

Indonesian Stemmer

Nama File	:	IndonesianStemmer.java
Dependensi File	:	resource/stemming/dictionary.txt
Cara menjalankan	:	

```
public static void main(String args[]){
    IndonesianStemmer stemmer = new IndonesianStemmer();

    String word = "diperbantukan";

    System.out.println("Basic word : " + stemmer.stem(word));

    for(int i = 0; i < stemmer.derivationalprefix.size(); i++){
        System.out.println("Derivational Prefix : " + stemmer.derivationalprefix.get(i));
    }

    System.out.println("Particle Suffix : " + stemmer.particlesuffix);

    System.out.println("Possessive Pronoun Suffix : " + stemmer.possessivepronounsuffix);

    System.out.println("Derivational Suffix : " + stemmer.derivationalsuffix);
}
```

Output :
Basic word : kurang
Derivational Prefix : pe
Particle Suffix :
Possessive Pronoun Suffix :
Derivational Suffix : an

Indonesian Sentence Tokenizer

Nama File	:	IndonesianSentenceTokenizer.java
Dependensi File	:	resource/tokenizer/acronym.txt resource/tokenizer/compositewords.txt
Cara menjalankan	:	

```
public static void main(String[] args){
    IndonesianSentenceTokenizer tokenizer = new IndonesianSentenceTokenizer();
    ArrayList<String> token = tokenizer.tokenizeSentence("Alvin pergi ke bagian kanan ruangan
Prof. Iping.");
    for(int i = 0; i < token.size(); i++){
        System.out.println(token.get(i));
    }

    System.out.println("=====");

    token = tokenizer.tokenizeSentenceWithCompositeWords("Alvin pergi ke bagian kanan ruangan
Prof. Iping.");
    for(int i = 0; i < token.size(); i++){
        System.out.println(token.get(i));
    }
}
```

Output :

Alvin
pergi
ke
bagian
kanan
ruangan
Prof.
Iping
.

=====

Alvin
pergi
ke
bagian kanan
ruangan
Prof.
Iping
.

Indonesian Sentence Detector

Nama File	:	IndonesianSentenceDetector.java
Dependensi File	:	resource/sentencedetector/acronym.txt resource/sentencedetector/delimiter.txt
Cara menjalankan	:	

```
public static void main(String[] args){
    IndonesianSentenceDetector detector = new IndonesianSentenceDetector();

    String sentence = "Ir. Soekarno dilahirkan di Surabaya, Jawa Timur, pada 6 Juni 1901 dan meninggal di Jakarta, pada 21 Juni 1970 adalah Presiden Indonesia pertama yang menjabat pada periode 1945–1966. Ia memainkan peranan penting untuk memerdekakan bangsa Indonesia dari penjajahan Belanda. Soekarno adalah penggali Pancasila karena ia yang pertama kali mencetuskan konsep mengenai dasar negara Indonesia itu dan ia sendiri yang menamainya Pancasila. Ia adalah Proklamator Kemerdekaan Indonesia (bersama dengan Mohammad Hatta) yang terjadi pada tanggal 17 Agustus 1945.";

    ArrayList<String> sentenceList = new ArrayList<String>();

    sentenceList = detector.splitSentence(sentence);

    for(int i = 0; i < sentenceList.size(); i++){
        System.out.println(sentenceList.get(i) + "\n");
    }
}
```

Output :

Ir. Soekarno dilahirkan di Surabaya, Jawa Timur, pada 6 Juni 1901 dan meninggal di Jakarta, pada 21 Juni 1970 adalah Presiden Indonesia pertama yang menjabat pada periode 1945-1966 .

Ia memainkan peranan penting untuk memerdekakan bangsa Indonesia dari penjajahan Belanda .

Soekarno adalah penggali Pancasila karena ia yang pertama kali mencetuskan konsep mengenai dasar negara Indonesia itu dan ia sendiri yang menamainya Pancasila .

Ia adalah Proklamator Kemerdekaan Indonesia (bersama dengan Mohammad Hatta) yang terjadi pada tanggal 17 Agustus 1945 .

Indonesian POS Tagger

Nama File	:	IndonesianSentenceDetector.java
Dependensi File	:	lib/ipostagger.jar resource/cattable.txt resource/Corpus.crp resource/inlex.txt resource/Lexicon.trn resource/Ngram.trn resource/Train.crp resource/tokenizer/acronym.txt
Cara menjalankan	:	

```
public static void main(String[] args){
    ArrayList<String[]> posTag = IndonesianPOSTagger.doPOSTag("Alexander Graham Bell dilahirkan
di Edinburgh , Skotlandia , Britania Raya pada 3 Maret 1847 .");

    for(int i = 0; i < posTag.size(); i++){
        System.out.println(posTag.get(i)[0] + " - " + posTag.get(i)[1]);
    }
}
```

Output :

Alexander - NNP
Graham - NNP
Bell - NNP
dilahirkan - VBT
di - IN
Edinburgh - NNP
, - ,
Skotlandia - NNP
, - ,
Britania - NNP
Raya - NNP
pada - IN
3 - CDP
Maret - NN
1847 - CDP
. - .

Indonesian Phrase Chunker

Nama File	:	IndonesianPhraseChunker.java
Dependensi File	:	IndonesianPOSTagger.java TreeNode.java resource/phrasechunker/grammar.txt
Cara menjalankan	:	

```
public static void main(String[] args){
    IndonesianPhraseChunker chunker = new IndonesianPhraseChunker();

    String sentence = "Ruth Graham dilahirkan di Qingjiang, Kiangsu, Tiongkok dengan nama Ruth
McCue Bell .";

    chunker.setSentence(sentence);
    chunker.extractPhrase();
    chunker.printPhraseTree(chunker.phraseTree);
}
```

Output :

```
Ruth Graham -- NP
Ruth -- NNP
Graham -- NNP
dilahirkan di Qingjiang, Kiangsu, Tiongkok dengan nama Ruth McCue Bell -- VP
dilahirkan -- VP
dilahirkan -- VBT
di Qingjiang, Kiangsu, Tiongkok -- PP
di -- IN
Qingjiang, Kiangsu, Tiongkok -- NP
Qingjiang, Kiangsu, -- NP
Qingjiang, -- NNP
Kiangsu, -- NNP
Tiongkok -- JJ
dengan nama Ruth McCue Bell -- PP
dengan -- IN
nama Ruth McCue Bell -- NP
nama -- NN
Ruth -- NNP
McCue -- NNP
Bell -- NNP
. -- .
```

Format File : grammar.txt

<TipeFrase> : <Urutan POS Tag dalam bentuk regular expression. Tiap POS Tag diberi akhiran '->
Contoh : NP : ((NN-|NNP-)+(CC-)(NN-|NNP-|NNG-)+)

Indonesian NE Tagger

Nama File	:	IndonesianNETagger.java
Dependensi File	:	IndonesianPOSTagger.java resource/tokenizer/acronym.txt resource/netagger/contextualfeature/*.txt resource/netagger/morphologicalfeature/*.txt resource/netagger/nerule.txt
Cara menjalankan	:	

```
public static void main(String[] args){
    IndonesianNETagger inner = new IndonesianNETagger();

    String sentece = "Ketua MPR pada tahun 1999-2004, Amien Rais, pergi ke Bandung pada tanggal
24 April 2001 menemui dr. Sofyan pada pukul 20:30 untuk memberikan uang sejumlah
Rp100.000.000,00.";

    ArrayList<String> Nelist = inner.extractNamedEntity(sentece);

    for(int i = 0; i < inner.token.size(); i++){
        System.out.println(inner.token.get(i) + " - " + inner.tokenKind.get(i) + " - " +
inner.contextualFeature.get(i) + " - " + inner.morphologicalFeature.get(i) + " - " +
inner.POSFeature.get(i) + " - " + Nelist.get(i));
    }
}
```

Output :

Ketua - WORD - OPOS - TitleCase - NN -
MPR - WORD - - UpperCase - NN - ORGANIZATION
pada - WORD - DPRE - LowerCase - IN -
tahun - WORD - DPRE - LowerCase - NN -
1999-2004 - NUM - - Numeric - CDP - DATETIME
, - PUNC - - - , -
Amien - WORD - - TitleCase - NNP - PERSON
Rais - WORD - - TitleCase - NNP - PERSON
, - PUNC - - - , -
pergi - WORD - - LowerCase - VBI -
ke - WORD - LOPP - LowerCase - IN -
Bandung - WORD - COUNTRY - TitleCase - NNP - LOCATION
pada - WORD - DPRE - LowerCase - IN -
tanggal - WORD - DPRE - LowerCase - NN -
24 - NUM - - Digit - CDP - DATETIME
April - WORD - MONTH - TitleCase - NN - DATETIME
2001 - NUM - - Digit - CDP - DATETIME
menemui - WORD - - LowerCase - VBT -
dr. - WORD - PPRE - - NN - PERSON
Sofyan - WORD - - TitleCase - NNP - PERSON

pada - WORD - DPRE - LowerCase - IN -
pukul - WORD - DPRE - LowerCase - NN -
20:30 - NUM - - TimeFormat - JJ - DATETIME
untuk - WORD - POLP - LowerCase - IN -
memberikan - WORD - - LowerCase - VBT -
uang - WORD - - LowerCase - NN -
sejumlah - WORD - QPRE - LowerCase - CDI -
Rp100.000.000,00 - WORD - - MoneyFormat - CDP - QUANTITY
. - PUNC - - - . -

Format File : nerule.txt

- Antara PreKondisi dan Aksi diberi pemisah '>'
- Antara Fitur dan nilai Fitur diberi pemisah '='
- Fitur yang tersedia : cont (contextual), morph (morphological), token, type (token kind), pos (pos tag), ne (named entity)
- Jika ingin mengakses fitur sebelum atau sesudah dari kata yang sedang diakses, gunakan '(+|-)<angka perbedaan posisi>'
- Jika terdapat lebih dari satu kemungkinan nilai fitur (logika OR), diberi pemisah '|'

Contoh :

cont=QPRE;type+1=WORD|NUM;morph+1=Digit|Numeric|NumStr|MoneyFormat>
ne+1=QUANTITY

cont=QSUF;type-1=WORD|NUM;morph-1=Digit|Numeric|NumStr|MoneyFormat>
ne=QUANTITY;ne-1=QUANTITY

Indonesian Sentence Formalization

Nama File	:	IndonesianSentenceFormalization.java
Dependensi File	:	IndonesianSentenceTokenizer.java resource/formalization/formalizationRule.txt resource/formalization/formalizationDict.txt
Cara menjalankan	:	

```
public static void main(String[] args){  
    IndonesianSentenceFormalization formalizer = new IndonesianSentenceFormalization();  
  
    String sentence = "kata2nya 4ku donk loecoe bagedh gt .";  
    System.out.println(formalizer.formalizeSentence(sentence));  
}
```

Output :
kata-katanya aku dong lucu banget begitu .

Format File : formalizationDict.txt

<kata tidak formal><tab><kata formal>

Contoh : 7an<tab>tujuan

Format File : formalizationRule.txt

Tipe 1 : Tanpa kondisi tambahan

```
@<nama_rule>  
[prekondisi 1 (regex)]>>>[aksi 1 (regex)]  
[prekondisi 2 (regex)]>>>[aksi 2 (regex)]  
....  
[prekondisi n (regex)]>>>[aksi n (regex)]
```

Contoh :

```
@formalize_word  
([a-z0-9]+)ny$>>>$1nya  
([a-z0-9]+)nk$>>>$1ng  
([a-z0-9]+)x$>>>$1nya  
([a-z0-9]+)dh$>>>$1t  
([a-z0-9]+)2$>>>$1-$1  
([a-z0-9]+)2([a-z0-9]*)$>>>$1-$1$2  
oe>>>u
```


Tipe 2 : Dengan kondisi tambahan

```
@@<nama_rule>  
##<kondisi (regex)>  
[prekondisi 1 (regex)]>>>[aksi 1 (regex)]  
[prekondisi 2 (regex)]>>>[aksi 2 (regex)]  
....  
[prekondisi n (regex)]>>>[aksi n (regex)]
```

Contoh :

```
@@convert_symbol_to_letter  
##[a-zA-Z0-9]+[\\@\\!\\$]+[\\@\\!\\$0-9a-zA-Z]*  
@>>>a  
!>>>i  
\\$>>>s
```

Indonesian Reference Resolution

Nama File	:	IndonesianReferenceResolution.java
Dependensi File	:	IndonesianPOSTagger.java IndonesianPhraseChunker.java IndonesianNETagger.java IndonesianSentenceDetector.java resource/reference/*.txt
Cara menjalankan	:	

```
public static void main(String[] args){
    String text = "Michael Jeffrey Jordan dilahirkan di Brooklyn, New York, Amerika Serikat, pada 17
    Februari 1963 adalah pemain bola basket profesional asal Amerika. Michael Jordan merupakan
    pemain terkenal di dunia dalam cabang olahraga itu. Setidaknya ia enam kali merebut kejuaraan
    NBA bersama kelompok Chicago Bulls (1991-1993, 1996-1998). Ia memiliki tinggi badan 198 cm dan
    merebut gelar pemain terbaik.";
    int idxResolvedSentence = 1;

    IndonesianSentenceDetector sentenceExtractor = new IndonesianSentenceDetector();
    ArrayList<String> sentenceList = sentenceExtractor.splitSentence(text);

    IndonesianReferenceResolution rr = new IndonesianReferenceResolution();

    System.out.println("Sentence : " + sentenceList.get(idxResolvedSentence));

    rr.solveReference(idxResolvedSentence, sentenceList);

    System.out.println("Resolved Sentence : " + sentenceList.get(idxResolvedSentence));
}
```

Output :

Sentence : Michael Jordan merupakan pemain terkenal di dunia dalam cabang olahraga itu .

Resolved Sentence : Michael Jeffrey Jordan merupakan pemain terkenal di dunia dalam cabang olahraga itu .

Indonesian Syntactic Parser

Nama File	:	IndonesianSyntacticParser/CYKParser.java
Dependensi File	:	ipostagger.jar IndonesianSyntacticParser/*.java PCFG/*.java resource/syntacticparser/grammarCNF.txt
Cara menjalankan	:	

```
public static void main(String args[]){
    try {
        CYKParser parser = new CYKParser("./resource/grammarCNF.txt");

        String sentence = "Alvin pergi ke pasar .";

        parser.setSentence(sentence);

        parser.CYKParsing();
        parser.getTree();

        System.out.println(parser.printBestTreeToString());
    } catch (FileNotFoundException ex) {
        Logger.getLogger(CYKParser.class.getName()).log(Level.SEVERE, null, ex);
    } catch (IOException ex) {
        Logger.getLogger(CYKParser.class.getName()).log(Level.SEVERE, null, ex);
    }
}
```

Output :

+++++Pohon Terbaik+++++

Probabilitas :1.3297482349640303E-5

<S>

```

    <S>
      <*NNP>
      <VP>
        <*VBI>
        <PP>
          <*IN>
          <*NN>
        <*.>
```

Indonesian Semantic Analyzer

Nama File	:	IndonesianSemanticAnalyzer/SemanticElmt.java
Dependensi File	:	IndonesianSyntacticParser/CYKParser.java IndonesianSemanticAnalyzer/*.java resource/semanticanalyzer/ruleFile.txt
Cara menjalankan	:	

```
public static void main(String args[]){
    try {
        CYKParser parser = new CYKParser("./resource/syntacticparser/grammarCNF.txt");

        String sentence = "Alvin membeli buku di pasar .";

        parser.setSentence(sentence);
        parser.CYKParsing();
        parser.getTree();
        String stringTree = parser.printBestTreeToString();

        SemanticElmt elmt = new SemanticElmt(sentence, stringTree,
"./resource/semanticanalyzer/ruleFile.txt");
        System.out.println(elmt.PrintLexicalSemantic());

        } catch (FileNotFoundException ex) {
            Logger.getLogger(SemanticElmt.class.getName()).log(Level.SEVERE, null, ex);
        } catch (IOException ex) {
            Logger.getLogger(SemanticElmt.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
}
```

Output :

```
<S>      ::      ?X?Y?Z?b event(X,membeli) ^ Actor(X,Y) ^ Patient(X,Z) ^ object(Z,buku) ^
Location(X,b) ^ place(b,pasar) ^ object(Y,Alvin)
<S>      ::      ?X?Y?Z?b event(X,membeli) ^ Actor(X,Y) ^ Patient(X,Z) ^ object(Z,buku) ^
Location(X,b) ^ place(b,pasar) ^ object(Y,Alvin)
<*NNP>    ::      ?Y object(Y,Alvin)
<VP>      ::      ?X?Y?Z?b event(X,membeli) ^ Actor(X,Y) ^ Patient(X,Z) ^ object(Z,buku) ^
Location(X,b) ^ place(b,pasar)
<*VBT>    ::      ?X?Y?Z event(X,membeli) ^ Actor(X,Y) ^ Patient(X,Z)
<*NN>     ::      ?Z object(Z,buku)
<PP>      ::      ?X?b Location(X,b) ^ place(b,pasar)
<*NN>     ::      ?h object(h,pasar)
```