MODUL 9 TUGAS BESAR

Raka Noorsyach (13218040)
Kevin Naoko (13218046)
Ian Azarya Aryanto (13218055)
Andy Lucky (13218058)

Asisten: Gabriella Hayley Tanudjaja
Tanggal Laporan: 08/04/2020
EL2208-Praktikum Pemecahan Masalah dengan C
Laboratorium Dasar Teknik Elektro - Sekolah Teknik Elektro dan Informatika ITB

Abstrak

1. DESKRIPSI PERMASALAHAN

a. Introduction to N-grams

Pemodelan linguistik, pada dasarnya berupa salah satu jenis pemodelan yang menetapkan probabilitas dalam beberapa deret kata. N-grams merupakan salah satu dari metode yang melakukan implementasi probabilitas dalam kalimat ataupun kata. Pada bidang *Natural Language Processing* (NLP), N- grams digunakan dalam berbagai hal. Salah satu contoh penggunaannya berupa *auto-completing text*, *spellcheck*, dan *plagiarism checker*.

b. N-grams Probability

Misalkan diberikan suatu input pada program "Terima kasih banyak atas bantuannya"dan diharapkan program mampu mengelompokan (*tokenize*) kumpulan dari kata – kata tersebut sesuai dengan berapa nilai dari n pada n-grams yang menentukan berapa banyak elemen yang terhimpun pada satu kelompok kata (*tokenize*).

Misalkan diberikan nilai n senilai n = 2. Maka dilakukan pengelompokan pada setiap 2 elemen kata pada kalimat tersebut dengan menggunakan n – 1 kata pertama dalam n-grams untuk memprediksi kata terakhir pada n – grams tersebut.

c. Background

Pada masalah yang diberikan, mahasiswa diharuskan untuk menyelesaikan permasalahan dengan mengeluarkan output berupa kumpulan kata – kata yang bersifat random, namun diharapkan kata dan kalimat random tersebut memiliki gaya atau sifat penulisan yang unik, sehingga digunakan konep dari N-grams yang merupakan salah satu konsep yang digunakan pada *Natural Language Programming* (NLP). Perlu diperhatikan juga input dari program berupa file external dengan format .txt, besarnya nilai N yang akan diproses dengan konsep N-grams, serta berapa banyak kata yang dicetak sebagai output.

2. ANALISIS PERMASALAHAN

a. Input External File As Reference (Toy Corpus)

Untuk memulai program, dilakukan input terhadap nama file external yang akan digunakan sebagai referensi. Dalam hal ini, file yang digunakan berupa text file (.txt). Kemudian input nama file tersebut digunakan sebagai file yang akan dibaca dalam program untuk dijadikan raw data untuk proses selanjutnya. Proses pembacaan dari file external dilakukan per baris dengan *fgets* yang kemudian disimpan dalam bentuk string *temporary*, untuk kemudian dilakukan tokenization.

b. Tokenization & Filtering

Tokenization merupakan sebuah proses memecah suatu text (dalam hal ini string temporary hasil *fgets*) menjadi per-kata. *Punctuation characters* atau tanda baca akan diasumsikan sebagai character biasa dan masuk menjadi satu kesatuan *word*, sehingga tidak perlu dilakukan *filtering* tanda baca. Pada permasalahan ini, proses tokenization menggunakan *strtok* untuk memecah satu persatu kata dari string kalimat temporary. Filtering yang dilakukan pada permasalahan ini adalah "\n" atau *new line*. Ketika ditemukan "\n" pada bagian token, maka "\n" akan diterminasi sehingga data token akan menjadi lebih bersih. Data yang telah ter-*tokenize* akan disimpan dalam array dalam bentuk *array of strings*.

c. Modeling

Modeling dimulai dengan input n-gram yang diinginkan oleh user. Dalam konsep dasar *N-gram word prediction*, semakin besar n yang digunakan, maka teks output yang dihasilkan akan semakin mendekati bentuk asli dari teks. Modeling dalam hal ini digunakan untuk menginisiasi pembentukan *language coverage* atau rasio total kemunculan *unique words*. Dalam hal ini, teks yang sudah ter-*tokenize* akan dikelompokkan kembali berdasarkan n-gram input.

Digunakan contoh permasalahan dalam naskah soal M9 soal nomor 2 (untuk genap). Contoh kalimat :

> "Ships at a distance have every man wish on board."

Untuk input n = 2, maka kalimat akan dikelompokkan dalam bentuk sebagai berikut :

Ships at
at a
a distance
distance have
have every
every man
man wish
wish on
on board.
board. Ships

Gambar 2.c.1 Contoh "N-Gram" Model Untuk n = 2

Untuk contoh lain digunakan teks sebagai berikut :

> "This is a short input file composed of spaces and of letters.
>
> It is also a file with end of line characters to help you to test"

Untuk input n = 4, akan dikelompokkan n-gram words sebagai berikut :

```
{This is a short }
{is a short input }
{a short input file }
{short input file composed }
{input file composed of }
{file composed of spaces }
{composed of spaces and }
{of spaces and of }
{spaces and of letters. }
{and of letters. It }
{of letters. It is }
{letters. It is also }
{It is also a }
{is also a file }
{also a file with }
{a file with end }
{file with end of }
{with end of line }
{end of line characters }
{of line characters to }
{line characters to help }
{characters to help you }
{to help you to }
{help you to test }
```

Gambar 2.c.2 Contoh "N-Gram" Model Untuk n = 4

Dari pengelompokkan tersebut, kelompok kata tersebut akan dimasukkan ke dalam suatu array tertentu untuk menginisiasi proses modeling untuk key dan value sebagai *look up table* yang akan digunakan dalam proses output.

d. Exploratory Analysis

Dalam proses sebelumnya, dilakukan pengelompokan data menjadi n-gram model sesuai dengan inputan yang dilakukan oleh user. Namun pengelompokkan tersebut belum mem-filter sekelompok kata yang duplikat, atau masih terdapat lebih dari satu n-gram model yang identik. Untuk mengatasi hal itu, dilakukan exploratory analysis untuk memfilter n-gram model yang masih memiliki duplikat menjadi key dan membentuk array value yang berisikan data n-gram value atau kemungkinan kemunculan kata selanjutnya. Untuk melakukannya, dilakukan komparasi antara array n-gram model dari proses modeling, dengan array key yang berisi data n-gram model yang eksklusif tanpa duplikat. Dalam hal ini, komparasi dilakukan dengan menggunakan *strcmp* untuk melihat kemiripan kedua string. Apabila *strcmp* menghasilkan integer tidak sama dengan 0 (!=0 / tidak sama), maka n-gram model dari array hasil Modeling akan dimasukkan ke array key. Dalam hal n-gram value, akan dilakukan pengambilan word dengan indeks +1 dari n-gram yang bersangkutan, dengan

menggunakan konsep *tokenization*. Word yang diambil adalah kata terakhir dari string dengan indeks +1, dan kemudian akan dijadikan value dari key pada indeks yang sedang dianalisis. Pada komparasi, apabila ditemukan 1 atau lebih kata yang identik dari array key sebelumnya, maka string tersebut tidak akan dimasukkan kembali kedalam key. Namun, akan dilakukan pengambilan word terakhir dari string dengan indeks +1, untuk dijadikan value, sehingga value dari key dapat berjumlah lebih dari satu. Key dan value yang diperoleh kemudian dijadikan *look up table* untuk proses selanjutnya.

Sebagai contoh, akan digunakan teks berikut :

"This is a short input file composed of spaces and of letters.

It is also a file with end of line characters to help you to test

your code and help you get started.

Isn't this the best assignment for you to solve?

Also, test your might, Mortal Kombat!"

Dari teks tersebut, akan diinput n = 2 sehingga terbentuk key dan value sebagai berikut :

Key	Value
This is	{ a }
is a	{ short }
a short	{ input }
short input	{ file }
input file	{ composed }
file composed	{ of }
composed of	{ spaces }
of spaces	{ and }
spaces and	{ of }
and of	{ letters. }
of letters.	{ It }
letters. It	{ is }
It is	{ also }
is also	{ a }
also a	{ file }
a file	{ with }
file with	{ end }
with end	{ of }
end of	{ line }
of line	{ characters }
line characters	{ to }
characters to	{ help }
to help	{ you }
help you	{ to , get }
you to	{ test , solve? }
to test	{ your }

Gambar 2.d.1 Contoh Hasil Exploratory Analysis Menjadi *Look Up Table*

Pada beberapa kasus, ditemukan key dengan value lebih dari 1 seperti pada "help you", "you to", dan "test your":

help you	{ to , get }
you to	{ test , solve? }
to test	{ your }
test your	{ code , might, }

Value-value tersebut yang nantinya akan dijadikan acuan sebagai prediksi keluaran berikutnya.

e. Frequency

Berjalan paralel dengan tahap sebelumnya, suatu array baru akan dibuat untuk merepresentasikan jumlah variasi dari value yang didapatkan dari proses Exploratory Analysis sebelumnya. Array ini nantinya akan digunakan untuk penentuan value yang akan dikeluarkan pada proses word prediction untuk memprediksi kata-kata yang akan keluar berikutnya.

f. *N-Gram Look Up Table*

Untuk memudahkan proses *debugging* dan validasi, akan dibuat suatu *Look Up Table* (LUT) yang merupakan pasangan dari array key dan value dari hasil pemrosesan pada proses Exploratory Analysis. LUT ini juga akan digunakan sebagai rujukan pada tahap *Output Phase*, yaitu tahap pemrosesan output.

g. Prediction

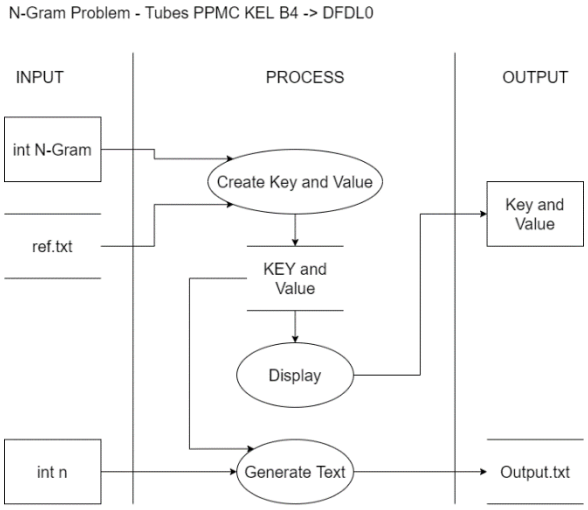
Apabila terdapat variasi value lebih dari 1 pada key yang sama, tentunya perlu dipilih salah satu dari variasi tersebut. Oleh karena itu, pada proses ini akan dilakukan model pengambilan value secara acak tanpa memperhitungkan frekuensi value tersebut muncul di teks. Hasil pada proses prediction ini merupakan word yang diambil dari array value untuk kemudian dijadikan patokan parameter pada *Output Phase*.

h. *Output Phase*

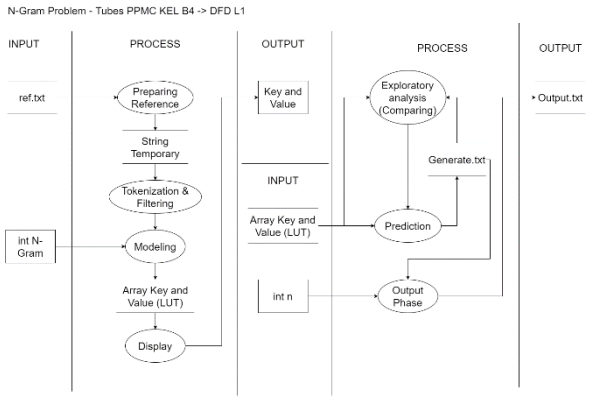
Pada tahap ini, akan dicetak hasil dari tabulasi data input dan hasil pemrosesan dari proses-proses sebelumnya. Program ini akan menerima masukkan pengguna berupa jumlah kata yang akan dicetak. Untuk kata pertama, kedua, hingga kata ke (n-1), akan dipilih dan di *print* kata-kata random yang akan di *sampling* dari kumpulan value yang ada. Setelah jumlah kata yang di pilih secara acak sama dengan jumlah n, maka program baru akan merujuk kumpulan key dari LUT yang telah dibuat pada proses Exploratory Analysis dan *N-Gram Look Up Table* untuk dilakukan penyusunan kata selanjutnya.

3. FLOWCHART USULAN ALGORITMA LENGKAP

3.1 DATA FLOW DIAGRAM



Gambar 3.1.1 Data Flow Diagram Level 0



Gambar 3.1.2 Data Flow Diagram Level 1

3.2 FLOWCHART ALGORTIMA LENGKAP

4. RENCANA PEMBAGIAN TUGAS

Tabel 4.1 Tabel rencana pembagian tugas sementara

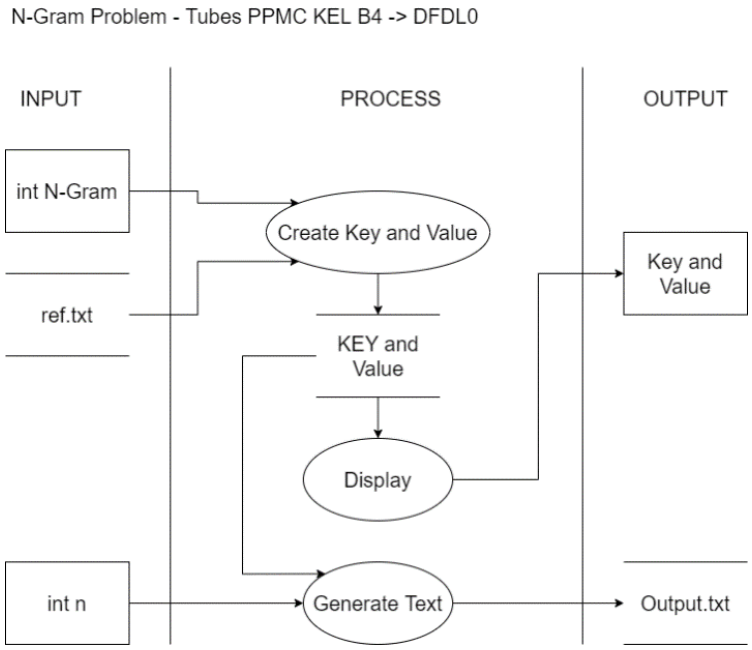
FUNGSI	BERKAS/	PROGRA	TESTER
	LIBRARY TERKAIT	MMER	
PREPARING REFERENCE	REF.H	IAN	KEVIN
TOKENIZATION & FILTERING	TOKEN.H	IAN	ANDY
MODELING	MODEL.H	KEVIN	RAKA
DISPLAY	DIS.H	ANDY	RAKA
EXPLORATORY ANALYSIS	EXP.H	KEVIN	IAN
PREDICTION	PREDICT.H	RAKA	ANDY
OUTPUT PHASE	OUT.H	RAKA	KEVIN
MENU USERINTERFACE	MAIN.C	ANDY	IAN

DAFTAR PUSTAKA

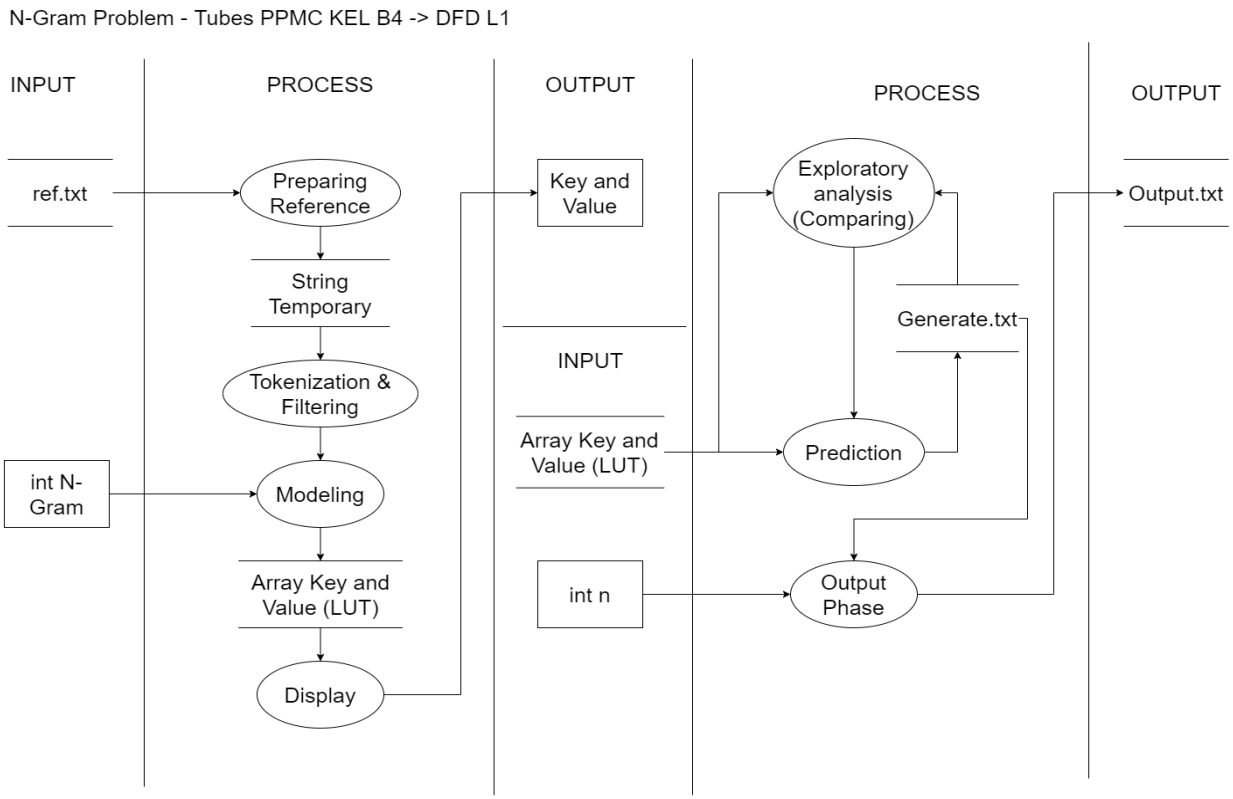
[1] Muhammad Naufal T., *Naskah Soal Tugas Besar: Tipe 2*, Program Studi Teknik Elektro ITB EL2208, Bandung, 2020.

[2] https://rstudio-pubs-static.s3.amazonaws.com/, diakses pada tanggal 8 April 2020 pada pukul 12.23.

Lampiran



Gambar 3.1.1 Data Flow Diagram Level 0



Gambar 3.1.1 Data Flow Diagram Level 1