mlflow==2.19.0
* Jika Anda menggunakan data unstructured dan menggunakan framework TensorFlow, silakan sesuaikan beberapa tahapan, tetapi tetap mengacu ke masing-masing objektif kriteria.
* Format pengiriman submission.

SMSML\_Nama-siswa

├── Eksperimen\_SML\_Nama-siswa.txt

├── Membangun\_model

    ├── modelling.py

    ├── modelling\_tuning.py (skilled/advanced)

    ├── namadataset\_preprocessing (bisa berupa file atau folder)

    ├── screenshoot\_dashboard.jpg

    ├── screenshoot\_artifak.jpg

    ├── requirements.txt

    ├── DagsHub.txt (berisikan tautan DagsHub jika menerapkan advanced)

├── Workflow-CI.txt

├── Monitoring dan Logging

    ├── 1.bukti\_serving

    ├── 2.prometheus.yml

    ├── 3.prometheus\_exporter.py

    ├── 4.bukti monitoring Prometheus (folder)

        └── 1.monitoring\_<metriks>

        └── 2.monitoring\_<metriks>

        └── dst (sesuaikan dengan poin yang diraih)

    ├── 5.bukti monitoring Grafana (folder)

        └── 1.monitoring\_<metriks>

        └── 2.monitoring\_<metriks>

        └── dst (sesuaikan dengan poin yang diraih)

    ├── 6.bukti alerting Grafana (folder)

        └── 1.rules\_<metriks>

        └── 2.notifikasi\_<metriks>

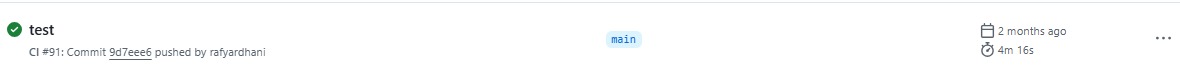
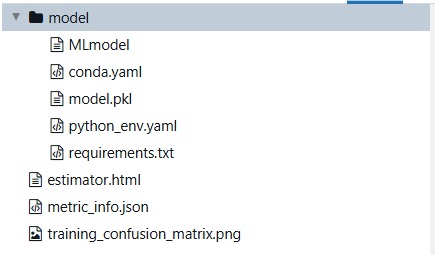
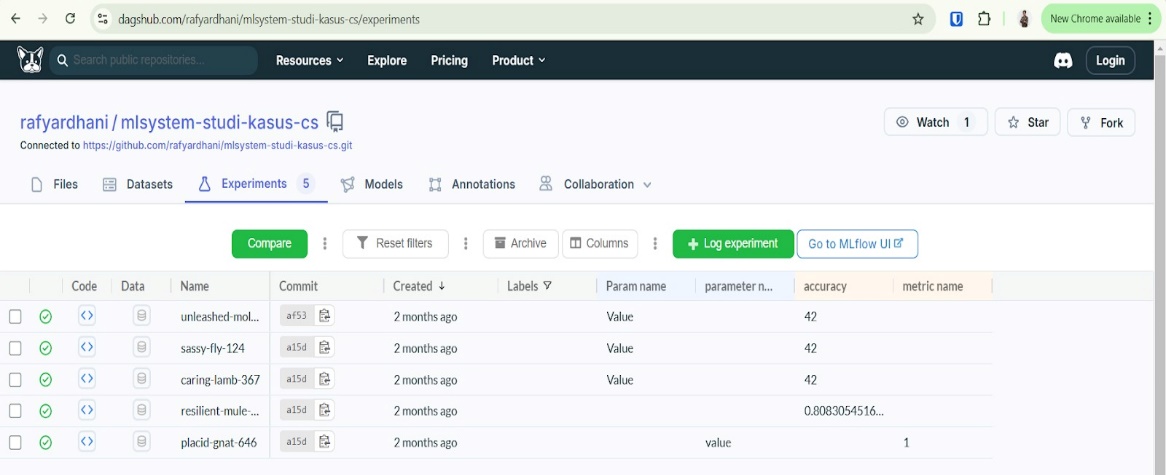
        └── 3.rules\_<metriks>

        └── 4.notifikasi\_<metriks>

        └── dst (sesuaikan dengan poin yang diraih)

    ├── 7.Inference.py

├── folder/file tambahan

* + **Catatan**
    - Eksperimen\_SML\_Nama-siswa.txt berisikan tautan ke repository GitHub kriteria pertama dengan format seperti yang sudah disampaikan pada halaman kriteria 1.
    - Workflow-CI.txt berisikan tautan ke repository GitHub kriteria ketiga dengan format seperti yang sudah disampaikan pada halaman kriteria 3.
    - **Pastikan Anda mengatur visibilitas public pada kedua repository tersebut.**
* Kriteria 1
  + Silakan Anda buat sebuah repository GitHub dengan visibilitas **Public** agar bisa diperiksa oleh tim reviewer
  + Pastikan Anda mengerjakan dan menjalankan seluruh tahapan tanpa ada error pada seluruh cell.
  + Jika Anda menerapkan **skilled**, silakan buat satu buah file .py berdasarkan workflow preprocessing yang dilakukan pada tahap eksperimen.
  + Jika Anda menerapkan **Advance**, silakan buat workflow yang sudah dijalankan dan minimal satu kali berhasil tanpa menghasilkan error.  
    
* Kriteria 2
  + Pastikan Anda menyimpan seluruh artefak pada MLflow Tracking UI dengan alamat *localhost* atau 127.0.0.1.  
    
  + Jika menerapkan **skilled**, pastikan Anda membuat file modelling\_tuning.py dan melakukan logging model yang menghasilkan struktur seperti berikut.  
    
  + Jika menerapkan **advanced**, pastikan Anda menambahkan minimal dua artefak selain pada tahapan **skilled**. Selain itu, Anda harus menyimpan file artefak MLflow ke DagsHub agar dapat diakses secara online.  
      
    Sebagai contoh, Anda dapat menggunakan kode berikut agar dapat menyimpan artefak pelatihan ke DagsHub.

import dagshub

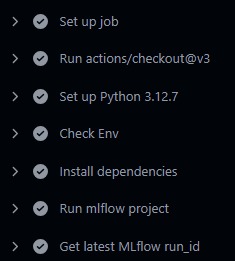
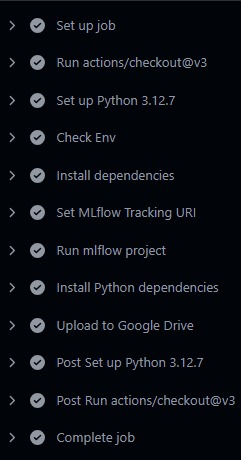
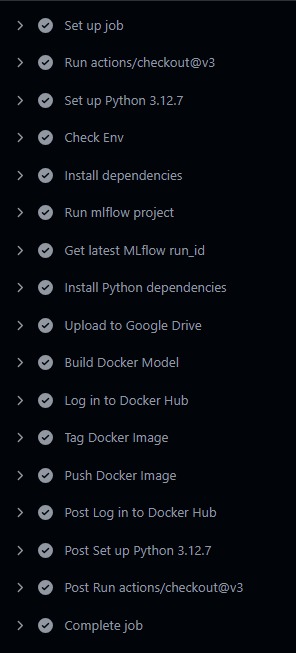
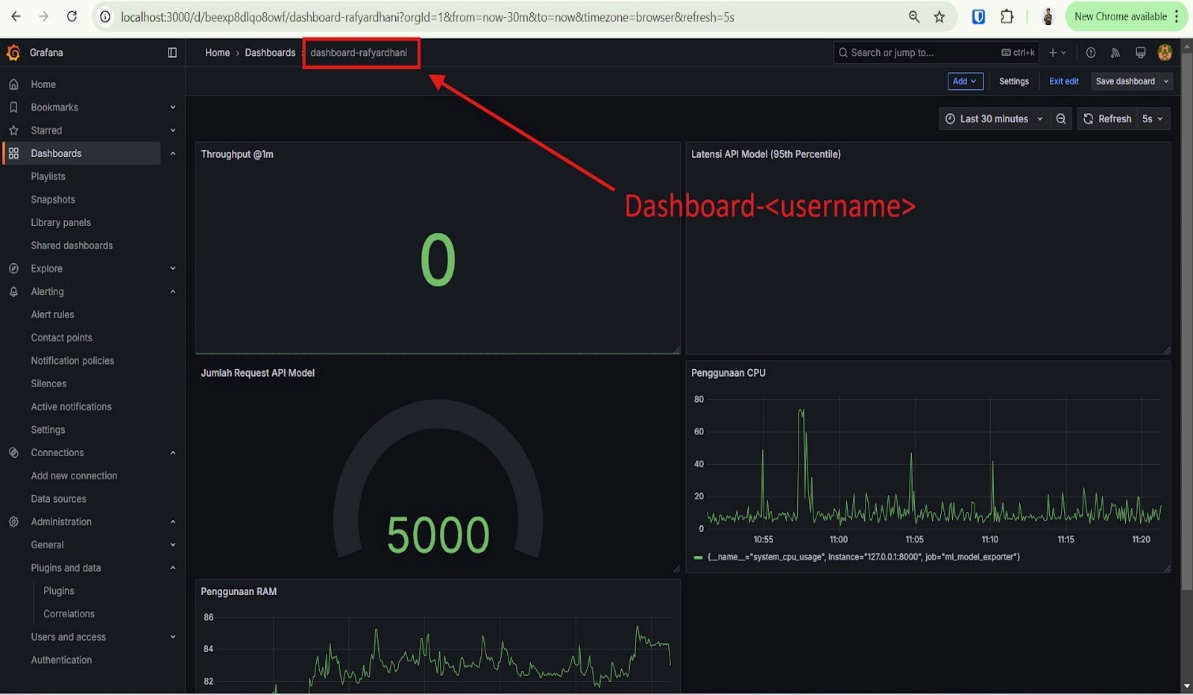
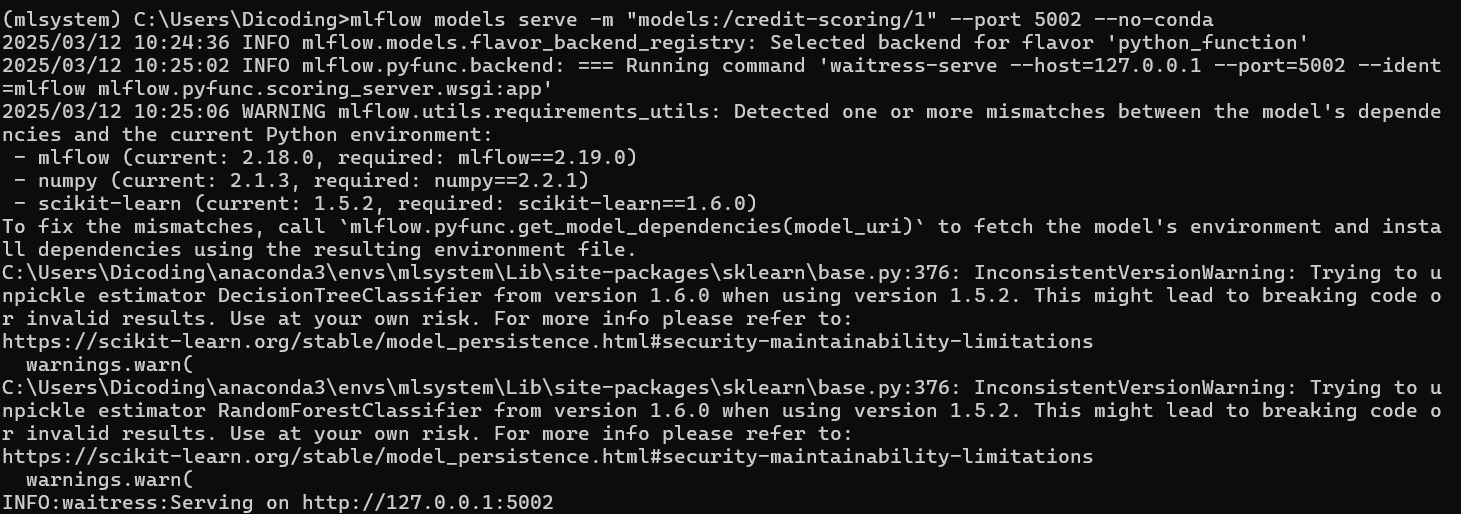
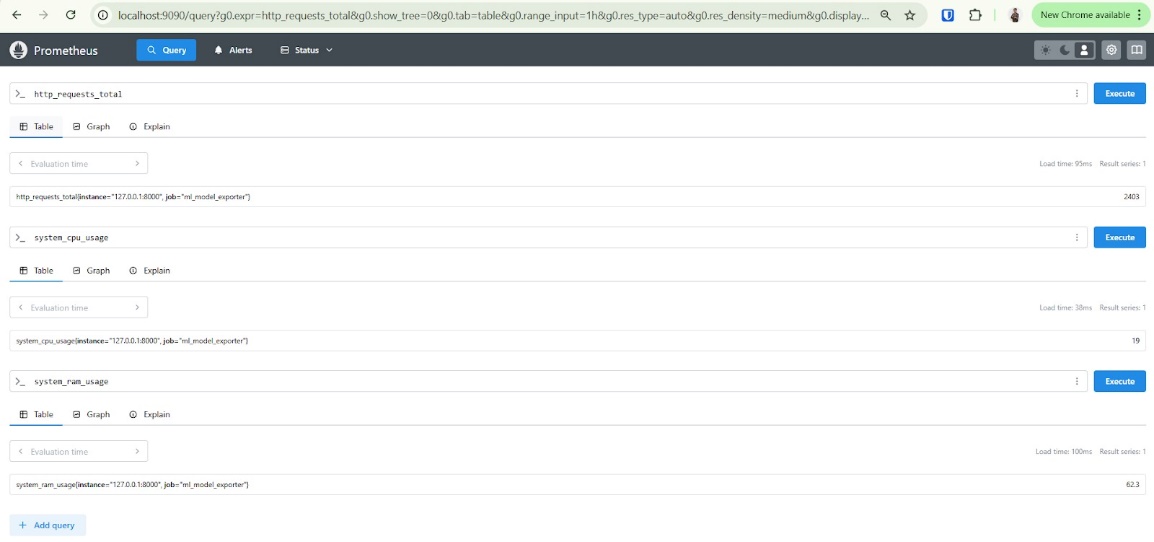
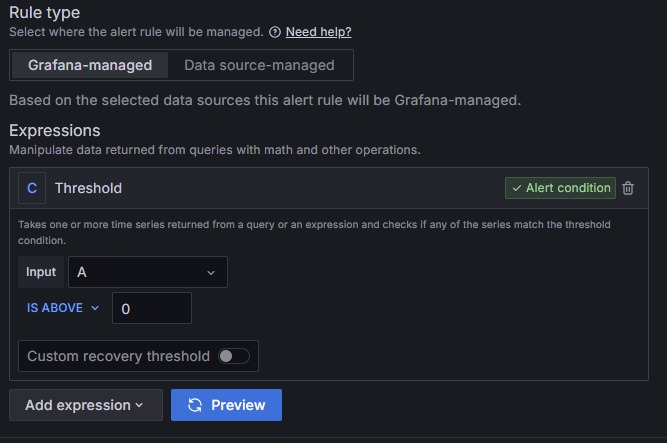
import mlflow

dagshub.init(repo\_owner=’<nama\_owner>', repo\_name='<nama\_repo>', mlflow=True)

with mlflow.start\_run():

  # Your training code here...

Jika Anda belum memasukkan kredensial apa pun, silakan *login* terlebih dahulu dengan mengikuti dokumentasi [DagsHub](https://dagshub.com/docs/integration_guide/mlflow_tracking/).

* Kriteria 3
  + Silakan Anda buat sebuah repository GitHub dengan visibilitas **Public** agar bisa diperiksa oleh tim reviewer.
  + Pastikan Anda membuat workflow dari nol agar dapat memastikan semuanya berjalan dengan baik.
  + Jangan lupa untuk memasukkan *secrets* agar informasi akun tidak disalahgunakan orang lain.
  + Jika Anda menerapkan **basic**, pastikan workflow CI yang dibuat memuat tahapan berikut.  
    
  + Jika Anda menerapkan  **skilled**, pastikan workflow CI yang dibuat memuat tahapan berikut.  
      
    Silakan sesuaikan tahapan “Upload to Google Drive” dengan metode penyimpanan yang Anda pilih seperti “Upload to GitHub” atau “Upload to GitHub LFS”
  + Jika Anda menerapkan **advanced**, pastikan workflow CI yang dibuat memuat tahapan berikut.  
      
    Silakan sesuaikan tahapan “Upload to Google Drive” dengan metode penyimpanan yang Anda pilih seperti “Upload to GitHub” atau “Upload to GitHub LFS”
* Kriteria 4
  + Pastikan tangkapan layar yang Anda kirim memiliki nama dashboard yang berisikan username akun Dicoding Anda seperti berikut.  
    
  + Jika menerapkan **basic**, silakan lakukan serving model baik itu menggunakan MLflow serve, membuat API menggunakan framework, dan lain sebagainya. Namun, pastikan Anda menyertakan bukti serving seperti berikut ini.  
      
    Atau ketika menggunakan Docker Images bisa seperti berikut.
    - Anda harus menyertakan bukti Prometheus sudah berjalan dengan membuat minimal tiga buah metriks monitoring seperti berikut ini.  
      
    - Selanjutnya silakan konversi metriks yang sudah Anda buat menggunakan Prometheus ke Grafana agar visualisasinya lebih baik.
  + Jika menerapkan Alerting, silakan sisipkan dua file seperti berikut ini.
    - **Bukti rules yang dibuat**  
      
    - **Bukti notifikasi**  
      