```
# coding: utf-8

# Import NLTK terlebih dahulu

# In[ ]:

import nltk


# Latihan tokenisasi

# In[ ]:

sentence = 'The operation began this morning and all the crews stay in
the location'
tokens = nltk.word_tokenize(sentence)


# In[ ]:

print(tokens)


# In[ ]:

print(len(tokens))


# In[ ]:

for token in tokens:
    print(token)
    print(str(len(token)))


# Latihan stemming, import Porter stemmer

# In[ ]:

from nltk.stem.porter import *


# In[ ]:

stemmer = PorterStemmer()
for token in tokens:
    stemmed_token = stemmer.stem(token)
    print(stemmed_token)


# Latihan lemmatization, import WordNet lemmatizer

# In[ ]:

from nltk.stem import WordNetLemmatizer
```

```
# In[ ]:

wnl = WordNetLemmatizer()
for token in tokens:
    lemmatized_token = wnl.lemmatize(token)
    print(lemmatized_token)


# Latihan membangun tabel vocab - frekuensi kemunculan kata, dengan
dictionary

# In[ ]:

freq_tab = {}


# In[ ]:

par = 'Pesilat Indonesia kembali meraih medali emas ke-12, setelah
Pipiet Kamelia berhasil menumbangkan pesilat Vietnam Thi Cam Nhi
Nguyen di babak final pertandingan cabang olahraga Pencak Silat di
ajang Asian Games 2018. '
par += 'Pipiet menang telak 5-0 atas Thi Cam di kelas D putri 60kg-65
kg, yang berlangsung di Padepokan Pencak Silat Taman Mini Indonesia
Indah (TMII), Jakarta Timur, Rabu petang. '
par += 'Dengan kemenangan ini, Pipiet berhasil membawa medali emas,
sementara Thi Cam harus puas juara kedua dengan medali perak. '

print(par)


# In[ ]:

lc_par = par.lower()
print(lc_par)


# In[ ]:

par_tokens = nltk.word_tokenize(lc_par)
print(par_tokens)


# In[ ]:

freq_tab = {}
total_count = 0
for token in par_tokens:
    if token in freq_tab:
        freq_tab[token] += 1
    else:
        freq_tab[token] = 1
    total_count += 1

print(freq_tab)
```

```
# In[ ]:

prob_tab = {}
for token in freq_tab:
    prob_tab[token] = freq_tab[token]/total_count

print(prob_tab)


# In[ ]:

test_sentence = 'Pesilat Indonesia meraih emas'
lc_test_sentence = test_sentence.lower()
test_tokens = nltk.word_tokenize(lc_test_sentence)
total_prob = 1.0
for test_token in test_tokens:
    total_prob = total_prob * prob_tab[test_token]
    print(test_token)

print(total_prob)


# In[ ]:
```