

# **SCRAPING OLX**

**"Mobil Bekas"**

Habil Nasution

2108107010089

# Tahapan Scraping OLX

01

Membuat  
environment

02

Menginstal  
Scrupy

03

Melakukan  
Scraping

04

Hasil  
Scraping

# 1. Membuat Environtment

```
noitanusx@noitanusx:~/studies$ python -m venv env-bigdata-course  
noitanusx@noitanusx:~/studies$ source env-bigdata-course/bin/activate  
(env-bigdata-course) noitanusx@noitanusx:~/studies$
```

Membuat environment dan mengaktifkan Python virtual environment

## 2. Menginstal Scrapy

```
(env-bigdata-course) noitusanx@noitusanx:~/studies$ mkdir scrapping-olx  
noitusanx@noitusanx:~/studies$ source env-bigdata-course/bin/activate  
(env-bigdata-course) noitusanx@noitusanx:~/studies/scrapping-olx$ pip install scrapy
```

Membuat direktori scrapping-olx dan melakukan instalasi scrapy

```
(env-bigdata-course) noitusanx@noitusanx:~/studies/scrapping-olx$ scrapy startproject scrapping
```

Membuat project Scrapy baru yang berisi kumpulan kode dan sumber daya yang digunakan Scrapy untuk menjelajahi situs web dan mengekstrak data

### 3. Melakukan Scraping

The screenshot shows a web browser displaying the OLX mobile website for used cars. The URL is [https://www.olx.co.id/mobil-bekas\\_c198](https://www.olx.co.id/mobil-bekas_c198). The search bar indicates "Selatan, Jakarta" and the filter "Hanya di Mobil Bekas" is selected. Two car listings are visible:

- Honda BR-V** - **Rp 148.000.000** (2016 - 80.000-85.000 km)
- Toyota Agya** - **Rp 131.000.000** (2021 - 40.000-45.000 km)

The DOM code for the first listing is shown below:

```
<li class=" _3V_Ww" data-aut-id="itemBox" data-aut-category-id="198"> event<br/><a class="" href="/item/olx-autos-tdp-15jt-honda-br-v-15-e-bensin-mt-2016-putih-metalik-iid-910790227"> event flex<br/><div class=" _1HlM1"><br/><figure class=" _3UrC5" data-aut-id="itemImage">...</figure><br/><div class=" _3OP3g">...</div> flex<br/><div class=" _2v8Tq"> flex<br/><span class=" _1zgtX" data-aut-id="itemPrice">Rp 148.000.000</span><br/><div class=" _21gnE" title="2016 - 80.000-85.000 km" data-aut-id="itemSubTitle">2016 - 80.000-85.000 km</div><br/><div class=" _2Gr10" title="[OLX AUTOS] TDP 15JT Honda BR-V 1.5 E Bensin-MT 2016 Putih Metalik" data-aut-id="itemTitle">Honda BR-V</div><br/><div class=" _3VRSm" data-aut-id="itemDetails">...</div> flex
```

Melakukan inspect pada situs web olx mobil bekas untuk mendapatkan class dari data-data yang diperlukan seperti data harga, tahun, kilometer dan brand dari mobil bekas

### 3. Melakukan Scraping



```
scrapping > scrapping > items.py > MyItem
1 # Define here the models for your scraped items
2 #
3 # See documentation in:
4 # https://docs.scrapy.org/en/latest/topics/items.html
5
6 import scrapy
7
8
9 class MyItem(scrapy.Item):
10     item_title = scrapy.Field()
11     item_subtitle = scrapy.Field()
12     item_price = scrapy.Field()
```

Mendefinisikan field yang akan diambil yang berfungsi untuk menyimpan data yang diambil di situs web

### 3. Melakukan Scraping

The screenshot shows a code editor with a dark theme. On the left is a file tree for a project named 'SCRAPPING-OLX'. The tree includes 'scrapping' (containing '\_pycache\_'), 'spiders' (containing '\_pycache\_'), and several files like '\_init\_.py', 'items.py', 'middlewares.py', 'pipelines.py', 'settings.py', and 'scrapy.cfg'. The right side of the editor displays a Python script named 'MySpider.py'. The script starts with comments about the package containing spiders and the documentation. It imports 'scrapy' and 'MyItem' from the 'scrapping.items' module. The spider class 'MySpider' extends 'scrapy.Spider' and is named 'myspider'. It has a single method 'parse' which takes a response object. Inside 'parse', it finds all list items containing a specific class ('\_3V\_Ww') using XPath. Then, for each item, it extracts the title, subtitle, and price by navigating through nested divs and spans using XPath expressions. Finally, it yields the extracted item.

```
scrapping > scrapping > spiders > __init__.py > MySpider > parse
1 # This package will contain the spiders of your Scrapy project
2 #
3 # Please refer to the documentation for information on how to create and manage
4 # your spiders.
5 import scrapy
6 from scrapping.items import MyItem
7
8 class MySpider(scrapy.Spider):
9     name = 'myspider'
10    start_urls = ['https://www.olx.co.id/mobil-bekas_c198']
11
12    def parse(self, response):
13        item_boxes = response.xpath('//li[contains(@class, "_3V_Ww")]')
14        for item_box in item_boxes:
15            item = MyItem()
16            item['item_title'] = item_box.xpath('.//div[@class="2Gr10"]/text()').get().strip()
17            item['item_subtitle'] = item_box.xpath('.//div[@class="21gnE"]/text()').get().strip()
18            item['item_price'] = item_box.xpath('.//span[@class="1zgtX"]/text()').get().strip()
19            yield item
```

Mengekstrak informasi daftar mobil bekas di situs olx yang terdiri dari brand, tahun, kilometer dan harga dari mobil bekas tersebut

# 3. Melakukan Scraping

```
(env-bigdata-course) noitusanx@noitusanx:~/studies/scrapping-olx/scrapping$ scrapy crawl myspider -o scrapping.json
2023-11-06 22:17:17 [scrapy.utils.log] INFO: Scrapy 2.11.0 started (bot: scrapping)
2023-11-06 22:17:17 [scrapy.utils.log] INFO: Versions: lxml 4.9.3.0, libxml2 2.10.3, cssselect 1.2.0, parsel 1.8.1, w3lib 2.1.2,
SSL 23.3.0 (OpenSSL 3.1.4 24 Oct 2023), cryptography 41.0.5, Platform Linux-6.2.0-36-generic-x86_64-with-glibc2.35
2023-11-06 22:17:17 [scrapy.addons] INFO: Enabled addons:
[]
2023-11-06 22:17:17 [asyncio] DEBUG: Using selector: EpollSelector
2023-11-06 22:17:17 [scrapy.utils.log] DEBUG: Using reactor: twisted.internet.asyncioreactor.AsyncioSelectorReactor
2023-11-06 22:17:17 [scrapy.utils.log] DEBUG: Using asyncio event loop: asyncio_unix_events._UnixSelectorEventLoop
2023-11-06 22:17:17 [scrapy.extensions.telnet] INFO: Telnet Password: b4b712a65f9bf8db
2023-11-06 22:17:17 [scrapy.middleware] INFO: Enabled extensions:
['scrapy.extensions.corestats.CoreStats',
 'scrapy.extensions.telnet.TelnetConsole',
 'scrapy.extensions.memusage.MemoryUsage',
 'scrapy.extensions.feedexport.FeedExporter',
 'scrapy.extensions.logstats.LogStats']
2023-11-06 22:17:17 [scrapy.crawler] INFO: Overridden settings:
{'BOT_NAME': 'scrapping',
 'FEED_EXPORT_ENCODING': 'utf-8',
 'NEWSPIDER_MODULE': 'scrapping.spiders',
 'REQUEST_FINGERPRINTER_IMPLEMENTATION': '2.7',
 'ROBOTSTXT_OBEY': True,
 'SPIDER_MODULES': ['scrapping.spiders'],
 'TWISTED_REACTOR': 'twisted.internet.asyncioreactor.AsyncioSelectorReactor'}
2023-11-06 22:17:17 [scrapy.middleware] INFO: Enabled downloader middlewares:
['scrapy.downloadermiddlewares.robotstxt.RobotsTxtMiddleware',
 'scrapy.downloadermiddlewares.httpauth.HttpAuthMiddleware',
```

Menjelajahi situs web menggunakan Scrapy dan menyimpan hasilnya dalam bentuk JSON

# 4. Hasil Scraping

```
ps scrapping.json  
(env-bigdata-course) noitusanx@noitusanx:~/studies/scrapping-olx/scrapping$ cat scrapping.json  
[  
{"item_title": "Nissan X-Trail", "item_subtitle": "2017 - 105.000-110.000 km", "item_price": "Rp 225.000.000"},  
 {"item_title": "Wuling Almaz", "item_subtitle": "2019 - 25.000-30.000 km", "item_price": "Rp 204.000.000"},  
 {"item_title": "Nissan Livina", "item_subtitle": "2019 - 60.000-65.000 km", "item_price": "Rp 198.000.000"},  
 {"item_title": "Wuling Cortez", "item_subtitle": "2019 - 25.000-30.000 km", "item_price": "Rp 152.000.000"},  
 {"item_title": "Toyota Yaris", "item_subtitle": "2019 - 45.000-50.000 km", "item_price": "Rp 212.000.000"},  
 {"item_title": "Nissan X-Trail", "item_subtitle": "2015 - 135.000-140.000 km", "item_price": "Rp 177.000.000"},  
 {"item_title": "Nissan Serena", "item_subtitle": "2013 - 80.000-85.000 km", "item_price": "Rp 152.000.000"},  
 {"item_title": "Suzuki Ertiga", "item_subtitle": "2018 - 130.000-135.000 km", "item_price": "Rp 180.000.000"},  
 {"item_title": "Mercedes-Benz E300", "item_subtitle": "2011 - 80.000-85.000 km", "item_price": "Rp 295.000.000"},  
 {"item_title": "Daihatsu Rocky", "item_subtitle": "2021 - 30.000-35.000 km", "item_price": "Rp 203.000.000"},  
 {"item_title": "Toyota Kijang Innova", "item_subtitle": "2019 - 60.000-65.000 km", "item_price": "Rp 342.000.000"},  
 {"item_title": "Toyota Yaris", "item_subtitle": "2017 - 65.000-70.000 km", "item_price": "Rp 176.000.000"},  
 {"item_title": "Honda CR-V", "item_subtitle": "2019 - 40.000-45.000 km", "item_price": "Rp 375.000.000"},  
 {"item_title": "Toyota Yaris", "item_subtitle": "2021 - 95.000-100.000 km", "item_price": "Rp 247.000.000"},  
 {"item_title": "BMW 320i", "item_subtitle": "2016 - 40.000-45.000 km", "item_price": "Rp 317.000.000"},  
 {"item_title": "Honda HR-V", "item_subtitle": "2018 - 40.000-45.000 km", "item_price": "Rp 262.000.000"},  
 {"item_title": "Honda HR-V", "item_subtitle": "2018 - 40.000-45.000 km", "item_price": "Rp 262.000.000"},  
 {"item_title": "Toyota Yaris", "item_subtitle": "2015 - 100.000-105.000 km", "item_price": "Rp 160.000.000"},  
 {"item_title": "Mitsubishi Pajero", "item_subtitle": "2014 - 120.000-125.000 km", "item_price": "Rp 250.000.000"},  
 {"item_title": "Mitsubishi Xpander", "item_subtitle": "2019 - 5.000-10.000 km", "item_price": "Rp 196.000.000"},  
 {"item_title": "Daihatsu Xenia", "item_subtitle": "2015 - 155.000-160.000 km", "item_price": "Rp 106.500.000"},  
 {"item_title": "Daihatsu Ayla", "item_subtitle": "2017 - 50.000-55.000 km", "item_price": "Rp 102.000.000"},  
 {"item_title": "Honda Mobilio", "item_subtitle": "2014 - 120.000-125.000 km", "item_price": "Rp 124.000.000"}  
(env-bigdata-course) noitusanx@noitusanx:~/studies/scrapping-olx/scrapping$ wc -l scrapping.json  
41 scrapping.json
```

Menampilkan hasil dan jumlah data scraping