# Tugas Materi 15

Zul Fauzi Oktaviansyah

2110181056

3 – D4 IT - B

# Code 1

In [2]:
```
data = []

for i in range(50):
    temp = i + 1
    try:
        f = open("textmining/news_dataset/data" +str(temp) + ".txt", "r", encoding="utf8")
        text=f.read()
        f.close()
    except:
        f = open("textmining/news_dataset/data" +str(temp) + ".txt", "r")
        text=f.read()
        f.close()
    text = text.lower()
    text = re.sub(r"\d+", "", text)
    text = text.translate(str.maketrans("","",string.punctuation))
    text = text.strip()
    data.append(text)
    print("\nText:\n----------------\n", text)
```

```
ketua umum gaikindo yohannes nangoi mengatakan kinerja pengapalan mobil pada maret  sempat tumbuh  persen secara tahunan namu
n raihan tersebut merupakan hasil timbunan pemesanan dari bulan sebelumnya

“sehingga kami proyeksikan ekspor pada tahun ini akan turun  persen padahal saat desember  kami diminta menaikan ekpsor hingg
a  juta unit pada ” ujarnya

Text:
----------------
 kabar berhembus daihatsu jepang sedang menyiapkan kombinasi suv dan mpv dengan  kursi baru majalah bestselling car informati
```

Membaca data 1 – 50, kemudian mengubah menjadi hurufkecil menhilangkan Plunktuasi, menghilangkan karakter kosong

# Code 2

```
In [5]: tokenData = []

        for item in data:
            tokens = word_tokenize(item)
            tokenData.append(tokens)
            print("\nTokenizing:\n----------------\n", tokens)
```

```
ementara', 'itu', 'ista', 'forum', 'merupakan', 'forum', 'diskusi', 'yang', 'menghadirkan', 'pembicara', 'internasional', 'da
n', 'nasional', 'ahli', 'di', 'bidang', 'sustainable', 'tourism', 'dan', 'marketing', 'yang', 'membahas', 'peluang', 'kerja',
'sama', 'dalam', 'pembangunan', 'pariwisata', 'berkelanjutan', 'ketua', 'indonesia', 'sustainable', 'tourism', 'council',
'i', 'gede', 'ardike', 'mengatakan', 'bahwa', 'para', 'pemenang', 'ista', 'dapat', 'didorong', 'untuk', 'mengikuti', 'sertifi
kasi', 'serta', 'secara', 'resmi', 'dapat', 'dipertanggungjawabkan', 'bahwa', 'destinasi', 'terkait', 'telah', 'menerapkan',
'konsep', 'pembangunan', 'kepariwisataan', 'berkelanjutan', 'dalam', 'malam', 'penganugerahan', 'ista', 'juga', 'diadakan',
'penandatanganan', 'nota', 'kesepahaman', 'mou', 'kerja', 'sama', 'penilitian', 'untuk', 'pariwisata', 'berkelanjutan', 'anta
ra', 'monash', 'university', 'dan', 'universitas', 'gajah', 'mada', 'ugm', 'di', 'sto', 'borobudur', 'yang', 'disaksikan', 'o
leh', 'menteri', 'pariwisata', 'deputi', 'pengembangan', 'destinasi', 'pariwisata', 'dan', 'tenaga', 'ahli', 'menteri', 'pari
wisata', 'bidang', 'pembangunan', 'pariwisata', 'berkelanjutan']

Tokenizing:
----------------
 ['tiga', 'negara', 'di', 'asia', 'timur', 'menjadi', 'pasar', 'penting', 'pariwisata', 'indonesia', 'baik', 'inbond', 'maupu
n', 'outbond', 'tiga', 'negeri', 'itu', 'adalah', 'jepang', 'korea', 'selatan', 'dan', 'taiwan', 'salah', 'upaya', 'menjaga',
'eksistensi', 'pariwisata', 'indonesia', 'di', 'tiga', 'negara', 'itu', 'kementerian', 'pariwisata', 'dan', 'ekonomi', 'kreat
if', 'kemenparekraf', 'menggelar', 'webinar', 'series', 'bersama', 'lebih', 'dari', 'pelaku', 'industri', 'pariwisata', 'dar
```

Tokenize data 1 - 50

# Code 3

In [6]:
```
from nltk.corpus import stopwords
```

In [7]:
```
# Filtering dengan Porter -------------------
listStopword = set(stopwords.words('indonesian'))

filteringResult = []

for tokens1 in tokenData:
    tmpstr = []
    for t in tokens1:
        if t not in listStopword:
            tmpstr.append(t)
    tokens1=tmpstr
    filteringResult.append(tokens1)
    print("\nSetelah filtering --> ", tokens1)
```

```
ysia', 'persen']

Setelah filtering -->  ['menteri', 'pariwisata', 'ekonomi', 'kreatif', 'menparekraf', 'wishnutama', 'kusubandio', 'diharapka
n', 'mengatasi', 'pekerjaan', 'rumahnya', 'bidang', 'industri', 'pariwisata', 'meningkatkan', 'pendapatan', 'devisa', 'terkai
t', 'pengeluaran', 'tinggal', 'wisatawan', 'mancanegara', 'pakar', 'marketing', 'pariwisata', 'halal', 'bayu', 'endro', 'wina
rko', 'jakarta', 'senin', 'november', 'bayu', 'ketua', 'bidang', 'analisis', 'kebijakan', 'publik', 'generasi', 'optimis', 'g
o', 'indonesia', 'menekankan', 'terkait', 'pengembangan', 'programprogram', 'meningkatkan', 'kunjungan', 'wisatawan', 'mancan
egara', 'indonesia', 'rendah', 'pr', 'ketiga', 'program', 'peningkatan', 'kesiapan', 'daya', 'dukung', 'destinasi', 'wisata',
'situasi', 'keamanan', 'transportasi', 'hotel', 'akomodasi', 'kesiapan', 'masyarakat', 'menyambut', 'tamu', 'daerah', 'keempa
t', 'program', 'peningkatan', 'kualitas', 'profesionalisme', 'sdm', 'pariwisata', 'bayu', 'tim', 'ahli', 'tourism', 'pekerjaa
n', 'rumah', 'pembenahan', 'infrastruktur', 'konektivitas', 'mudah', 'nyaman', 'wisatawan', 'mancanegara', 'bayu', 'mendoron
g', 'menparekraf', 'wishnutama', 'wamenparekraf', 'angela', 'tanoesudibjo', 'belajar', 'portugal', 'sukses', 'the', 'world',
```

Filtering token data 1 - 50

# Code 4

In [8]:
```
from nltk.stem import PorterStemmer
```

In [9]:
```
# Stemming dengan Porter ---------------

stemmingResult = []

for tokens2 in filteringResult:
    tmpstr = []
    ps = PorterStemmer()
    for k in tokens2:
        tmpstr.append(ps.stem(k))
    tokens2=tmpstr
    stemmingResult.append(tokens2)
    print("\nOutput stemming:\n", tokens2)
```

```
Output stemming:
 ['sebulan', 'harga', 'bawang', 'putih', 'bombay', 'melonjak', 'turun', 'stabil', 'ratarata', 'rp', 'kg', 'berkah', 'harga',
'murah', 'dinikmati', 'masyarakat', 'terusik', 'naiknya', 'harga', 'bawang', 'putih', 'pasar', 'pemerhati', 'pertanian', 'sya
iful', 'bahari', 'kenaikan', 'harga', 'komod', 'terkait', 'impor', 'bawang', 'putih', 'bombay', 'gula', 'disebabkan', 'kebija
kan', 'restriksi', 'pembatasan', 'diberlakukan', 'pemerintah', 'contoh', 'bawang', 'putih', 'komod', 'menuai', 'gugatan', 'kp
pu', 'operasi', 'tangkap', 'tangan', 'kpk', 'sumber', 'spi', 'riph', 'regulasi', 'menyuburkan', 'permainan', 'kuota', 'penimb
unan', 'rekayasa', 'harga', 'sekelompok', 'mafia', 'pangan', 'syaiful', 'sabtu', 'bawang', 'putih', 'bombay', 'syaiful', 'rel
aksasi', 'diberlakukan', 'terbukti', 'harga', 'turun', 'drasti', 'bombay', 'rp', 'kilo', 'gram', 'rp', 'rp', 'kilo', 'gram',
'komod', 'menyumbang', 'deflasi', 'syaiful', 'tinggal', 'tergantung', 'pemerintah', 'membiarkan', 'harga', 'bergejolak', 'mer
ugikan', 'jutaan', 'masyarakat', 'konsumen', 'dipaksa', 'menerima', 'harga', 'wajar', 'membuka', 'relaksasi', 'membenahi', 't
ata', 'niaga', 'pangan', 'nasion', 'gejolak', 'harga', 'terulang', 'terangnya', 'informasi', 'dihimpun', 'kondisi', 'harga',
```

Stemming filtering data 1 - 50

# Code 5

```
In [11]: frequencies = []

for tokens3 in stemmingResult:
    tf = FreqDist(tokens3)
    frequencies.append(tf.most_common())
    print("\nTerm Frequency:\n----------------\n", tf.most_common())
```

```
og', 1), ('disiapkan', 1), ('asdp', 1), ('garuda', 1), ('peningkatan', 1), ('tren', 1), ('digit', 1), ('perangkat', 1), ('ech
annel', 1), ('crm', 1), ('kondisi', 1), ('alternatif', 1), ('bertransaksi', 1), ('masyarakat', 1), ('mengaks', 1), ('brimo',
1), ('brilink', 1), ('meliputi', 1), ('cek', 1), ('saldo', 1), ('pemindahbukuan', 1), ('penggantian', 1), ('blokir', 1), ('de
liveri', 1), ('order', 1), ('spbu', 1), ('tutup', 1), ('bulanan', 1), ('jaringan', 1), ('edc', 1), ('menurunkan', 1), ('pelos
ok', 1), ('membantu', 1), ('keuangan', 1), ('pln', 1), ('bpj', 1), ('pembelian', 1), ('pulsa', 1), ('central', 1), ('asia',
1), ('menerbitkan', 1), ('penyesuaian', 1), ('operasiaon', 1), ('matraman', 1), ('galaxi', 1), ('palembang', 1), ('denpasar',
1), ('malang', 1), ('bandung', 1), ('semarang', 1), ('solo', 1), ('slamet', 1), ('riyadi', 1), ('medan', 1), ('madiun', 1),
('khusu', 1), ('blitar', 1), ('pengaturan', 1), ('bukatutup', 1), ('pandemi', 1), ('from', 1), ('home', 1), ('channel', 1),
('electronik', 1), ('klikbca', 1), ('fasilita', 1), ('diaks', 1), ('mengetik', 1), ('fitur', 1), ('qr', 1)]

Term Frequency:
----------------
 [('rupiah', 5), ('pasar', 4), ('pergerakan', 3), ('ibrahim', 3), ('rp', 3), ('negara', 3), ('amerika', 3), ('menguat', 2),
('poin', 2), ('perdagangan', 2), ('mengalami', 2), ('penguatan', 2), ('mei', 2), ('intern', 2), ('ekstern', 2), ('sisi', 2),
('milik', 2), ('surat', 2), ('normal', 2), ('faktor', 2), ('masyarakat', 2), ('pemerintah', 2), ('perekonomian', 2), ('bank',
2), ('sentral', 2), ('kabar', 2), ('serikat', 2), ('isu', 2), ('spot', 1), ('pascalebaran', 1), ('diproyeksikan', 1), ('direk
tur', 1), ('trfx', 1), ('garuda', 1), ('berjangka', 1), ('posisi', 1), ('level', 1), ('rentang', 1), ('resisten', 1), ('penut
```

Menghitung frekuensi tiap kata dari data 1 - 50

# Code 6

In [12]:
```
score = []

for common in frequencies:
    temp = []
    print("\nKeseluruhan keywords:\n----------------\n")
    for word, frequency in common:
        check = common[0][1] /2
        if frequency >= check:
            temp.append((word, frequency))
            print(word, ":", frequency)

    score.append(temp)
```

```
Keseluruhan keywords:
----------------

harga : 20
ema : 20
gram : 17
rp : 15

Keseluruhan keywords:
----------------

saham : 9
```

Menampilkan score selain dibawah 50% dari score tertinggi

# Code 7

```
In [13]: query = "pertumbuhan ekonomi, perkembangan pasar dan pergerakan harga saham"
         categoryQuery = "ekonomi"
         query = query.lower()
         query = re.sub(r"\d+", "", query)
         query = query.translate(str.maketrans("","",string.punctuation))
         query = query.strip()
         query

Out[13]: 'pertumbuhan ekonomi perkembangan pasar dan pergerakan harga saham'
```

Menset query dan category query

# Code 8

```
In [14]: queryList = word_tokenize(query)
         queryList

Out[14]: ['pertumbuhan',
          'ekonomi',
          'perkembangan',
          'pasar',
          'dan',
          'pergerakan',
          'harga',
          'saham']
```

Tokenize query

# Code 9

In [15]:
```
listStopword = set(stopwords.words('indonesian'))
tmpstr = []
for t in queryList:
    if t not in listStopword:
        tmpstr.append(t)
queryList=tmpstr
print("\nSetelah filtering --> ", queryList)
```

```
Setelah filtering -->  ['pertumbuhan', 'ekonomi', 'perkembangan', 'pasar', 'pergerakan', 'harga', 'saham']
```

Filtering queryList

# Code 10

In [16]:
```
tmpstr = []
ps = PorterStemmer()
for k in queryList:
    tmpstr.append(ps.stem(k))
queryList=tmpstr
print("\nOutput stemming:\n", queryList)
```

```
Output stemming:
 ['pertumbuhan', 'ekonomi', 'perkembangan', 'pasar', 'pergerakan', 'harga', 'saham']
```

Stemming hasil filtering queryList

# Code 11

In [17]:
```
def checkQuery(index, data):
    temp = 0
    result = {}

    for item in queryList:
        for word, freq in data:
            if item == word:
                temp+=freq

    name = 'data' + str(index+1)
    result[name] = temp
    return name,temp
```

Fungsi untuk pengecekan query dan mengembalikan total score

# Code 12

In [18]:
```
total = []
nameData = []

for i in range(50):
    temp = checkQuery(i, score[i])
    name, freq = checkQuery(i, score[i])
    nameData.append(name)
    total.append(freq)
```

Menampung total score dan nama data

# Code 13

```
In [20]: rankdocs = pd.DataFrame(dataRank)
         rankdocs
```

Out[20]:

| | name | score |
|---|---|---|
| 0 | data1 | 20 |
| 1 | data2 | 9 |
| 2 | data3 | 21 |
| 3 | data4 | 0 |
| 4 | data5 | 7 |
| 5 | data6 | 11 |
| 6 | data7 | 0 |
| 7 | data8 | 5 |
| 8 | data9 | 8 |
| 9 | data10 | 0 |
| 10 | data11 | 0 |
| 11 | data12 | 0 |
| 12 | data13 | 0 |
| 13 | data14 | 0 |

Menampilkan nama data dan total score kemiripan

# Code 14

In [21]:
```
zeroPos = rankdocs[rankdocs['score'] == 0].index
zeroPos
```

Out[21]:
```
Int64Index([ 3,  6,  9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23,
            24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40,
            42, 44, 45, 47, 48, 49],
           dtype='int64')
```

In [22]:
```
rankdocs = rankdocs.drop(zeroPos)
rankdocs
```

Out[22]:

| | name | score |
|---|---|---|
| 0 | data1 | 20 |
| 1 | data2 | 9 |
| 2 | data3 | 21 |
| 4 | data5 | 7 |
| 5 | data6 | 11 |
| 7 | data8 | 5 |
| 8 | data9 | 8 |
| 41 | data42 | 8 |

Mendrop data dengan score 0

# Code 15

In [23]:
```
label = pd.read_csv('label.csv')
label
```

Out[23]:

| | name | category |
|---|---|---|
| 0 | data1 | ekonomi |
| 1 | data2 | ekonomi |
| 2 | data3 | ekonomi |
| 3 | data4 | ekonomi |
| 4 | data5 | ekonomi |
| 5 | data6 | ekonomi |
| 6 | data7 | ekonomi |
| 7 | data8 | ekonomi |
| 8 | data9 | ekonomi |
| 9 | data10 | ekonomi |
| 10 | data11 | sepakbola |
| 11 | data12 | sepakbola |
| 12 | data13 | sepakbola |

Membaca label csv

# Code 16

In [24]:
```
temp = label
temp = temp.drop(rankdocs.index)
temp
```

Out[24]:

| | name | category |
|---|---|---|
| 3 | data4 | ekonomi |
| 6 | data7 | ekonomi |
| 9 | data10 | ekonomi |
| 10 | data11 | sepakbola |
| 11 | data12 | sepakbola |
| 12 | data13 | sepakbola |
| 13 | data14 | sepakbola |
| 14 | data15 | sepakbola |
| 15 | data16 | sepakbola |
| 16 | data17 | sepakbola |
| 17 | data18 | sepakbola |
| 18 | data19 | sepakbola |

Mengambil data label selain rankdoc

# Code 17

In [25]:
```
rankdocLabel = label.drop(temp.index)
rankdocLabel
```

Out[25]:

| | name | category |
|---|---|---|
| 0 | data1 | ekonomi |
| 1 | data2 | ekonomi |
| 2 | data3 | ekonomi |
| 4 | data5 | ekonomi |
| 5 | data6 | ekonomi |
| 7 | data8 | ekonomi |
| 8 | data9 | ekonomi |
| 41 | data42 | pariwisata |
| 43 | data44 | pariwisata |
| 46 | data47 | pariwisata |

Mendrop data label selain rankdoc

# Code 18

In [26]:
```
rankdocs['category'] = rankdocLabel['category']
rankdocs = rankdocs.sort_values(by=['score'], ascending=False)
rankdocs
```

Out[26]:

| | name | score | category |
|---|---|---|---|
| **2** | data3 | 21 | ekonomi |
| **0** | data1 | 20 | ekonomi |
| **5** | data6 | 11 | ekonomi |
| **46** | data47 | 10 | pariwisata |
| **1** | data2 | 9 | ekonomi |
| **8** | data9 | 8 | ekonomi |
| **41** | data42 | 8 | pariwisata |
| **4** | data5 | 7 | ekonomi |
| **43** | data44 | 7 | pariwisata |
| **7** | data8 | 5 | ekonomi |

Mengisi nilai category dan mengurutkan dari yang terbesar

# Code 19

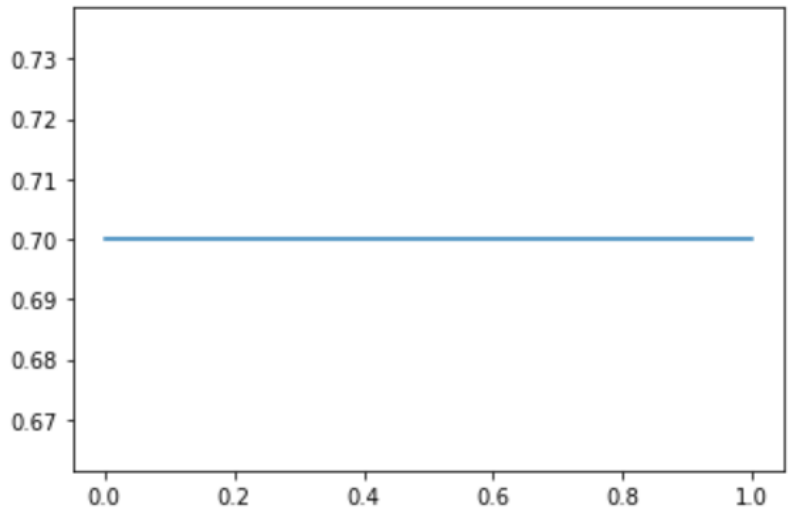
```
In [27]: categoryRelate = len(rankdocs[rankdocs['category'] == categoryQuery])
         precision = categoryRelate / len(rankdocs)
         precision

Out[27]: 0.7
```

```
In [28]: relateCategoryInLabel = len(label[label['category'] == categoryQuery])
         recall = categoryRelate / relateCategoryInLabel
         recall

Out[28]: 0.7
```

Menghitung nilai recall dan precision

# Code 20

```
In [29]: import matplotlib.pyplot as plt

In [30]: plt.plot([recall,precision])
         plt.show()
```




Menampilkan visualisasi recall dan precision