**1.** Pertanyaan #1

Which of the following are correct about the Extract, Transform, Load (ETL) procedure?

A picture containing text, screenshot, font

Description automatically generated

**2.** Pertanyaan #2

What does the following code block achieve?

tfds.load(name="mnist", split="train")

**.**

Pertanyaan #2

What does the following code block achieve?

1

tfds.load(name="mnist", split="train")

**1 / 1 poin**



Splits the downloaded mnist data into train and test sets



Downloads and extracts training records from the mnist dataset



Extracts the mnist training dataset from a zip file



Loads mnist labels and assign them to any training dataset

**Benar**

Correct!

It downloads the dataset (if not already stored in your local TensorFlow directory) and ***split="train"*** parameter tells it to return only training records

**3.** Pertanyaan #3

Can you explore 10 records from the dataset by loading them into an iterator like this?

iterator = dataset.take(10)

**1 / 1 poin**



Yes



No

**Benar**

Correct!

take() method allows you to select the n examples from the dataset, where n is a passed as a parameter.

**4.** Pertanyaan #4

What is the role of the tfds.list\_builders() function?

**1 / 1 poin**



To return a list of files in the dataset



To build a list of multiple datasets to load at a time



To create an empty dataset for creating a custom dataset



To return string names of all available datasets in Tensorflow

**Benar**

Correct!

tfds.list\_builders() returns the string names of all  [tfds.core.DatasetBuilder](https://www.tensorflow.org/datasets/api_docs/python/tfds/core/DatasetBuilder) which is the baseclass defined to handle all datasets present in Tensorflow APIs.

**5.** Pertanyaan #5

How would you inspect the metadata and the details of a TensorFlow dataset?

**1 / 1 poin**



Load the data using tfds.load() with the parameter as\_supervised=False , and then inspect the DataSetInfo property



There’s no API for this, read the docs instead



Load the data using tfds.load() with the parameter with\_info=true, and then inspect the showCoreData property.



Load the data using tfds.load() with the parameter with\_info=true, and then inspect the DataSetInfo property

**Benar**

Correct!

DatasetInfo documents datasets, including its name, version, and features. ***with\_info=true*** is the parameter to pass in tfds.load() to get the metadata.

**6.** Pertanyaan #6

Which of the following ways are used to load mnist dataset with major version 1, minor version 2 and any patch version ?

**1 / 1 poin**



Specify the exact version for a patch version in the load parameter like this: ***tfds.load(“mnist:1.2.1”)***

**Benar**

This also works as it loads specifically the version 1.2.1. If it is not pre-cached, it gets downloaded, extracted and loaded even if there are different versions like 1.2.2 present in your TensorFlow installation folder.



You’ll need to install the matching version of TFDS that installs that dataset, and then load it



Specify the desired version with asterisk in patch version in the string in the load parameter like this:

***tfds.load(“mnist”).version(“1.2.\*”)***

**Benar**

The asterisk helps in identifying any dataset with 1.2.x as a version meaning any patch version is identified



Specify the desired version as a split, like this:

***tfds.load(name=”mnist”, split=”1.\*.\*”)***

**7.** Pertanyaan #7

The fashion MNIST is a relatively simple example of a dataste used in computer vision modelling tasks used with or without TFDS. If you load the data using TensorFlow Keras datasets in TensorFlow 2.0 and above, what would the code look like?

data = tf.keras.dataset.fashion\_mnist

(training\_images,training\_labels),(test\_images,test\_labels) = data.load\_data()

A screenshot of a computer

Description automatically generated with medium confidence

**8.** Pertanyaan #8

Which of the following code blocks would successfully create "Horses and Humans" test batches of 10 by shuffling 100 data samples?

data = tf.keras.dataset.fashion\_mnist

(training\_images,training\_labels),(test\_images,test\_labels) = data.load\_data()

A picture containing text, font, screenshot, line

Description automatically generated