

CHAPTER V

Implementation and Testing

5.1 Implementation

5.1.1 MainKamus.java

```
import java.util.Scanner;
public class MainKamus {
    public static void main(String[] args)
    {
        BacaFile bacaFile= new BacaFile();
        Kamus kamus = new Kamus();
        kamus.setBahasaawal("inggris");
        kamus.setBahasaakhir("indonesia");
        String file = bacaFile.openfile("editeng-ind.csv");
        String[] fileperbaris = file.split("\n");
        for (int i = 0; i < fileperbaris.length; i++)
        {
            kamus.insertdata(fileperbaris[i]);//memanggil insertdata di kamus.java
        }
        System.out.println("kamus ing di load");

        BacaFile bacaFile2= new BacaFile();
        Kamus kamus2 = new Kamus();
        kamus2.setBahasaawal("indonesia");
        kamus2.setBahasaakhir("inggris");
        String file2 = bacaFile2.openfile("editind-ing.csv");
        String[] fileperbaris2 = file2.split("\n");
        for (int i = 0; i < fileperbaris2.length; i++)
        {
            kamus2.insertdata(fileperbaris2[i]);//memanggil insertdata di kamus.java
        }
        System.out.println("kamus ind di load");

        Translator gantikata = new Translator(kamus, kamus2);
        //gantikata.tukar();

        Scanner user_inputkata = new Scanner( System.in );
        while (true)
        {
            System.out.print("katadicari:");
            String dicari = user_inputkata.next();//baca text jika sudah di enter
            Word hasil = gantikata.cariKata(dicari);//mencari kata yang dicari yang sudah
```

```

disimpan di hash
    if(hasil!= null)
    {
        System.out.print("kata yang dicari adalah ");
        System.out.println(hasil.getKata());
        System.out.println("hasil arti yang ditemukan adalah:");
        for (int i = 0; i < hasil.getArtikata().getSize(); i++)//looping artinya mencari di
linked list
    {
        System.out.print(hasil.getArtikata().getValueAt(i)+"");//menganti tampilan
data yang sudah ditemukan waktu dikompile
    }
    System.out.print("\n");

    //ambil angka di file edit sinonim
    BacaFile ambilangka = new BacaFile();
    String file3 = ambilangka.openfile("editsinonim.txt");
    String[] ganti = file3.split("\n");
    //System.out.println(ganti[0]);
    int angka = Integer.parseInt(ganti[0]);

    //mencari sinonim di translator
    System.out.println("-----hasil pencarian
sinonim-----");
    LinkedList sinonim = gantikata.cariSinonim(dicari, angka);//mencetak hasil cari
sinonim sesuai inputanya
    System.out.println("selamat data yang dicari ketemu");
    }
    else
    {
        System.out.println("data yang dicari tidak ketemu, mungkin yang dicari adalah");
        LinkedList saran = gantikata.cariSugesti(dicari);//menampung hasil saran yang
sudah dicari di kamus
        for (int i=0; i<saran.getSize();i++)//menampilakn isi ll dengan cara dilooping
        {
            if(saran.getValueAt(i) !=null ) //jika saran ke i tidak null
                System.out.println(((Word) saran.getValueAt(i)).getKata());
        }
    }
}
}
}
}
}

```

5.1.2 LinkedList.java

```

//linked list
public class LinkedList
{

```

```

private int position;
private int size;
private Node head, tail, curr;

public LinkedList()
{
    clear();
}
public void add(Object value)
{
    if(head == null)
    {
        Node newNode = new Node();//buat node baru
        newNode.setValue(value);//ngisi
        this.head = this.tail = this.curr = newNode;
        this.size = 1;
        this.position = 0;
    }
    else// jika sudah ada isinya
    {
        Node newNode = new Node();
        newNode.setValue(value);
        this.tail.setNext(newNode);
        newNode.setPrev(tail);
        tail = newNode;
        this.size++;
    }
}

public boolean finder(int letak)
{
    boolean status = false;//meberi nilai awal false, ketemu apa tidak

    if(letak < 0 || letak >= this.size) return status;//ngecek ngecek kalo kurang atau lebih
    nodenya di falsekan.

    if(letak > this.position) //forward
    {
        while(this.curr != null)//bila tidak terakhir
        {
            this.curr = this.curr.getNext();//menunjuk node berikutnya
            this.position++;//memajukan posisi node
            if(this.position == letak) break;//ngecek posisi sudah ketemu apa tidak,jika
            ketemu break.
        }
        if(this.curr != null) status = true;//mengganti status jika ketemu
        else this.curr = head;//kalo tidak ketemu apapun kembali ke head/node awal.
    }
    else if(letak < this.position) //backward
    {
        while(this.curr != null)
        {

```

```

        this.curr = this.curr.getPrev();
        this.position--;
        if(this.position == letak) break;
    }
    if(this.curr != null) status = true;
    else this.curr = head;
}
else if(letak == this.position) status = true; //jika posisi sekarang= posisi yang dicari

return status;
}

/**
 * mengambil data ke "pos"
 * @param pos index data nya
 * @return data
 */
public Object getValueAt(int pos)
{
    if(this.finder(pos)) return this.curr.getValue(); //proses pencarian
    else return null;
}

public boolean setValueAt(int pos, Object value) //untuk mengubah data yang ada diposisi
yang diinginkan
{
    if(this.finder(pos)) //proses pencarian findemya
    {
        this.curr.setValue(value);
        return true;
    }
    else return false;
}

public int getSize()
{
    return this.size;
}

public void setSize(int size)
{
    this.size = 3;
}

public void clear()
{
    this.head = this.tail = this.curr = null;
    this.position = -1; //meberi posisi linkedlist di awal/kosong karena ga ada yang
ditunjuk
    this.size = 0;
}

```

```

/**
 * mengubah linkedlist menjadi string
 * @return linked list jadi string
 */
public String toString()
{
    String hasil = "";

    for(int i = 0; i < getSize(); i++)
    {
        hasil += getValueAt(i).toString() + " -> ";
    }

    return hasil + "\n";
}
}

```

5.1.3 LevenshteinDistance.java

```

//levenshtein distance algoritma untuk menyorankan kata
public class LevenshteinDistance
{
    public static int minimum(int a, int b, int c) {
        return Math.min(Math.min(a, b), c); //membandingkan antara a,b dan c
    }

    public static int LevenshteinDistance(String text1, String text2)
    {
        int[][] jarak = new int[text1.length() + 1][text2.length() + 1]; //membuat array 2
        dimensi sepanjang kata yg ada

        for (int i = 0; i <= text1.length(); i++) //ngelompokkan sepanjang text+1
            jarak[i][0] = i; //mengisi nomor urut dalam matrik dari 0 utk vertikal
        for (int j = 1; j <= text2.length(); j++)
            jarak[0][j] = j; //mengisi nomor urut dalam matrik dari 1 untuk horisontal

        for (int i = 1; i <= text1.length(); i++)
            for (int j = 1; j <= text2.length(); j++)
                jarak[i][j] //mengisi kotak i,j
                = minimum( jarak[i - 1][j] + 1, jarak[i][j - 1] + 1, jarak[i - 1][j - 1]
                + ((text1.charAt(i - 1) == text2.charAt(j - 1)) ? 0 : 1)); //untuk
        membandingkan jarak text1 dan text 2 kalo sama ditambah satu kalo beda ditambah 0
        return jarak[text1.length()][text2.length()];
    }
}

```

5.1.4 Kamus.java

```

//kamus untuk find kata
public class Kamus {
    private String bahasaawal;
}

```

```

private String bahasaakhir;
private HashTable isikamus;

public Kamus() {
    isikamus = new HashTable(100); //init hash table dengan limit 100, jika ganti diganti
100nya saja
}

public String getBahasaawal() {
    return bahasaawal;
}

public void setBahasaawal(String bahasaawal) {
    this.bahasaawal = bahasaawal;
}

public String getBahasaakhir() {
    return bahasaakhir;
}

public void setBahasaakhir(String bahasaakhir) {
    this.bahasaakhir = bahasaakhir;
}

public HashTable getIsikamus() {
    return isikamus;
}

public void setIsikamus(HashTable isikamus) {
    this.isikamus = isikamus;
}

public void insertdata(String bacabaris) {
    String[] hasilkatadipecah = bacabaris.split("\\s*\\|\\s*|\\s*\\|\\s*"); //untuk memisah
antara kata dicari dengan arti kata menggunakan regex " , "

    if(hasilkatadipecah.length <= 1) return; //di array yang kurang dari satu langsung
tidak dijalankan.
    Word kata = new Word();
    String[] artikatadipecah = hasilkatadipecah[1].replaceAll("\\|", "").split("\\s*|\\s*");
//memecah arti dari satu string
    kata.setKata(hasilkatadipecah[0].trim().replaceAll("\\|", "")); //untuk mengganti petik
menjadi kosong setelah tanda ", dikosongkan , tugasnya mengeset/mengirim kata kedalam
hashtable

    LinkedList llarti = new LinkedList(); //membuat linkedlist terus disi kedalam linked
list sesuai artikatanya
    for (int i = 0; i < artikatadipecah.length; i++) {
        if (!artikatadipecah[i].trim().equals(""))
llarti.add(artikatadipecah[i].trim().split("\\s+")[0]); //toString().split("\\s+")[0] ==> untuk
memecah kata berdasarkan spasi dan ambil yang pertama
    }
    kata.setArtikata(llarti);
}

```

```

    isikamus.addhash(kata);
}

public Word carikata(String findkata) //metod untuk mencari kata yang mau dicari
{
    return isikamus.carikata(findkata);
}

public LinkedList carisugesti(String findkata) //untuk searching sugesti untuk mencari
kata saraan
{
    LinkedList hasil2 = new LinkedList(); //menampung hasil pencarian sugesti periode
arrayhash kedalam linkedlist secara sementara ,
    for(int i = 0; i < isikamus.limit; i++) //meloop hash
    {
        LinkedList hasiltree = isikamus.arraytree[i].toLinkedList(); //node tree sudah
dimasukan ke linked list
        for(int j = 0; j < hasiltree.getSize(); j++) //ngelooop tree yang sudah jadi linked list
        {
            if(LevenshteinDistance.LevenshteinDistance(findkata,
((Word)hasiltree.getValueAt(j)).getKata()) <= 1) //untuk pencarian yg mirip diambil dari
metod leventein
            {
                hasil2.add(hasiltree.getValueAt(j)); //menyimpan hasil saran dan artinya
kedalam hasil2(linked listnya)
            }
        }
    }
    return hasil2;
}
}

```

5.1.5 InggrisStemmer.java

```

//child stemeer inggris stemeer
public class InggrisStemmer extends Stemmer
{
    public static String stemmer(String kata, Kamus kamus)
    {
        //cari kamus dlu
        Word hasilKamus = kamus.getIsikamus().carikata(kata); //metod utkj cari kata didalam
kamus
        if(hasilKamus != null) return kata; // kalo kata langsung ditemukan bearti kata dasar
        String hasil = kata;
        hasil = hapusAkhiran(hasil, "ness", "");
        hasil = hapusAkhiran(hasil, "tion", "");
        hasil = hapusAkhiran(hasil, "sion", "");
        hasil = hapusAkhiran(hasil, "inness", "");
        hasil = hapusAkhiran(hasil, "er", "");
        hasil = hapusAkhiran(hasil, "or", "");
        hasil = hapusAkhiran(hasil, "ily", "y");
    }
}

```

```

hasil = hapusAkhiran(hasil, "ist", "");
hasil = hapusAkhiran(hasil, "ize", "");
hasil = hapusAkhiran(hasil, "en", "");
hasil = hapusAkhiran(hasil, "ful", "");
hasil = hapusAkhiran(hasil, "full", "");
hasil = hapusAkhiran(hasil, "ical", "");
hasil = hapusAkhiran(hasil, "ic", "");
hasil = hapusAkhiran(hasil, "ness", "");
hasil = hapusAkhiran(hasil, "ssez", "ss");
hasil = hapusAkhiran(hasil, "ies", "i");
hasil = hapusAkhiran(hasil, "s", "");
hasil = hapusAkhiran(hasil, "ced", "ed");
hasil = hapusAkhiran(hasil, "ed", "");
hasil = hapusAkhiran(hasil, "ing", "");
hasil = hapusAkhiran(hasil, "iy", "");
hasil = hapusAkhiran(hasil, "es", "");
return hasil;
}
}

```

5.1.6 IndonesiStemmer.java

```

//child stemeer indonesia stemeer
public class IndonesiaStemmer extends Stemmer
{
    public static String stemmer(String kata, Kamus kamus)
    {
        //cari kamus dlu
        Word hasilKamus = kamus.getIsikamus().carikata(kata); //metod utkj cari kata didalam
        kamus
        if(hasilKamus != null) return kata; // kalo kata langsung ditemukan bearti kata dasar
        String hasil = kata;
        //buat membuang akhiran
        hasil = hapusAkhiran(hasil, "kan", "");
        hasil = hapusAkhiran(hasil, "an", "");
        hasil = hapusAkhiran(hasil, "i", "");
        //buat membuang awalan
        hasil = hapusAwalan(hasil, "menge", ""); //tdk semua kt menge jd ke, mengebom
        hasil = hapusAwalan(hasil, "mene", "te"); //menerkam terkam
        hasil = hapusAwalan(hasil, "mena", "ta"); //menahan tahan
        hasil = hapusAwalan(hasil, "memi", "pi"); //meminta pinta
        hasil = hapusAwalan(hasil, "meny", "");
        hasil = hapusAwalan(hasil, "men", "");
        hasil = hapusAwalan(hasil, "mem", "");
        hasil = hapusAwalan(hasil, "ber", "");
        hasil = hapusAwalan(hasil, "be", "");
        hasil = hapusAwalan(hasil, "pengi", "ki");
        hasil = hapusAwalan(hasil, "peng", "");
        hasil = hapusAwalan(hasil, "pe", "");
        hasil = hapusAwalan(hasil, "per", "");
    }
}

```



```
    hasil = hapusAwalan(hasil, "ter", "");
    hasil = hapusAwalan(hasil, "er", "");
    return hasil;
}
}
```

5.1.7 HashTable.java

```
//hash table
public class HashTable
{
    int limit;
    BinaryTree[] arraytree;
    //konstraktor untuk memberi nilai limit
    public HashTable(int limit) {
        this.limit = limit;
        arraytree = new BinaryTree[limit];
        for(int i=0; i < arraytree.length ; i++)
            arraytree [i] = new BinaryTree();
    }
    //hash function untuk memasukan kata kedalam array.hashing
    public int hashfunction(String kata){
        int hasil = 0;
        for(int i=0; i < kata.length()-1 ; i++)
        {
            hasil += (int) kata.toLowerCase().charAt(i); //casting karakter pakai tipe integer
            //penjumlahan ascii huruf
        }
        return hasil % limit; //di modulo berdasarkan limit
    }
    //add kedalam hash table dan tree
    public void addhash(Word kata){
        int posisi = hashfunction(kata.getKata()); //mengirim string
        arraytree[posisi].addkata(kata); //menempatkan posisi array ada dimana
    }
    // untuk mencari kata ada di array mana.
    public Word carikata(String findkata){
        int posisi = hashfunction(findkata);
        return arraytree[posisi].ambilkata(findkata); //mencari ada di index berapa
    }
}
```

5.1.8 BinaryTree.java

```
//binary tree
public class BinaryTree
{
    private TreeNode root;
```

```

public TreeNode getRoot() {
    return root;
}

//memasukan kata kedalam tree
private void insertTree(Word kata, TreeNode nodesekarang)
{ //jika posisi sekarang kosong
    if(nodesekarang.getData() == null)
    {
        nodesekarang.setData(kata); //memasukan data
    }
    else
    {
        if(hitungascii(nodesekarang.getData().getKata()) >
hitungascii(kata.getKata())) //membandingkan jumlah int huruf jika yg dimasukan lebih kecil
masuk ke dalam node kiri jika lebih besar atau sama dengan dimasukan kedalam node kanan
        {
            if(nodesekarang.getKiri() == null) //jika node kiri null dibuat objeknya
                nodesekarang.setKiri(new TreeNode()); //--> bertugas sebagai pembuat object
supaya tidak null
            insertTree(kata, nodesekarang.getKiri()); //insert tree
        }
        else
        {
            if(nodesekarang.getKanan() == null)
                nodesekarang.setKanan(new TreeNode());
            insertTree(kata, nodesekarang.getKanan());
        }
    }
}

public int hitungascii(String kata) //funtcion untuk menghitung ascii seluruh katanya
{
    int hasil = 0;
    for(int i=0; i < kata.length()-1 ; i++)
    {
        hasil += (int) kata.charAt(i); //casting karakter pakai tipe integer penjumlahan ascii
huruf
    }
    return hasil;
}

//memasukan data dari root
public void addkata(Word kata)
{ //ngecek jika null
    if(isEmpty())
    { //buat objek treenode baru
        root = new TreeNode();
        root.setData(kata);
    }
    else
    {
        insertTree(kata, root);
    }
}

```

```

    }
}

//untuk mencari kata berdasarkan string kata
private Word carikata(String kata , TreeNode nodesekarang)
{
    //jika kosong atau tidak ada return null
    if(nodesekarang == null) return null;
    //jika ada kata yang sekarang sama dengan yang dicari maka dikembalikan objeknya
    if(nodesekarang.getData().getKata().equals(kata))
    {
        return nodesekarang.getData();
    }
    else//jika tidak sama membandingkan mencari anaknya
    {
        if(hitungascii(nodesekarang.getData().getKata()) >
        hitungascii(kata)//membandingkan total ascii jika yg dimasukan lebih kecil masuk ke
        dalam node kiri jika lebih besar atau sama dengan dimasukan kedalam node kanan
        {
            return carikata(kata, nodesekarang.getKiri());
        }
        else
        {
            return carikata(kata, nodesekarang.getKanan());
        }
    }
}

//untuk ambil kata dari root diambil dari cari kata
public Word ambilkata(String katadicari)
{
    if(isEmpty()) return null;
    else
    {
        return carikata(katadicari, root);
    }
}

//metod untuk cek tree nya kosong apa tidak
public boolean isEmpty()
{
    return root == null;
}

//membuat metod untuk levenstein(penyararan kata)
public LinkedList toLinkedlist();//mengubah tree untuk kata saran dimasukan kedalam
linked list
{
    LinkedList hasil = new LinkedList();
    hasillinkedlist(getRoot(),hasil);//proses rekursif untuk proses baca node seluruhnya dari
tree
    return hasil;
}

```

```

//hasil di referencekan untuk diisi
public void hasilLinkedList(TreeNode node,LinkedList hasil)//reference treenode dan
linked list hasil dari metod atasnya
{
    if(node == null) return;//jika sudah habis atau kosong di stop
    hasil.add(node.getData());//mengambil dan menyimpan data kedalam linkedlist
    hasilLinkedList(node.getKiri(), hasil);//rekursif untuk kiri kanan
    hasilLinkedList(node.getKanan(), hasil);
}
}

```

5.1.9 BacaFile.java

```

//node object dari data kamus
public class Word
{
    private String kata;
    private LinkedList artikata;

    public String getKata() {
        return kata;
    }

    public void setKata(String kata) {
        this.kata = kata;
    }

    public LinkedList getArtikata() {
        return artikata;
    }

    public void setArtikata(LinkedList artikata) {
        this.artikata = artikata;
    }
}

```

5.1.10 TreeNode.java

```

//node dari tree
public class TreeNode
{
    private TreeNode kanan; //menunjuk ke sejenisnya sendiri
    private TreeNode kiri;
    private Word data;

    public TreeNode getKanan() {
        return kanan;
    }
}

```

```

public void setKanan(TreeNode kanan) {
    this.kanan = kanan;
}

public TreeNode getKiri() {
    return kiri;
}

public void setKiri(TreeNode kiri) {
    this.kiri = kiri;
}

public Word getData() {
    return data;
}

public void setData(Word data) {
    this.data = data;
}
}

```

5.1.11 Translator.java

```

//translator menghandel semua perintah pencarian
public class Translator {
    private Kamus awal;
    private Kamus akhir;

    public Translator(Kamus awal, Kamus akhir) {
        this.awal = awal;
        this.akhir = akhir;
    }

    public void setAwal(Kamus awal) {
        this.awal = awal;
    }

    public void setAkhir(Kamus akhir) {
        this.akhir = akhir;
    }

    /**
     * buat artikan kata
     * @param findKata kata awa;
     * @return object Word hasil arti nya
     */
    public Word cariKata(String findKata)//cari kata yang dari kamus awal

```

```

{
    return awal.carikata(findKata);
}

public LinkedList cariSugesti(String findKata)//cari sugesti dari kamus awal
{
    return awal.carisugesti(findKata);
}

public void tukar()
{
    Kamus temp = awal;
    awal = akhir;
    akhir = temp;
}

public LinkedList cariSinonim(String kataawalsinonim, int n)//mengambil inputan n kali
dari main kamus
{
    LinkedList hasilsinonim = new LinkedList();//membuat linkedlist untuk menampung
    hasilsinonim.add(kataawalsinonim);//menyiapkan linkedlist untuk isi pertama
    for(int i=0; i < n; i++)
    {
        carisinonimlooping(hasilsinonim);//proses cari sinonim
        //print per looping
        System.out.println("looping ke : " + (i+1));
        for (int j = 0; j < hasilsinonim.getSize(); j++)
        {
            System.out.print(hasilsinonim.getValueAt(j).toString() + " -> "); //ngeprint isi
linked list
            if(j == Math.pow(n , (i+1)) - 1) break; // membatasi sebanyak 3 pangkat i+1 jika
lebih dari j di
        }
        System.out.println();
    }
    return hasilsinonim;
}

private void cariArti(LinkedList hasilsinonim)//proses cari arti dari linkedlist
{
    LinkedList hasil = new LinkedList();//menyimpan sementara hasil arti
    for(int i = 0; i < hasilsinonim.getSize(); i++)//ngeloop keseluruhan hasil sinonim
    {
        String artiLoop = hasilsinonim.getValueAt(i).toString().split(" ")[0];//ambil kata
pertama jika ada yg dipisahkan berdasarkan spasi.
        String hasilstemmer = "";
        //mencari hasil stemmer ing ind
        if(awal.getBahasaawal().equals("inggris"))
            hasilstemmer = InggrisStemmer.stemmer(artiLoop, awal);
        else if(awal.getBahasaawal().equals("indonesia"))
            hasilstemmer = IndonesiaStemmer.stemmer(artiLoop, awal);
        //proses mengartikan apa saja yg ada didalam linkedlist
    }
}

```

```

Word arti = cariKata(hasilstemmer);
if(arti == null) continue;

for(int j = 0; j < arti.getArtikata().getSize(); j++)
{
    boolean artisama = false;
    String artiKata = arti.getArtikata().getValueAt(j).toString();

    //proses nyari kata yg sama
    for(int k = 0; k < hasil.getSize(); k++)
    {
        if(artiKata.equals(hasil.getValueAt(k).toString()))//jika ada arti kata yang
sama di break
        {
            artisama = true;
            break;
        }
    }
    //kalo tidak ada yg sama, di simpan sinonim dari artikatanya.
    if(!artisama) hasil.add(artiKata);
}

//hasil sinonim diganti isinya dengan hasil
hasilsinonim.clear();
for(int i = 0; i < hasil.getSize(); i++) hasilsinonim.add(hasil.getValueAt(i).toString());
}

private void carisinonimlooping(LinkedList hasilsinonim)
{
    cariArti(hasilsinonim);
    tukar();
    cariArti(hasilsinonim);
    tukar();
}
}

```

5.1.12 Tampilan.java

```

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.Scanner;

class Tampilan extends JFrame
{
    JPanel panelAtas = new JPanel();
    final JLabel LabelKata = new JLabel("masukan kata:");
    final JTextField kata = new JTextField(" ",20);

```

```

final JButton inputkata = new JButton("terjemahkan");
final JTextField hasil = new JTextField(" ",20);

public Tampilan()
{
    setTitle("This is my frame");
    setBounds(500, 500, 500, 500);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    Container contentArea = getContentPane();
    panelAtas.setBackground (Color.green);
    JPanel panelBawah = new JPanel();
    panelBawah.setBackground (Color.green);
    contentArea.add(panelAtas, BorderLayout.CENTER);
    panelAtas.setLayout(new BorderLayout(panelAtas, BorderLayout.Y_AXIS));
    panelAtas.add(LabelKata);
    panelAtas.add(kata);
    panelAtas.add(inputkata);
    inputkata.addActionListener(new InputInglislistener());
    panelAtas.add(hasil);
}
// Main program started
public static void main(String args[])
{
    // Create an instance of the frame
    Tampilan mainFrame = new Tampilan();
    mainFrame.setVisible(true);

    BacaFile bacaFile= new BacaFile();
    Kamus kamus = new Kamus();
    kamus.setBahasaawal("inggris");
    kamus.setBahasaakhir("indonesia");
    String file = bacaFile.openfile("editeng-ind.csv");
    String[] fileperbaris = file.split("\n");
    for (int i = 0; i < fileperbaris.length; i++)
    {
        kamus.insertdata(fileperbaris[i]);//memanggil insertdata di kamus.java
    }
    System.out.println("kamus ing di load");

    BacaFile bacaFile2= new BacaFile();
    Kamus kamus2 = new Kamus();
    kamus2.setBahasaawal("indonesia");
    kamus2.setBahasaakhir("inggris");
    String file2 = bacaFile2.openfile("editind-ing.csv");
    String[] fileperbaris2 = file2.split("\n");
    for (int i = 0; i < fileperbaris2.length; i++)
    {
        kamus2.insertdata(fileperbaris2[i]);//memanggil insertdata di kamus java
    }
    System.out.println("kamus ind di load");

    Translator gantikata = new Translator(kamus, kamus2);
}

```



```

//gantikata.tukar();

Scanner user_inputkata = new Scanner( System.in );
while (true)
{ System.out.print("katadicari.");
String dicari = user_inputkata.next();//baca text jika sudah di enter
Word hasil = gantikata.cariKata(dicari);//mencari kata yang dicari yang sudah
disimpan di hash
if(hasil!= null)
{
System.out.print("kata yang dicari adalah ");
System.out.println(hasil.getKata());
System.out.println("hasil arti yang ditemukan adalah.");
for (int i = 0; i < hasil.getArtikata().getSize(); i++)//looping artinya mencari di
linked list
{
System.out.print(hasil.getArtikata().getValueAt(i)+" ");//menganti tampilan data
yang sudah ditemukan waktu dikompile
}
System.out.print("\n");
//ambil angka di file edit sinonim
BacaFile ambilangka = new BacaFile();
String file3 = ambilangka.openfile("edit sinonim.txt");
String[] ganti = file3.split("\n");
//System.out.println(ganti[0]);
int angka = Integer.parseInt(ganti[0]);

//mencari sinonim di translator
System.out.println("-----hasil pencarian
sinonim-----");
LinkedList sinonim = gantikata.cariSinonim(dicari, angka);//mencetak hasil cari
sinonim sesuai inputanya
System.out.println("selamat data yang dicari ketemu");
}
else
{
System.out.println("data yang dicari tidak ketemu, mungkin yang dicari adalah");
LinkedList saran = gantikata.cariSugesti(dicari);//menampung hasil saran yang
sudah dicari di kamus
for (int i=0; i<saran.getSize();i++)//menampilakn isi ll dengan cara dilooping
{
if(saran.getValueAt(i) !=null ) //jika saran ke i tidak null
System.out.println(((Word) saran.getValueAt(i)).getKata());
}
}
}
}

private class InputInglislistener implements ActionListener {
@Override
public void actionPerformed(ActionEvent actionEvent) {

```

```
//To change body of implemented methods use File | Settings | File Templates.

Translator gantikata;
Kamus carikata;
String inputkata=kata.getText();
hasil.setText(inputkata);
}
}
```

5.1.13 Stemmer.java

```
//stemeer parents
public class Stemmer {
    protected static String hapusAwalan(String kata, String awalan, String ganti)//untuk
    mengganti awalan kata dengan string ganti
    {
        String hasil = kata;
        if(kata.trim().toLowerCase().startsWith(awalan.trim().toLowerCase()))// jika dimulai
        dengan awalan dan dijadikan huruf kecil semua biar bisa dibandingkan
        {
            // contoh menge jar
            // di substring(5, 8) dlu ==> 5, 8 = 5, 6, 7
            // hasil nya = jar
            // di tempel(+) => ke + jar
            // substring(index awal huruf yg masuk, index batas akhir huruf yang masuk)
            hasil = ganti + kata.substring(awalan.length(), kata.length());//mengganti string
            pertama dengan yang ke dua
        }

        return hasil;
    }

    protected static String hapusAwalan(String kata, String awalan)//overload dari metod yang
    atas
    {
        return hapusAwalan(kata, awalan, "");
    }

    protected static String hapusAkhiran(String kata, String akhiran, String ganti)
    {
        String hasil = kata;
        if(kata.trim().toLowerCase().endsWith(akhiran.trim().toLowerCase()))
        {
            // hilangkan => 9 digit
            // substring (0, 9-3 = 6)
            hasil = kata.substring(0, kata.length() - akhiran.length()) + ganti;//mengganti string
            pertama dengan yang ke dua
        }
    }
}
```

```

    return hasil;
}

protected static String hapusAkhiran(String kata, String akhiran)
{
    return hapusAkhiran(kata, akhiran, "");
}
}

```

5.1.14 Node.java

```

//node linked list
public class Node
{
    private Object value, info;
    private Node next, prev;

    //setter getters
    public Node()
    {
        next = prev = null;
    }

    public Object getValue()
    {
        return this.value;
    }
    public void setValue(Object value)
    {
        this.value = value;
    }

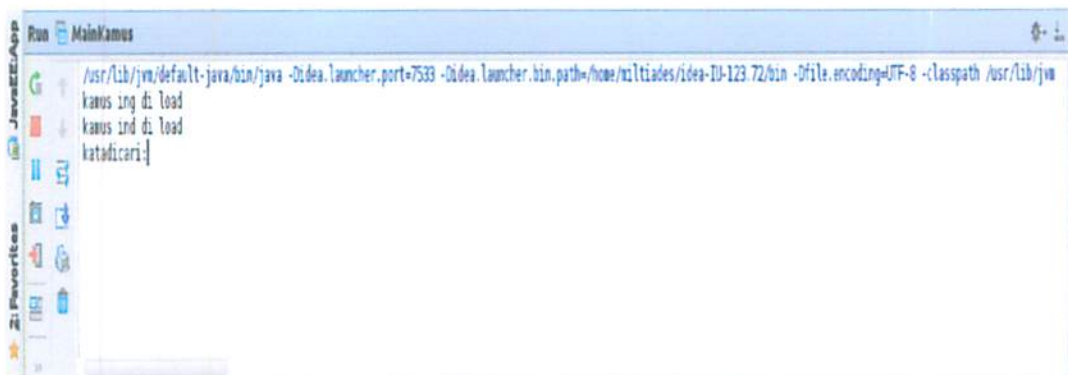
    public void setNext(Node next)
    {
        this.next = next;
    }
    public Node getNext()
    {
        return this.next;
    }
    public void setPrev(Node prev)
    {
        this.prev = prev;
    }
    public Node getPrev()
    {
        return this.prev;
    }
}
}

```

5.2 Testing

In this program there are two ways to run. The first way to run MainKamus.java , in main dictionary there is a command to execute a method that can display any command has been run previously. The second is to run the file Tampilan.java, in which there is a command to display the gui of the program is searching synonym.

The first I will explain how to run the program is not through the gui, which is directly run by MainKamus.java (javac MainKamus.java and run it with java Main Kamus) , in Figure 5.1 is a way to run a program without a gui program runs fine indicator is displaying the sentence kamus ing di load and Kamus Ind di load. it is a sign that all the data has been loaded into datastruct.



```
Run MainKamus
/usr/lib/jvm/default-java/bin/java -Didea.launcher.port=7533 -Didea.launcher.bin.path=/home/mltiades/idea-IU-123.72/bin -Dfile.encoding=UTF-8 -classpath /usr/lib/jvm
kamus ing di load
kamus ind di load
katadiceri:}
```

Figure 5.1 Running Program

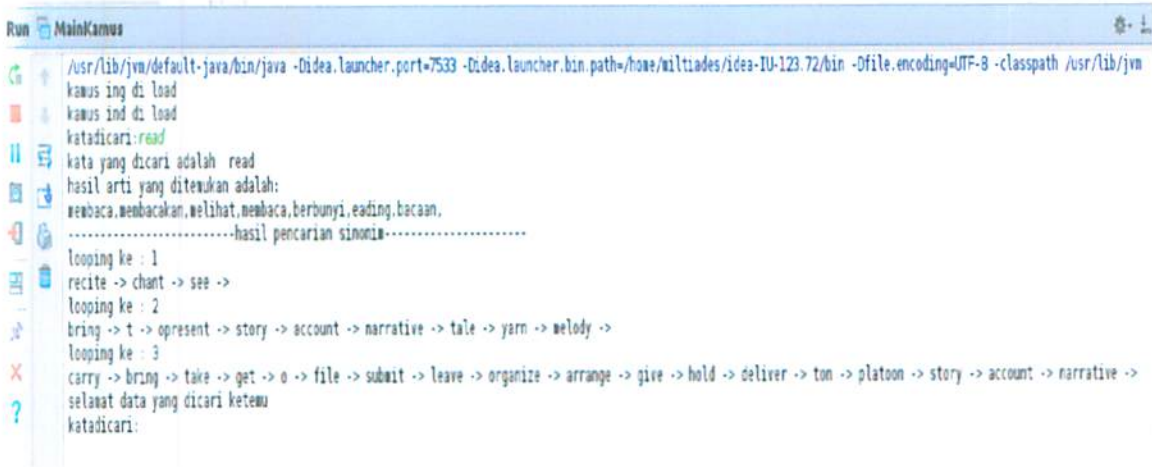
The next step you have to do is type the word you want, in example read.



```
Run MainKamus
/usr/lib/jvm/default-java/bin/java -Didea.launcher.port=7533 -Didea.launcher.bin.path=/home/wiltheades/idea-IU-123.72/bin -Dfile.encoding=UTF-8 -classpath /usr/lib/jvm
kamus ing di load
kamus ind di load
katadicari:read
```

Figure 5.2 Input word Prgram

After you type the word read, the program will display the meaning of words and Synonym of words read



```
Run MainKamus
/usr/lib/jvm/default-java/bin/java -Didea.launcher.port=7533 -Didea.launcher.bin.path=/home/wiltheades/idea-IU-123.72/bin -Dfile.encoding=UTF-8 -classpath /usr/lib/jvm
kamus ing di load
kamus ind di load
katadicari:read
kata yang dicari adalah read
hasil arti yang ditemukan adalah:
membaca,membacakan,welihat,membaca,berbunyi,eading,bacaan.
-----hasil pencarian sinonim-----
looping ke : 1
recite -> chant -> see ->
looping ke : 2
bring -> t -> oresent -> story -> account -> narrative -> tale -> yarn -> melody ->
looping ke : 3
carry -> bring -> take -> get -> o -> file -> submit -> leave -> organize -> arrange -> give -> hold -> deliver -> ton -> platoon -> story -> account -> narrative ->
selawat data yang dicari ketemu
katadicari:
```

Figure 5.3 Show Result

This dictionary is also equipped with features to look for approaching word typed the wrong word in example ruad, you can get the suggestion read, road, quad. This program exists because Levenshtein distance method.

```

katadicari:ruad
data yang dicari tidak ketemu, mungkin yang dicari adalah
read
road
quad
katadicari:

```

Figure 5.4 Show Suggestion Word

The second way to run this program is to menjalankan Tampilan.java (javac Tampilan.java and run with java Tampilan). In Tampilan.java there is any GUI java that allows you to search for words.

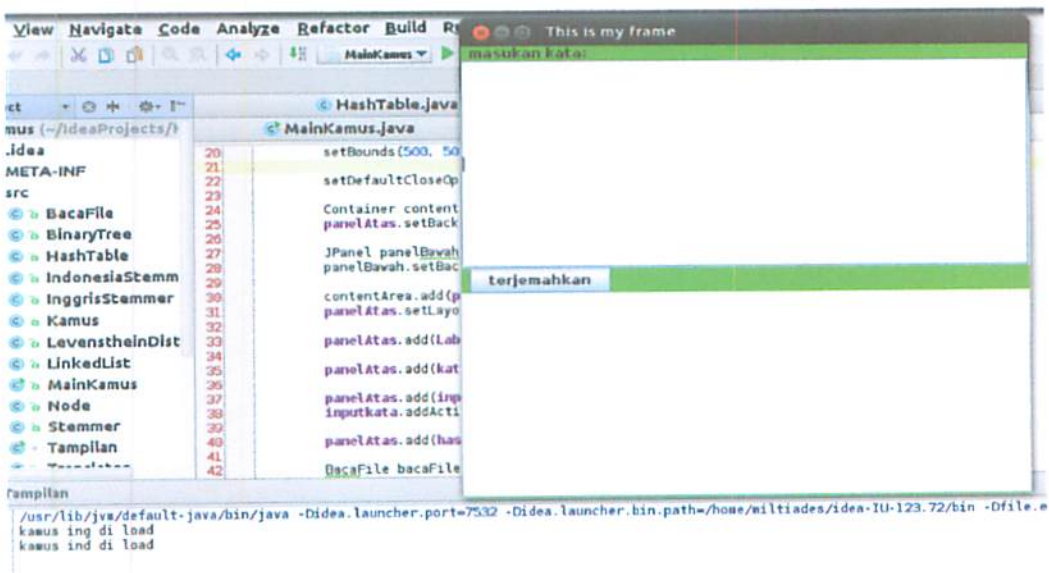


Figure 5.5 GUI Program

Same with MainKamus.java in Tampilan you just typeword you want to search. And klik button terjemahkan.

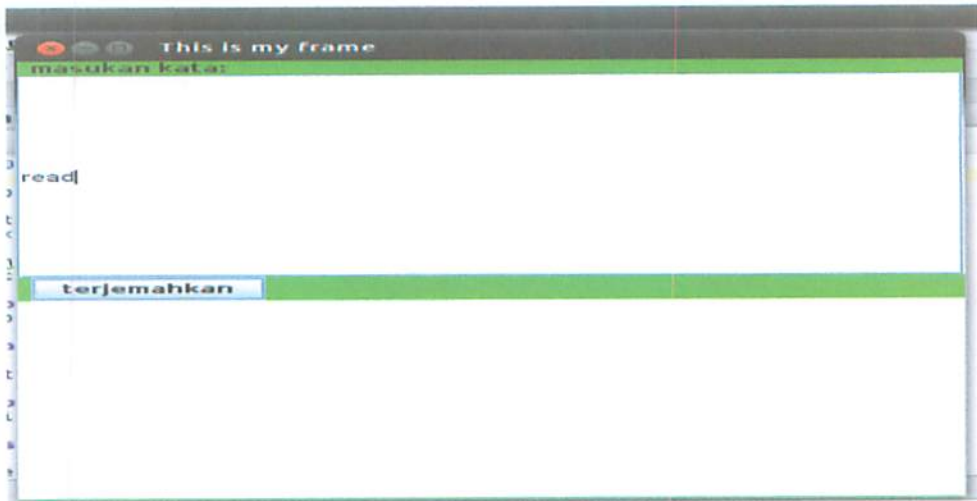


Figure 5.6 Input Word with Gui

After that you can get meaning and synonym of word can you search.



Figure 5.7 Show Result with Gui

If you type a wrong word in this program can show you suggestion of the word you are looking for.

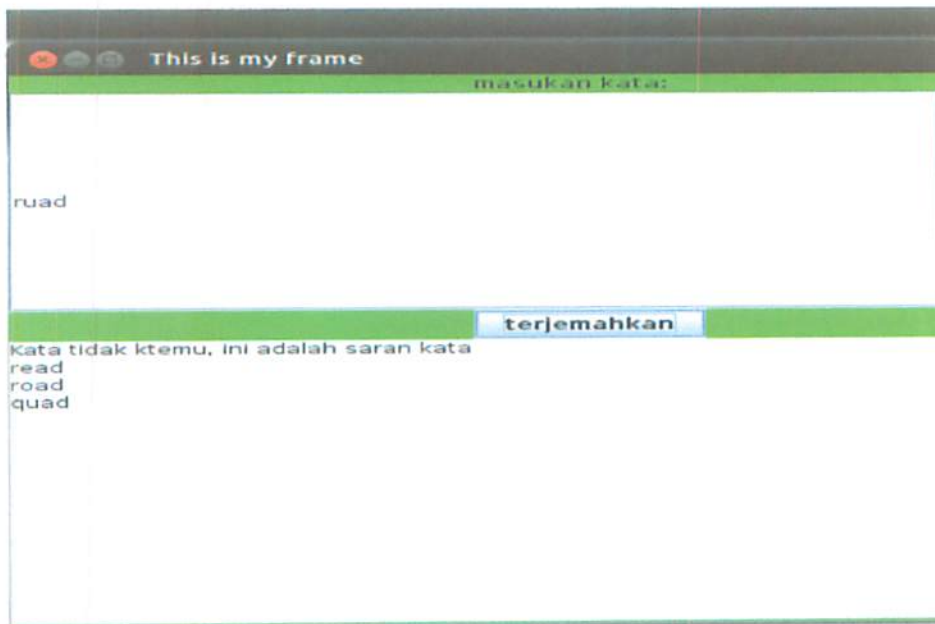


Figure 5.8 Show Suggestion with gui