

<<<<

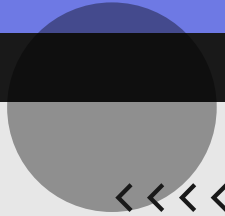
# ARTIFICIAL INTELLIGENCE (AI)

## PRAKTIKUM 6

Dosen Pengampu : Renovita Edelani S.ST., M.Tr.Kom

---

oleh : Dukhaan Kamimpangan  
2 D4 IT A  
3122600003



01.

# PRAKTIKUM



# TEXT BERITA



## VALORANT

Valorant adalah game yang dikembangkan dan diterbitkan oleh Riot Games. Permainan tembak-menembak taktis daring yang telah menarik perhatian para penggemar game di seluruh dunia. Valorant telah menjadi salah satu game yang paling banyak dimainkan dan disukai di dunia e-sports.

# TOKENIZED WORD




```
from nltk.tokenize import sent_tokenize, word_tokenize

text="Valorant adalah game yang dikembangkan dan diterbitkan oleh Riot Games. Permainan tembak-menembak taktis daring yang telah menarik perhatian para penggemar game di seluruh dunia. V
tokenized_text=sent_tokenize(text)
print(tokenized_text)


['Valorant adalah game yang dikembangkan dan diterbitkan oleh Riot Games.', 'Permainan tembak-menembak taktis daring yang telah menarik perhatian para penggemar game di seluruh dunia.', "
```

Output : ['Valorant', 'adalah', 'game', 'yang', 'dikembangkan', 'dan', 'diterbitkan', 'oleh', 'Riot', 'Games', '.', 'Permainan', 'tembak-menembak', 'taktis', 'daring', 'yang', 'telah', 'menarik', 'perhatian', 'para', 'penggemar', 'game', 'di', 'seluruh', 'dunia', '.', 'Valorant', 'telah', 'menjadi', 'salah', 'satu', 'game', 'yang', 'paling', 'banyak', 'dimainkan', 'dan', 'disukai', 'di', 'dunia', 'e-sports', '.']

# FILTERED LIST

```
0s  #REMOVING STOPWORDS
e_filtered_list = []
for word in tokenized_word:
    if word.casefold() not in i_stopwords:
        e_filtered_list.append(word)

e_filtered_list

 ['Valorant',
'game',
'dikembangkan',
'diterbitkan',
'Riot',
'Games',
'',
'Permainan',
'tembak-menembak',
'taktis',
'daring',
'menarik',
'perhatian',
'penggemar',
'game',
'dunia',
'',
'Valorant',
'salah',
'game',
'dimainkan',
'disukai',
'dunia',
'e-sports',
'']
```

Filter kata Stopwords Bahasa Indonesia yang ada di tokenized word menggunakan filtered list seperti 'di' main 'kan',

<- Output

# STREMMER

```
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory

factory = StemmerFactory()
i_stemmer = factory.create_stemmer()
i_stemmed_words = [i_stemmer.stem(word) for word in i_filtered_list]

i_stemmed_words

['valorant',
 'game',
 'kembang',
 'terbit',
 'riot',
 'games',
 '',
 'main',
 'tembak',
 'taktis',
 'daring',
 'tarik',
 'perhati',
 'gemar',
 'game',
 'dunia',
 '',
 'valorant',
 'salah',
 'game',
 'main',
 'suka',
 'dunia',
 'e-sports',
 '']
```

Proses Stremming akan mengurangi kata dari parent seperti 'ber' kembang menjadi kembang

<- Output

# DISTRIBUSI STEMMED WORD



```
from nltk.probability import FreqDist
fdist = FreqDist(lemmatized_words)
fdist

FreqDist({'game': 3, '.': 3, 'Valorant': 2, 'dunia': 2, 'dikembangkan': 1, 'diterbitkan': 1, 'Riot': 1, 'Games': 1, 'Permainan': 1, 'tembak-menembak': 1, ...})
```

Output : FreqDist({'game': 3, '.': 3, 'Valorant': 2, 'dunia': 2, 'dikembangkan': 1, 'diterbitkan': 1, 'Riot': 1, 'Games': 1, 'Permainan': 1, 'tembak-menembak': 1, ...})

02.

PRAKTIKUM



# PENUGASAN

Rubah CFG menjadi NLTK CFG fromstring:

Kalimat → Subyek Predikat

Kalimat → Subyek Predikat Keterangan

Subyek → Determinan KataBenda

Predikat → KataKerja KataBenda

Determinan → seorang, seekor

KataBenda → manusia, anjing, nasi, daging, air

KataKerja → makan, minum

Keterangan → kemaren, tadi

- lakukan percobaan dengan kalimat berikut  
"seekor anjing makan daging tadi" dan "seorang manusia minum air  
kemaren"
- Tampilkan tree-nya keduanya



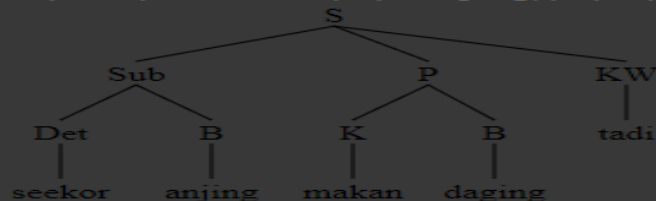
# KALIMAT 1



```
text = "seekor anjing makan daging tadi"
tokenized_word = word_tokenize(text)
print(tokenized_word)
groucho_grammar = nltk.CFG.fromstring("""
S -> Sub P | Sub P KW
Sub -> Det B
P -> K B
Det -> 'seorang' | 'seekor'
B -> 'manusia' | 'anjing' | 'nasi' | 'daging' | 'air'
K -> 'makan' | 'minum'
KW -> 'kemaren' | 'tadi'
""")
parser = nltk.ChartParser(groucho_grammar)
trees = list(parser.parse(tokenized_word))
print(trees[0])
trees[0]
```



```
['seekor', 'anjing', 'makan', 'daging', 'tadi']
(S (Sub (Det seekor) (B anjing)) (P (K makan) (B daging)) (KW tadi))
```



## KALIMAT 2



```
text = "seorang manusia minum air kemaren"
tokenized_word = word_tokenize(text)
print(tokenized_word)
groucho_grammar = nltk.CFG.fromstring("""
S -> Sub P | Sub P KW
Sub -> Det B
P -> K B
Det -> 'seorang' | 'seekor'
B -> 'manusia' | 'anjing' | 'nasi' | 'daging' | 'air'
K -> 'makan' | 'minum'
KW -> 'kemaren' | 'tadi'
""")
parser = nltk.ChartParser(groucho_grammar)
trees = list(parser.parse(tokenized_word))
print(trees[0])
trees[0]
```

```
['seorang', 'manusia', 'minum', 'air', 'kemaren']
(S
  (Sub (Det seorang) (B manusia))
  (P (K minum) (B air))
  (KW kemaren))
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data] Package punkt is already up-to-date!
```



/[AI]/[AI]/



**TERIMA  
KASIH**

ARTI  
CIAL  
INTE  
IGEN  
[AI]