

LAPORAN FINAL PROJECT

# **Transformasi Database dan Data Mining Terhadap Dataset MyAnimeList**

Disusun untuk memenuhi Ujian Akhir Semester

Mata Kuliah : Basis Data Lanjut

Dosen Pengampu : Kusnawi, S.Kom, M.Eng



Oleh :

Aqsal Harris Pratama 19.11.3201

Sofyan Adi Saputra 19.11.3196

Farhan Raldifan 19.11.3244

**PROGRAM STUDI S1 INFORMATIKA  
UNIVERSITAS AMIKOM YOGYAKARTA  
2022**

## BAB 1. PENDAHULUAN

### 1.1 Tentang MyAnimeList

MyAnimeList, sering disingkat MAL, adalah situs jejaring sosial anime dan manga serta aplikasi katalog sosial yang dijalankan oleh volunteer. Situs ini menyediakan penggunanya sistem seperti daftar untuk mengatur dan menilai anime dan manga [1]. Situs ini juga memfasilitasi pencarian pengguna yang memiliki selera yang sama dan menyediakan database besar tentang anime dan manga [2]. Pada tahun 2008, situs tersebut menyatakan bahwa memiliki 4,4 juta anime dan 775.000 entri manga [3]. Pada tahun 2015, situs ini menerima 120 juta pengunjung setiap bulan [1]

### 1.2 Informasi Dataset MyAnimeList

Dataset yang digunakan adalah dataset dengan nama anime.csv yang diambil dari Kaggle , dengan jumlah 13.379 data anime yang terbagi dalam 35 kolom, dengan penjelasan kolom pada tabel di bawah ini [4] :

Tabel 1.1 Kolom dalam Dataset Anime

Kolom	Keterangan
anime_id	Nomor identitas anime
anime_url	Url myanimelist dari anime
title	Nama anime
synopsis	Deskripsi singkat dari plot anime
main_pic	Url ke gambar sampul anime
type	Jenis anime (contoh TV, Movie, OVA dll...)
source_type	Jenis sumber anime (contoh Manga, Light Novel dll.)
num_episodes	Jumlah episode di anime
status	Status penayangan anime
start_date	Tanggal mulai anime

end_date	Tanggal akhir anime
season	Musim anime mulai ditayangkan
studios	Daftar studio yang membuat anime
genres	Daftar genre anime (Action, Shonen dll.)
score	Skor rata-rata anime di myanimelist
score_count	Jumlah pengguna yang mencetak anime
score_rank	Peringkat anime berdasarkan skornya di myanimelist
popularity_rank	Peringkat anime berdasarkan popularitasnya di myanimelist
members_count	Jumlah pengguna yang menjadi anggota anime
favorites_count	Jumlah pengguna yang menambahkan sebagai anime favorit
watching_count	Jumlah pengguna yang menonton anime
completed_count	Jumlah pengguna yang telah menyelesaikan anime
onholdcount	Jumlah pengguna yang menahan anime
dropped_count	Jumlah pengguna yang telah menjatuhkan anime
plantowatch_count	Jumlah pengguna yang berencana untuk menonton anime
total_count	Jumlah total pengguna yang menyelesaikan, berencana untuk menonton, menonton, menjatuhkan, atau menahan anime
score10count	Jumlah pengguna yang mendapat skor anime 10
score09count	Jumlah pengguna yang mendapat skor anime 9
score08count	Jumlah pengguna yang mendapat skor anime 8
score07count	Jumlah pengguna yang mendapat skor anime 7
score06count	Jumlah pengguna yang mendapat skor anime 6
score05count	Jumlah pengguna yang mendapat skor anime 5

score04count	Jumlah pengguna yang mendapat skor anime 4
score03count	Jumlah pengguna yang mendapat skor anime 3
score02count	Jumlah pengguna yang mendapat skor anime 2
score01count	Jumlah pengguna yang mendapat skor anime 1
clubs	Daftar klub MAL yang menjadi bagian dari anime
pics	Daftar url juga gambar anime

### 1.3 Batasan Project

Beberapa batasan dalam project ini adalah sebagai berikut.

1. Data yang digunakan adalah data dari dataset MyAnimeList dari Kaggle.
2. Data yang digunakan terbatas pada data anime dalam dataset MyAnimeList.

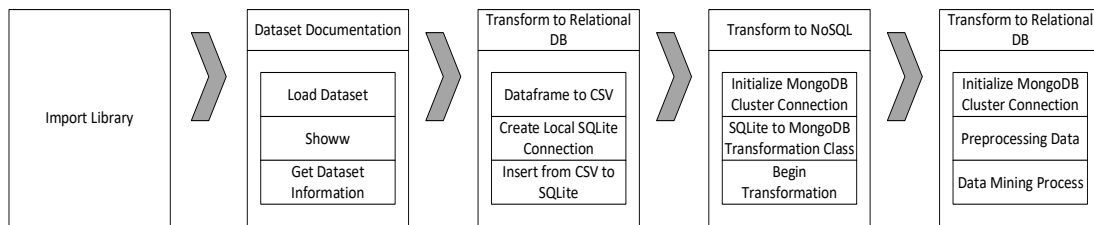
### 1.4 Tujuan Project

Maksud dan tujuan dari project ini adalah sebagai berikut.

1. Mengetahui proses transformasi dataset dengan format CSV menjadi relational database (SQLite)
2. Mengetahui proses transformasi relational database menjadi NoSQL (MongoDB)
3. Mengetahui proses datamining terhadap dataset MyAnimeList untuk menemukan :
  - Genre populer anime dalam jangka waktu lima tahunan.
  - Faktor yang berpengaruh terhadap popularitas anime

## BAB 2. SKEMA Pengerjaan Project

Project ini terdiri atas lima tahapan, dimulai dari tahap import library, dokumentasi dataset, konversi ke relational database, konversi ke NoSQL, dan datamining. Adapun alur project ditunjukkan sebagai berikut pada Gambar 1.



Gambar 2.1. Diagram alur project

### 2.1 Import Library

Dengan melakukan instalasi module dan import library, program dapat menjalankan berbagai fitur dari package yang disediakan oleh library dan modul.

### 2.2 Dataset Documentation

Dokumentasi dataset bertujuan untuk mengetahui berbagai informasi dalam dataset dengan tujuan menemukan gambaran informasi untuk proses data mining.

#### 2.2.1 Load Dataset

Dataset anime.csv diambil dari Kaggle kemudian dipindah ke dalam Google Drive. Pengambilan file dari Google Drive menggunakan metode split url yang mana identitas file dipisah dari URL awal untuk diunduh.

#### 2.2.2 Show Dataset Information

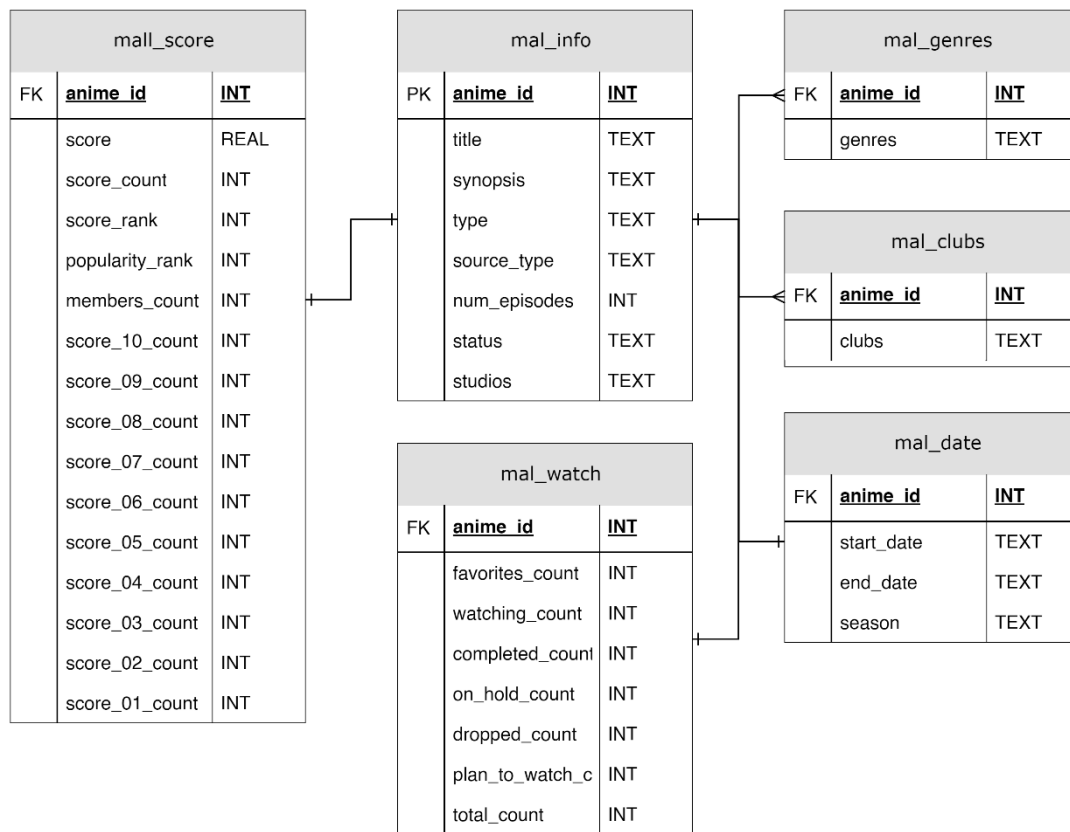
Tahapan ini akan menghasilkan berbagai informasi umum mengenai dataset seperti jumlah baris dan kolom data, tipe data, jumlah baris tiap kolom yang bernilai null, jumlah baris pada tiap kolom yang bernilai unique, dan deskripsi statistika data.

#### 2.2.3 Drop Unused Column

Menghapus kolom yang tidak dibutuhkan dilakukan dengan tujuan agar proses data mining lebih terfokus dan pengelolaan datanya lebih mudah.

## 2.3 Transform to Relational Database

Dataframe dengan kolom yang sudah difilter akan diubah ke dalam bentuk relational database dengan mengelompokkan setiap kolom yang memiliki hubungan dengan kolom lain sehingga terbentuk relasi antar tabel dengan rancangan ERD sebagai berikut.



Gambar 2.2 Entity Relationship Diagram

### 2.3.1 Convert Dataframe to CSV

Dataframe sebelumnya dipisah menjadi beberapa file CSV dengan kolom sesuai pada ERD di atas. Index dan Header dari dataframe sebelumnya dihapus dan digantikan dengan header baru.

### 2.3.2 Initialize SQLite Local Connection and Create Table

Relational database SQLite yang digunakan pada project ini bersifat local sehingga tidak dapat diakses melalui jaringan. Selanjutnya tabel dibuat dengan kolom

dan tipe data sesuai dengan ERD yang ada. Tiap kolom tidak didefinisikan panjang datanya agar saat proses input data lebih dinamis.

### 2.3.3 Insert form CSV to SQLite

Setelah inisialisasi koneksi telah berhasil dan tabel telah dibuat, proses input data dilakukan terhadap setiap tabel dengan data yang diambil dari tiap-tiap file CSV yang telah dipisah.

### 2.4 Transform to NoSQL (MongoDB)

Proses ini akan menggabungkan setiap baris pada masing-masing tabel dalam database SQLite menjadi sebuah dokumen dalam bentuk NoSQL yang disimpan pada cluster MongoDB yang telah dibuat. Adapun skema dokumen konversi yang nantinya dihasilkan adalah sebagai berikut :

_ id	:	ObjectId
anime_id	:	mal_info.anime
title	:	mal_info.title
synopsis	:	mal_info.synopsis
type	:	mal_info.type
source_type	:	mal_info.source_type
num_episodes	:	mal_info.num_episodes
status	:	mal_info.status
studios	:	mal_info.studios[array]
date	:	
start_date	:	mal_date.start_date
end_date	:	mal_date.end_date
season	:	mal_date.season
score	:	
popularity_rank	:	mal_score.popularity_rank
members_count	:	mal_score.members_count
score_10_count	:	mal_score.score_10_count
score_09_count	:	mal_score.score_09_count
score_08_count	:	mal_score.score_08_count
score_07_count	:	mal_score.score_07_count
score_06_count	:	mal_score.score_06_count
score_05_count	:	mal_score.score_05_count
score_04_count	:	mal_score.score_04_count
score_03_count	:	mal_score.score_03_count
score_02_count	:	mal_score.score_02_count
score_01_count	:	mal_score.score_01_count
score	:	mal_score.score
score_count	:	mal_score.score_count

score_rank	: mal_score.score_rank
watch	:
favorites_count	: mal_watch.favorites_count
watching_count	: mal_watch.watching_count
completed_count	: mal_watch.completed_count
on_hold_count	: mal_watch.on_hold_count
dropped_count	: mal_watch.dropped_count
plan_to_watch_count	: mal_watch.plan_to_watch_count
total_count	: mal_watch.total_count
genres	:
mal_genres	[array]
clubs	:
mal_clubs	[array]

#### 2.4.1 Initialize Cluster MongoDB Connection

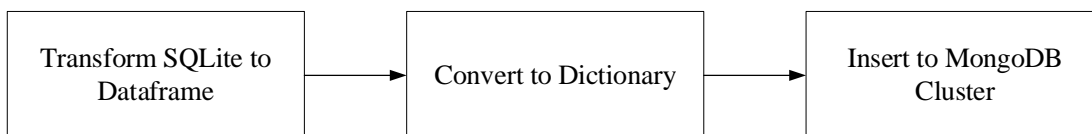
Proses ini akan memulai koneksi ke Cluster MongoDB yang nantinya akan digunakan untuk proses input data yang telah dikonversi menjadi bentuk skema dokumen MongoDB di atas.

#### 2.4.2 Create SQLite to MongoDB Transformation Class

Transformation class ini digunakan untuk melakukan konversi dari beberapa tabel dalam SQLite menjadi Collection dalam cluster MongoDB. Proses transformasi ini menggunakan tiga method, yaitu inisialisasi, konversi ke dataframe, dan insert ke cluster MongoDB.

#### 2.4.3 Begin Transformation

Proses transformasi menggunakan metode sebagai berikut :



##### 2.4.3.1 Transform SQLite to Dataframe

Setiap tabel dalam MongoDB cluster diubah ke dalam bentuk Dataframe sehingga terbentuk enam dataframe yaitu info, genres, clubs, date, score, dan watch.

##### 2.4.3.2 Convert to Dictionary

Setiap baris dalam dataframe diubah menjadi dictionary, kemudian beberapa kolom yang berisi array dipisahkan sehingga menjadi beberapa object tersendiri.



#### 2.4.3.3 Insert to MongoDB Cluster

Dictionary yang telah dibuat kemudian diinput ke dalam cluster MongoDB menggunakan method yang ada di dalam transformation class yang telah dibuat.

### 2.5 Data Mining

Proses data mining pada project ini tidak menerapkan suatu algoritma apapun, namun proses data mining dilakukan pengambilan setiap kolom yang dibutuhkan sesuai dengan tujuan awal data miningnya, yaitu mengetahui genre populer anime dalam jangka waktu lima tahunan dan menemukan faktor yang berpengaruh terhadap popularitas anime.

#### 2.5.1 Initialize Cluster MongoDB Connection and Import Collection to Dataframe

Proses ini akan memulai koneksi ke Cluster MongoDB yang telah dibuat dan berisi collection hasil konversi pada proses sebelumnya yang nantinya akan digunakan untuk proses data mining. Setiap dokumen dalam collection yang diambil akan dikonversi ke dalam bentuk dataframe.

#### 2.5.2 Preprocessing

Seluruh dokumen yang sebelumnya telah diinput pada cluster MongoDB tidak dapat langsung digunakan dalam proses data mining karena terdapat data yang bernilai tidak sesuai atau kosong. Preprocessing ini bertujuan untuk membersihkan data sehingga siap digunakan dalam proses data mining. Tahapan yang dilakukan dalam proses ini antara lain :

##### 2.5.2.1 Normalize Column

Beberapa kolom dalam dataframe yang didapatkan dari hasil import document pada collection MongoDB masih berisi objek-objek embedded document sehingga dengan normalisasi kolom ini akan memisahkan setiap objek dalam embedded dokumen menjadi kolom sendiri.

##### 2.5.2.2 Check Null Values and Drop Column

Jumlah null dari setiap kolom akan digunakan sebagai pertimbangan untuk melakukan drop row yang berisi null dari suatu kolom. Drop row tersebut dipakai

dengan tujuan agar data yang penting dan dibutuhkan untuk proses datamining nanti seluruhnya dapat dipakai.

#### 2.5.2.3 Replace Specific Column Value and Add Column

Beberapa baris dalam kolom yang bernilai null memiliki opsi untuk diisi, seperti kolom `date.season` yang dapat diisi berdasarkan bulan dari `date.start_date`. Kolom `year` juga ditambahkan dengan mengambil data tahun dari `date.start_date`.

#### 2.5.2.4 Fill Null Column

Kolom yang tidak memungkinkan untuk diisi dengan nilai lainnya dapat diisi dengan angka nol sehingga tidak ada data yang bernilai null pada dataframe yang akan digunakan dalam proses data mining.

### 2.5.3 Data Mining Process

#### 2.5.3.1 Genre Populer 5 Tahunan

Proses data mining untuk mencari informasi ini adalah sebagai berikut :

- Mencari tahun rilis minimal dan maksimal seluruh anime dan menggunakannya sebagai parameter untuk menentukan periode tahun rilisnya
- Membuat list genre untuk menampung setiap genre (dapat satu atau lebih) dari setiap anime dan list periode untuk menampung tahun rilis. Value dari periode diperoleh dari looping dengan parameter range yang berisi tahun minimal dan maksimal anime, sedangkan genre didapatkan dari proses loop kolom genres dan memisahnya apabila terdapat lebih dari satu genre.
- Pembuatan dataframe dengan kolom periode dan genres beserta masing-masing jumlahnya
- Melakukan filter dataframe untuk hanya menunjukkan periode dengan jumlah  $\geq 5\%$  dari keseluruhan data dan hanya menampilkan genre yang  $\geq$  mean seluruh genre
- Membuat plot dari dataframe yang telah difilter dalam bentuk bar.

#### 2.5.3.2 Faktor yang Berpengaruh Popularitas Anime

Secara umum, konsep dalam proses data mining untuk menemukan faktor yang berpengaruh dalam popularitas anime yaitu dengan memetakan kolom yang memungkinkan ke dalam bentuk scatter kemudian mencari nilai mayoritas dalam plot tersebut dengan rincian langkah sebagai berikut :

- Mengambil 100 data teratas dari dataframe yang telah diurutkan berdasarkan score.popularity\_rank
- Membuat scatter dari kolom tahun rilis
- Membuat scatter dari kolom jumlah episode
- Membuat scatter dari kolom sumber anime
- Membuat scatter dari kolom studio
- Membuat scatter dari kolom tahun rilis
- Membuat scatter dari kolom genre

## BAB 3. IMPLEMENTASI DAN PEMBAHASAN

Keseluruhan proses implementasi dari skema yang telah dibuat dalam project ini menggunakan bahasa pemrograman Python dan Google Colaboratory sebagai IDE, kecuali pencarian dataset, penyimpanan dataset, pembuatan dan konfigurasi cluster MongoDB dilakukan secara manual berturut-turut menggunakan website Kaggle, Google Drive, dan MongoDB Cloud pada browser Google Chrome.

### 3.1 Install Module and Import Library

Library yang diinstal pada bagian ini nantinya akan digunakan pada keseluruhan project. Pandas digunakan untuk pengolahan dataframe, sqlite3 untuk menjalankan sqlite, csv untuk membaca file csv, pymongo dan pymongo[srv] merupakan modul yang diperlukan agar dapat melakukan import MongoClient yang nantinya akan digunakan untuk mengakses cluster MongoDB, dan library matplotlib.pyplot untuk representasi data dalam bentuk plot

### 3.2 Dataset Documentation

#### 3.2.1 Load Dataset

Dataset yang telah diambil dari kaggle diupload pada folder di Goole Drive, kemudian akan didownload pada notebook menggunakan metode split url. Metode split url dipilih dengan tujuan agar notebook dapat dijalankan oleh siapapun. Kemudian file yang telah didownload akan ditampilkan dalam bentuk dataframe dengan memanfaatkan library pandas. Dataset anime.csv pada project ini disimpan dalam object dataframe dengan nama df

	anime_id	anime_url	title	synopsis	main_pic	type	source_type	num_episodes	status	start_date	end_date
0	2366	<a href="https://myanimelist.net/anime/2366/Touma_Kishinden">https://myanimelist.net/anime/2366/Touma_Kishinden</a>	Touma Kishinden Oni	Shuramaru is hated and feared by the villagers...	<a href="https://cdn.myanimelist.net/images/anime/9/829...">https://cdn.myanimelist.net/images/anime/9/829...</a>	TV	Game	25.0	Finished Airing	1995-10-05 00:00:00	1996-03-21 00:00:00
1	4940	<a href="https://myanimelist.net/anime/4940/Sabaku_no_Kaizoku_Captain_Kuppa">https://myanimelist.net/anime/4940/Sabaku_no_Kaizoku_Captain_Kuppa</a>	Sabaku no Kaizoku! Captain Kuppa	Sometime in the future, the world was complete...	<a href="https://cdn.myanimelist.net/images/anime/9/736...">https://cdn.myanimelist.net/images/anime/9/736...</a>	TV	Manga	26.0	Finished Airing	2001-08-13 00:00:00	2002-02-11 00:00:00
2	50285	<a href="https://myanimelist.net/anime/50285/On_Air_Dekinai!">https://myanimelist.net/anime/50285/On_Air_Dekinai!</a>	On Air Dekinai!	Set in 2014, the anime follows the adventures ...	<a href="https://cdn.myanimelist.net/images/anime/1021/...">https://cdn.myanimelist.net/images/anime/1021/...</a>	TV	Manga	12.0	Currently Airing	2022-01-10 00:00:00	2022-03-28 00:00:00

Gambar 3.1 Dataset Anime.csv

### 3.2.2 Show Dataset Information

Bagian ini akan menunjukkan seluruh informasi pada dataframe. Shape akan menunjukkan jumlah baris dan kolom pada dataset, yaitu 13379 baris data dan 38 kolom.

```
df.shape  
  
(13379, 38)
```

Gambar 3.2 Data Shape

Method info() akan menampilkan informasi berupa nama kolom, jumlah data yang tidak null dan tipe data dari kolom tersebut

```
df.info()  
  
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 13379 entries, 0 to 13378  
Data columns (total 38 columns):  
#   Column                Non-Null Count  Dtype  
---  ---  
0   anime_id              13379 non-null  int64  
1   anime_url             13379 non-null  object  
2   title                 13379 non-null  object  
3   synopsis              13375 non-null  object  
4   main_pic              13379 non-null  object  
5   type                  13379 non-null  object  
6   source_type           13379 non-null  object  
7   num_episodes          12948 non-null  float64  
8   status                13379 non-null  object
```

Gambar 3.3 Data Info

Method nunique() akan menampilkan jumlah data yang memiliki nilai unique dari masing-masing kolom

```
df.nunique()  
  
anime_id          13379  
anime_url         13379  
title             13379  
synopsis           12046  
main_pic          13379  
type               5  
source_type        14  
num_episodes       189  
status             3
```

Gambar 3.4 Data Nunique

Method describe() dari library pandas akan menghasilkan tabel statistika dari seluruh kolom pada dataframe.

```
df.describe()
```

	anime_id	num_episodes	score	score_count	score_rank	popularity_rank	members_count	favorites_count
count	13379.000000	12948.000000	10714.000000	1.071400e+04	9465.000000	13379.000000	1.337900e+04	13379.000000
mean	23185.896928	12.327850	6.561258	3.728666e+04	5544.653883	7889.506017	5.787081e+04	684.553629
std	16646.297648	44.955752	0.985277	1.229652e+05	3528.083235	5333.044230	1.831933e+05	5225.720400
min	1.000000	1.000000	1.850000	1.010000e+02	1.000000	0.000000	0.000000e+00	0.000000
25%	5433.000000	1.000000	6.020000	8.315000e+02	2483.000000	3380.500000	9.270000e+02	1.000000
50%	27417.000000	3.000000	6.650000	3.486500e+03	5208.000000	7094.000000	5.239000e+03	8.000000
75%	37771.000000	12.000000	7.250000	1.989225e+04	8394.000000	11694.500000	3.048950e+04	69.000000
max	51333.000000	3057.000000	9.050000	2.380891e+06	12406.000000	19552.000000	3.329298e+06	197551.000000

Gambar 3.5 Data Describe

### 3.2.3 Drop Unused Column

Kolom yang berisi link (anime\_url, main\_pic, dan pics) akan dihapus dan dataframe baru akan dibuat dengan nama **data**. Sehingga dataframennya akan menjadi seperti gambar di bawah :

```
data = df.drop(columns=['anime_url', 'main_pic', 'pics'])
data.head()
```

	anime_id	title	synopsis	type	source_type	num_episodes	status	start_date	end_date	season	studios	genres	score	score_count	score_rank
0	2366	Touma Kishinden Oni	Shuramaru is hated and feared by the villagers...	TV	Game	25.0	Finished Airing	1995-10-05 00:00:00	1996-03-21 00:00:00	Fall 1995	J.C.Staff	Supernatural	NaN	NaN	NaN
1	4940	Sabaku no Kaizoku! Captain Kupp	Sometime in the future, the world was complete...	TV	Manga	26.0	Finished Airing	2001-08-13 00:00:00	2002-02-11 00:00:00	Summer 2001	NaN	Action Adventure	NaN	NaN	NaN
2	50285	On Air Dekinai!	Set in 2014, the anime follows the adventures ...	TV	Manga	12.0	Currently Airing	2022-01-10 00:00:00	2022-03-28 00:00:00	Winter 2022	Space Neko Company Jinnnan Studio	Comedy	NaN	NaN	NaN

Gambar 3.6 Dataframe Setelah Drop Column

## 3.3 Transform to Relational Database

### 3.2.1 Convert Dataframe to CSV

Proses ini akan menghasilkan output berupa file CSV dengan nama tabel\_informasi.csv, tabel\_genres.csv, tabel\_clubs.csv, tabel\_date.csv, tabel\_score.csv, dan tabel\_watch.csv.

### 3.2.2 Initialize SQLite Local Connection and Create Table

Query yang digunakan dalam pengolahan SQLite pada python ini hampir sama dengan query SQL pada umumnya, yang menjadi pembedanya adalah SQLite ini memanfaatkan cursor dalam menjalankan querynya.

```
conn = sqlite3.connect('myanimelist.sqlite')
cur = conn.cursor()

cur.execute('DROP TABLE IF EXISTS mal_info')
cur.execute('''
CREATE TABLE "mal_info"(
    "anime_id" INTEGER,
    "title" TEXT,
    "synopsis" TEXT,
    "type" TEXT,
    "source_type" TEXT,
    "num_episodes" INTEGER,
    "status" TEXT,
    "studios" TEXT
)
''')
```

Gambar 3.7 Query Inisialisasi Koneksi dan Pembuatan Salah Satu Tabel

Tabel yang telah dibuat dapat dilihat menggunakan query seperti gambar di bawah :

```
cur.execute("SELECT name FROM sqlite_master WHERE type='table';")
print(cur.fetchall())

[('mal_info',), ('mal_genres',), ('mal_clubs',), ('mal_date',), ('mal_score',), ('mal_watch',)]
```

Gambar 3.8 Tabel pada SQLite

### 3.2.3 Insert form CSV to SQLite

Insert data ke dalam tabel SQLite dilakukan secara berurutan sesuai dengan kolom pada file CSV yang telah dipisah tadi.

```
tabel_informasi = open("tabel_informasi.csv")
rows_informasi = csv.reader(tabel_informasi)
cur.executemany('INSERT INTO "mal_info" VALUES (?, ?, ?, ?, ?, ?, ?, ?)', rows_informasi)
```

Gambar 3.9 Insert Data SQLite

Untuk dapat melihat data yang telah diinputkan, dapat menggunakan query select seperti SQL pada umumnya.

```
cur.execute("SELECT * FROM mal_info LIMIT 3")
print(cur.fetchall())

cur.execute("SELECT * FROM mal_genres LIMIT 3")
print(cur.fetchall())

cur.execute("SELECT * FROM mal_clubs LIMIT 3")
print(cur.fetchall())

cur.execute("SELECT * FROM mal_date LIMIT 3")
print(cur.fetchall())

cur.execute("SELECT * FROM mal_score LIMIT 3")
print(cur.fetchall())

cur.execute("SELECT * FROM mal_watch LIMIT 3")
print(cur.fetchall())

[(2366, 'Touma Kishinden Oni', 'Shuramaru is hated and feared
[('Supernatural',), ('Action|Adventure',), ('Comedy',)]
[('14045|10778|342',), ('8494',), ('27907|8652',)]
[('1995-10-05 00:00:00', '1996-03-21 00:00:00', 'Fall 1995'),
[('', '', '', 11743, 884, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0), ('',
[(0, 61, 173, 31, 75, 544, 884), (2, 26, 124, 18, 73, 250, 49
```

Gambar 3.10 Select Data

Apabila SQLite sudah selesai digunakan dan data telah berhasil diinput keseluruhan, maka selanjutnya perlu melakukan commit dan menutup koneksi dengan menggunakan method commit() dan close() dari sqlite3.

### 3.4 Transform to NoSQL (MongoDB)

#### 3.4.1 Initialize Cluster MongoDB Connection

Akses ke cluster mongoDB memerlukan URI atau Connection String. URI tersebut dapat diakses melalui button Connect pada halaman cluster MongoDB. Konsep akses ke clusternya memanfaatkan cursor yang mengacu kepada URI tersebut.

```
uri = 'mongodb+srv://bdli1f0:bdli1f0@bdli1f0.vexcmfi.mongodb.net/?retryWrites=true&w=majority'
client = MongoClient(uri)

#show connection status
client.stats

Database(MongoClient(host=['ac-g0rzzal-shard-00-01.vexcmfi.mongodb.net:27017', 'ac-g0rzzal-shard-00-02.vexcmfi.mongodb.net:27017'], document_class=dict, tz_aware=False, connect=True, retrywrites=True, w='majority', authsource='admin', replicaset='atlas-ccxbv5-shard-0', tls=True), 'stats')
```

Gambar 3.11 Inisialisasi Koneksi dan Status Cluster MongoDB



### 3.4.2 Create SQLite to MongoDB Transformation Class

Class transformation ini memiliki tiga method, yaitu init yang berfungsi untuk melakukan inisialisasi koneksi setiap kali object dalam class tersebut dibuat, sqlite\_to\_df untuk melakukan konversi dari tabel sqlite menjadi dataframe, dan method mongodb\_insert berfungsi untuk melakukan input data ke MongoDB cluster.

```
class transformation:
    def __init__(self, mongo_uri='mongodb+srv://bdlif10:bdLAMIKOMif10@bdlif10.vexc
        sqlite_conn = sqlite3.connect('myanimelist.sqlite')
        sqlite_cur = sqlite_conn.cursor()
        mongodb_client = pymongo.MongoClient(mongo_uri)
        self.sqlite_conn = sqlite_conn
        self.sqlite_cur = sqlite_cur
        self.mongodb_client = mongodb_client
        print("Objek class berhasil dibuat")

    def sqlite_to_df(self, tabel_name):
        query_getdata = "select * from" + " " + tabel_name
        df_from_sql = pd.read_sql_query(query_getdata, self.sqlite_conn)
        print('Transformasi '+tabel_name+' ke DataFrame berhasil')
        return df_from_sql

    def mongodb_insert(self, database_mongo_name, collection_name, dict_insert):
        db = self.mongodb_client[database_mongo_name]
        collection = db[collection_name]
        collection.insert_one(dict_insert)
```

Gambar 3.11 Transformation Class

### 3.4.3 Begin Transformation

Proses transformasi ini dimulai dengan membuat sebuah object dari class transformation tersebut dengan nama transform\_to\_MongoDB.

```
transform_to_MongoDB = transformation()

Objek class berhasil dibuat
```

Gambar 3.12 Object Transformation Class

#### 3.4.3.1 Transform SQLite to Dataframe

Dengan menggunakan method sqlite\_to\_df pada class tersebut akan dibuat lima dataframe yang berasal dari tiap-tiap tabel pada database di SQLite.

```
df_info = transform_to_MongoDB.sqlite_to_df('mal_info')
df_genres = transform_to_MongoDB.sqlite_to_df('mal_genres')
df_clubs = transform_to_MongoDB.sqlite_to_df('mal_clubs')
df_date = transform_to_MongoDB.sqlite_to_df('mal_date')
df_score = transform_to_MongoDB.sqlite_to_df('mal_score')
df_watch = transform_to_MongoDB.sqlite_to_df('mal_watch')
```

Transformasi mal\_info ke DataFrame berhasil  
 Transformasi mal\_genres ke DataFrame berhasil  
 Transformasi mal\_clubs ke DataFrame berhasil  
 Transformasi mal\_date ke DataFrame berhasil  
 Transformasi mal\_score ke DataFrame berhasil  
 Transformasi mal\_watch ke DataFrame berhasil

Gambar 3.13 Konversi Tabel SQLite ke DataFrame

#### 3.4.3.2 Convert to Dictionary

Setelah dataframe berhasil dibuat, langkah selanjutnya adalah melakukan konversi dari tiap baris dari dataframe yang telah dikonversi menjadi nested dictionary, yang mana setiap dictionary ini akan menjadi sebuah embedded document pada MongoDB. Dictionary yang dibuat akan diinput ke dalam MongoDB Cluster.

```
{'anime_id': 2366,
 'clubs': [14045, 10778, 342],
 'date': {'end_date': '1995-10-05',
 'season': 'Fall',
 'start_date': '1995-10-05'},
 'genres': ['Supernatural'],
 'num_episodes': 25,
 'score': {'members_count': 884,
 'popularity_rank': 11743,
 'score_01_count': 0,
 'score_02_count': 0,
 'score_03_count': 0,
 'score_04_count': 0,
 'score_05_count': 0,
 'score_06_count': 0,
 'score_07_count': 0,
 'score_08_count': 0,
 'score_09_count': 0,
 'score_10_count': 0},
 'source_type': 'Game',
 'status': 'Finished Airing',
 'studios': ['J.C.Staff'],
 'synopsis': 'Shuramaru is hated and feared by the village'}
```

Gamar 3.14 Contoh Dictionary yang Diinput

#### 3.4.3.3 Insert to MongoDB Cluster

Input dictionary akan dilakukan dengan menggunakan method `mongodb_insert` pada class transformation. Proses insert ini akan memakan waktu yang cukup lama tergantung pada jumlah data.

## 3.5 Data Mining

### 3.5.1 Initialize Cluster MongoDB Connection and Import Collection to Dataframe

Proses inisialisasi sama seperti pada tahapan transformasi SQLite ke MongoDB. Setelah inisialisasi berhasil, maka seluruh dokumen pada collection di cluster MongoDB dapat diimport ke dalam bentuk dataframe. Tujuan dijadikan dataframe ini adalah agar pengolahan data lebih mudah untuk dilakukan.

	_id	anime_id	title	synopsis	type	source_type	num_episodes	status	studios	date	score	watch	clubs
0	62d4f035aa797eed7380354f	2366	Touma Kishinden Oni	Shuramaru is hated and feared by the villagers...	TV	Game	25.0	Finished Airing	[J.C.Staff]	{start_date: '1995-10-05', end_date: '1995-...	{popularity_rank: 11743, members_count: 88...	{favorites_count: 0, watching_count: 61, '...	[14045, 10778, 342]
1	62d4f036aa797eed73803550	4940	Sabaku no Kaizoku! Captain Kupp	Sometime in the future, the world was complete...	TV	Manga	26.0	Finished Airing	NaN	{start_date: '2001-08-13', end_date: '2001-...	{popularity_rank: 13321, members_count: 49...	{favorites_count: 2, watching_count: 26, '...	[8494]
2	62d4f036aa797eed73803551	50285	On Air Dekinai!	Set in 2014, the anime follows the adventures...	TV	Manga	12.0	Currently Airing	[Space Neko Company, Jimnan Studio]	{start_date: '2022-01-10', end_date: '2022-...	{popularity_rank: 9603, members_count: 205...	{favorites_count: 8, watching_count: 203, '...	[27907, 8652]

Gambar 3.15 Document MongoDB dalam Bentuk Dataframe

### 3.5.2 Preprocessing

#### 3.5.2.1 Normalize Column

Pada gambar 3.15 dapat dilihat bahwa kolom data, score, dan watch, masih berisi dictionary lagi, maka dengan melakukan normalisasi kolom ini akan membuat dictionary di dalam kolom menjadi kolom individu.

```
df = pd.json_normalize(query_list, max_level=1)
df.head(3)
```

	_id	anime_id	title	synopsis	type	source_type	num_episodes	status	studios	clubs	genres	date.start_date	date.end_date	date.season
0	62d4f035aa797eed7380354f	2366	Touma Kishinden Oni	Shuramaru is hated and feared by the villagers...	TV	Game	25.0	Finished Airing	[J.C.Staff]	[14045, 10778, 342]	[Supernatural]	1995-10-05	1995-10-05	Fall
1	62d4f036aa797eed73803550	4940	Sabaku no Kaizoku! Captain Kupp	Sometime in the future, the world was complete...	TV	Manga	26.0	Finished Airing	NaN	[8494]	[Action, Adventure]	2001-08-13	2001-08-13	Summer

Gambar 3.16 Penerapan Normalisasi Kolom

#### 3.5.2.2 Check Null Values and Drop Row

Baris yang bernilai null dan tidak memungkinkan untuk diisi namun berjumlah sedikit dapat dihapus, seperti baris pada kolom synopsis, num\_episodes, dan date.start\_date pada gambar di bawah.

synopsis	4	synopsis	0
type	0	type	0
source_type	0	source_type	0
num_episodes	431	num_episodes	0
status	0	status	0
studios	3036	studios	2803
clubs	1761	clubs	1685
genres	1761	genres	1685
date.start_date	181	date.start_date	0

Gambar 3.17 Jumlah Baris Null Sebelum dan Sesudah Drop Row

### 3.5.2.3 Replace Specific Column Value and Add Column

Pengisian kolom season dilakukan dengan mengambil bulan (karakter ke 6 dan 7 pada kolom date.start\_date) kemudian mengisinya dengan season. Kolom date.start\_date.year ditambahkan dari 4 karakter pertama kolom date.start\_date.

```
def fill_season(month):
    if (month in [3,4,5]):
        return 'Spring'
    elif (month in [6,7,8]):
        return 'Summer'
    elif (month in [9,10,11]):
        return 'Fall'
    elif (month in [12,1,2]):
        return 'Winter'

def get_year(date):
    try:
        return int(date[0:4:1])
    except:
        pass

df['date.season'] = df['date.start_date'].str[5:7:1].astype(int).apply(fill_season)
df['date.start_date.year'] = df['date.start_date'].apply(get_year)
```

Gambar 3.18 Method fill\_season dan get\_year

### 3.5.2.4 Fill Null Column

Baris pada kolom lain yang memiliki nilai null lainnya akan diisi dengan nilai 0 untuk memudahkan membedakan data dalam proses data mining.

```
df = df.fillna(0)
df.isna().sum()

_id                0
anime_id          0
title             0
synopsis          0
type              0
source_type       0
```

Gambar 3.19 Mengisi Null dengan Nilai 0

### 3.5.3 Data Mining Process

#### 3.5.3.1 Genre Populer 5 Tahunan

- Dataframe yang diperoleh dari perhitungan genre per lima tahun periode ditunjukkan pada gambar di bawah :

	period	Action	Adventure	Avant Garde	Award Winning	Boys Love	Cars	Comedy	Demons	Drama	Ecchi	Erotica	Fantasy	Game	Girls Love	Gourmet	Harem	Hentai	Historical	Horror	Josei	Kids
0	1917-1921	0.00	0.00	0.00	0.00	0.00	0.00	3.00	0.00	4.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00
1	1922-1926	0.00	2.00	0.00	0.00	0.00	0.00	2.00	0.00	1.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	0.00	3.00	0.00	0.00	0.00
2	1927-1931	7.00	6.00	0.00	0.00	0.00	0.00	6.00	1.00	3.00	0.00	0.00	3.00	0.00	0.00	0.00	0.00	0.00	4.00	0.00	0.00	5.00
3	1932-1936	13.00	9.00	3.00	0.00	0.00	1.00	10.00	0.00	1.00	0.00	0.00	4.00	0.00	0.00	0.00	0.00	0.00	6.00	0.00	0.00	8.00
4	1937-1941	4.00	2.00	0.00	0.00	0.00	0.00	5.00	0.00	2.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	3.00	0.00	0.00	2.00
5	1942-1946	3.00	0.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	7.00	0.00	0.00	1.00

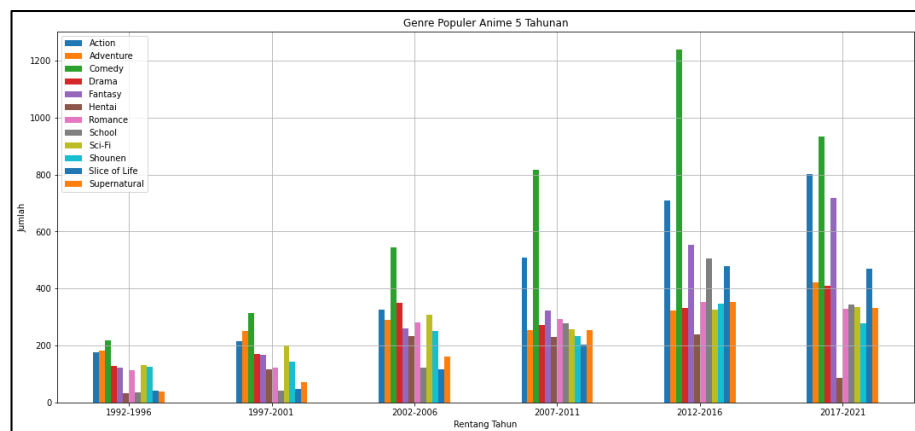
Gambar 3.20 Perhitungan Genre dan Pembagian Periode

- Dataframe yang telah melalui proses filter data dan kolom akan menjadi seperti gambar di bawah :

	period	Action	Adventure	Comedy	Drama	Fantasy	Hentai	Romance	School	Sci-Fi	Shounen	Slice of Life	Supernatural
15	1992-1996	177.0	183.0	217.0	129.0	121.0	33.0	113.0	36.0	132.0	125.0	40.0	38.0
16	1997-2001	215.0	252.0	314.0	171.0	167.0	115.0	123.0	40.0	200.0	142.0	46.0	71.0
17	2002-2006	326.0	289.0	545.0	351.0	259.0	233.0	282.0	121.0	308.0	252.0	117.0	160.0
18	2007-2011	508.0	253.0	817.0	272.0	322.0	230.0	294.0	279.0	257.0	234.0	204.0	254.0
19	2012-2016	710.0	324.0	1240.0	333.0	552.0	239.0	353.0	504.0	326.0	347.0	477.0	354.0
20	2017-2021	803.0	421.0	933.0	411.0	717.0	86.0	329.0	344.0	335.0	279.0	469.0	331.0

Gambar 3.21 Dataframe Setelah Filter Jumlah Data

- Dataframe yang telah difilter tersebut diinputkan ke dalam plot bar sehingga menghasilkan output sebagai berikut :



Gambar 3.22 Diagram Bar

### 3.5.3.2 Faktor yang Berpengaruh Popularitas Anime

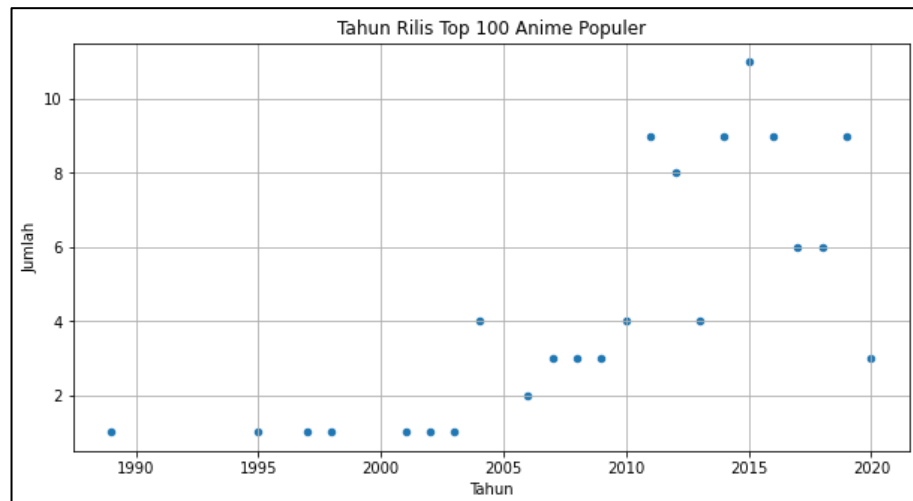
- Mengambil 100 data teratas berdasarkan kolom score.popularity\_rank

```
df_popular = df.sort_values(by=['score.popularity_rank']).loc[df['score.popularity_rank'] != 0].head(100)
df_popular.head()
```

	_id	anime_id	title	synopsis	type	source_type	num_episodes	status	studios	clubs	genres	date.start_date	date.end_date	date.season
3672	6204f381aa797eed738043a7	16498	Shingeki no Kyojin	Centuries ago, mankind was slaughtered to near...	TV	Manga	25.0	Finished Airing	[Wit Studio]	[20081, 19736, 70446, 38603, 35833, 35120, 290...	[Action, Drama, Fantasy, Mystery, Military, Su...	2013-04-07	2013-04-07	Spring
1979	6204f1fcaa797eed73803d0a	1535	Death Note	Brutal murders, petty thefts, and senseless vi...	TV	Manga	37.0	Finished Airing	[Madhouse]	[20081, 70446, 158, 38603, 25565, 413, 887, 4...	[Mystery, Supernatural, Suspense, Police, Psyc...	2006-10-04	2006-10-04	Fall

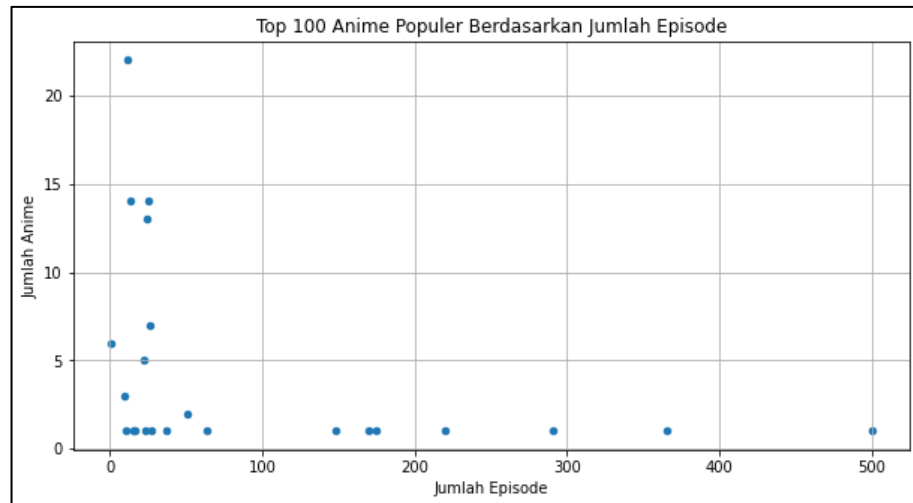
Gambar 3.23 df\_popular

- Membuat scatter dari kolom tahun rilis



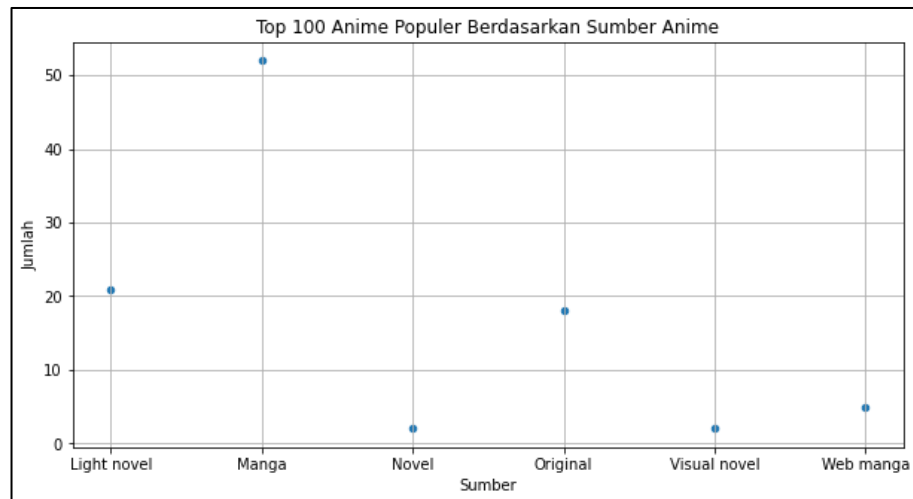
Gambar 3.24 Scatter Tahun Rilis Anime Populer

- Membuat scatter dari kolom jumlah episode



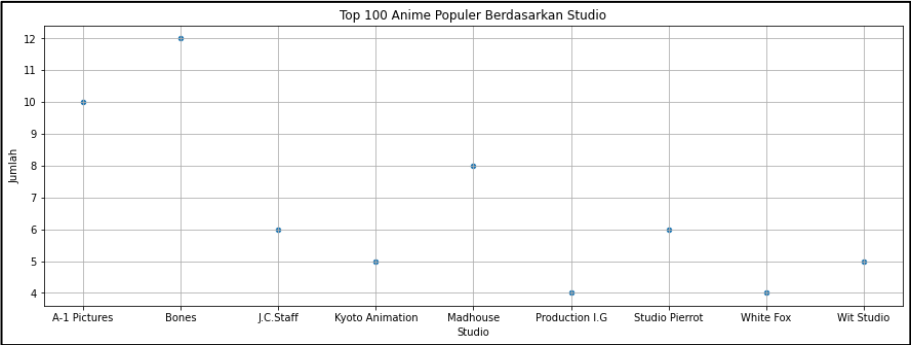
Gambar 3.25 Scatter Jumlah Episode Anime Populer

- Membuat scatter dari kolom sumber anime



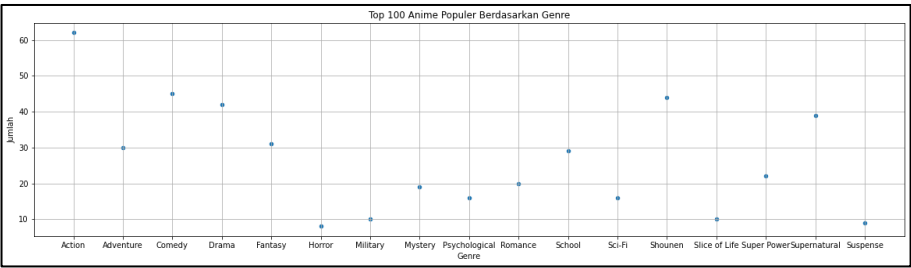
Gambar 3.26 Scatter Sumber Anime Populer

- Membuat scatter dari kolom studio



Gambar 3.27 Scatter Studio Produksi Anime Populer

- Membuat scatter dari kolom genre



Gambar 3.28 Scatter Genre Anime Populer



## BAB 4. HASIL DAN KESIMPULAN

Projek ini bertujuan untuk mengetahui proses transformasi data dari yang awalnya berbentuk CSV menjadi relational database SQLite, kemudian menjadi NoSQL MongoDB, dan dilanjutkan dengan datamining.

Transformasi data dari CSV menjadi relational database SQLite dapat dilakukan dengan memisahkan setiap kolom yang berhubungan atau memiliki konteks yang sama menjadi file CSV individu yang akan digunakan sebagai tabel RDB SQLite.

Transformasi data dari SQLite menjadi MongoDB dapat dilakukan dengan menggunakan dictionary yang diperoleh dari tiap-tiap baris tabel yang telah dikonversi kembali menjadi dataframe.

Kedua proses transformasi tersebut sangat bergantung pada dataframe, artinya dataframe disini berguna sebagai jembatan dalam melakukan konversi bentuk database pada python.

Proses data mining dilakukan sesuai dengan skema yang telah dibuat dengan tujuan mencari genre populer lima tahunan dan faktor yang berpengaruh terhadap popularitas suatu anime. Adapun hasil yang didapatkan dari proses data mining yang telah dilakukan adalah sebagai berikut :

### 4.1 Genre Populer 5 Tahunan

Popularitas jumlah anime mulai terlihat pada tahun 1992 dengan jumlah 1344 atau lebih dari 5% keseluruhan anime. Dari seluruh anime yang ada dapat dikelompokkan menjadi beberapa genre yang menjadi favorit. Seperti yang telah ditunjukkan diagram pada gambar 3.22, genre anime populer 5 tahunan terus berubah. Tabel di bawah ini menunjukkan insight yang diambil dari diagram gambar 3.22.

Tabel 4.2 Kolom dalam Dataset Anime

Genre	Informasi
Action	Action terus mengalami kenaikan yang stabil sejak tahun 1992. Genre ini memungkinkan untuk terus diminati.

Adventure	Adventure tidak begitu stabil, dapat dilihat dari tahun 2007-2011 mengalami penurunan, mungkin dikarenakan banyaknya muncul anime action yang lebih menarik. Hingga menarik banyak minat dari masyarakat.
Comedy	Comedy memiliki grafik yang stabil untuk kenaikannya dari tahun 1992-2016 dan bisa dikategorikan genre paling banyak diminati lebih dari 2 dekade. Namun untuk 5 tahun terakhir peminat anime comedy mulai menurun.
Drama	Mungkin untuk genre drama tidak banyak diminati tetapi grafiknya selalu mengalami kenaikan, meskipun tidak terlalu signifikan dari tahun 1992-2021.
Fantasy	Fantasy memiliki grafik kenaikan yang paling stabil diantara lainnya. Dari tahun 1992-2021 grafik dari genre fantasy terus berkembang hingga dapat membuat genre comedy mengalami penurunan. Mungkin dapat didasari dari banyaknya studio yang membuat anime genre fantasy menarik 5 tahun terakhir ini.
Hentai	Hentai adalah genre yang mengalami banyak penuruna dari tahun 2002-2021. Mungkin dikarenakan banyaknya peminat yang berpindah ke genre lain sehingga genre ini sedikit ditinggalkan.
Romance	Romance pada awalnya genre yang sedikit memiliki grafik yang baik dalam kenaikannya. Namun pada tahun 2017-2021 mengalami penurunan.
School	School memiliki grafik yang hampir sama dengan comedy namun berbeda dengan jumlahnya.
Sci-Fi	Sci-fi mengalami penurunan pada tahun 2007-2011. Namun setelah tahun itu sci-fi mengalami kenaikan grafik lagi.

Shounen	Shounen sedikit mengalami penurunan pada tahun 2017-2021, mungkin semakin berkurangnya jumlah penonton dari tahun tersebut maka anime yang diproduksi semakin menurun.
Slice of Life	Genre ini dapat dikatakan memiliki grafik yang stabil dari tahun 1992-2021. Selalu mengalami peningkatan hingga 5 tahun terakhir.
Supernatural	Sama dengan genre slice of life, selalu mengalami peningkatan. Namun pada genre ini sedikit mengalami penurunan pada tahun 2017-2021

#### 4.2 Faktor yang Berpengaruh Popularitas Anime

Analisis terhadap scatter pada point 3.5.3.2 adalah sebagai berikut :

No. Plot	Hasil
3.24	Anime terpopuler paling banyak dirilis pada tahun 2010-2020.
3.25	Anime terpopuler rata-rata memiliki jumlah episode 12-24.
3.26	Anime populer banyak yang berasal dari manga, dengan jumlah lebih dari 50 anime.
3.27	Studio produksi Bones memiliki 12 anime populer
3.28	Genre teratas anime populer yaitu Action, Comedy, dan Shounen

Berdasarkan analisis di atas, dapat diambil kesimpulan bahwa anime yang baru dirilis memiliki kemungkinan lebih tinggi untuk masuk ke dalam anime populer. Anime dengan jenis serial lebih banyak diminati dibanding dengan movie. Anime yang diadaptasi dari Manga juga lebih diminati. Studio produksi anime tidak memberikan pengaruh yang signifikan terhadap popularitas anime. Selain itu, genre populer 5 tahunan juga relevan sebagai penentu popularitas anime, sehingga dengan rilisnya anime bergenre populer akan memiliki kemungkinan lebih tinggi untuk masuk ke dalam daftar anime populer.

## REFERENSI

- [1] “MyAnimeList Passes Third Day Of Unexpected Downtime.”  
<https://www.forbes.com/sites/laurenorsini/2018/05/30/myanimelist-passes-third-day-of-unexpected-downtime/?sh=2e5f35417e9e> (accessed Jul. 25, 2022).
- [2] “Why Some Fans Watch Anime At Double Speed.”  
<https://www.kotaku.com.au/2018/01/why-some-fans-watch-anime-at-double-speed/> (accessed Jul. 25, 2022).
- [3] “CraveOnline Buys Manga Social Net MyAnimeList - CBS News.”  
<https://www.cbsnews.com/news/craveonline-buys-manga-social-net-myanimelist/> (accessed Jul. 25, 2022).
- [4] “MyAnimeList Dataset | Kaggle.”  
<https://www.kaggle.com/datasets/svanoo/myanimelist-dataset> (accessed Jul. 25, 2022).

## LINK TAMBAHAN

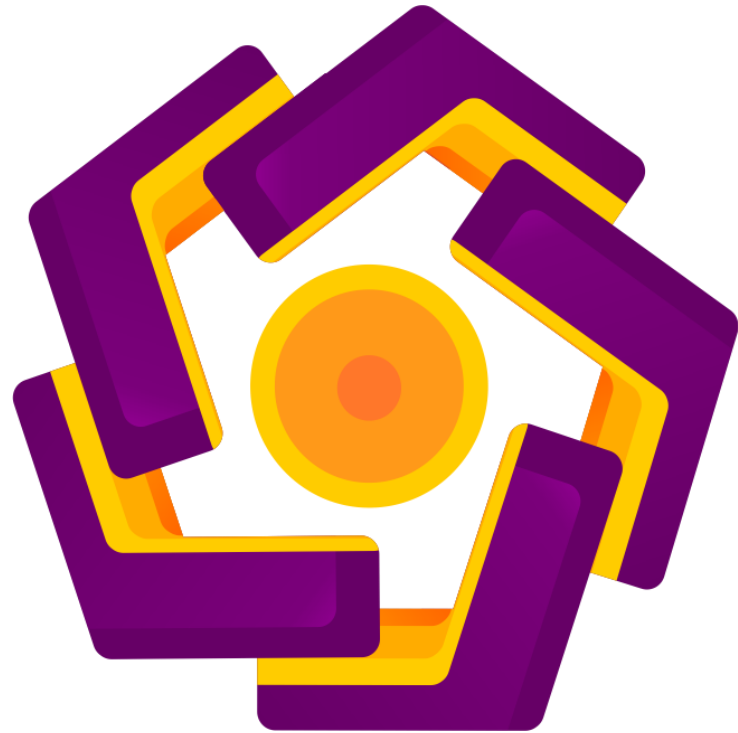
[Google Drive Project](#)

[Google Colab](#)

[Dataset \(Drive\)](#)

[Dataset \(Kaggle\)](#)

[Slide Presentasi](#)



# Transformasi Database dan Data Mining Terhadap Dataset MyAnimeList

Oleh :

Aqsal Harris Pratama	19.11.3201
Sofyan Adi Saputra	19.11.3196
Farhan Raldifan	19.11.3244

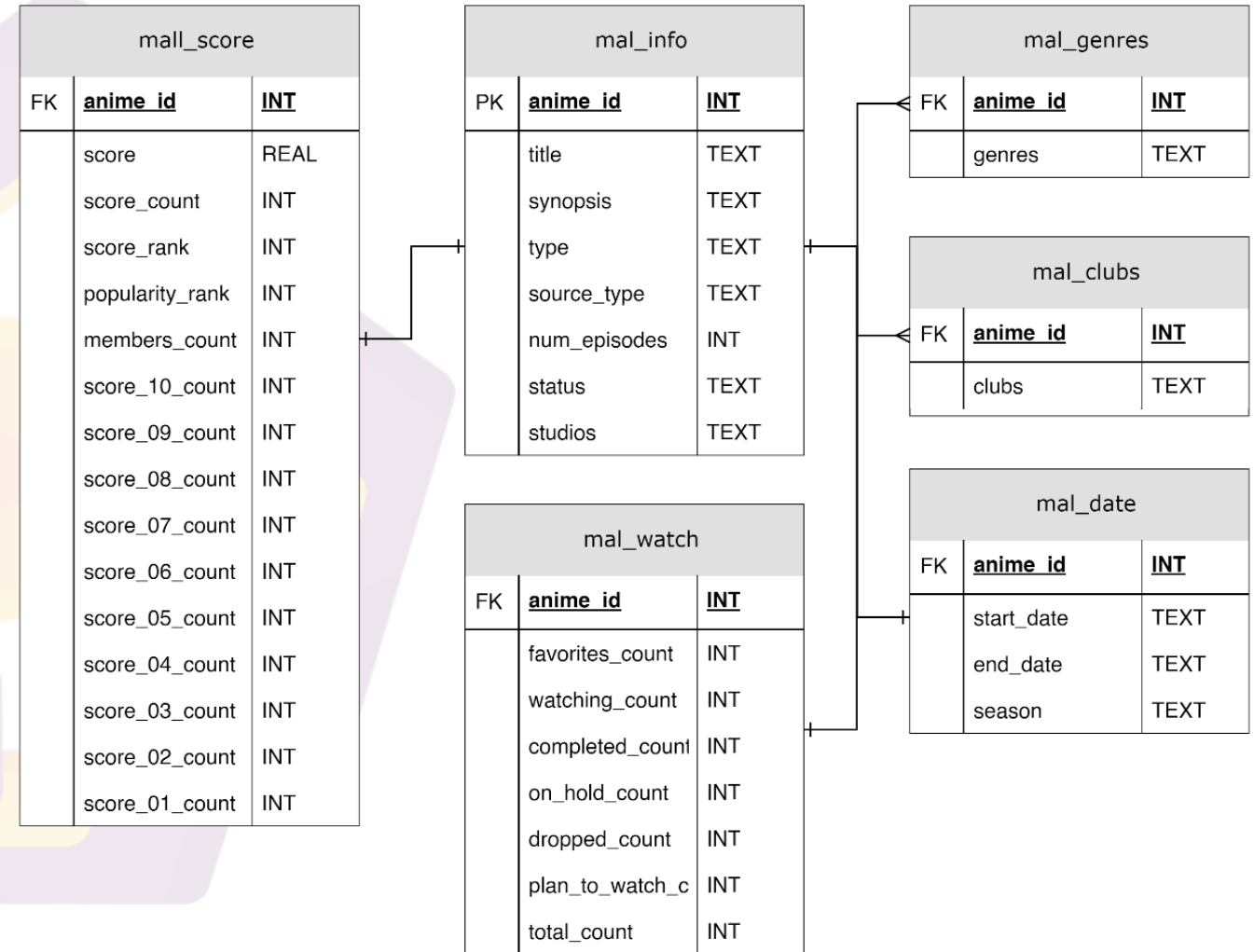
19S1IF-BasisData5

# Informasi Dataset

Dataset MyAnimeList memiliki 13.379 data anime yang terbagi dalam 35 kolom.

Transformasi dataset menjadi relational database SQLite dilakukan dengan memisahkan setiap kolom yang berhubungan atau memiliki konteks yang sama menjadi file CSV individu yang akan digunakan sebagai tabel RDB SQLite.

Adapun rancangan ERD relational database yang akan dibuat seperti gambar di sebelah kanan.



# Konversi ke MongoDB

Transformasi data dari SQLite menjadi MongoDB dilakukan dengan menggunakan dictionary yang diperoleh dari tiap-tiap baris tabel yang telah dikonversi kembali menjadi dataframe.

Setiap dictionary yang dibuat kemudian diinput ke MongoDB cluster yang telah disiapkan.

Rancangan bentuk dokumen yang akan dibuat pada gambar di sebelah kanan.

_id	: ObjectId
anime_id	: mal_info.anime
title	: mal_info.title
synopsis	: mal_info.synopsis
type	: mal_info.type
source_type	: mal_info.source_type
num_episodes	: mal_info.num_episodes
status	: mal_info.status
studios	: mal_info.studios[array]
date	: start_date : mal_date.start_date end_date : mal_date.end_date season : mal_date.season
score	: popularity_rank : mal_score.popularity_rank members_count : mal_score.members_count score_10_count : mal_score.score_10_count score_09_count : mal_score.score_09_count score_08_count : mal_score.score_08_count score_07_count : mal_score.score_07_count score_06_count : mal_score.score_06_count score_05_count : mal_score.score_05_count score_04_count : mal_score.score_04_count score_03_count : mal_score.score_03_count score_02_count : mal_score.score_02_count score_01_count : mal_score.score_01_count score : mal_score.score score_count : mal_score.score_count score_rank : mal_score.score_rank
watch	: favorites_count : mal_watch.favorites_count watching_count : mal_watch.watching_count completed_count : mal_watch.completed_count on_hold_count : mal_watch.on_hold_count dropped_count : mal_watch.dropped_count plan_to_watch_count : mal_watch.plan_to_watch_count total_count : mal_watch.total_count
genres	: mal_genres[array]
clubs	: mal_clubs[array]

# Data Cleaning

Data cleaning ini merupakan proses membersihkan data yang akan digunakan dalam proses data mining

Proses yang dilakukan meliputi normalisasi kolom (memisahkan setiap objek dalam embedded dokumen menjadi kolom sendiri), drop column, replace specific column, fill null column

```
df = pd.json_normalize(query_list, max_level=1)
df.head(3)
```

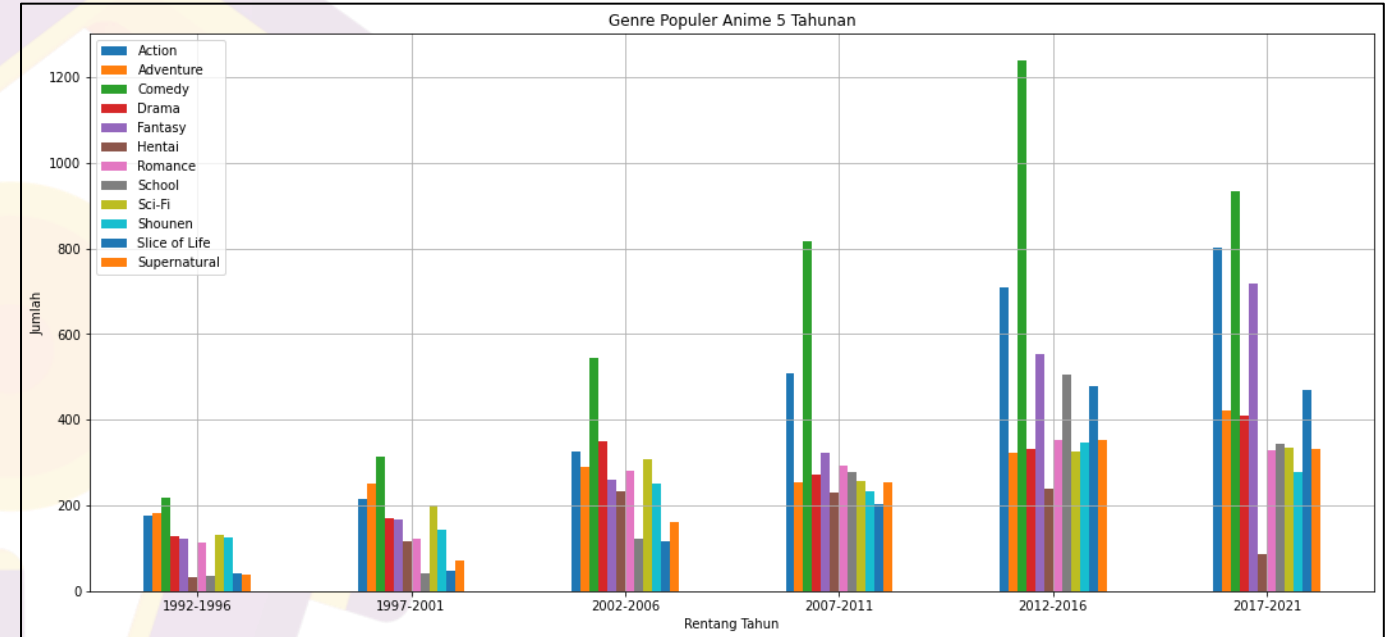
	_id	anime_id	title	synopsis	type	source_type
0	62d4f035aa797eed7380354f	2366	Touma Kishinden Oni	Shuramaru is hated and feared by the villagers...	TV	Game
1	62d4f036aa797eed73803550	4940	Sabaku no Kaizoku! Captain Kuno	Sometime in the future, the world was complete	TV	Manga



# Data Mining

Data Mining yang dilakukan dalam project ini yaitu mencari genre populer anime dalam jangka waktu lima tahunan dan faktor yang berpengaruh terhadap popularitas anime

Proses data mining dilakukan dengan pengelompokan data (klasifikasi) kemudian membuat representasi data tersebut.



# Kesimpulan

## Genre Populer Periode 5 Tahunan

Popularitas anime mulai terlihat pada tahun 1992 dengan jumlah 1344 atau lebih dari 5% keseluruhan anime. Dari seluruh anime yang ada dapat dikelompokkan menjadi beberapa genre yang menjadi favorit.

Period 2017-2021 menunjukkan bahwa genre Action, Comedy, dan Fantasy paling banyak diminati.

## Faktor yang Berpengaruh Terhadap Popularitas Anime

Anime yang baru dirilis memiliki kemungkinan lebih tinggi untuk masuk ke dalam anime populer. Anime dengan jenis serial lebih banyak diminati dibanding dengan movie. Anime yang diadaptasi dari Manga juga lebih diminati. Studio produksi anime tidak memberikan pengaruh yang signifikan terhadap popularitas anime. Selain itu, genre populer 5 tahunan juga relevan sebagai penentu popularitas anime, sehingga dengan rilisnya anime bergenre populer akan memiliki kemungkinan lebih tinggi untuk masuk ke dalam daftar anime populer.