CHAPTER V

IMPLEMENTATION AND TESTING

5.1 Implementation

5.1.1 Read the Database

The communication between all of these class is starts from the Home class. The method of readData will load all of the data from KATA2.txt and PARITTA.txt files.

```
public void readData()
{
        AssetManager asset_manager = this.getAssets();
        InputStream in_stream1 = null, in_stream2 = null;

        //open file
        try
        {
                in_stream1 = asset_manager.open("KATA2.txt");
                in_stream2 = asset_manager.open("PARITTA.txt");
        }
        catch (FileNotFoundException fnfe)
        {
                Toast.makeText(this, "File Not Found", Toast.LENGTH_SHORT).show();
                return; //quit from method
        }
        catch (IOException e)
        {
                // TODO Auto-generated catch block
                e.printStackTrace();
        }
```

Figure 5.1.1 Code of Open File in Asset

5.2.2 Load the Database into Hash Table

After all of the data are loaded then it will open those file and process each row from each files and save it to the hash table. For the data from KATA2.txt, it will split the sentence from each row by the space character and separate it to Indonesian word and Pali word. After that, the Indonesian word and Pali word will be save to the hash table of Indonesia-Pali.

```
//read file & show the result to console
try
{
        DataInputStream in = new DataInputStream(in_stream1);
        BufferedReader br = new BufferedReader(new InputStreamReader(in)); //membaca di konsul
        String strLine; //tampungan untuk membaca per baris dari data txt

        while ((strLine = br.readLine()) != null)
        {
                int panj=strLine.length(); //banyak karakter dalam satu kalimat
                int batas=0; //index dari spasi
                String pali="";
                String indo="";

                //ambil B.Pali
                for(int i=0;i<panj;i++)
                {
                        if(strLine.charAt(i)==32) //32=biner spasi
                        {
                                batas=i;
                                pali=strLine.substring(0,batas);
                                break;
                        }
                }

                //ambil B.indonesia
                for(int i=0;i<panj;i++)
                {
                        if(strLine.charAt(i)==32)
                        {
                                batas=i+1;
                                indo=strLine.substring(batas);
                                break;
                        }
                }

                abc.addHash1(indo, pali);
}
}
catch (IOException ioe)
{
        System.out.println(ioe.getMessage());
        return;
}
```

Figure 5.1.2 Code of Load File from KATA2.txt into Hash Table

This is the code of load the data from PARITTA.txt, it will split the sentence from each row by the colon character and separate it to Paritta's titile and Paritta's content. The Paritta's content will be splitted again by space character and then separate each word. Each of this word will be bundled with its title ad then saved to the hash table.

```
try
{
        DataInputStream in2 = new DataInputStream(in_stream2);
        BufferedReader brd = new BufferedReader(new InputStreamReader(in2)); //membaca di konsul
        String strLine; //tampungan untuk membaca per baris dari data txt

        while ((strLine = brd.readLine()) != null)
        {
                int pjg=strLine.length(); //banyak karakter dalam satu kalimat
                int bts=0; //index dari titik dua
                String pali1="";
                String pali2="";

                //ambil judul doa Pali
                for(int k=0;k<pjg;k++)
                {
                        if(strLine.charAt(k)==58) //58=biner titik dua
                        {
                                bts=k;
                                pali1=strLine.substring(0,bts);
                                break;
                        }
                }

                //ambil isi doa Pali
                for(int k=0;k<pjg;k++)
                {
                        if(strLine.charAt(k)==58)
                        {
                                bts=k+1;
                                pali2=strLine.substring(bts);
                                String[] parts = pali2.split(" ");
                                for(int l=0; l<parts.length; l++)
                                {
                                        bcd.addHash2(pali1, parts[l]);
                                }
                                break;
                        }
                }
        }
}
catch (IOException ioe)
{
        System.out.println(ioe.getMessage());
        return;
}
```

Figure 5.1.3 Code of Load File from PARITTA.txt into Hash Table

5.2.3 Displaying the Hash Table

This is the code HashTable class and the attribute is hashArray which is an array of LinkList object. The method of displayTable1 is used to display all of the items in hash table of Indonesia-Pali which contain Indonesian words and Pali words. The method of displayTable2 is used to display all of the items in hash table of Paritta which contain Paritta's title and Paritta's content.

```
 1 package com.example.tugasakhir;
 2
 3 import java.util.ArrayList;
 4
 5 class HashTable
 6 {
 7         private LinkList[] hashArray;
 8
 9         public HashTable(int size)
10         {
11                 hashArray = new LinkList[size];
12                 for(int i=0; i<hashArray.length; i++)
13                         hashArray[i] = new LinkList();
14         }
15
16         public void displayTable1()
17         {
18                 for(int i=0; i<hashArray.length; i++)
19                 {
20                         System.out.print(" " + i + " : ");
21                         hashArray[i].displayList1();
22                 }
23         }
24
25         public void displayTable2()
26         {
27                 for(int j=0; j<hashArray.length; j++)
28                 {
29                         System.out.print(" " + j + " : ");
30                         hashArray[j].displayList2();
31                 }
32         }
```

Figure 5.1.4 Code of Display the Hash Table

The method of hashFunction1 is used to calculate the index in the hash table of Indonesia-Pali for the given word and hashFunction2 method is used to calculate the index in the hash table of Paritta. Those function sums the ASCII values of the letters in a string and apply the modulus operator using size of hash to generate a value within the table range. The method of addHash1 is used to add an item to the hash table of Indonesia-Pali in a calculated index before and the method of addHash2 is used to add an item to the hash table of Paritta in a calculated index before.

```
public int hashFunction1(String indo)
{
        int jml=0;
        int index=0;
        int panj2=indo.length();
        for(int j=0;j<panj2;j++)
        {
                jml=jml+indo.charAt(j);
        }
        index=jml%25;
        return index;
}

public void addHash1(String indo, String pali)
{
        int hashVal = hashFunction1(indo);
        hashArray[hashVal].addLink1(indo, pali);
}

public int hashFunction2(String pali1)
{
        int jumlah=0;
        int idx=0;
        int pjg2=pali1.length();
        for(int h=0;h<pjg2;h++)
        {
                jumlah=jumlah+pali1.charAt(h);
        }
        idx=jumlah%20;
        return idx;
}

public void addHash2(String pali1, String pali2)
{
        int hashValue = hashFunction2(pali1);
        hashArray[hashValue].addLink2(pali1, pali2);
}
```

Figure 5.1.5 Code of Hash Function & Add Hash

5.2.4 Searching Process

This is the code of searching process, the find method will calculate the index of the search keyword and use it to search the hash table of Indonesia-Pali for the given search keyword in the given row index.

```
public String find(String indo)
{
        int hashVal = hashFunction1(indo);
        System.out.println("At index " + hashVal + " of Pali's Hash Table");
        return hashArray[hashVal].searchLink(indo);
}
```

Figure of 5.1.6 Code of Searching Process I

Meanwhile, the method of find2 will search the Paritta's title in each row in the hash table of Paritta with call the method of searchLink2 in LinkList class and will receive the list of Paritta's title which contain the translated word. After that, the method will search the Paritta's content for each item in the above list of Paritta's title in each row of the hash table through the method of displayList3 in the LinkList class. It will receive the combined of Paritta's word content and save it to the list of Paritta's word content. Then the list of Paritta's title and Paritta's content will be passed to the Search class to be processed.

```
public ArrayList<ArrayList<String>> find2(String pali)
{
    ArrayList<ArrayList<String>> value = new ArrayList<ArrayList<String>>();
    ArrayList<String> list_title = new ArrayList<String>();
    ArrayList<String> list_paritta = new ArrayList<String>();

    for(int j=0; j<hashArray.length; j++)
    {
        ArrayList<String> title = hashArray[j].searchLink2(pali);
        if(title.size() > 0)
        {
            for(int i=0; i<title.size(); i++)
            {
                String pali1 = title.get(i);
                list_title.add(pali1);
                list_paritta.add(hashArray[j].displayList3(pali1));
            }
        }
    }
    value.add(list_title);
    value.add(list_paritta);
    return value;
}
```

Figure 5.1.7 Code of Searching Process II

This is the code of LinkList class which have two attributes. Head and tail are the object of Link class, meanwhile the hd and tl are the object of Link2 class. The method of isEmpty and isEmpty2 method is used to check whether the linked list is empty. The method of addLink1 is used to insert an item to the Indonesia-Pali linked list and the method of displayList1 is used to display content from each item in the linked list.

```
1  package com.example.tugasakhir;
2
3  import java.util.ArrayList;
4
5  class LinkList
6  {
7          private Link head;
8          private Link tail;
9          private Link2 hd;
10         private Link2 tl;
11
12         public LinkList()
13         {
14                 head = null;
15                 tail = null;
16                 hd = null;
17                 tl = null;
18         }
19
20         public boolean isEmpty()
21         {
22                 return (head == null);
23         }
24
25         public boolean isEmpty2()
26         {
27                 return (hd == null);
28         }
29
30         //add data to the end
31         public void addLink1(String id, String pl)
32         {
33                 Link baru = new Link(id, pl);
34                 if (isEmpty())
35                 {
36                         head = baru;
37                         tail = baru;
38                 }
39                 else
40                 {
41                         tail.setNext(baru);
42                         baru.setPrev(tail);
43                         tail=baru;
44                 }
45         }
46
47         public void displayList1()
48         {
49                 Link p = head;
50                 while (p != null)
51                 {
52                         p.displayLink1();
53                         p = p.getNext();
54                 }
55                 System.out.println("");
56         }
```

Figure 5.1.8 Code of addLink1 and displayLink1

This is the code of the method of addLink2 which is used to insert an item to the Pariita's linked list and the method of displayList2 is used to display content from each item in the linked list.

```
public void addLink2(String jpali, String cpali)
{
        Link2 baru2 = new Link2(jpali, cpali);
        if (isEmpty2())
        {
                hd = baru2;
                tl = baru2;
        }
        else
        {
                tl.setLanjut(baru2);
                baru2.setBalik(tl);
                tl=baru2;
        }
}

public void displayList2()
{
        Link2 h = hd;
        while (h != null)
        {
                h.displayLink2();
                h = h.getLanjut();
        }
        System.out.println("");
}
```

Figure 5.1.9 Code of addLink2 and displayLink2

This is the method of searchLink which is used to search through the Indonesia-Pali linked list for given search keyword and will return the Pali word for the given keyword.

```
public String searchLink(String searchWord)
{
        Link a = head;
        String result = "";
        System.out.print("Searching : " + searchWord + " ------> ");

        while(a!=null)
        {
                if(!a.getIndo().equals(searchWord))
                {
                        if(a.getNext() == null)
                        {
                                System.out.println("Result : DATA NOT FOUND!");
                                System.out.println("\t");
                                break;
                        }
                }
                else if(a.getIndo().equals(searchWord))
                {
                        result = a.getPali();
                        System.out.println("Result : " +result);
                        System.out.println("\t");
                        break;
                }
                a = a.getNext();
        }
        return result;
}
```

Figure 5.1.10 Code of searchLink

This is the method of searchLink2 which is used to search through the Paritta linked list for given the Pali word and will return the list of Paritta's title. The method of displayList3 will be used to combine all of the Paritta's word content for its corresponding Paritta's title.

```
public ArrayList<String> searchLink2(String searchDoa)
{
        ArrayList<String> a = new ArrayList<String>();
        Link2 h = hd;
        while (h != null)
        {
                String temp = h.findLink2(searchDoa);
                if(!temp.equals(""))
                {
                        if(!a.contains(temp))
                        a.add(temp);
                }
                h = h.getLanjut();
        }
        return a;
}

public String displayList3(String palil)
{
        String combined_pali2 = "";
        Link2 h = hd;
        while (h != null)
        {
                if(h.getPalil().equals(palil)) combined_pali2 += (h.displayLink3() + " ");
                h = h.getLanjut();
        }
        return combined_pali2;
}
```

Figure 5.1.11 Code of searchLink2 and displayList3

This is the Link class which have setter and getter for each attribute. Indo and pali are the object of String class, meanwhile prev and next are the object of Link class. The method of displayLink1 is used to display the Indonesia-Pali word for the current Link.

```
1  package com.example.tugasakhir;
2
3  class Link
4  {
5          private String indo;
6          private String pali;
7          private Link prev;
8          private Link next;
9
10         public Link(String id, String pl)
11         {
12                 indo = id;
13                 pali = pl;
14                 prev = null;
15                 next = null;
16         }
17
18         public void displayLink1()
19         {
20                 if (next == null)
21                         System.out.print(" (" + getIndo() + " : " + getPali() + "," + next + ") ");
22                 else
23                         System.out.print(" (" + getIndo() + " : " + getPali() +") --->");
24         }
25
26         public void setIndo(String id)
27         {
28                 indo = id;
29         }
30
31         public String getIndo()
32         {
33                 return indo;
34         }
35
36         public void setPali(String pl)
37         {
38                 pali = pl;
39         }
40
41         public String getPali()
42         {
43                 return pali;
44         }
45
46         public void setPrev(Link n)
47         {
48                 prev = n;
49         }
50
51         public Link getPrev()
52         {
53                 return prev;
54         }
55
56         public void setNext(Link n)
57         {
58                 next = n;
```

Figure 5.1.12 Code of Link Class

This is the Link2 class which have setter and getter for each attribute. Pali1 and pali2 are the object of String class, meanwhile balik and lanjut are the object of Link2 class. The method of displayLink2 is used to display the Paritta's content word and its corresponding title for the current Link2. The method of findLink2 is used to search each item of the linked list and compare the Paritta's word content with the given search keyword.

```
1  package com.example.tugasakhir;
2
3  class Link2
4  {
5          private String pali1;
6          private String pali2;
7          private Link2 balik;
8          private Link2 lanjut;
9
10         public Link2(String jpali, String cpali)
11         {
12                 pali1 = jpali;
13                 pali2 = cpali;
14                 balik = null;
15                 lanjut = null;
16         }
17
18         public void displayLink2()
19         {
20                 if (lanjut == null)
21                         System.out.print(" (" + getPali1() + " : " + getPali2() + "," + lanjut + ") ");
22                 else
23                         System.out.print(" (" + getPali1() + " . " + getPali2() +") ----");
24         }
25
26         public String findLink2(String searchDoa)
27         {
28                 String result = "";
29                 String pali2 = getPali2();
30
31                 if(pali2.toLowerCase().contains(searchDoa.toLowerCase())
32                         || pali1.toLowerCase().contains(searchDoa.toLowerCase()))
33                         result = getPali1();
34
35                 return result;
36         }
37
38         public String displayLink3()
39         {
40                 return getPali2();
41         }
42
43         public void setPali1(String jpali)
44         {
45                 pali1 = jpali;
46         }
47
48         public String getPali1()
49         {
50                 return pali1;
51         }
52
53         public void setPali2(String cpali)
54         {
55                 pali2 = cpali;
56         }
57
58         public String getPali2()
```

Figure 5.1.13 Code of Link2 Class

This Search class have txt_search object of EditText class and will be used to control the input field of search keyword in the form layout. The method of onCreate is called as soon as the form is started and will change the layout to activity_search layout file. The method of handleClick will determine what will happen when the user press the search button on the layout form. It will get the search keyword from the input field, then it will search the result through the searchKata method. After the search result is received, then the result will be passed to Result form through the extra value of an intent and start the Result form.

```
13 public class Search extends Activity
14 {
15         private EditText txt_search;
16
17     @Override
18     public void onCreate(Bundle savedInstanceState)
19     {
20         super.onCreate(savedInstanceState);
21         setContentView(R.layout.activity_search);
22
23         txt_search = (EditText)findViewById(R.id.txt_search);
24     }
25
26     @Override
27     public boolean onCreateOptionsMenu(Menu menu)
28     {
29         getMenuInflater().inflate(R.menu.activity_search, menu);
30         return true;
31     }
32
33     public void handleClick(View view)
34     {
35         switch(view.getId())
36         {
37         case R.id.btn_search:
38                 String search_keyword = txt_search.getText().toString();
39                 ArrayList<ArrayList<String>> search_result = null;
40                 if(!search_keyword.equals(""))
41                         {
42                         search_result = searchKata(search_keyword);
43                         if(search_result != null)
44                         {
45
46                                 ArrayList<String> list_title = search_result.get(0);
47                                 ArrayList<String> list_paritta = search_result.get(1);
48
49                                 Intent intent_search = new Intent(Search.this, Result.class);
50                                 intent_search.putExtra("search_keyword", search_keyword);
51                                 intent_search.putExtra("list_title", list_title);
52                                 intent_search.putExtra("list_paritta", list_paritta);
53                                 startActivity(intent_search);
54                         }
55                         else Toast.makeText(this, "Data not found", Toast.LENGTH_SHORT).show();
56                         }
57                 else Toast.makeText(this, "Search keyword must not be empty", Toast.LENGTH_SHORT).show();
58                 break;
59         }
60     }
61
62     public ArrayList<ArrayList<String>> searchKata(String indo)
63         {
64                 String pali = Home.abc.find(indo);
65                 if(!pali.equals(""))
66                         return Home.bcd.find2(pali);
67                 else return null;
68         }
```

Figure 5.1.14 Code of Search Class

This Result class is used to display the list of search result for the given search keyword. The Result class will contain list of Paritta's title and its corresponding content.

```
import android.os.Bundle;
import android.app.Activity;
import android.content.Intent;
import android.view.Menu;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.TextView;

public class Result extends Activity
{
        private ListView list_view;
        private TextView view_result;

    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_result);

        list_view = (ListView)findViewById(R.id.listView1);

        Intent intent_search = getIntent();
        final ArrayList<String> list_title = intent_search.getStringArrayListExtra("list_title");
        final ArrayList<String> list_paritta = intent_search.getStringArrayListExtra("list_paritta");
        String search_keyword = intent_search.getStringExtra("search_keyword");

        list_view.setAdapter(new ArrayAdapter<String>(this, android.R.layout.simple_list_item_1, list_title));

        list_view.setOnItemClickListener(new OnItemClickListener()
        {

                        public void onItemClick(AdapterView<?> parent, View view, int position, long id)
                        {
                                // TODO Auto-generated method stub
                                String clicked_item = (String) parent.getItemAtPosition(position);

                                Intent intent_detail = new Intent(Result.this, ResultDetail.class);
                                intent_detail.putExtra("title", clicked_item);
                                intent_detail.putExtra("paritta", list_paritta.get(position));
                                startActivity(intent_detail);
                        }
                });

        view_result = (TextView)findViewById(R.id.view_result);
        view_result.setText("Result search for " + search_keyword + ": ");
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu)
    {
        getMenuInflater().inflate(R.menu.activity_result, menu);
        return true;
    }
}
```

Figure 5.1.15 Code of Result Class

This Result class have attribute of list_view object of ListView class which is used for controlling the List View on the layout and view_result of TextView class object which is used for controlling the Text View on the layout. The method of onCreate will be called as soon as the form is started and will change the layout to activity_result layout file. This class will receive two search result value from Search class. These values are the list of Paritta's titile and the list of Paritta's content. The list of Paritta's content will be displayed on the above list view through an abject of ArrayAdapter class. The list view is given a listener named onItemClickListener which will conntrol about what will happen when the user choose one item from the list view. The listener has been customized so when the user choose one item from the list view, it will pass the chosen Paritta's title and Paritta's content to the ResultDetail form and start a ResultDetail form. The view_result attribute will display the search keyword typed by the user before.

The last class is the ResultDetail class which is used to display the choosen Paritta's title and its corresponding content passed by the Result form. The attribute of this class is view_title and view_paritta object from TextView class which is used for controlling the TextView on the form layout. The method of onCreate is called as soon as the ResultDetail form is started. The values which is passed by Result form will be caught in the form of extra value of an intent and will bring two additional values of Paritta's title and Paritta's content. The value of Paritta's title will be displayed on view_title object of TextView class and the value of Paritta's content will be displayed on view_paritta object of TextView class. Meanwhile, onCreateOptionsMenu will be used for controlling the menu item on ResultDetail form.

```
 1 package com.example.tugasakhir;
 2
 3 import android.os.Bundle;
 4 import android.app.Activity;
 5 import android.content.Intent;
 6 import android.view.Menu;
 7 import android.widget.TextView;
 8
 9 public class ResultDetail extends Activity
10 {
11         private TextView view_title, view_paritta;
12
13     @Override
14     public void onCreate(Bundle savedInstanceState)
15     {
16         super.onCreate(savedInstanceState);
17         setContentView(R.layout.activity_result_detail);
18
19         Intent intent_detail = getIntent();
20         String title = intent_detail.getStringExtra("title");
21         String paritta = intent_detail.getStringExtra("paritta");
22
23         view_title = (TextView)findViewById(R.id.view_title);
24         view_paritta = (TextView)findViewById(R.id.view_paritta);
25
26         view_title.setText(title);
27         view_paritta.setText(paritta);
28     }
29
30     @Override
31     public boolean onCreateOptionsMenu(Menu menu)
32     {
33         getMenuInflater().inflate(R.menu.activity_result_detail, menu);
34         return true;
35     }
36 }
```

Figure 5.1.16 Figure of ResultDetail Class

5.2 Testing

This is the application interface which have a Enter button for start search the Paritta and Quit button if the user want to quit from the application. This interface is set in layout of the activity_home which is in Home class.

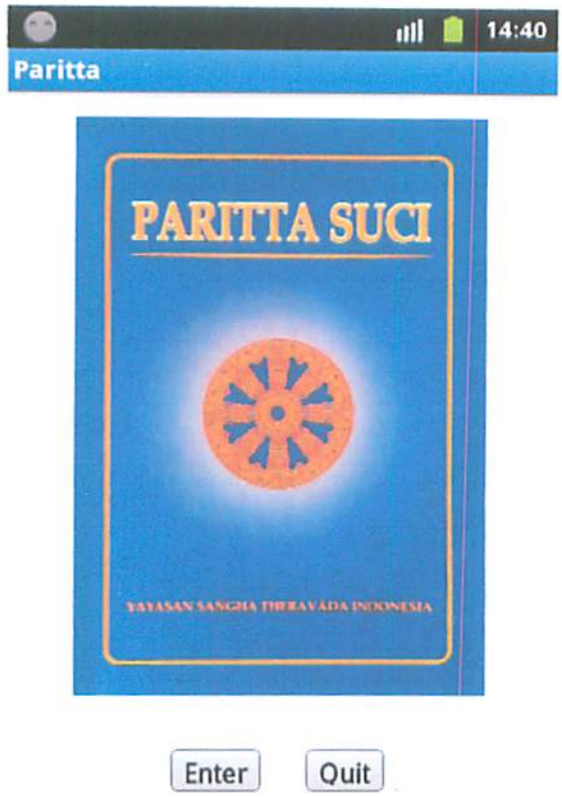

Figure 5.2.1 The Application Interface of the Home Class

If the user choose the Enter button, it will change the layout of activity_home to the layout of activity_search form from Search class. It will have a input field of search keyword in the form layout. For the example, if the search keyword is "berkah" then it will get the search keyword from the input field and will search the result through the searchKata method. The method of searchKata will search through the saved hash table for the given keyword

and will return the search result in the form of list of Paritta's title and list of Paritta's content.

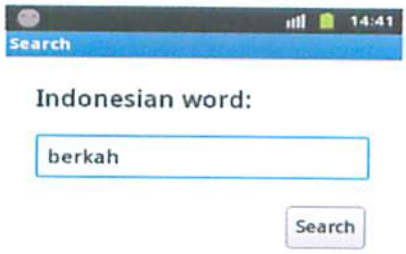

Figure 5.2.2 The Application Interface of the Search Class

After the search result is received, then this result will be passed to Result form through the extra value of an intent and will change to the layout of activity_result in the Result class. The Result class will receive two search result value from the Search class before. These values are the list of Paritta's title and the list of Paritta's content. The list of Paritta's content will be displayed on the above list view via an object of ArrayAdapter class.

Figure 5.2.3 The Application Interface of the Result Class

The list view is given a listener named onItemClickListener which will control when the user choose one item from the list view. The listener has been customized so when the user choose one item from the list view, it will pass the chosen Paritta's title and Paritta's content to the ResultDetail form and start a ResultDetail form. For the example, if the user choose "RATANATTAYANUBHAVENATIADI GATHA", then the ResultDetail class will call the layout of activity_result_detail. The values which is passed by the Result form will be caught in the form of extra value of an intent. The intent will bring two additional values, Paritta's title and Paritta's content. The value of Paritta's title will be displayed on view_title objct of TextView class and the value of Paritta's content will be displayed on view_paritta object of TextView class.

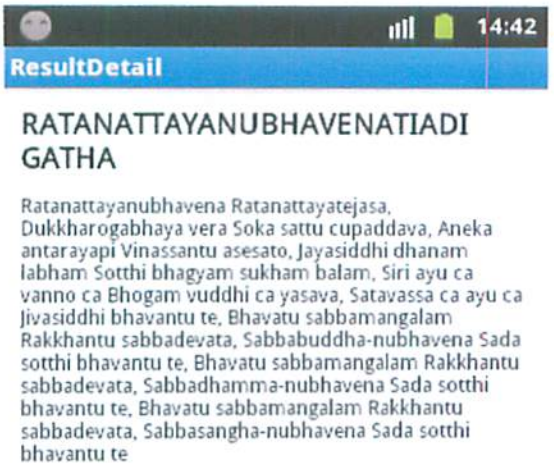

Figure 5.2.4 The Application Interface of ResultDetail Class