**Muhammad Faisal Amir**

**1301198497**

**Tugas Pekan ke-9 *Word Sense Disambiguation***

**Program dan Analisis yang Harus Dibuat**

1. Buatlah program untuk menghitung skor kemiripan/similarity antara 2 buah teks dengan algoritma Lesk, yaitu berdasarkan jumlah kata yang beririsan pada 2 buah teks tersebut.
2. Pilihlah **tiga** contoh kata ambigu, kemudian buat contoh kalimat yang mengandung kata tersebut. Jadi Anda akan memiliki 6 (3 x 2 kalimat), karena masing-masing sense kata diwakili oleh 1 kalimat. Tentukan sense mana yang paling tepat untuk kata pada tiap kalimat, berdasarkan algoritma Lesk. Perbandingan dengan algoritma Lesk dilakukan berdasar definisi dan contoh kalimat dari Kamus Besar Bahasa Indonesia daring, <https://kbbi.kemdikbud.go.id>. Pendefinisian sense, definisi, dan contoh konteks dapat dilakukan secara *hard-code* (dituliskan langsung pada kode program).
3. Lakukan analisis terhadap hasil identifikasi sense yang tepat pada keenam contoh kalimat yang disebutkan pada nomor 2.

Jawab

import nltk  
  
nltk.download(**'wordnet'**)  
nltk.download(**'punkt'**)  
  
word\_1 = **'pintar'**word\_2 = **'makan'**word\_3 = **'cinta'**gloss = **'gloss'**example = **'example'**arti\_kata = **'arti kata'**sentences\_1\_first = **"rupanya pencuri itu lebih pintar daripada polisi"**sentences\_1\_second = **"mereka sudah pintar membuat baju sendiri"**sentences\_2\_first = **"pembangunan jembatan ini makan waktu lama"**sentences\_2\_second = **"upacara adat itu makan ongkos besar"**sentences\_3\_first = **"orang tuaku cinta kepada kami semua"**sentences\_3\_second = **"cinta kepada sesama makhluk"**word\_ambigu = {  
 word\_1: [  
 {gloss: arti\_kata, example: sentences\_1\_first},  
 {gloss: arti\_kata, example: sentences\_1\_second}],  
 word\_2: [  
 {gloss: arti\_kata, example: sentences\_2\_first},  
 {gloss: arti\_kata, example: sentences\_2\_second}],  
 word\_3: [  
 {gloss: arti\_kata, example: sentences\_3\_first},  
 {gloss: arti\_kata, example: sentences\_3\_second}]}  
  
length\_dict = {key: len(value) for key, value in word\_ambigu.items()}  
  
  
def lesk(kata, sentence):  
 length\_dict = {key: len(value) for key, value in word\_ambigu.items()}  
 length\_key = length\_dict[kata]  
 best\_sense = word\_ambigu[kata][0]  
 max\_overlap = 0  
 for i in range(length\_key):  
 overlap = 0  
 for each in nltk.word\_tokenize(sentence):  
 overlap += sum([1 for d in word\_ambigu[kata][i].values() if each in d])  
 if overlap > max\_overlap:  
 max\_overlap = overlap  
 best\_sense = word\_ambigu[kata][i]  
 return best\_sense  
  
  
print(lesk(word\_1, sentences\_1\_first))  
print(lesk(word\_1, sentences\_1\_second))  
  
print(lesk(word\_2, sentences\_2\_first))  
print(lesk(word\_2, sentences\_2\_second))  
  
print(lesk(word\_3, sentences\_3\_first))  
print(lesk(word\_3, sentences\_3\_second))

