



TUGAS ASINKRON 2 MINGGU 1 KELAS DAI-002
Use automated machine learning in Azure Machine Learning

Oleh :

KELOMPOK 4

- Dhea Amalia Ariantoputri
 - Dinda Ega Fajarwati
 - Fadhil Rausyanfikir
- Dwira Kurnia Larasati
 - Renaka Agusta
 - Eka Aperinda Putri
 - Dinna Nurfadlillah

Mentor :

Noviyanti Tri Maretta Sagala, S.TI., M.Sc

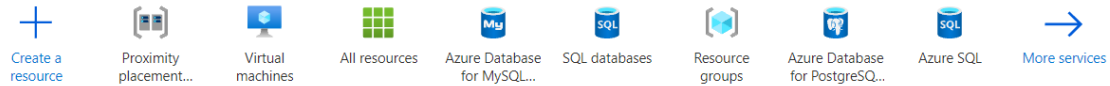
Asisten Mentor :

Ainul Zakiy

STUDI INDEPENDEN PT. MICROSOFT INDONESIA
OKTOBER 2021

Pada tugas asinkron ini setiap kelompok diminta untuk melatih sebuah model Machine Learning dan menggunakan Azure Machine Learning Designer sebagai sebuah layanan mengikuti pada modul di Microsoft Learn dengan beberapa perubahan yang diminta. Adapun langkah-langkah yang harus dilakukan diantaranya :

Azure services



1. Masuk ke Portal Azure dan buat resource baru dengan menekan tombol "Create a resource"

Machine Learning

Microsoft



Machine Learning [Add to Favorites](#)

Microsoft

★ 4.3 (113 Azure ratings)

[Create](#)

2. Dikarenakan kita akan melakukan training model maka pilihlah layanan Machine Learning dengan menekan tombol create pada bagian tersebut

[Beranda](#) > [Create a resource](#) > [Machine Learning](#) >

Pembelajaran mesin ...

Buat ruang kerja pembelajaran mesin

mengatur dan mengelola semua sumber daya Anda.

Langganan * ⓘ

Azure for Students



Grup sumber daya * ⓘ

(Baru) mlkel4



[Buat baru](#)

Detail ruang kerja

Tentukan nama dan kawasan untuk ruang kerja.

Nama ruang kerja * ⓘ

mlkel4011021



Kawasan * ⓘ

Southeast Asia



Akun penyimpanan * ⓘ

(baru) mlkel40110218944939393



[Buat baru](#)

Key vault * ⓘ

(baru) mlkel40110214058373628



[Buat baru](#)

Application Insights * ⓘ

(baru) mlkel40110217214965608



[Buat baru](#)

[Tinjau + buat](#)

[< Sebelumnya](#)

[Berikutnya : Jaringan](#)

Microsoft Azure

Cari sumber daya, layanan, dan dokumen (G +/)

[Beranda](#) > [Create a resource](#) > [Machine Learning](#) >

Pembelajaran mesin

Buat ruang kerja pembelajaran mesin

✓ Lolos validasi

Dasar

Jaringan

Lanjutan

Tag

Tinjau + buat

Dasar

Langganan	Azure for Students
Grup sumber daya	(Baru) mlkel4
Kawasan	Southeast Asia
Nama ruang kerja	mlkel4011021
Akun penyimpanan	(baru) mlkel40110218944939393
Key vault	(baru) mlkel40110214058373628
Application Insights	(baru) mlkel40110217214965608
Registri kontainer	Tidak ada

Jaringan

Metode konektivitas	Titik akhir publik (jaringan yang dipilih)
---------------------	--

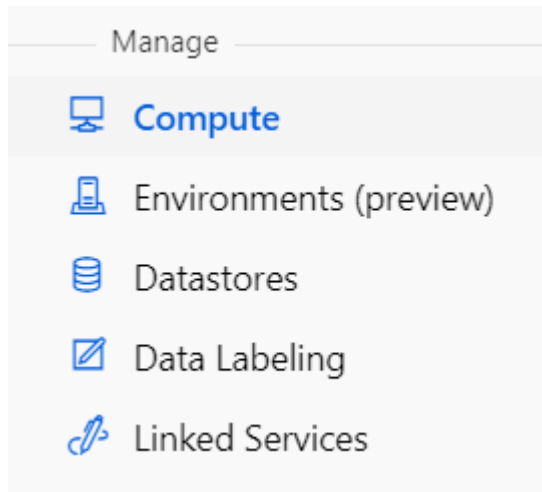
Buat

< Sebelumnya

Berikutnya >

[Unduh templat untuk otomatisasi](#)

- Sebelum memanfaatkan layanan Azure Machine Learning Designer, diharuskan mengisi beberapa data seperti resource group yang digunakan dan workspace. Pada workspace kali ini region yang dipilih adalah southeastasia dikarenakan merupakan region terdekat. Setelah seluruh data diisi masuk ke tab review+create dan tekan tombol create setelah seluruh data yang diisi dirasa sudah benar.



4. Pada bagian pemilihan jenis instance pilihlah "Compute" dikarenakan kita membutuhkan layanan komputasi untuk mentraining model

Compute instances Compute clusters Inference clusters Attached computes

Studio Pembelajaran Mesin Microsoft Azure

Buat instans komputasi

☒ Pengaturan yang Diperlukan
☐ Pengaturan Lanjutan

Konfigurasi pengaturan yang diperlukan
dapat mengubahnya ke pengguna lain di bagian pengaturan tingkat lanjut.

Nama komputasi * ⓘ
cikel4010

Lokasi ⓘ
southeastasia

Tipe mesin virtual ⓘ
☒ CPU ☐ GPU

Ukuran mesin virtual ⓘ
☐ Pilih dari opsi yang disarankan ☒ Pilih dari semua opsi

+ Tambah filter

Cari berdasarkan nama VM...

Buat Kembali Selanjutnya: Pengaturan Tingkat Lanjut

5. Pada tahap ini kita akan melakukan konfigurasi terhadap Compute instance,
- Pada penggunaan nama, kami menginput compute name dengan : “cikel4010”
 - Pada tipenya disini kami memilih menggunakan CPU
 - Untuk Ukuran Mesin Virtual, kami memilih semua opsi dan menggunakan Standard_DS11_v2

Buat kluster komputasi ⓘ

Mesin Virtual

Pilih mesin virtual
Pilih ukuran mesin virtual yang ingin Anda gunakan untuk kluster komputasi Anda.

Lokasi *
Southeast Asia

Prioritas mesin virtual ⓘ
☒ Khusus ☐ Prioritas rendah

Tipe mesin virtual ⓘ
☒ CPU ☐ GPU

Ukuran mesin virtual ⓘ
☐ Pilih dari opsi yang disarankan ☒ Pilih dari semua opsi

+ Tambah filter

Cari berdasarkan nama VM...

Kembali Berikutnya Batal

6. Gambar diatas menunjukkan konfigurasi pada tab Computer Cluster yang berisi
- lokasi pada Southeast Asia
 - prioritas Virtual Machine (VM) adalah dedicated
 - tipe VM adalah CPU
 - untuk ukuran Virtual machine kami memilih select from all option, lalu cari dan pilih Standard_DS11_v2

Buat kluster komputasi ⓘ

Mengonfigurasi Pengaturan

Nama	Kategori	Inti	Kuota yang tersedia	RAM	Penyimpanan	Biaya/Node
Standard_DS11_v2	Memori dioptimalkan	2	2 inti	14 GB	28 GB	\$0.19/jam

Nama komputasi * ⓘ
cckel4011021

Jumlah node minimum * ⓘ
0

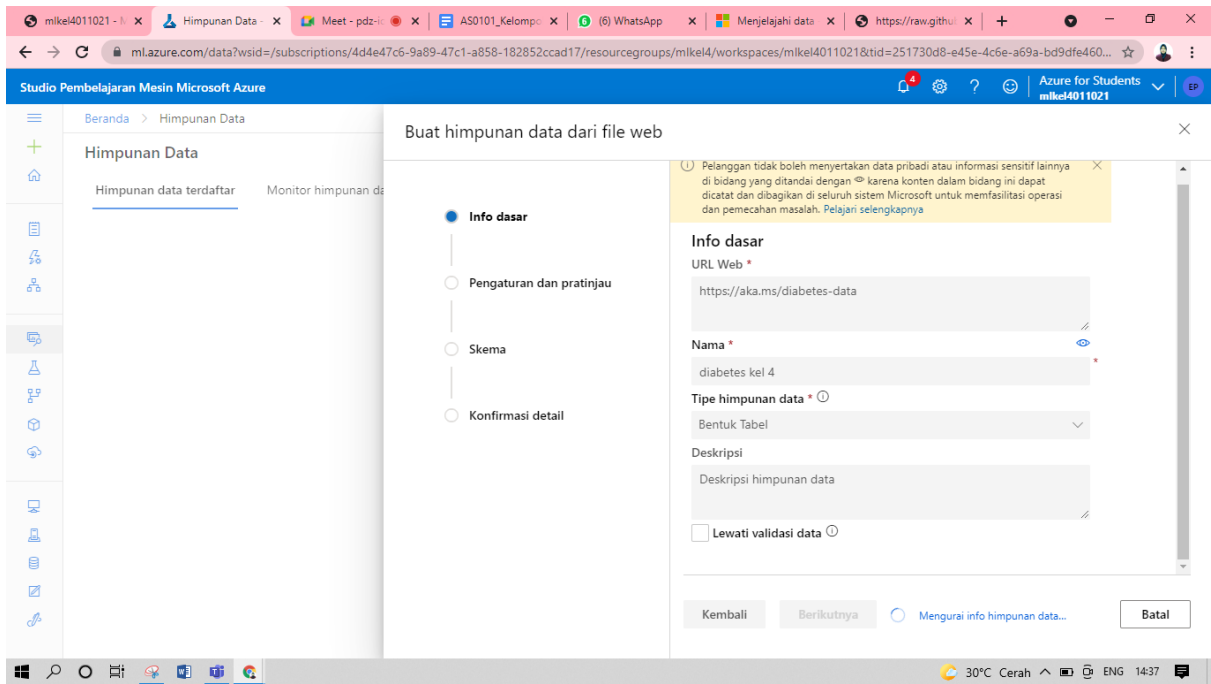
Jumlah node maksimum * ⓘ
2

ⓘ Anda tidak memiliki kuota langganan yang memadai untuk meningkatkan skala lebih dari 1 node.

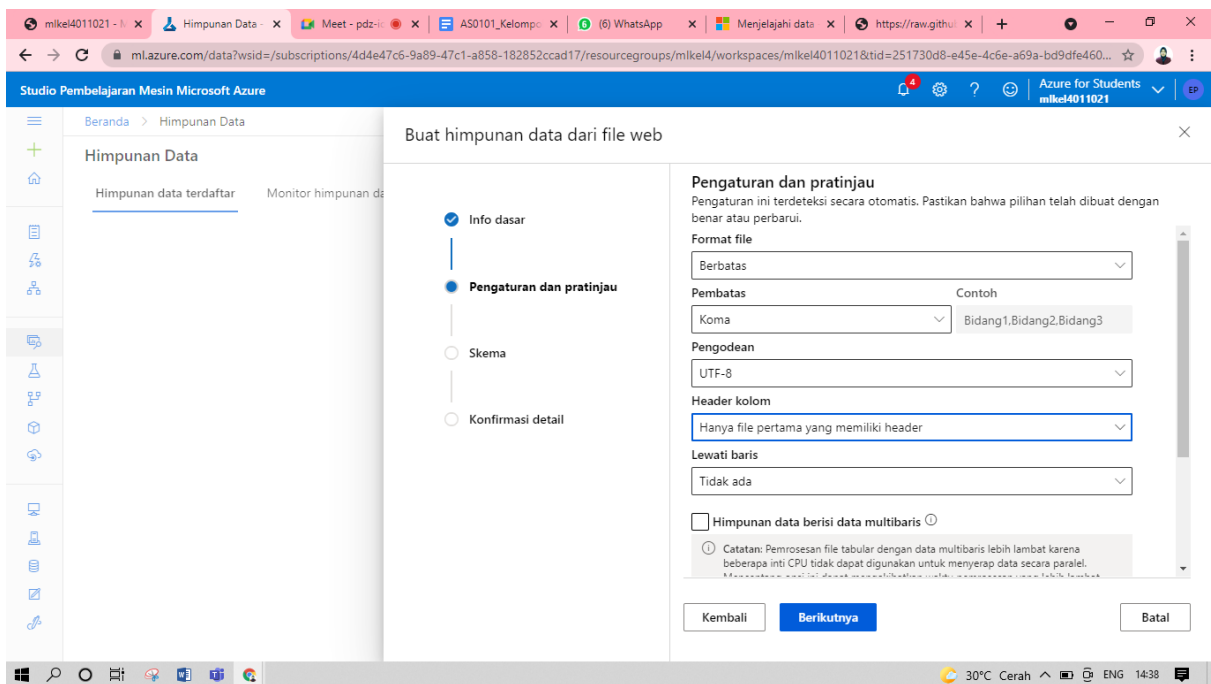
Detik diam sebelum menurunkan skala * ⓘ
120

Kembali Buat Unduh templat untuk otomatisasi Batal

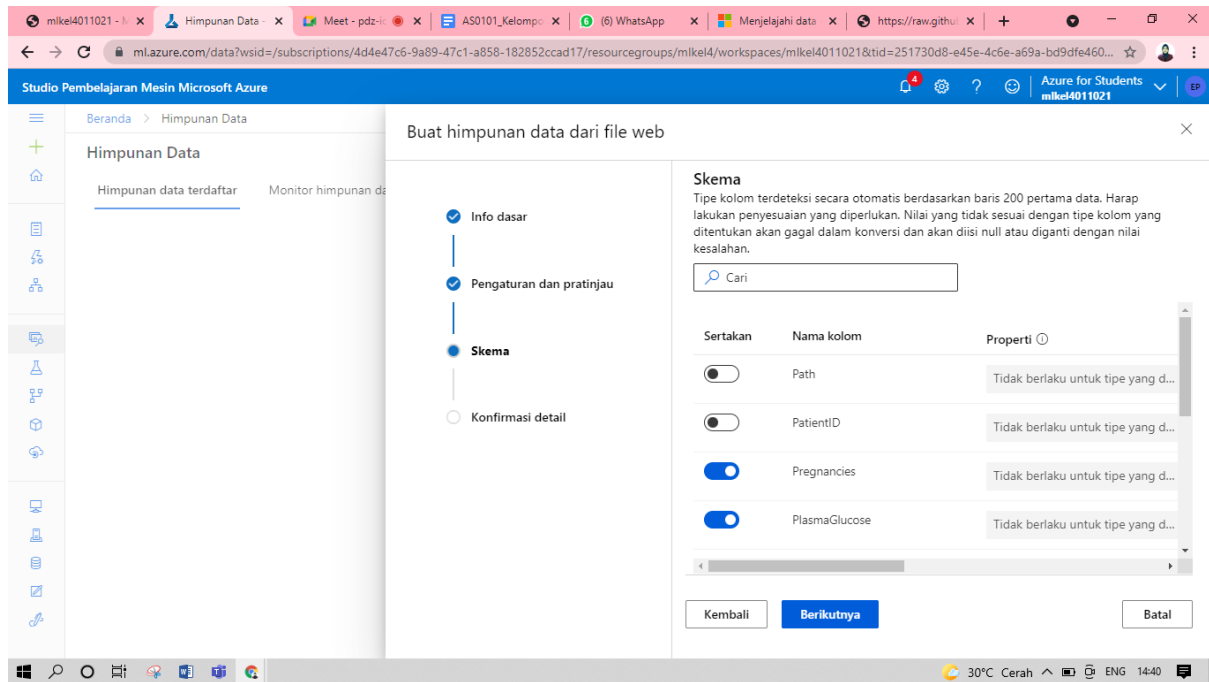
7. Setelah itu klik next, atur konfigurasi nama komputasi, jumlah node maksimum dan detik diam sebelum menurunkan skala lalu klik buat.



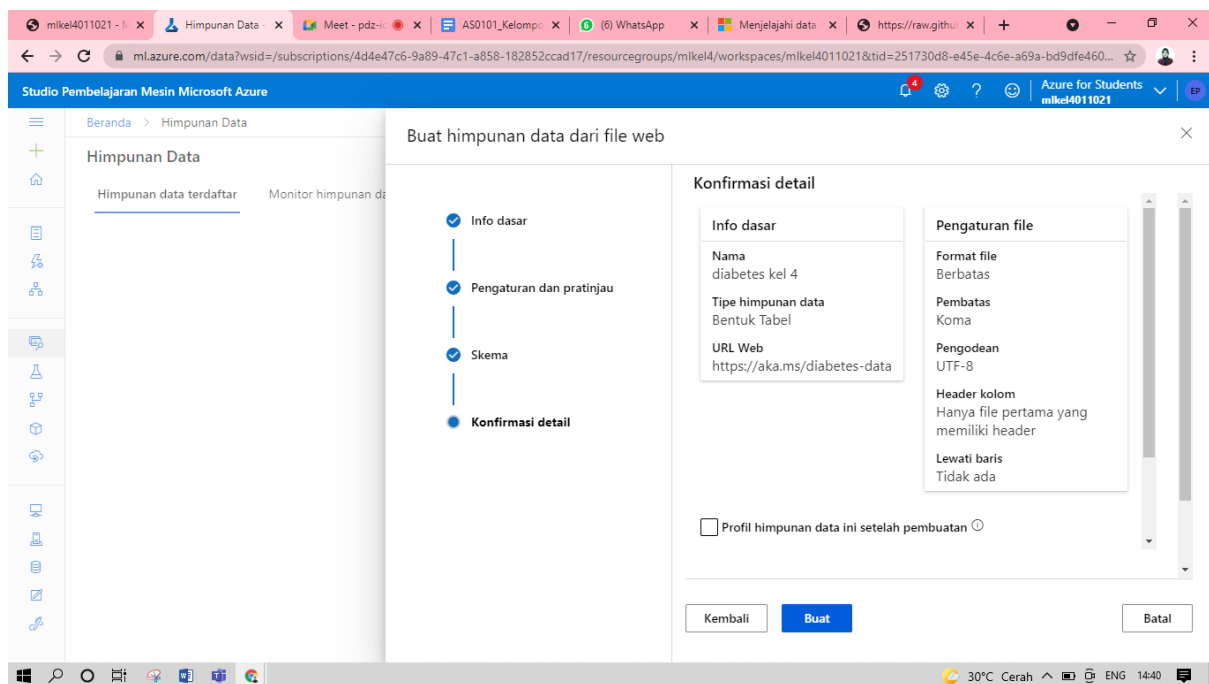
8. Buat dataset dan pilih from web files. Masukkan link dataset, beri nama untuk datasetnya



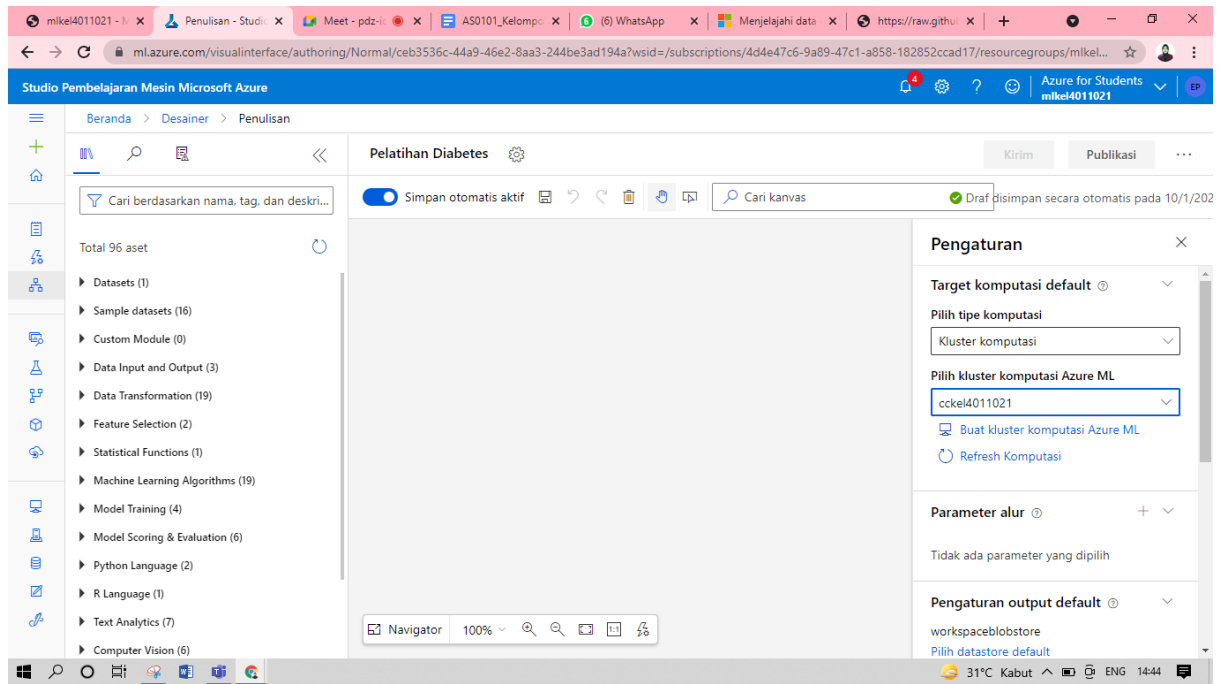
9. Klik next lalu atur header kolom menjadi seperti pada gambar



10. Untuk schema disable PatientID dan Path, kemudian cek apakah tipe data per kolom sudah benar atau belum



11. Periksa sekali lagi, pastikan sudah benar semua lalu klik buat



12. Klik tab designer lalu setting tipe komputasi, pilih compute cluster dan pilih compute cluster yang sudah dibuat sebelumnya: cckel4011021



13. Pilih module dataset dengan cara drag and drop lalu pilih modul normalisasi data lalu hubungkan dataset ke module

Columns to transform



Pilih kolom ☒ Dengan aturan ☐ Berdasarkan nama

Izinkan duplikat dan pertahankan urutan kolom dalam pilihan ☐

Sertakan

Nama kolom

Pregnancies PlasmaGlucose
DiastolicBloodPressure
TricepsThickness
SerumInsulin BMI
DiabetesPedigree Age



Simpan

Batal

14. edit kolom pada normalisasi data dan pilih kolom yang ada pada di gambar. normalisasi dibutuhkan karena pada kolom-kolom diatas memiliki skala pengukuran yang berbeda, sehingga harus disamakan (normalisasi).

Siapkan eksekusi alur

×

Percobaan

☐ Pilih yang ada ☒ Buat baru

Nama percobaan baru *

Pelatihan_diabetes_kel4

Deskripsi eksekusi *

Pelatihan Diabetes

☒ Lanjutkan pada langkah kegagalan

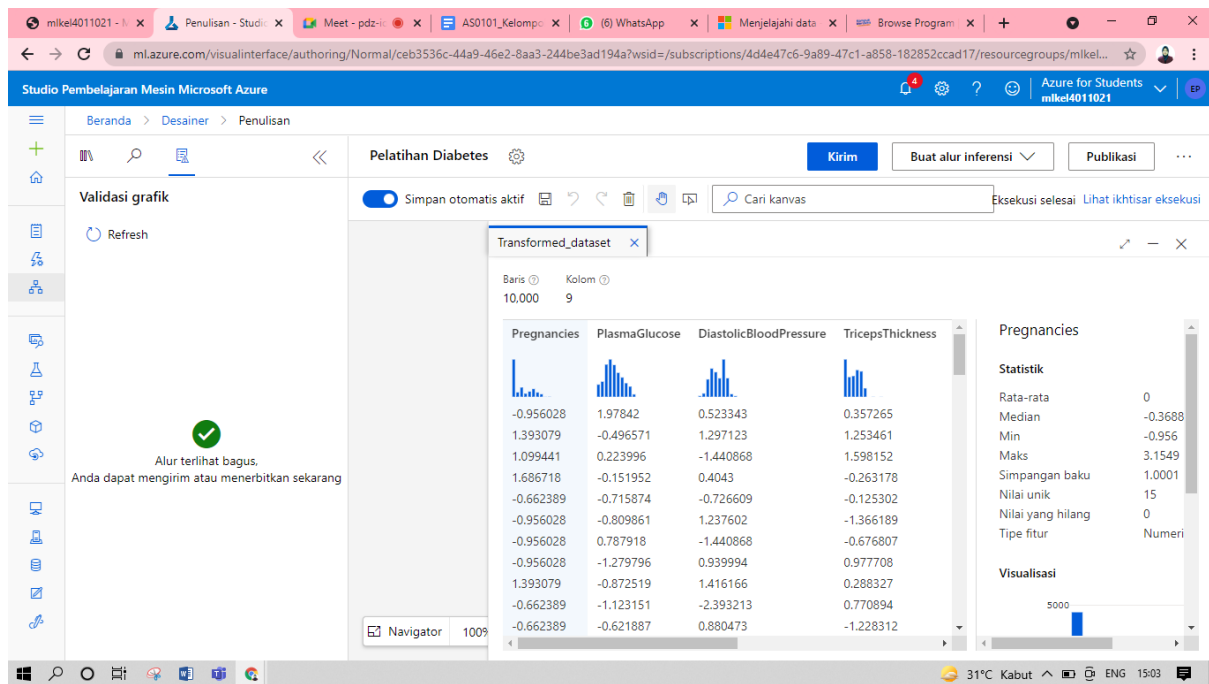
Target komputasi

Default	cckel4011021
---------	--------------

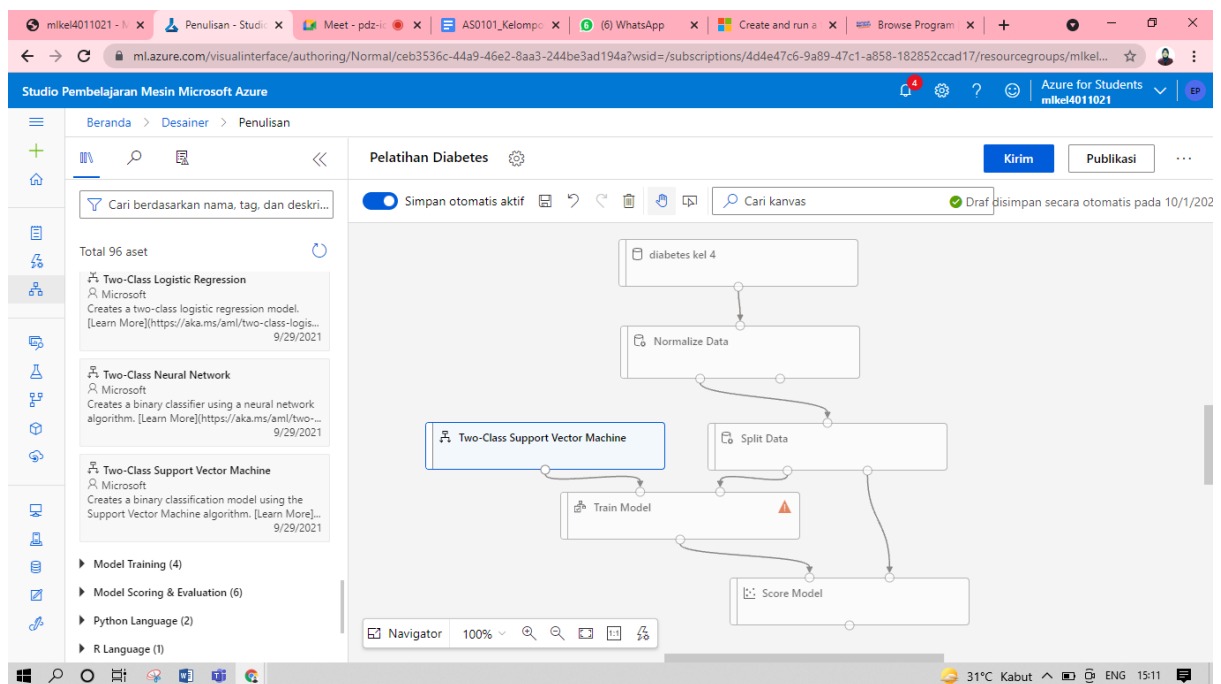
Kirim

Batal

15. setelah itu klik submit dan pilih buat baru dan beri nama lalu klik kirim



16. setelah submit berhasil maka kita dapat melihat hasil visualisasi data yang telah ditransform.



17. Pilih beberapa modul, modul dapat dilihat pada gambar. Hubungkan sesuai dengan apa yang dibutuhkan

Train Model



Label column ? *

Edit kolom

Nama kolom: Diabetic

Model explanations ?

False



Pengaturan output



Pengaturan eksekusi



Komentar



Informasi modul



18. Pada train model, edit kolom dan pilih kolom diabetic. hal ini karena kita hendak memberi label berdasarkan kolom diabetic.

Split Data ✕

Splitting mode ? *

Split Rows ✓

Fraction of rows in the first output dat...

0.8

Randomized split ? *

True ✓

Random seed ? *

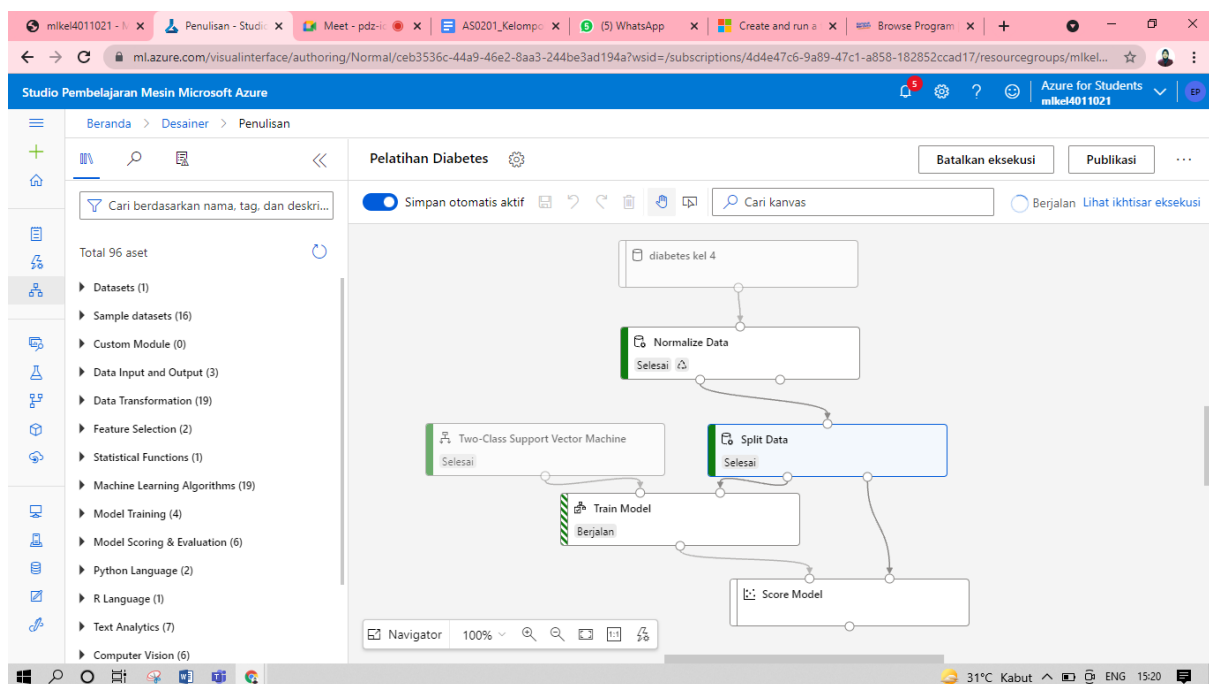
123

Stratified split ? *

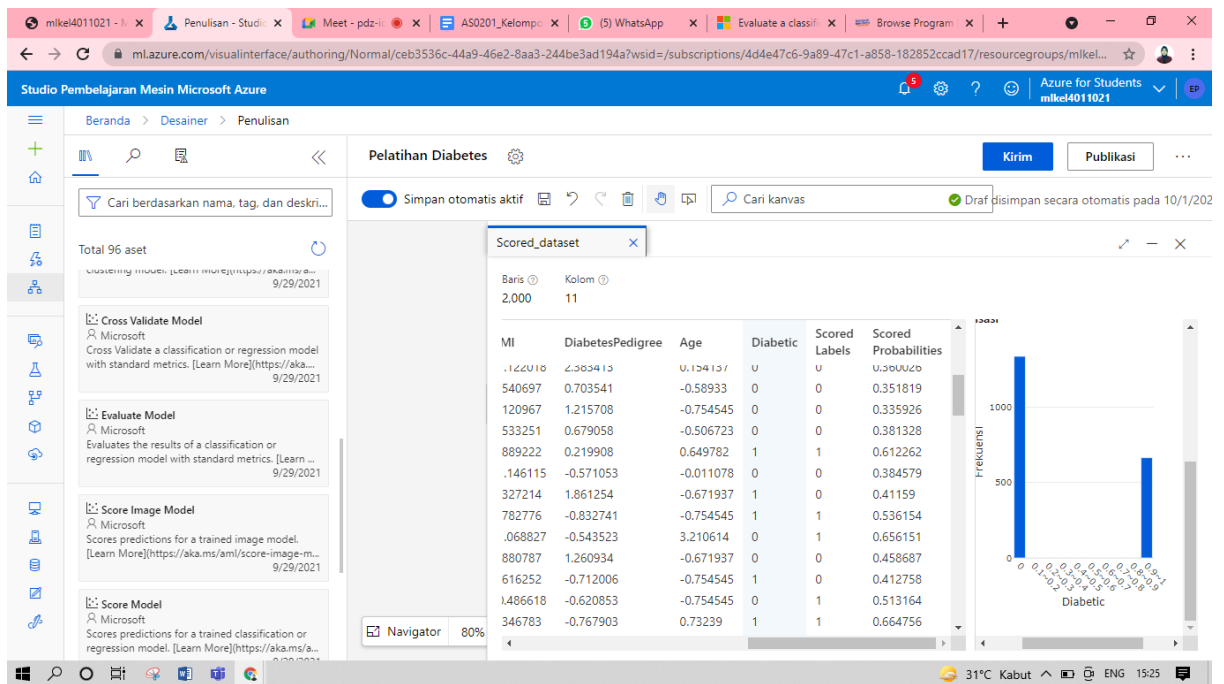
False ✓

Penetapan output

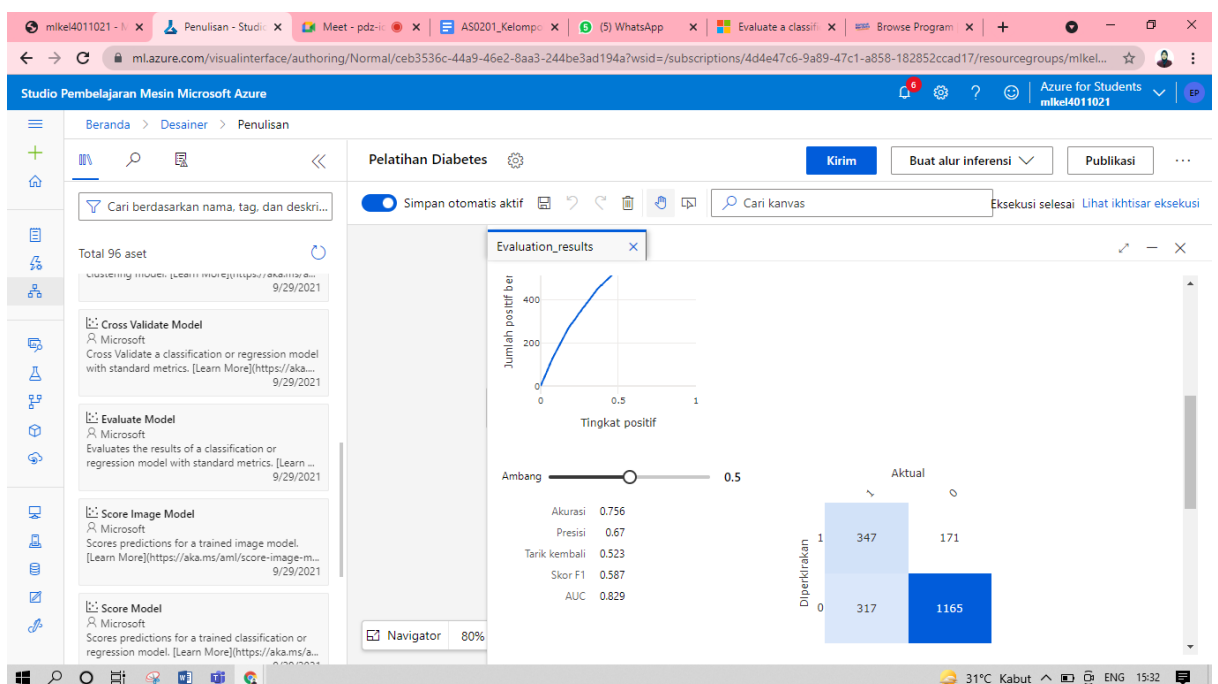
19. Melakukan split data untuk tahap train dan validasi dengan rasio data yang digunakan untuk training adalah 0.8 (80% data untuk dilatih) sedangkan validasi 0.2 adalah untuk testing data.



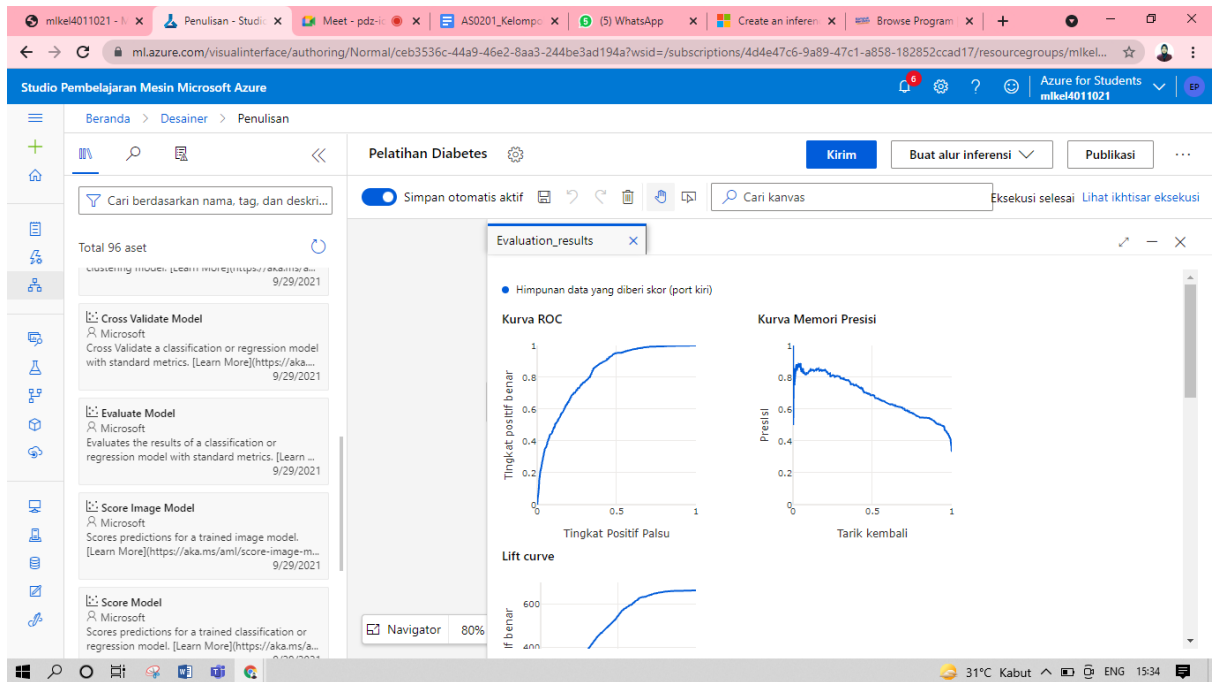
20. Setelah itu klik submit dan pilih dari yang sudah ada lalu create



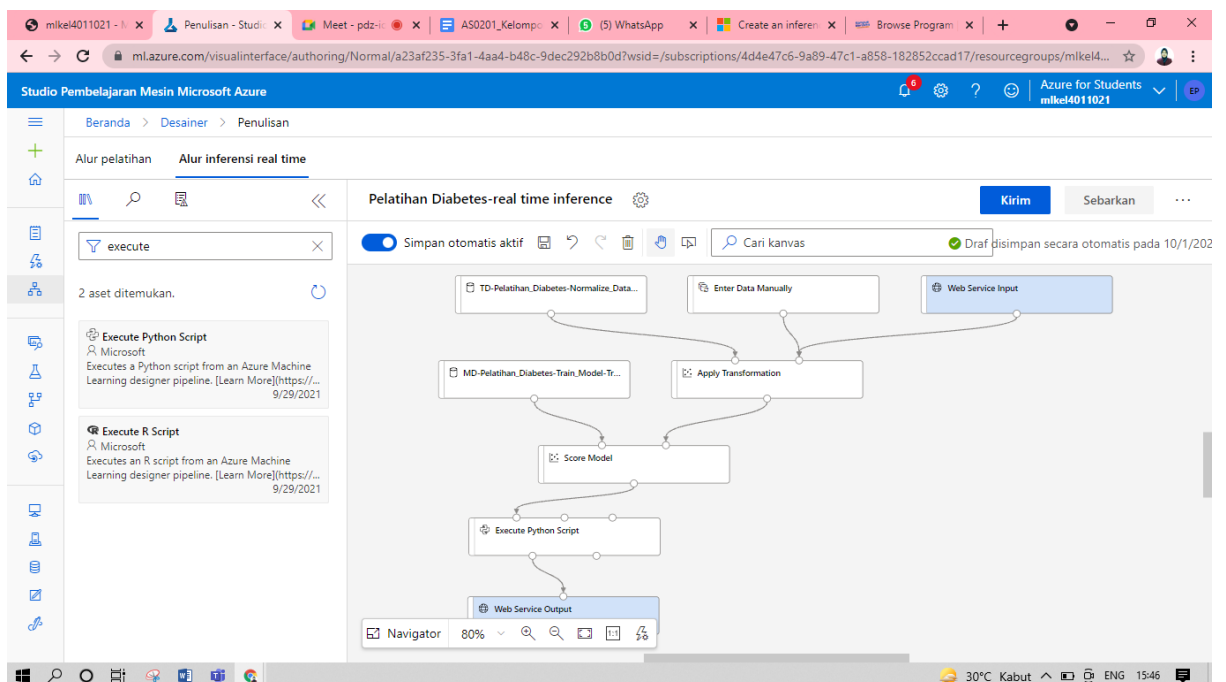
21. berikut hasil evaluasi data pada score model, terlihat ada sebagian pasien yang terprediksi diabetes dan tidak.



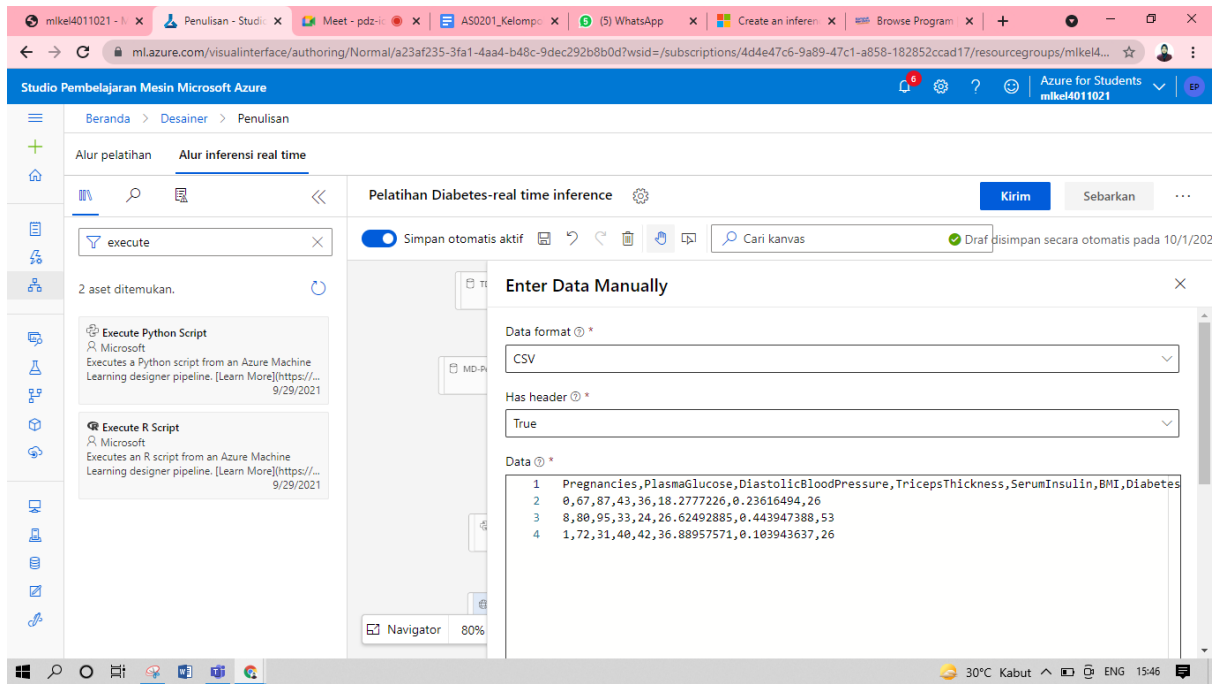
22. confusion matrix merupakan gambaran prediksi dan aktual. confusion matrix terdiri dari 4 keadaan, yaitu True Positive (TP), False Positive (FP), True Negative (TN), dan False Negative (FN). TP adalah keadaan dimana keadaan positif diprediksi positif, sedangkan FP adalah jika keadaan negatif diprediksi positif. TN adalah keadaan dimana keadaan negatif diprediksi negatif, dan FN adalah jika keadaan positif diprediksi negatif. pada kasus ini, keadaan positif adalah penderita mengalami diabetes (bernilai 1) dan sebaliknya.



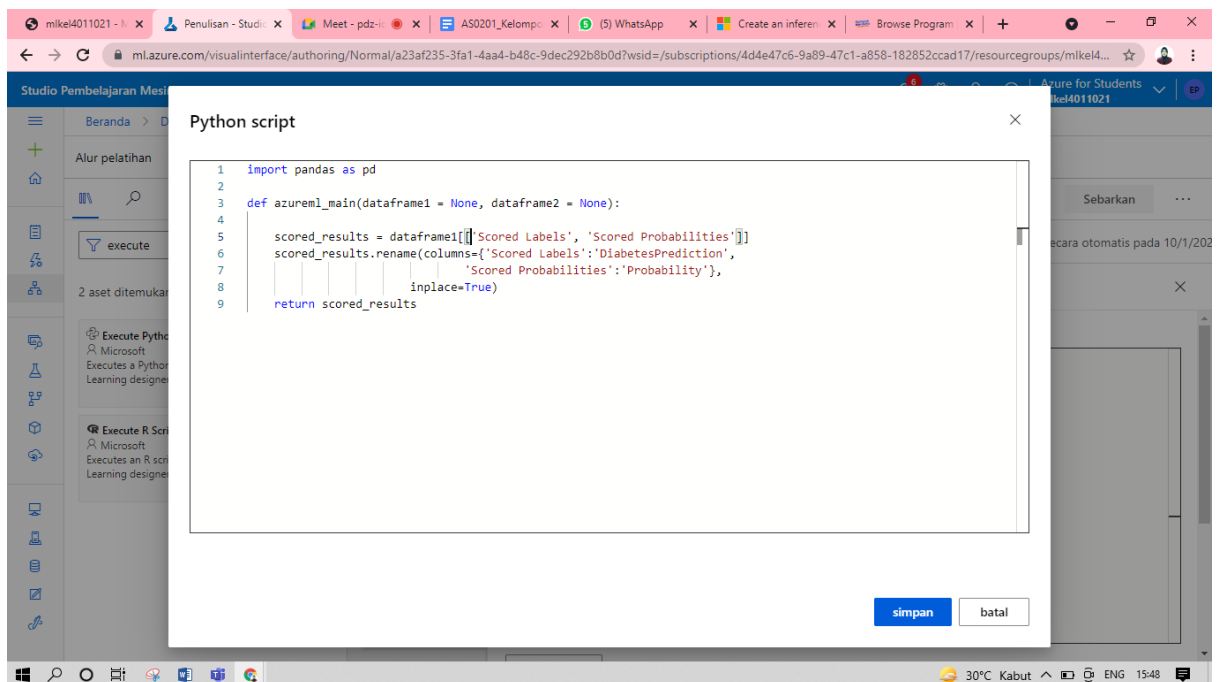
23. grafik ROC digunakan untuk menampilkan kinerja algoritma klasifikasi dalam bentuk grafik. Kurva ROC dibuat berdasar confusion matrix (lihat gambar no 24)



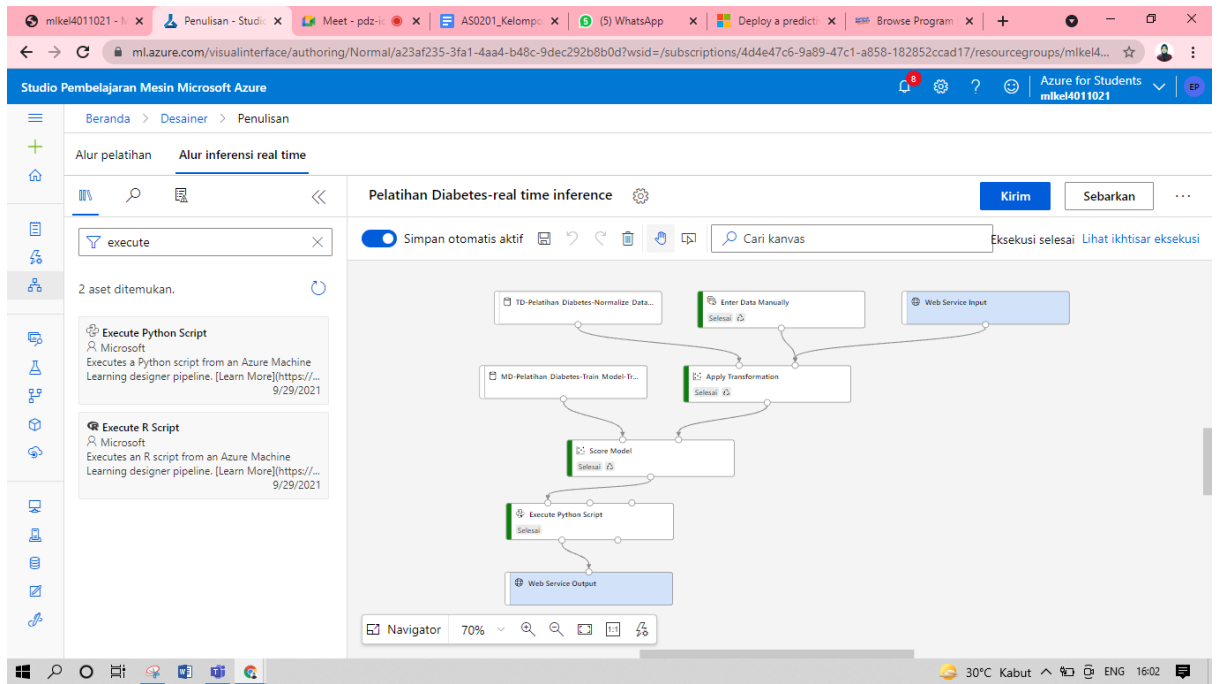
24. Setelah itu buat interface pipeline, lalu pilih real time interface. Setelah itu maka akan muncul beberapa modul baru. Disini hilangkan modul dataset awal kita dan evaluate model lalu tambahkan module enter data manually dan execute python script



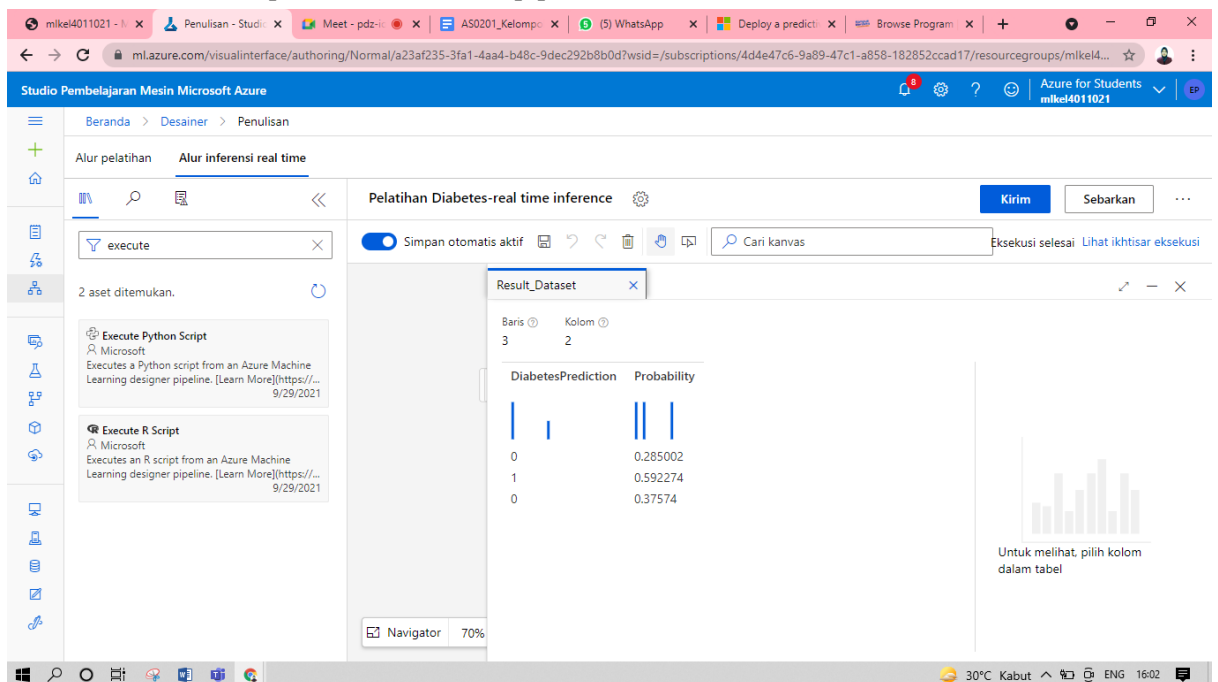
25. masukkan kode ke dalam box Data berdasarkan soal nomor 5



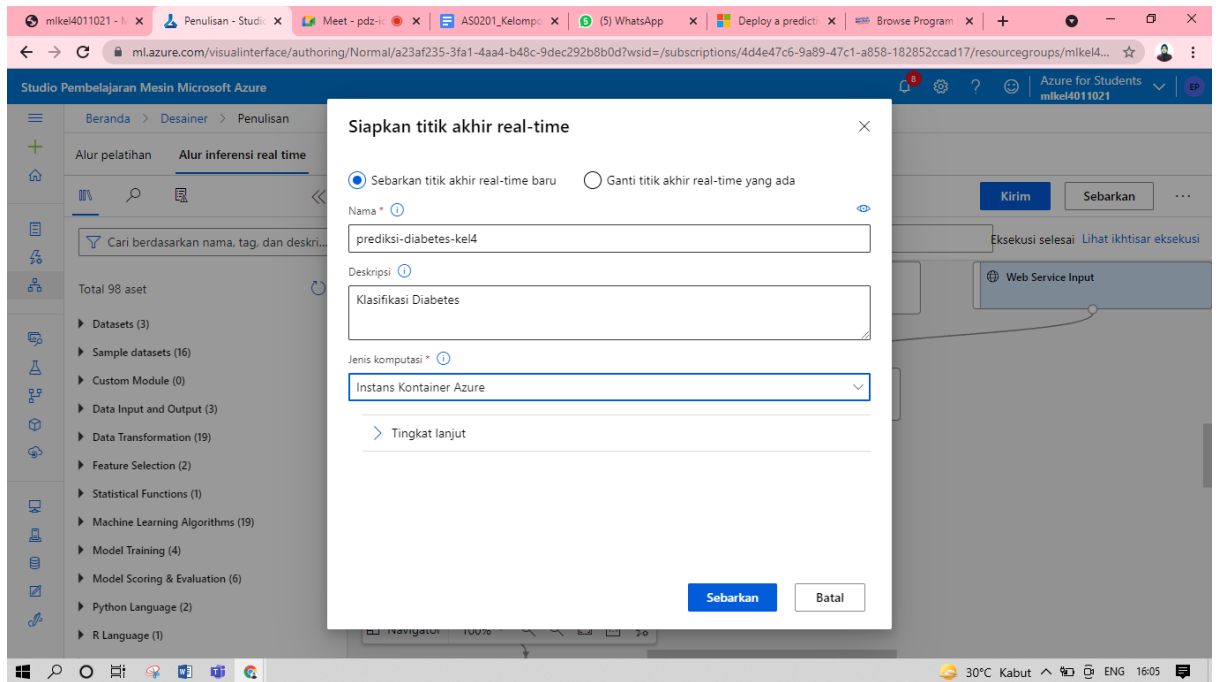
26. masukkan script python pada module execute python script. hilangkan patient ID karena dari awal, kolom pasien ID tidak digunakan



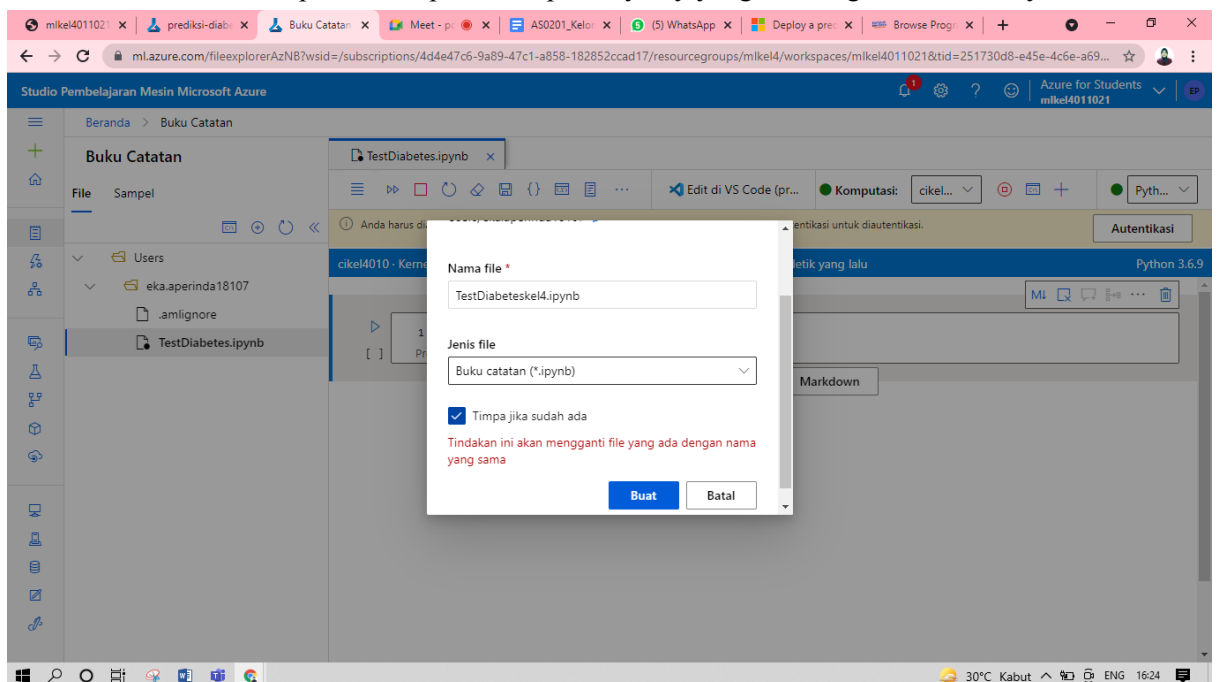
27. Hasil eksekusi setelah pembuatan interface pipeline.



28. Berikut visualisasi result dataset dari inference pipeline yang terdapat pada execute python script >> output+logs >> visualisasi result dataset. terlihat bahwa dari 3 pasien, pasien pertama diprediksikan tidak mengalami diabetes, begitupula dengan pasien ketiga. sedangkan, pasien kedua diprediksi mengalami diabetes.



29. Setelah itu deploy endpoint. Beri nama endpoint dan deskripsi, pilih jenis komputasi instans kontainer azure. Setelah itu deploy. Setelah selesai deploy, klik tab endpoints, lalu klik consume, maka akan tampil rest endpoint dan primary key yang akan digunakan nantinya



30. langkah selanjutnya adalah membuat notebook untuk menjalankan code.

```
1 endpoint = 'http://4dabf768-91e6-4875-bfab-4c13c7d2d7eb.southeastasia.azurecontainer.io/score' #Replace with your en
2 key = 'RWEjGkeertDD0ko7hz4nz8kCz0M0bd80' #Replace with your key
3
4 import urllib.request
5 import json
6 import os
7
8 data = {
9     "Inputs": {
10         "WebServiceInput0":
11             [
12                 {
13                     'Pregnancies': 4,
14                     'PlasmaGlucose': 115,
15                     'DiastolicBloodPressure': 50,
16                     'TricepsThickness': 29,
17                     'SerumInsulin': 243,
18                     'BMI': 34.69,
19                     'DiabetesPedigree': 0.741159962,
20                     'Age': 59,
21                 },
22             ],
23     }
```

31. Menjalankan kode untuk melakukan request HTTP ke endpoint yang telah diperoleh dari service yang telah dibuat pada langkah sebelumnya

```
28 body = str.encode(json.dumps(data))
29
30 headers = {'Content-Type': 'application/json', 'Authorization': ('Bearer ' + key)}
31 req = urllib.request.Request(endpoint, body, headers)
32
33 try:
34     response = urllib.request.urlopen(req)
35     result = response.read()
36     json_result = json.loads(result)
37     output = json_result["Results"]["WebServiceOutput0"][0]
38     print("\nPrediction: {} \nProbability: {:.2f}".format(output["DiabetesPrediction"],
39                                                         output["Probability"]))
40 except urllib.error.HTTPError as error:
41     print("The request failed with status code: " + str(error.code))
42
43     # Print the headers to help debug
44     print(error.info())
45     print(json.loads(error.read().decode("utf8", 'ignore')))
```

[1] ✓ <1 sec

...

Prediction: 1.0
Probability: 0.66

32. Setelah di run maka akan muncul prediksi dan probabilitas

```
[1] ✓ <1 sec
...
Prediction: 1.0
Probability: 0.66
```

33. Pada kasus ini, kita memeriksa seorang pasien dengan umur 59 tahun dengan data-data seperti di atas dan diperiksa dengan ML yang telah kita buat. Setelah di prediksi orang tersebut kemungkinan terkena penyakit diabetes dengan hasil dari probabilitasnya adalah 0.66 dimana berada di atas 0.5 dan diprediksikan 1.0