



UTS TEMU KEMBALI INFORMASI

MODIFIKASI SEARCH ENGINE DARI TUGAS I

220411100013

MISHBAHUS SURUR

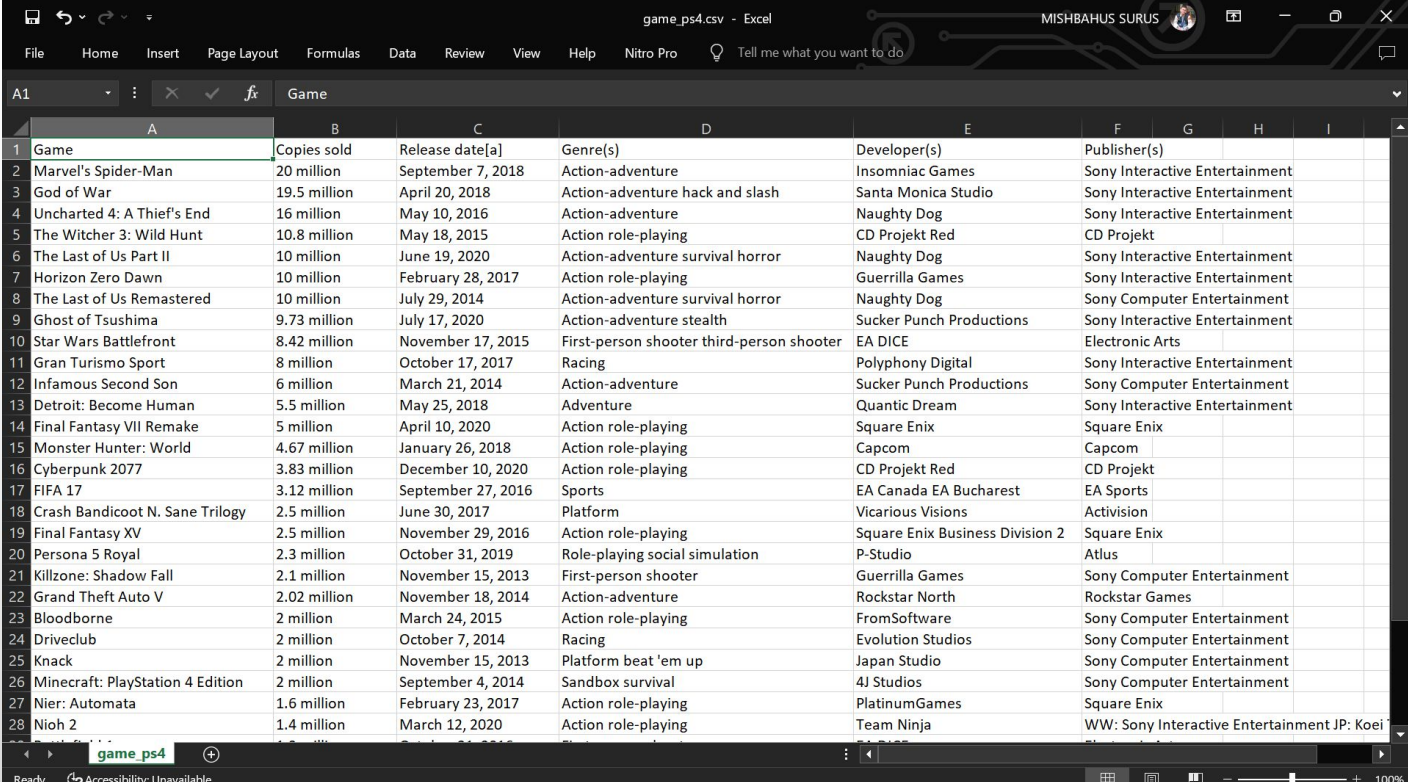


LANGKAH – LANGKAH

- 1. MENCARI DATASET (BISA DARI KAGGLE ATAU DARI WEBSITE YANG MENYEDIAKAN DATASET)
- 2. MEMBUAT DATA .JSON DARI DATASET YANG SUDAH DIDAPATKAN
- 3. MEMODIFIKASI TF-IDF.PY PADA TUGAS 1
- MEMODIFIKASI QUERY.PY PADA TUGAS 1
- MEMODIFIKASI FOLDER LARAVEL PADA TUGAS 1

MENCARI DAN MENYIAPKAN DATASET

- DATASET BISA DIDAPATKAN DARI WEBSITE YANG MENYEDIAKAN DATASET, BISA DARI DATA.GO.ID, KAGGLE, DAN LAINNNYA
- SAYA MENDAPATKAN DATASET DARI KAGGLE YANG BEREKSTENSI .CSV
- DATASET YANG SAYA GUNAKAN ADALAH GAME_PS4.CSV



The screenshot shows an Excel spreadsheet titled 'game_ps4.csv' with the following data:

Game	Copies sold	Release date[a]	Genre(s)	Developer(s)	Publisher(s)
Marvel's Spider-Man	20 million	September 7, 2018	Action-adventure	Insomniac Games	Sony Interactive Entertainment
God of War	19.5 million	April 20, 2018	Action-adventure hack and slash	Santa Monica Studio	Sony Interactive Entertainment
Uncharted 4: A Thief's End	16 million	May 10, 2016	Action-adventure	Naughty Dog	Sony Interactive Entertainment
The Witcher 3: Wild Hunt	10.8 million	May 18, 2015	Action role-playing	CD Projekt Red	CD Projekt
The Last of Us Part II	10 million	June 19, 2020	Action-adventure survival horror	Naughty Dog	Sony Interactive Entertainment
Horizon Zero Dawn	10 million	February 28, 2017	Action role-playing	Guerrilla Games	Sony Interactive Entertainment
The Last of Us Remastered	10 million	July 29, 2014	Action-adventure survival horror	Naughty Dog	Sony Computer Entertainment
Ghost of Tsushima	9.73 million	July 17, 2020	Action-adventure stealth	Sucker Punch Productions	Sony Interactive Entertainment
Star Wars Battlefront	8.42 million	November 17, 2015	First-person shooter third-person shooter	EA DICE	Electronic Arts
Gran Turismo Sport	8 million	October 17, 2017	Racing	Polyphony Digital	Sony Interactive Entertainment
Infamous Second Son	6 million	March 21, 2014	Action-adventure	Sucker Punch Productions	Sony Computer Entertainment
Detroit: Become Human	5.5 million	May 25, 2018	Adventure	Quantic Dream	Sony Interactive Entertainment
Final Fantasy VII Remake	5 million	April 10, 2020	Action role-playing	Square Enix	Square Enix
Monster Hunter: World	4.67 million	January 26, 2018	Action role-playing	Capcom	Capcom
Cyberpunk 2077	3.83 million	December 10, 2020	Action role-playing	CD Projekt Red	CD Projekt
FIFA 17	3.12 million	September 27, 2016	Sports	EA Canada EA Bucharest	EA Sports
Crash Bandicoot N. Sane Trilogy	2.5 million	June 30, 2017	Platform	Vicarious Visions	Activision
Final Fantasy XV	2.5 million	November 29, 2016	Action role-playing	Square Enix Business Division 2	Square Enix
Persona 5 Royal	2.3 million	October 31, 2019	Role-playing social simulation	P-Studio	Atlus
Killzone: Shadow Fall	2.1 million	November 15, 2013	First-person shooter	Guerrilla Games	Sony Computer Entertainment
Grand Theft Auto V	2.02 million	November 18, 2014	Action-adventure	Rockstar North	Rockstar Games
Bloodborne	2 million	March 24, 2015	Action role-playing	FromSoftware	Sony Computer Entertainment
Driveclub	2 million	October 7, 2014	Racing	Evolution Studios	Sony Computer Entertainment
Knack	2 million	November 15, 2013	Platform beat 'em up	Japan Studio	Sony Computer Entertainment
Minecraft: PlayStation 4 Edition	2 million	September 4, 2014	Sandbox survival	4J Studios	Sony Computer Entertainment
Nier: Automata	1.6 million	February 23, 2017	Action role-playing	PlatinumGames	Square Enix
Nioh 2	1.4 million	March 12, 2020	Action role-playing	Team Ninja	WW: Sony Interactive Entertainment JP: Koei

MERUBAH DATA .CSV MENJADI .JSON

- MERUBAH DARI EKSTENSI .CSV MENJADI EKSTENSI .JSON
- DENGAN MENGGUNAKAN KODE DISAMPING AKAN MENDAPATKAN OUTPUT DATA.JSON (FILE BARU), YANG ISINYA BERUPA DATA DARI DATASET GAME_PAS4.CSV.

```
convert_csv-json.py > ...
1  import csv
2  import json
3
4  csv_file = open("game_ps4.csv", 'r')
5  csv_reader = csv.reader(csv_file, delimiter=';')
6
7  field_names = next(csv_reader)
8  print(field_names)
9
10 data = []
11
12 for row in csv_reader:
13     data.append(dict(zip(field_names, row)))
14
15 print(data)
16
17
18 json_data = json.dumps(data)
19
20 json_file = open("data.json", 'w')
21 json_file.write(json_data)
22
23 csv_file.close()
24 json_file.close()
25
26
```

MERUBAH DATA .CSV MENJADI .JSON

- DATA.JSON BERHASIL DIDAPATKAN SETELAH MENJALANKAN KODE DIATAS
- NAMUN ADA SEDIKIT KEBINGUNGAN DALAM KOLOM GAMENYA

```
{ data.json > ...
1  [
2  {
3    "\u00ef\u00bb\u00bfGame": "Marvel's Spider-Man",
4    "Copies sold": "20 million",
5    "Release date[a]": "September 7, 2018",
6    "Genre(s)": "Action-adventure",
7    "Developer(s)": "Insomniac Games",
8    "Publisher(s)": "Sony Interactive Entertainment"
9  },
10 {
11   "\u00ef\u00bb\u00bfGame": "God of War",
12   "Copies sold": "19.5 million",
13   "Release date[a]": "April 20, 2018",
14   "Genre(s)": "Action-adventure hack and slash",
15   "Developer(s)": "Santa Monica Studio",
16   "Publisher(s)": "Sony Interactive Entertainment"
17 },
18 {
19   "\u00ef\u00bb\u00bfGame": "Uncharted 4: A Thief's End",
20   "Copies sold": "16 million",
21   "Release date[a]": "May 10, 2016",
22   "Genre(s)": "Action-adventure",
23   "Developer(s)": "Naughty Dog",
24   "Publisher(s)": "Sony Interactive Entertainment"
25 },
26 {
27   "\u00ef\u00bb\u00bfGame": "The Witcher 3: Wild Hunt",
28   "Copies sold": "10.8 million",
29   "Release date[a]": "May 18, 2015",
30   "Genre(s)": "Action role-playing",
31   "Developer(s)": "CD Projekt Red",
32   "Publisher(s)": "CD Projekt"
33 },
34 {
35   "\u00ef\u00bb\u00bfGame": "The Last of Us Part II",
36   "Copies sold": "10 million",
37   "Release date[a]": "June 19, 2020",
38   "Genre(s)": "Action-adventure survival horror",
39   "Developer(s)": "Naughty Dog",
40   "Publisher(s)": "Sony Interactive Entertainment"
41 }
```


MERUBAH DATA .CSV MENJADI .JSON

- MELAKUKAN FILTER LAGI DENGAN KODE YANG ADA DI FILTER.PY
- DAN MENDAPATKAN FILE BARU LAGI YANG DINAMAKAN GAME.JSON

```

1  import json
2
3  def filter_json(input,output):
4
5      with open(input,'r') as f:
6          data = json.load(f)
7
8
9          filtered = []
10         for item in data:
11             filtered_item = {
12                 "Game" : item["\u00ef\u00bb\u00bfGame"],
13                 "Copies_Sold" : item["Copies sold"],
14                 "Realease_Date" : item["Release date[a]"],
15                 "Genre" : item["Genre(s)"],
16                 "Developer" : item["Developer(s)"],
17                 "Publisher" : item["Publisher(s)"]
18             }
19             filtered.append(filtered_item)
20
21
22         with open(output,'w') as f:
23             json.dump(filtered,f,indent=2)
24
25
26     filter_json("data.json","game.json")
27

```

```

1  [
2  {
3      "Game": "Marvel's Spider-Man",
4      "Copies_Sold": "20 million",
5      "Realease_Date": "September 7, 2018",
6      "Genre": "Action-adventure",
7      "Developer": "Insomniac Games",
8      "Publisher": "Sony Interactive Entertainment"
9  },
10 {
11     "Game": "God of War",
12     "Copies_Sold": "19.5 million",
13     "Realease_Date": "April 20, 2018",
14     "Genre": "Action-adventure hack and slash",
15     "Developer": "Santa Monica Studio",
16     "Publisher": "Sony Interactive Entertainment"
17 },
18 {
19     "Game": "Uncharted 4: A Thief's End",
20     "Copies_Sold": "16 million",
21     "Realease_Date": "May 10, 2016",
22     "Genre": "Action-adventure",
23     "Developer": "Naughty Dog",
24     "Publisher": "Sony Interactive Entertainment"
25 },
26 {
27     "Game": "The Witcher 3: Wild Hunt",
28     "Copies_Sold": "10.8 million",
29     "Realease_Date": "May 18, 2015",
30     "Genre": "Action role-playing",
31     "Developer": "CD Projekt Red",
32     "Publisher": "CD Projekt"
33 },
34 {
35     "Game": "The Last of Us Part II",
36     "Copies_Sold": "10 million",
37     "Realease_Date": "June 19, 2020",
38     "Genre": "Action-adventure survival horror",
39     "Developer": "Naughty Dog",
40     "Publisher": "Sony Interactive Entertainment"
41 },

```

MELAKUKAN MODIFIKASI PADA TF-IDF.PY

- ADA SEDIKIT MODIFIKASI PADA FILE TF-IDF.PY
- MELAKUKAN INDEXING
- MENENTUKAN KOLOM MANA YANG AKAN DI GUNAKAN UNTUK DILAKUKANNYA PENCARIAN

```
for data in content:
    tf = {}
    # Clean and list word
    clean_title = clean_str(data['Game'])
    list_word = clean_title.split(" ")

    for word in list_word:
        if word in sw:
            continue

        # TF term frequency
        if word in tf:
            tf[word] += 1
        else:
            tf[word] = 1

        # DF document frequency
        if word in df_data:
            df_data[word] += 1
        else:
            df_data[word] = 1

    tf_data[data['Game']] = tf
```

```
# Calculate IDF
total_docs = len(content)
for word, doc_freq in df_data.items():
    idf_data[word] = 1 + math.log10(total_docs / doc_freq)

tf_idf = {}

for word in df_data:
    list_doc = []
    for data in content:
        tf_value = 0

        if word in tf_data[data['Game']]:
            tf_value = tf_data[data['Game']][word]

        weight = tf_value * idf_data[word]

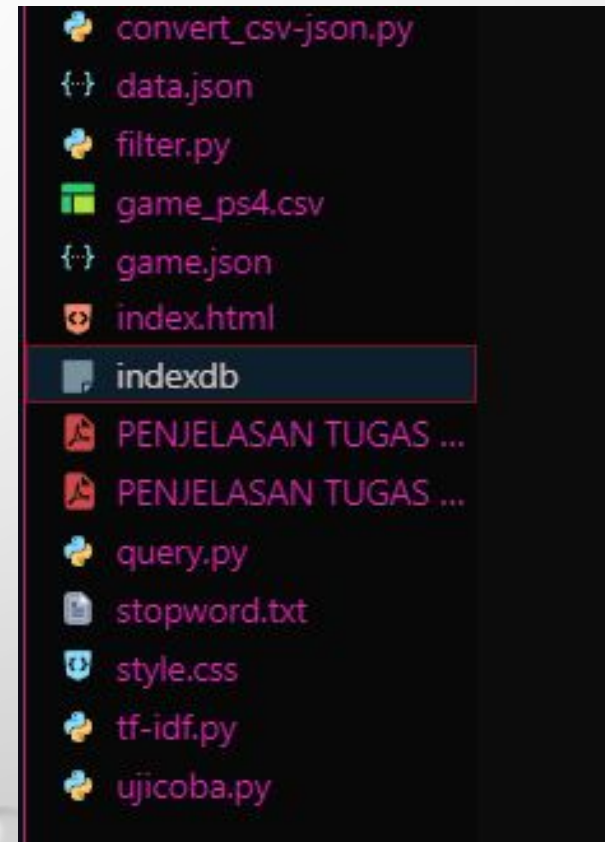
        doc = {
            'Game': data['Game'], # Menggunakan nama game se
            'Copies_Sold': data['Copies_Sold'], # Menggunakan
            'Realease_Date': data['Realease_Date'],
            'Genre': data['Genre'],
            'Developer': data['Developer'],
            'Publisher': data['Publisher'],
            'score': weight # Bobot TF-IDF
        }

        if doc['score'] != 0:
            if doc not in list_doc:
                list_doc.append(doc)

    tf_idf[word] = list_doc
```

MELAKUKAN MODIFIKASI PADA TF-IDF.PY

- MEMASUKKAN PERINTAH “PYTHON TF-IDF.PY GAME.JSON INDEXDB” DI TERMINAL
- PYTHON = MENGGUNAKAN PYTHON
- TF-IDF.PY = FILE YANG AKAN DIJALANKAN
- GAME.JSON = FILE YANG AKAN DIHUBUNGKAN
- INDEXDB = DATABASE YANG DIGUNAKAN UNTUK MENYIMPAN NILAI DARI DATA GAME.JSON
- OUTPUT TERSEBUT AKAN MEMBUAT FILE BARU YAITU INDEXDB



MELAKUKAN MODIFIKASI PADA FILE QUERY.PY

- DIGUNAKAN UNTUK MELAKUKAN PENCARIAN
- MENGGUNAKAN PERINAH “PYTHON QUERY.PY INDEXDB 3 THE” PADA TERMINAL
- QUERY.PY = FILE YANG AKAN DIJALANKAN
- INDEXDB = FILE DATABASE YANG AKAN DIGUNAKAN
- 3 = JIKA BERHASIL MENCARI DITAMPILKAN SEBANYAK 3 DATA
- THE = KATA YANG AKAN DICARI (YANG ADA KATA

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\smt 4\Temu kemabali Informasi\UTS> python query.py indexdb 3 the
{"Game": "The Witcher 3: Wild Hunt", "Copies_Sold": "10.8 million", "Realease_Date": "May 18, 2015", "Genre": "Action role-playing", "Developer": "CD Projekt Red", "Publisher": "CD Projekt", "score": 1.8195439355418688}
{"Game": "The Last of Us Part II", "Copies_Sold": "10 million", "Realease_Date": "June 19, 2020", "Genre": "Action-adventure survival horror", "Developer": "Naughty Dog", "Publisher": "Sony Interactive Entertainment", "score": 1.8195439355418688}
{"Game": "The Last of Us Remastered", "Copies_Sold": "10 million", "Realease_Date": "July 29, 2014", "Genre": "Action-adventure survival horror", "Developer": "Naughty Dog", "Publisher": "Sony Computer Entertainment", "score": 1.8195439355418688}
PS D:\smt 4\Temu kemabali Informasi\UTS> 
```

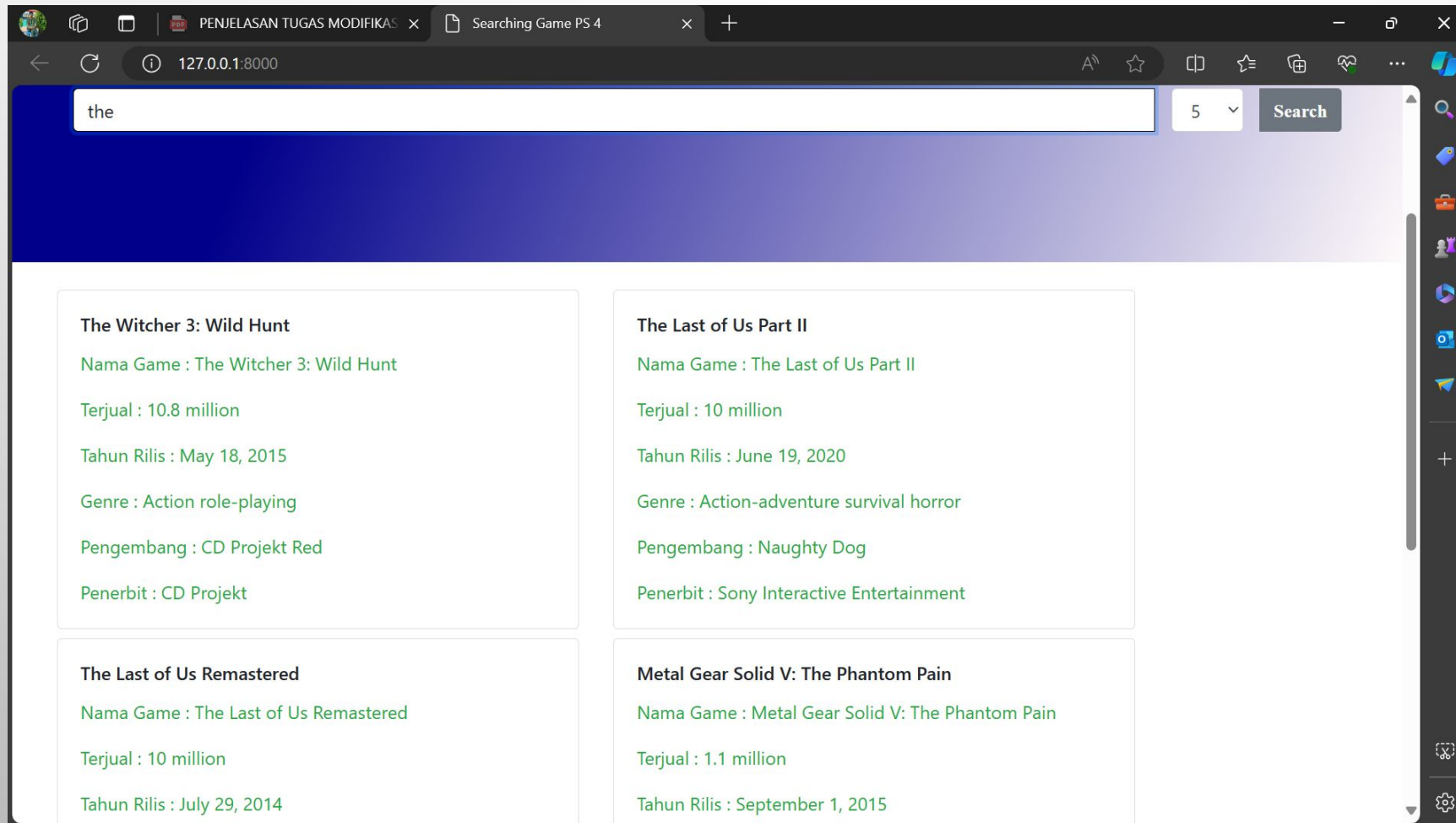
```
query.py > ...
1 import sys
2 import json
3 import pickle
4
5 # Argumen check
6 if len(sys.argv) != 4:
7     print("\nPenggunaan:\n\tquery.py [index] [n] [query]..\n")
8     sys.exit(1)
9
10 query = sys.argv[3].split(" ")
11 n = int(sys.argv[2])
12
13 with open(sys.argv[1], 'rb') as indexdb:
14     indexFile = pickle.load(indexdb)
15
16 # Query
17 list_doc = {}
18 for q in query:
19     try:
20         for doc in indexFile[q]:
21             if doc['Game'] in list_doc: # Mengganti 'url' dengan 'Game'
22                 list_doc[doc['Game']]['score'] += doc['score'] # Mengganti 'url' dengan 'Ga
23             else:
24                 list_doc[doc['Game']] = doc # Mengganti 'url' dengan 'Game'
25     except:
26         continue
27
28 # Convert to list
29 list_data = list(list_doc.values())
30
31 # Sorting List descending
32 count = 1
33 for data in sorted(list_data, key=lambda k: k['score'], reverse=True):
34     y = json.dumps(data)
35     print(y)
36     if count == n:
37         break
38     count += 1
39
```

MELAKUKAN MODIFIKASI PADA FOLDER LARAVEL

- MENGGANTI FILE INDEXDB DAN FILE QUERY.PY YANG DIGUNAKAN PADA SAAT MENGERJAKAN TUGAS 1, DAN MENGGANTINYA DENGAN FILE INDEXDB DAN QUERY.PY YANG SUDAH DIBUAT
- MEMODIFIKASI DIDALAM FILE LANDINGCONTROLLER.PHP DENGAN MENGGUNAKAN KODE DISAMPING
- MENGGUNAKAN DATA YANG SUDAH DI PERSIAPKAN DI TF-IDF YANG SUDAH DI SIMPAN DALAM INDEXDB

```
9 class LandingController extends Controller
10 {
11     public function search(Request $request)
12     {
13         // $category = Input::get('category', 'default category');
14         $query = $request->input('q');
15         $rank = $request->input('rank');
16
17         $process = new Process(['python', 'query.py', 'indexdb', $rank, $query],
18             null,
19             ['SYSTEMROOT' => getenv('SYSTEMROOT'), 'PATH' => getenv('PATH')]);
20         $process->run();
21
22         // executes after the command finishes
23         if (!$process->isSuccessful()) {
24             throw new ProcessFailedException($process);
25         }
26
27         $list_data = array_filter(explode("\n", $process->getOutput()));
28
29         $data = array();
30
31         foreach ($list_data as $book) {
32             $dataj = json_decode($book, true);
33             array_push($data, '
34                 <div class="col-lg-5">
35                     <div class="card mb-2">
36                         <div style="display: flex; flex: 1 1 auto;">
37                             <div class="card-body">
38                                 <h6 class="card-title">'. $dataj['Game']. '</h6>
39                                 <p class="card-text text-success">Nama Game : '. $dataj['Game']. '</p>
40                                 <p class="card-text text-success">Terjual : '. $dataj['Copies_Sold']. '</p>
41                                 <p class="card-text text-success">Tahun Rilis : '. $dataj['Release_Date']. '</p>
42                                 <p class="card-text text-success">Genre : '. $dataj['Genre']. '</p>
43                                 <p class="card-text text-success">Pengembang : '. $dataj['Developer']. '</p>
44                                 <p class="card-text text-success">Penerbit : '. $dataj['Publisher']. '</p>
45                             </div>
46                         </div>
47                     </div>
48                 </div>
49             ');
50         }
51         echo json_encode($data);
52     }
53 }
54 }
```

TAMPILAN



- MENGGUNAKAN KATA KUNCI 'THE' UNTUK MELAKUKAN PENCARIAN
- DAN MENDAPATKAN 5 DATA YANG ADA KATA KUNCI 'THE'