

9_Prototipe Model Generatif Dengan LSTM

by Erizul Eriz

Submission date: 10-Apr-2023 12:20PM (UTC+0700)

Submission ID: 2060290221

File name: 9_Prototipe_Model_Generatif_Dengan_LSTM_Untuk.pdf (3.32M)

Word count: 3846

Character count: 22532

Prototipe Model Generatif Dengan LSTM Untuk Penciptaan Lagu Campur Sari Didi Kempot

R.Soelistijadi, Eri Zuliarso, Isworo Nugroho

^{1,2,3}Universitas Stikubank; Jl. Tri Lomba Juang No 1 Semarang, (024)8311668

^{1,2}Progdi Teknik Informatika, FTII UNISBANK, Semarang

³Progdi Sistem Informasi, FTII UNISBANK, Semarang

e-mail: *¹r.soelistijadi@edu.unisbank.ac.id, ²eri299@edu.unisbank.ac.id, ³wwk6259@edu.unisbank.ac.id

Abstrak

Penyanyi musik campursari Alm. Didi Kempot meninggalkan karya berupa 80 album berisi sekitar 700 lagu dimana 98% lagunya diciptakan sendiri. Oleh karenanya sepeninggal beliau dibutuhkan penciptaan lagu campursari baru agar eksistensi musik tersebut tetap terjaga. Berdasarkan latar belakang tersebut maka penelitian ini mencoba membangun model generatif untuk menghasilkan lagu baru yang memiliki lirik yang hampir sama dengan lagu-lagu sebelumnya. Tahap pertama diambil berbagai lirik lagu campursari sebanyak 56 lirik lagu sebagai data testing kemudian dilakukan cleansing data dan disimpan dalam format excell sebagai dataset-nya. Selanjutnya masuk ke platform Kaggle dan menggunakan library Pandas untuk membaca dataset tersebut dan melakukan clustering untuk melihat top of terms. Langkah berikutnya dengan menggunakan library Keras dibangun model sequential dengan arsitektur Neural Network pada LongShort-TermMemory (LSTM). Agar sistem dapat menghasilkan lirik lagu sesuai yang diinginkan maka di generate per baris lirik lagu dengan pemrosesan jumlah epochs sebanyak 100 kali. Hasil yang diperoleh dari penelitian ini yaitu berupa prototipe karena dalam tahap eksperimen hasil akhirnya bisa memiliki banyak varian lagu baru. Oleh karenanya ukuran kinerja yang digunakan yaitu tingkat keberhasilan dalam menghasilkan musik campursari baru yang berbeda lirik lagunya namun memiliki tema yang sama.

Kata kunci— Campursari, LSTM, Kaggle

Abstract

The campursari music singer Alm. Didi Kempot left a work in the form of 80 albums containing about 700 songs of which 98% of the songs were composed by himself. Therefore, after his death, it was necessary to create a new campursari song so that the existence of the music was maintained. Based on this background, this research tries to build a generative model to produce new songs that have almost the same lyrics as the previous songs. The first stage was taken various song lyrics of campursari as many as 56 song lyrics as data testing then data cleansing was carried out and stored in excel format as the dataset. Then go to the Kaggle platform and use the Pandas library to read the dataset and perform clustering to see the top of terms. The next step using the Keras library is to build a sequential model with the Neural Network architecture on LongShort-TermMemory (LSTM). In order for the system to produce song lyrics as desired, they are generated per line of song lyrics by processing the number of epochs 100 times. The results obtained from this study are in the form of prototypes because in the experimental stage the final results can have many variants of new songs. Therefore, the performance measure used is the level of success in producing new campursari music that has different song lyrics but has the same theme.

Keywords— Campursari, LSTM, Kaggle

1 PENDAHULUAN

Sejak dulu kehidupan manusia tidak lepas dari seni musik. Secara linguistik, arti seni musik terdiri dari dua kata, yaitu “seni” dan “musik”. Seni adalah suatu dimensi ciptaan atau perasaan manusia yang dituangkan ke dalam media tertentu untuk dibagikan kepada orang lain. Sedangkan kata “musik” berasal dari kata *mousikos* (red. Yunani) yang berarti dewa keindahan yang memiliki kekuatan di bidang seni dan ilmu pengetahuan [1]. Selanjutnya pengertian seni musik itu sendiri diartikan sebagai suatu bidang keilmuan atau aliran seni yang menggunakan nada dan suara untuk menyampaikan pesan, ekspresi, atau nilai seni kepada orang lain dalam satu kesatuan [2]. Sesuai dengan perkembangan zaman maka jenis seni musik semakin bervariasi, salah satunya adalah musik campursari yang merupakan perpaduan antara gamelan dan musik modern.

Menurut [3] musik campursari merupakan kolaborasi musik gamelan tradisional Jawa dengan alat musik modern, yaitu musik elektrik (elemen *keyboard*) yang saat ini sedang sangat digemari masyarakat hingga ke luar Jawa. Salah satu tokoh penyanyi yang fenomenal dalam musik campursari yaitu Alm. Didi Kempot yang meninggal pada tahun 2020. Selain sebagai penyanyi, ia juga seorang pencipta lagu yang produktif dengan lebih dari 80 album berisi sekitar 700 lagu. Sebagian besar lagu yaitu sekitar 98%, digubah sendiri, sedangkan sisanya 2% diciptakan oleh musisi lain [4]. Oleh karena itu pada saat beliau meninggal, diperlukan seorang pengganti yang bisa menciptakan lagu-lagu campursari baru agar eksistensi music tersebut tetap terjaga.

Permasalahan yang muncul yaitu bahwa selama ini penciptaan lagu-lagu campursari hanya dapat dilakukan oleh mereka yang berkecimpung dalam musik tersebut. Jika orang awam akan menciptakannya, akan sulit karena mereka tidak begitu hapal macam-macam lirik lagu yang terkandung di dalamnya sehingga membutuhkan lebih banyak waktu untuk membuatnya. Selain itu, musik campursari yang berkembang sangat pesat juga mengalami apa yang disebut *disrupsi teknologi digital*. Yang dimaksud yaitu sebuah lompatan perubahan dari sistem lama ke cara-cara baru, dimana hal ini merupakan fenomena yang menyebabkan terjadinya perubahan pemahaman konvensional masyarakat dengan segala aktivitasnya ke sistem digital [5]. *Disrupsi* juga mengubah teknologi lama yang lebih banyak menggunakan teknologi fisik / manual menjadi digital / robot sehingga menghasilkan sesuatu yang benar-benar baru, lebih efisien dan lebih cepat.

Berkaitan dengan hal itu maka solusi yang ditawarkan yaitu penggunaan model algoritma generatif yang berisi instruksi atau langkah dalam membuat rancangan lirik lagu baru dengan menggunakan komputer. Algoritma generatif sendiri merupakan suatu algoritma yang berisi instruksi atau langkah dalam membuat atau menggenerasi rancangan baru dengan menggunakan komputer sehingga memungkinkan *user* membuat sebuah model dari yang simple sampai yang rumit tanpa pengetahuan pemrograman atau *script*. [6]. Sebagai contoh, untuk menciptakan sebuah lagu campursari baru maka si pencipta tidak perlu tahu dan hapal lirik maupun syair campursari, dia cukup memproses berbagai lirik lagu yang telah ditulis sebelumnya ke dalam sistem komputer. Hasil pemrosesan tersebut berupa prediksi ataupun ciptaan syair lagu campursari baru yang didasarkan pada data lirik lagu yang sudah disimpan sebelumnya. Dalam hal ini lirik lagu tersebut direpresentasikan dalam data yang berurutan yang diekspresikan dalam rangkaian lirik lagu yang merentang sepanjang dimensi waktu. Dalam bentuk data sekuensial semacam ini, informasi baru dapat diperoleh jika serangkaian data sekuensial dapat diolah. Untuk dapat menangani data sekuensial seperti ini, arsitektur yang digunakan yaitu *Recurrent Neural Network (RNN)* yang merupakan salah satu bentuk arsitektur Jaringan Syaraf Tiruan (JST) yang dirancang khusus untuk mengolah data sekuensial [7]. Pada model ini, RNN dilatih untuk menghafalkan suatu *dataset* lalu RNN disuruh mengeluarkan apa yang telah diingatnya

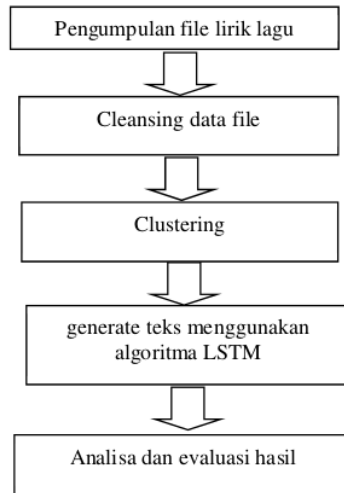
sebagai sebuah keluaran dimana dalam RNN digunakan n ump: balik dari nilai output sebelumnya untuk digunakan pada output saat ini. Dengan kata lain keluarannya menjadi salah satu masukan bagi proses selanjutnya.

Agar lebih optimal, RNN akan melakukan *unrolling* yang akan mengurai *loop* pada RNN menjadi untaian *clone* sepanjang jumlah iterasi sehingga bentuknya menjadi lebih linier. Dalam hal ini jumlah repetisi berubah menjadi jumlah *layer* dimana jika jumlah *layer* mencapai lebih dari lima maka komposisi lirik lagu akan menghadapi masalah *vanishing gradient*. Untuk mengatasi hal tersebut maka dikembangkan suatu arsitektur dengan memanfaatkan sistem gerbang [8]. Sistem gerbang yang digunakan yaitu LongShort-TermMemory (LSTM) yang pertama kali dibuat oleh Sepp Hochreiter dan Jürgen Schmidhuber pada tahun 1997. Dalam hal ini, metode LSTM untuk menangani masalah *vanishing gradient* yaitu dengan menggunakan *cell state* yang berfungsi sebagai memori untuk *layer* di mana nilai *cell state* dapat dimanipulasi menggunakan sistem gerbang (*gating system*). Sistem gerbang ini sendiri terdiri dari beberapa arsitektur jaringan syaraf tiruan yang digunakan untuk mengatur kapan data harus disimpan, digunakan, atau dilupakan [8].

Sementara itu berkaitan dengan pengumpulan data (*dataset*) maka digunakan *data science* yang merupakan media untuk mengolah data diberbagai bidang. *Data science* sendiri merupakan metode *forecasting* yang digunakan untuk mencari nilai yang diperoleh di waktu akan datang dengan menggunakan data-data sebelumnya untuk menghasilkan kinerja hasil berikutnya dengan menggunakan suatu metode [9]. Sedangkan untuk pengoperasian *dataset*-nya dalam penelitian ini digunakan platform *Kaggle* yang merupakan situs yang menyediakan *dataset*, kode *data science*, dan *notebook* dalam satu paket komputasi yang dapat dijalankan secara online [10]. Saat ini *Kaggle* merupakan salah satu situs yang terkenal di dunia Data Science dan Machine Learning yang memiliki lebih dari 6000 *dataset* yang dapat diunduh dalam format CSV [11]. Oleh karena itu berdasarkan uraian diatas maka penelitian ini akan menggunakan model generatif RNN dan arsitektur LSTM dengan platform *Kaggle* untuk merancang lirik lagu baru campursari berdasarkan lirik-lirik lagu campursari yang sudah ada sebelumnya.

2. METODE PENELITIAN

Penelitian ini dirancang dengan metode sebagai gambar 1 berikut ini:

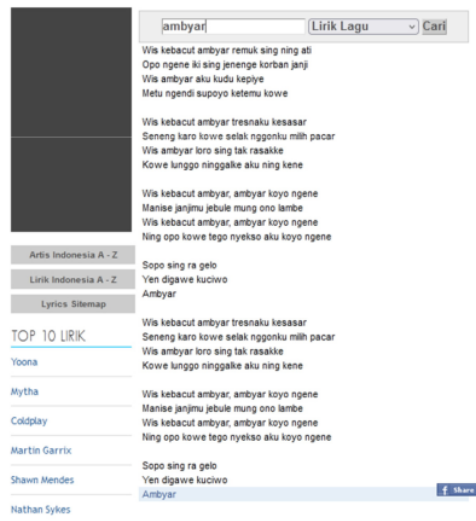


Gambar 1. Diagram Blok Metode Perancangan

Berikut uraian dari tahapan-tahapannya metode perancangan sistemnya yaitu:

2.1. Pengumpulan Lirik-lirik lagu

Data dikumpulkan secara manual dengan mengambil berbagai lirik lagu campursari sebanyak 56 lirik lagu sebagai data *testing* seperti contoh pada gambar 2 sebuah lirik lagu yang berjudul *Ambyar*.

Gambar 2. Lirik Lagu *Ambyar*

2.2. *Cleansing Data File*

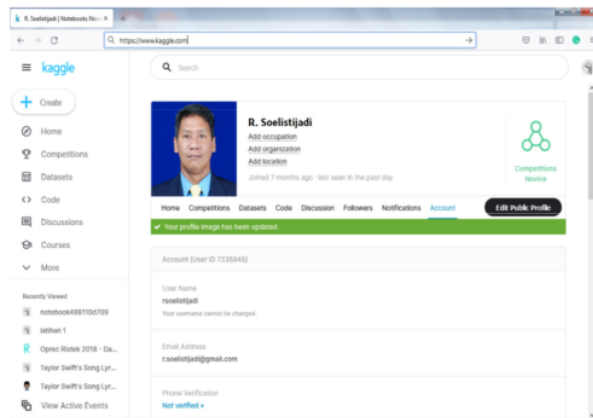
Untuk memudahkan pengolahan datanya dilakukan proses *case folding*, *tokenizing*, dan *stopword removal* untuk menghasilkan data set yang disimpan dalam format excell dengan nama *didikempot.csv*. Proses yang dilakukan yaitu pembersihan data dengan cara proses mendeteksi dan memperbaiki (atau menghapus) data set yang salah, rusak, tidak akurat, tidak lengkap dan salah format seperti : tag html, tanda baca, karakter selain huruf. Sebagai contoh pada tabel 1 tentang lirik lagu campursari yang berjudul *Ambyar*.

Tabel 1. *Cluster lirik lagu Ambyar*

line	track_title	lyric
1	ambyar	Wis kebacut ambyar remuk sing ning ati
2	ambyar	Opo ngene iki sing jenenge korban janji
3	ambyar	Wis ambyar aku kudu kepiye
4	ambyar	Metu ngendi supoyo ketemu kowe
5	ambyar	Wis kebacut ambyar tresnaku kesasar
6	ambyar	Seneng karo kowe selak nggonku milih pacar
7	ambyar	Wis ambyar loro sing tak rasakke
8	ambyar	Kowe lunggo ninggalke aku ning kene
9	ambyar	Wis kebacut ambyar, ambyar koyo ngene
10	ambyar	Manise janjimu jebule mung ono lambe
11	ambyar	Wis kebacut ambyar, ambyar koyo ngene
12	ambyar	Ning opo kowe tego nyekso aku koyo ngene
13	ambyar	Sopo sing ra gelo
14	ambyar	Yen digawe kuciwo
15	ambyar	Ambyar
16	ambyar	Wis kebacut ambyar tresnaku kesasar
17	ambyar	Seneng karo kowe selak nggonku milih pacar
18	ambyar	Wis ambyar loro sing tak rasakke
19	ambyar	Kowe lunggo ninggalke aku ningkene
20	ambyar	Wis kebacut ambyar, ambyar koyo ngene
21	ambyar	Manise janjimu jebule mung ono lambe
22	ambyar	Wis kebacut ambyar, ambyar koyo ngene
23	ambyar	Ning opo kowe tego nyekso aku koyo ngene
24	ambyar	Sopo sing ra gelo
25	ambyar	Yen digawe kuciwo
26	ambyar	Ambyar

2.3. *Clustering*

Selanjutnya *dataset* tersebut dilakukan *clustering* dengan cara masuk ke platform *Kaggle* seperti pada gambar 3 berikut ini.



Gambar 3. Platform *Kaggle* Untuk *Clustering* Lagu Campur Sari

Setelah masuk dalam platform *Kaggle* maka dilakukan uji coba dengan cara membuat perintah dengan library *Pandas* untuk membaca *didikempot.csv* sebagai *dataset*-nya yang ditunjukkan pada gambar 4. Untuk selanjutnya perintah dalam aplikasi ini direalisasikan dengan serangkaian unit program berbentuk CLI (*Command Line Interface*).

```
#Load the dataset
dataset = pd.read_csv('../input/lirikcsv/didikempot.csv', encoding = "latin1")
```

Gambar 4. Library *Pandas*

Langkah berikutnya berdasarkan data mentah sebanyak 56 lagu kemudian dilakukan *clustering* dengan *K means* sebagai *top of terms* perintah program seperti terlihat pada gambar 5 berikut ini.

```
:
KMeans(n_clusters=5, n_jobs=-1, random_state=99)

:
labels = model.labels_
cluster_center=model.cluster_centers_

:
#Reference credit - to find the top 10 features of cluster centriod
#https://stackoverflow.com/questions/47452119/kmean-clustering-top-terms-in-cluster
print("Top terms per cluster:")
order_centroids = model.cluster_centers_.argsort()[:, :-1]
terms = count_vect.get_feature_names()
for i in range(5):
    print("Cluster %d:" % i, end='')
    for ind in order_centroids[i, :20]:
        print(' %s' % terms[ind], end='')
    print()
```

Gambar 5. *Top Terms Per Cluster*

2.4. Generate Teks

Setelah proses clustering dilakukan maka tahap selanjutnya membangun arsitektur *Neural Network* pada LSTM berdasarkan *library Keras* dengan perintah:

```
# keras module for building LSTM
from keras.preprocessing.sequence import pad_sequences
from keras.layers import Embedding, LSTM, Dense, Dropout
from keras.preprocessing.text import Tokenizer
from keras.callbacks import EarlyStopping
from keras.models import Sequential
import keras.utils as ku

import pandas as pd
import numpy as np
import string, os

import warnings
warnings.filterwarnings("ignore")
warnings.simplefilter(action="ignore", category=FutureWarning)
```

Gambar 6. Perintah library Keras Untuk Membangun Arsitektur LSTM

Hasil dari perintah diatas digunakan untuk mengimport LSTM sehingga dihasilkan Model Sequential berupa tabel seperti terlihat pada gambar 7 berikut ini.

```
Model: "Sequential"
Layer (type)                 Output Shape         Param #
-----
Embedding (Embedding)        (None, 668)         67468
LSTM (LSTM)                   (None, 668)         1000000
Dense (Dense)                 (None, 668)         445376
Dropout (Dropout)            (None, 668)         0
Dense (Dense)                 (None, 668)         445376
Total params: 2119830
```

```
model, epochs=100, verbose=5)
```

Gambar 7. Model Sequential LSTM

Agar sistem dapat menghasilkan lirik lagu sesuai yang diinginkan maka perlu *generate* per baris lirik dari lagu sebelumnya dengan menggunakan algoritma LSTM tersebut. Untuk itu dilakukan pemrosesan / eksekusi untuk membangun model LSTM dengan jumlah *epochs* sebanyak 100 kali seperti gambar berikut ini:


```
model.fit(predictors, label, epochs=100, verbose=5)

Epoch 1/100
Epoch 2/100
Epoch 3/100
Epoch 4/100
Epoch 5/100
Epoch 6/100
Epoch 7/100
Epoch 8/100
Epoch 9/100
Epoch 10/100
Epoch 11/100
Epoch 12/100
Epoch 13/100
Epoch 14/100
Epoch 15/100
Epoch 16/100
Epoch 17/100
Epoch 18/100
Epoch 19/100
Epoch 20/100
Epoch 21/100
Epoch 22/100
Epoch 23/100
Epoch 24/100
Epoch 25/100
Epoch 26/100
Epoch 27/100
Epoch 28/100
Epoch 29/100
Epoch 30/100
Epoch 31/100
Epoch 32/100
Epoch 33/100
-----
```

Gambar 8. Uji coba *epochs*

Secara keseluruhan dalam tahap *Generate Teks* ini terdapat tiga bagian utama yaitu mencari lagu yang memiliki lirik lagu mirip dengan teks yang ditampilkan, memisahkan lirik yang diambil menjadi per baris, menampilkan lirik lanjutan setelah teks yang ditampilkan sebelumnya.

2.4. Analisa dan Evaluasi Hasil

Kemudian program untuk penghitungan frekuensi kata (*Term Frekequency*) dari sejumlah 54 syair lagu campur sari tersebut yang dapat dilihat pada gambar 9 berikut ini.

```
# Generating bag of words features.
from sklearn.feature_extraction.text import CountVectorizer
count_vect = CountVectorizer()
bow = count_vect.fit_transform(lyrics)
bow.shape
```

Gambar 9. Perintah untuk frekuensi kata

Dilanjutkan dengan program untuk mencari kata-kata yang sering muncul dari sejumlah syair lagu campur sari tersebut dengan batasan 20 kata teratas. Perintahnya dapat dilihat pada gambar 10 berikut ini.

```
cv = CountVectorizer()  
words = cv.fit_transform(lyrics)  
sum_words = words.sum(axis=0)  
  
words_freq = [(word, sum_words[word, idx]) for word, idx in cv.vocabulary_.items()]  
words_freq = sorted(words_freq, key = lambda x: x[1], reverse = True)  
frequency = pd.DataFrame(words_freq, columns=['word', 'freq'])  
  
color = plt.cm.twilight(np.linspace(0, 1, 20))  
frequency.head(20).plot(x='word', y='freq', kind='bar', figsize=(15, 7), color = color)  
plt.title("Most Frequently Occuring Words - Top 20")
```

Text(0.5, 1.0, 'Most Frequently Occuring Words - Top 20')

Gambar 10. Perintah Untuk Mencari 20 Kata Teratas

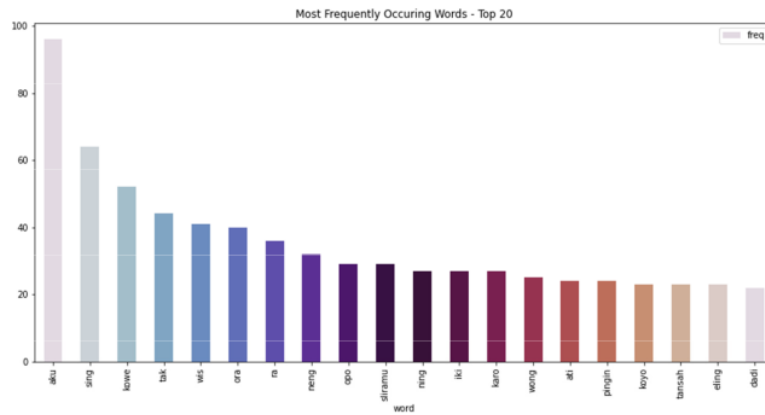
Dari hasil penghitungan tersebut maka akan dihasilkan *Words Cloud* tentang kata-kata yang paling sering muncul yaitu 20 kata yang meliputi:



Gambar 11. *Word Cloud* 20 kata teratas

3. HASIL DAN PEMBAHASAN

Berdasarkan frekuensi kata (*Term Frekequency*) maka kata yang paling sering muncul yaitu kata “*Aku*” seperti yang ditunjukkan pada gambar 12 dibawah ini.



Gambar 12. Histogram Term Frekuensi

Mengacu gambar Histogram di atas maka banyaknya jumlah kata “Aku” menunjukkan bahwa syair lagu campur sari tersebut seolah-olah menggambarkan pada orientasi cerita pengalaman yang ingin disampaikan oleh penyanyinya. Selain itu kata “Aku” merupakan kata ganti orang pertama tunggal yang menunjukkan status *egaliter* atau minimal setingkat dengan lawan bicara yang biasa digunakan dalam komunikasi searah seperti puisi, lagu dan lainnya. Hal ini akan berbeda apabila kata ‘aku’ sebagai kata ganti orang pertama tunggal diganti dengan kata lain seperti kata “kulo” atau kata “dalam” yang menunjukkan status yang tinggi rendah dalam masyarakat. Oleh karena disebabkan minimal setingkat dengan lawan bicara tersebut maka tidak heran kalau syair lagu campur sari yang dinyanyikan alm. Didi Kempot terlihat lebih santai dan akrab sehingga bisa diterima oleh masyarakat.

Selanjutnya apabila diurutkan berdasarkan urutan kata-kata yang paling sering muncul sampai dengan urutan kata yang hanya muncul cuma 1 kali maka bisa ditunjukkan pada gambar 13 berikut ini.

```
tokens = nltk.tokenize.word_tokenize(raw_text)
kemunculan = nltk.FreqDist(tokens)
print(kemunculan.most_common())#
```

```
[('aku', 96), ('sing', 64), ('kome', 52), ('tak', 44), ('wis', 41), ('ora', 40), ('ra', 36), ('neng', 32), ('siramu', 29), ('opo', 28), ('ning', 27), ('ik
i', 27), ('karo', 27), ('', 25), ('wong', 25), ('pingin', 24), ('koyo', 23), ('tansah', 23), ('eling', 23), ('ati', 22), ('dadi', 22), ('sopo', 21), ('ati
ku', 21), ('jme', 19), ('nganti', 19), ('mung', 17), ('rasono', 17), ('ambay', 16), ('ngene', 16), ('ono', 16), ('trusaku', 15), ('gelo', 15), ('lan', 1
5), ('wangi', 15), ('loro', 14), ('rasane', 14), ('lali', 14), ('game', 14), ('biso', 13), ('kangen', 13), ('kesetru', 13), ('ngerti', 12), ('kudu', 11),
('malah', 11), ('nanging', 10), ('pancen', 10), ('ayu', 10), ('kebacut', 9), ('ing', 9), ('kudi', 9), ('usid', 9), ('saki', 9), ('janji', 8), ('ketemu',
8), ('lambe', 8), ('sak', 8), ('melak', 8), ('kapan', 8), ('dalam', 8), ('amper', 8), ('nyaman', 8), ('nglarami', 8), ('tu', 7), ('seprene', 7), ('saka', 7),
('nelingso', 7), ('uriku', 7), ('cah', 7), ('ngendi', 6), ('keme', 6), ('rondo', 6), ('anak', 6), ('bedo', 6), ('merapi', 6), ('waliko', 6), ('janjine',
6), ('lungane', 6), ('esuk', 6), ('tekane', 6), ('liang', 6), ('lilakno', 6), ('kok', 6), ('kepiye', 5), ('tego', 5), ('...', 5), ('muto', 5), ('nangis
', 5), ('akuh', 5), ('sadayne', 5), ('kembang', 5), ('soko', 5), ('trusamu', 5), ('abot', 5), ('udun', 5), ('karepe', 5), ('adu', 5), ('krungu', 5), ('k
elingan', 5), ('ngiri', 5), ('margo', 5), ('kulun', 5), ('terminal', 5), ('karp', 5), ('turu', 5), ('nggonku', 4), ('liyo', 4), ('gusti
', 4), ('anakku', 4), ('ange-ange', 4), ('orong-orong', 4), ('nggawe', 4), ('monong', 4), ('nadyan', 4), ('nggo', 4), ('gadang', 4), ('deme', 4), ('neh',
4), ('seje', 4), ('ademe', 4), ('gumung', 4), ('purbo', 4), ('lungo', 4), ('mulih', 4), ('nasibu', 4), ('cidro', 4), ('dheue', 4), ('nggare', 4), ('tekan
', 4), ('upu', 4), ('cabay', 4), ('bulan', 4), ('sedung', 4), ('kresnan', 4), ('kebaye', 4), ('lintang', 4), ('Lungga', 4), ('sesandingan', 4), ('nggo
usi', 4), ('nangisi', 4), ('salahku', 4), ('nyatane', 4), ('mugo', 4), ('kebangetan', 4), ('awen', 4), ('lagi', 4), ('kiepek', 4), ('sansoyo', 4), ('jengeng
e', 3), ('seneng', 3), ('kongo', 3), ('adus', 3), ('kringet', 3), ('mesti', 3), ('siji', 3), ('ming', 3), (''limo', 3), ('e', 3), ('ompong',
3), ('sawo', 3), ('podo', 3), ('wae', 3), ('kabet', 3), ('gandrung', 3), ('nggileki', 3), ('malika', 3), ('waw', 3), ('sesejan', 3), ('ebe', 3), ('bungah
ing', 3), ('blyen', 3), ('maliboro', 3), ('seksenono', 3), ('lelakonku', 3), ('ngenteni', 3), ('lingsir', 3), ('angin', 3), ('ngelingake', 3), ('tresnan
i', 3), ('seng', 3), ('sore', 3), ('yo', 3), ('wonge', 3), ('saben', 3), ('mblerjani', 3), ('maneh', 3), ('tebu', 3), ('ngawi', 3), ('jodone', 3), ('dikapak
ke', 3), ('mak', 3), ('dipekso', 3), ('segganing', 3), ('ugesi', 3), ('kaengan', 3), ('entenono', 3), ('adok', 3), ('peroyo', 3), ('sae', 3), ('jane',
3), ('sih', 3), ('kerop', 3), ('kang', 3), ('prassat', 3), ('kesasar', 2), ('selak', 2), ('milih', 2), ('pacar', 2), ('rasakke', 2), ('lunggo', 2), ('nin
galke', 2), ('manise', 2), ('janjmu', 2), ('jebile', 2), ('nyekso', 2), ('digame', 2), ('kuciso', 2), ('lilo', 2), ('penyumuuku', 2), ('puji', 2), ('priy
o', 2), ('adu', 2), ('netes', 2), ('peresan', 2), ('kadir', 2), ('randene', 2), ('ndane', 2), ('kingkong', 2), ('sak', 2), ('wayah', 2), ('teman', 2), ('
kulinno', 2), ('ngentake', 2), ('ino', 2), ('dionong', 2), ('tonggo', 2), ('gunjingane', 2), ('konco', 2), ('luntur', 2), ('urip', 2), ('mugil', 2), ('kerso
', 2), ('lakuku', 2), ('leaku', 2), ('nuruti', 2), ('siram', 2), ('tetesing', 2), ('eluhku', 2), ('suk', 2), ('pirang-pirang', 2), ('sasi', 2), ('pen
dak', 2), ('menjil', 2), ('sakueto', 2), ('telang', 2), ('ningu', 2), ('kabarumu', 2), ('nggal', 2), ('nggolek', 2), ('gantine', 2), ('suono', 2), ('pa
ntes', 2), ('dawe', 2), ('ro', 2), ('eluh', 2), ('ueruh', 2), ('ngaleno', 2), ('ngalen', 2), ('banyu', 2), ('langit', 2), ('dhuwur', 2), ('shyangan', 2),
('watu', 2), ('gedhe', 2), ('kalingan', 2), ('mendunge', 2), ('telesono', 2), ('atine', 2), ('kassarane', 2), ('setyo', 2), ('melu', 2), ('suaramu', 2), ('n
gomogke', 2), ('langgan', 2), ('wosowari', 2), ('Sugakarta', 2), ('sawe-sawe', 2), ('pamit', 2), ('semene', 2), ('suwene', 2), ('tenten', 2), ('ndino
```

Gambar 13. Urutan Kata-kata yang Sering Muncul pada Term Frekuensi

Dari gambar di atas maka terlihat kata yang sering keluar yaitu kata “aku” sebanyak 96 kata yang diikuti kata “sing” (64 kata), kata “kowe” (52) dan seterusnya. Dalam hal ini apabila 3 kata yang muncul pertama tersebut digabung akan menjadi kalimat “Aku Sing Kowe....” yang dapat diterjemahkan sebagai “Aku Yang Kamu....”. Selanjutnya apabila ditambahkan kata berikutnya berdasarkan urutannya yaitu 4 kata berikutnya: kata “tak” (44 kata), kata “wis” (41 kata), kata “ora” (40 kata), kata “ra” (36 kata) yang kesemuanya secara eksplisit mengacu pada arti awalan “kata keterangan”. Oleh karena itu berdasarkan urutan kata-kata tersebut apabila ditarik polanya maka akan muncul syair lagu yang menyiratkan bahwa “Aku” menerima apa “Yang” dilakukan oleh “Kamu”.

Disisi lain Almarhum Didi Kempot sering dijuluki sebagai *Lord of Broken Heart* mengingat lagu-lagu yang diciptakan maupun dinyanyikan sebagian besar bernada patah hati. Penelitian ini sejalan dengan pendapat tersebut dimana dibuktikan dari beberapa *cluster* (*K means using TFIDF*) yang telah dibuat maka rata-rata menunjukkan syair lagu yang bernada hidup susah karena masalah asmara. Sebagai contoh hasil *cluster* dengan *top of terms* sebanyak 20 kata tampak terlihat pada tabel 2 berikut ini :

Tabel 2. *Top of terms* Dalam Masing-masing Cluster

Cluster 0: tak	Cluster 1: aku	Cluster 2:sing	Cluster 3: nglarani	Cluster 4: wong
kowe	kowe	ra	uripku	koyo
ora	pingin	dadi	atiku	ngene
wis	tansah	opo	kesetrum	sing
neng	ning	tak	khayangan	ambyar
karo	nyawan	nganti	ketiga	wis
ati	sliramu	lungane	ketiban	ra
sliramu	neng	wis	ketemu	kebacut
iki	ngerti	suwe	ketara	klepek
atiku	opo	sopo	ketaman	liyo
wengi	kangen	ning	kesiram	ora
mung	iki	ing	yèn	ono
tresnaku	eling	janjine	kingkong	seprene
yen	gelo	eling	kesasar	ning
rasane	tresno	ono	kertonegoro	kang
loro	karo	ati	kerso	seng
opo	to	lali	kerjo	ketaman
eling	sesandingan	ngene	kerep	ilang
sopo	ngarepe	ambyar	kepungkur	kasmaran
lali	gawe	kembang	kepriye	tuamu

Untuk lebih jelas maka dilakukan pengujian dengan melakukan serangkain percobaan sebanyak 3 kali. Pada percobaan ke 1 dilakukan *generate text* dengan 6 *cluster* dengan panjang kata masing-masing 11 kata. Hasil uji coba mengenerate lirik lagu yang baru tersebut maka dihasilkan syair lagu campur sari seperti gambar 14 berikut ini :

```
print (generate_text("tresno", 10, model, max_sequence_len))
print (generate_text("cah ayu", 10, model, max_sequence_len))
print (generate_text("sliramu", 10, model, max_sequence_len))
print (generate_text("janji", 10, model, max_sequence_len))
print (generate_text("lara ati", 10, model, max_sequence_len))
print (generate_text("jebul", 10, model, max_sequence_len))
```

Tresno Sing Tulus Pungkasane Dadi Ngene Ngene Ngene Ngene Gadah Dudu
Cah Ayu Entenono Tekaku Suaramu Aku Dodo Gadah Gadah Gadah Gadah
Sliramu Cidro Janji Sholat Ati Nggo Ati Lambe Gadah Sangganing Janji
Janji Banyune Enak Akeh Mendunge Udah Kuliahe Gadah Gadah Gadah
Lara Ati Janjimu Jebule Mung Ono Lambe Ati Gadah Gadah Janji Aku
Jebul Atiku Ngjarani Uripku Iki Ngene Ngene Aku Koyo Wong Andhuke

+ Code + Markdown

Gambar 14. *Generate Text 6 Cluster Dengan Panjang 11 Kata*

Berdasarkan percobaan di atas terlihat banyak ketidak sesuaian arti kata antara kata-kata dalam setiap baris maupun keterkaitan masing-masing baris yang satu dengan lainnya. Selanjutnya dilakukan percobaan ke 2 dengan *generate text* masih dengan 6 *cluster* dengan panjang kata masing-masing 6 kata hanya saja perbedaannya pada jumlah kata yaitu 2 kata pada setiap *cluster* sehingga secara keseluruhan menjadi 7 kata seperti yang terlihat pada gambar 15 berikut ini.

```
print (generate_text("tresno sliramu", 5, model, max_sequence_len))
print (generate_text("cah ayu", 5, model, max_sequence_len))
print (generate_text("sliramu janji", 5, model, max_sequence_len))
print (generate_text("janji palsu", 5, model, max_sequence_len))
print (generate_text("lara ati", 5, model, max_sequence_len))
print (generate_text("jebul ngapusi", 5, model, max_sequence_len))
```

Tresno Sliramu Biso Lali Eling Naliko Kae
Cah Ayu Entenono Tekaku Suaramu Dudu Telo
Sliramu Janji Seprene Sing Kepungkur Ra Gadang
Janji Palsu Awan Wengi Tansah Mbedo Ati
Lara Ati Koyo Ngene Kuwi Ing Ing
Jebul Ngapusi Aku Opo To Salahku Salahku

Gambar 15. *Generate Text 6 Cluster Dengan Panjang 7 Kata*

Dari percobaan ke 2 ternyata masih ditemui ketidak sesuaian arti kata pada setiap barisnya. Agar dapat diperoleh lirik lagu yang memiliki sedikit kesesuaian arti maka dilanjutkan dengan percobaan ke 3 yang dilakukan penyederhanaan jumlah kata hanya 1 kata setiap *cluster* dengan 3 kata pada variabel pada setiap barisnya. Hal didasarkan kenyataan bahwa pada umumnya tembang campur sari cenderung terdiri dari 3-5 kata setiap barisnya. Sebagai perbandingan perhatikan gambar berikut ini dengan 6 *cluster* dengan panjang kata masing-masing 3 kata seperti yang terlihat pada gambar 16 berikut ini.

```
print (generate_text("tresno", 2, model, max_sequence_len))  
print (generate_text("cah ayu", 2, model, max_sequence_len))  
print (generate_text("sliramu", 2, model, max_sequence_len))  
print (generate_text("janji", 2, model, max_sequence_len))  
print (generate_text("lara ati", 2, model, max_sequence_len))  
print (generate_text("jebul", 2, model, max_sequence_len))
```

Tresno Sing Tulus
Cah Ayu Entenono Tekaku
Sliramu Cidro Janji
Janji Banyune Enak
Lara Ati Janjimu Jebule
Jebul Atiku Nglarani

+ Code + Markdown

Gambar 16. *Generate Text* 6 Cluster Dengan Panjang 3 Kata

Apabila diperhatikan untaian syair pada gambar di atas maka arti dari terjemahan bebasnya yaitu tentang "... seseorang yang tulus mencintai seorang gadis dan berharap untuk menunggu kedatangannya yang ternyata gadis tersebut mengingkari janji sehingga membuat sakit hati ...". Oleh karena itu berdasarkan ke 3 percobaan diatas maka perlu juga diperhatikan pentingnya panjang kata dalam setiap baris.

4. KESIMPULAN

Almarhum Didi Kempot sering dijuluki sebagai *Lord of Broken Heart* mengingat lagu-lagu yang diciptakan maupun dinyanyikan sebagian besar bernada patah hati. Penelitian ini sejalan dengan pendapat tersebut dimana dibuktikan dari beberapa *cluster (K means using TFIDF)* yang telah dibuat maka rata-rata menunjukkan lirik lagu yang bernada hidup susah karena masalah asmara. Berdasarkan eksplorasi data yang ditunjukkan dengan gambar Histogram maka kata yang paling sering muncul pada lirik lagu campur sari alm. Didi Kempot yaitu kata "Aku" yang menunjukkan status *egaliter* yang berkesan lebih santai dan akrab sehingga bisa diterima oleh masyarakat. Selain itu berdasarkan percobaan *generate text 6 cluster* sebanyak 3 kali maka hasilnya yaitu panjang kata dalam setiap baris yang dianjurkan sebanyak 3-4 kata agar diperoleh lirik lagu yang memiliki sedikit kesesuaian arti setiap barisnya. Hasil akhir capaian dalam penelitian ini yaitu **dibangunnya sistem cerdas yang bertindak sebagai artis digital yang dapat menciptakan lirik lagu campursari baru yang memiliki kemiripan dengan lagu campursari buatan manusia.**

5. SARAN

Sedangkan saran yang dapat disampaikan dalam penelitian ini yaitu :

1. Model dalam penelitian ini dapat digunakan untuk menciptakan beberapa tipe / macam jenis lagu berdasarkan lirik lagu-lagu sebelumnya yang sudah ada.
2. Perlu penggunaan algoritma lain untuk membandingkan hasil model generatifnya.

UCAPAN TERIMA KASIH

Ijinkan kami mengucapkan terimakasih kepada Pimpinan Fakultas Tehnologi Informasi dan Industri UNISBANK yang sudah support sarana dan prasarana dalam penelitian serta Editor dan Reviewer Jurnal JATISI yang telah memberikan review dan koreksi sehingga artikel ini dapat terpublikasikan.

DAFTAR PUSTAKA

- [1] Gramedia, 2022, *Pengertian Seni Musik: Sejarah, Unsur, Jenis dan Fungsinya (On Line)*, URL : <https://www.gramedia.com/literasi/seni-musik/>
- [2] Bayu Satya D.S, 2021, *Pendidikan dan Seni Budaya Musik*. Andi offset.
- [3] D. Pariwisata, 2019, Campursari, Seni Musik, Dinas Pariwisata dan Kebudayaan DKI Jakarta, URL: <http://encyclopedia.jakarta-tourism.go.id/post/campursari--seni-musik?lang=id>
- [4] Rahmawati, L.P., 2017, *Eksistensi Musik Campursari karya Didi Kempot, Pend. Seni Musik-SI*, Vol.6, No.1 (2017) Fak. Bahasa dan Seni, UNY, Yogyakarta.
- [5] Udayana, Anak Agung Gde Bagus, 2020, "*Disrupsi Teknologi Digital: Tumbuh Kembangnya Industri Kreatif*", *Seminar Nasional ENVISI 2020: Industri Kreatif*, Universitas Ciputra, Jakarta
- [6] Ahmad Rizani Saputra, 2018, "*Penerapan Algoritma Generatif Sebagai Sarana Untuk Menjabatani Non-Enginerr User Dalam Menggunakan Additive Manufacturing pada Studi Kasus Pembuatan Orthosis*", Fak.. Rekayasa Industri, Univ. TELKOM
- [7] Aditya, Yanuar. R, 2018, *Menara Ilmu Machine Learning: Recurrent Neural Network (RNN)*, MIPA UGM. URL: <https://machinelearning.mipa.ugm.ac.id/2018/07/01/recurrent-neural-network-rnn/>
- [8] Lukman Zaman, Surya Sumpeno, Mochamad Hariadi, 2019, *Analisis Kinerja LSTM dan GRU Sebagai Model Generatif untuk Tari Remo, JNTETI*, Vol. 8, No. 2, Mei 2019
- [9] Witak, D.D., dkk, 2020, *Analisis Data Science pada Struktur Data kepadatan Penduduk Kota Malang, KURAWAL Jurnal Teknologi, Informasi dan Industri*, Vol. 3 No. 2 Oktober 2020
- [10] Purwono, 2022, *Belajar Data Science Dengan Kaggle Dilengkapi Dengan Praktikum Latihan Data Science Untuk Pemula*, UHB Press, [On Line]. URL : <https://www.researchgate.net/publication/362058470>
- [11] Kaggle, 2019, *Kaggle*, [On Line].URL : <https://www.kaggle.com/ronitf/heart-disease-uci>.

9_Prototipe Model Generatif Dengan LSTM

ORIGINALITY REPORT

13%

SIMILARITY INDEX

12%

INTERNET SOURCES

0%

PUBLICATIONS

8%

STUDENT PAPERS

MATCH ALL SOURCES (ONLY SELECTED SOURCE PRINTED)

7%

★ www.researchgate.net

Internet Source

Exclude quotes On

Exclude matches < 2%

Exclude bibliography On