

CHAPTER 5

IMPLEMENTATION AND TESTING

5.1 Implementation

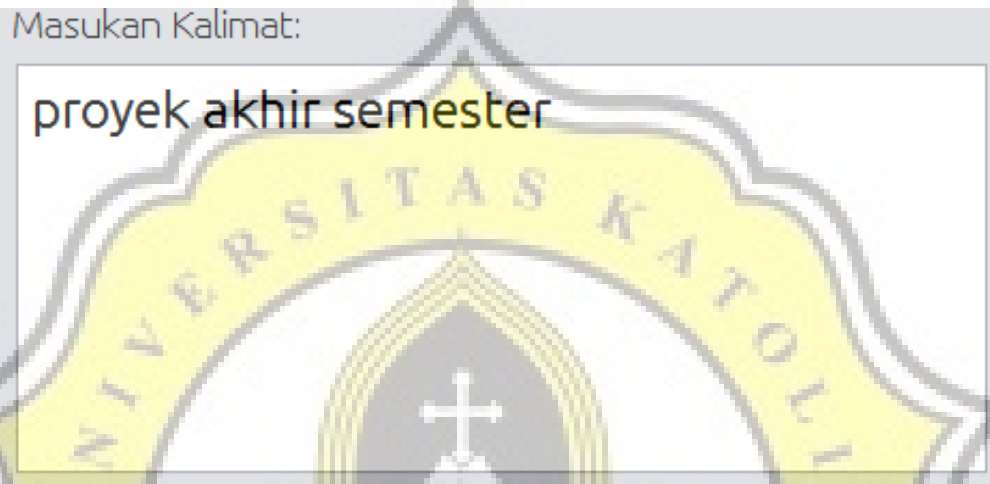


Illustration 5.1: Input user

This project uses Java Language with array as the data structure. The array is used in every process in this project from splitting the word until store the syllables sound. For example :

```
1. String kalimat = GUI.jTextArea1.getText();  
2. String kataBesar=kalimat.toUpperCase();  
3. String[] a = kataBesar.split(spasi);
```

the first line is to get the user input from text area. The next line is to normalize that input into upper case letter. And then the last row is the declaration for array to store each word from user input using split to break the text.

After get all the words, the process continue to check each syllables. First the system check the first letter of the word. If the first letter is vowel, the word will break between 3 different syllables classification.

```
4. if(isVokal(k.charAt(i))){  
5.   if(!isVokal(k.charAt(i+1))){  
6.     if(!isVokal(k.charAt(i+2))&&isVokal(k.charAt(i+3))){  
7.       if(k.charAt(i+2)=='Y' || k.charAt(i+2)=='y'){
```

```

8.         sukukata[cskt]=k.substring(i,i+1);
9.         GUI.jTextArea2.setText(GUI.jTextArea2.getText()
        +sukukata[cskt]+"\\n");
10.        cskt++;
11.    }
12.    else{
13.        sukukata[cskt]=k.substring(i,i+2);
14.        GUI.jTextArea2.setText(GUI.jTextArea2.getText()
        +sukukata[cskt]+"\\n");
15.        cskt++;
16.        i++;
17.    }
18. }
19. elseif(!isVokal(k.charAt(i+2))&&!isVokal(k.charAt(i+3))){
20.     sukukata[cskt]=k.substring(i,i+3);
21.     GUI.jTextArea2.setText(GUI.jTextArea2.getText()
        +sukukata[cskt]+"\\n");
22.     cskt++;
23.     i=i+2;
24. }
25. else{
26.     sukukata[cskt]=k.substring(i,i+1);
27.     GUI.jTextArea2.setText(GUI.jTextArea2.getText()
        +sukukata[cskt]+"\\n");
28.     cskt++;
29. }
30. }
31. else{
32.     sukukata[cskt]=k.substring(i,i+1);
33.     GUI.jTextArea2.setText(GUI.jTextArea2.getText()
        +sukukata[cskt]+"\\n");
34.     cskt++;
35. }
36. }
37.

```

the first line is to check the first letter of the word if the first letter is vowel, the word continue to the next condition to determine the syllabels. In line 4, the word will be checked is the next letter is vowel or consonant. If the next letter is consonant, the word continue to the next condition. In line 6-18, if the next word is consonant and the next of the next word is vowel, the system will check is the next letter is G or Y. If the next letter is G or Y, the word will break in V syllabels (example: ANYAR, A-NYAR). If not, the word will break in VC syllabels (example: ANDA, AN-DA) and back to the loop to check the other syllabels of the word. In line 19-24, if the next letter is consonant and the next again is consonant too, the word will break in VCC syllabels (example: ANGKA,

ANG-KA) and back to the loop. In line 25-29, if the next word is vowel, the word will break in V syllables (example: AKU, A-KU) and back to the loop. And in the line 31-35, if the first next word is vowel, the word is break in V syllables (example: AIR, A-IR) and back to the loop.

If the first letter of the word is consonant, the word will break between 6 different syllables classification.

```

38.     if(!isVokal(k.charAt(i+3))&&!isVokal(k.charAt(i+4))){
39.         sukukata[cskt]=k.substring(i,i+4);
40.         GUI.jTextArea2.setText(GUI.jTextArea2.getText()+sukukata[cskt]+"\n");
41.         cskt++;
42.         i=i+3;
43.     }

```

This code is to break the word in CVCC syllables (example: RANGKA, RANG-KA).

```

44.     elseif(!isVokal(k.charAt(i+2))&&!
45.         isVokal(k.charAt(i+3))&&isVokal(k.charAt(i+4))){
46.         if(k.charAt(i+3)=='G' || k.charAt(i+3)=='Y' ||
47.             k.charAt(i+3)=='R'){
48.             GUI.jTextArea2.setText(GUI.jTextArea2.getText()+sukukata[cskt]+"\n");
49.             cskt++;
50.             i++;
51.         }
52.         else{
53.             sukukata[cskt]=k.substring(i,i+3);
54.             GUI.jTextArea2.setText(GUI.jTextArea2.getText()+sukukata[cskt]+"\n");
55.             cskt++;
56.             i=i+2;
57.         }
58.     }

```

Code above is to break word in CV and CVC syllables. Line 44-48 is the CV syllables (example: RAPI, RA-PI), and line 49-54 is the CVC syllables (example: BANTAL, BAN-TAL).

```

57.     if(!isVokal(k.charAt(i+4))&&!isVokal(k.charAt(i+5))){
58.         sukukata[cskt]=k.substring(i,i+5);
59.         GUI.jTextArea2.setText(GUI.jTextArea2.getText()+sukukata[cskt]+"\n");
60.         cskt++;
61.         i=i+4;
62.     }

```

This code is to break word in CCVCC syllables (example: PRANGKO, PRANG-KO)

```

63.     else if(!isVokal(k.charAt(i+4))){
64.         if(k.charAt(i+4)=='G' || k.charAt(i+4)=='Y' ||
        k.charAt(i+4)=='R'){
65.             GUI.jTextArea2.setText(GUI.jTextArea2.getText()
        +sukukata[cskt]+"\\n");
66.             cskt++;
67.             i=i+2;
68.         }
69.     } else{
70.         sukukata[cskt]=k.substring(i,i+4);
71.         GUI.jTextArea2.setText(GUI.jTextArea2.getText(
        )+sukukata[cskt]+"\\n");
72.         cskt++;
73.         i=i+3;
74.     }
75. }

```

The code above is to break word in CCV and CCVC syllables. Line 63-67 is the CCV syllables (example: PRAKATA, PRA-KA-TA), and line 68-73 is the CCVC syllables (example: PRAKTEK, PRAK-TEK).

Hasil Check Suku Kata:

PRO YEK AK HIR SE MES TER

Illustration 5.2: hasil pemecahan kata

To play the sound, this project use java library “Javax.sound.saple” to locate the sound file and play the sound.

```

76.     for(String j: isi){
77.         File wavFile2 = new File("src/suara/"+j+".wav");

```

```

78.         if(!wavFile2.exists()){
79.             continue;
80.         }
81.         else{
82.             AudioInputStream stream;
83.             AudioFormat format;
84.             DataLine.Info info;
85.             Clip clip;
86.             stream = AudioSystem.getAudioInputStream(wavFile2);
87.             format = stream.getFormat();
88.             info = new DataLine.Info(Clip.class, format);
89.             clip = (Clip) AudioSystem.getLine(info);
90.             clip.open(stream);
91.             clip.start();
92.             while (!clip.isRunning())
93.                 Thread.sleep(10);
94.             while (clip.isRunning())
95.                 Thread.sleep(10);
96.             clip.close();
97.         }

```

In line 75, the system start looping for each array to check the syllabels sound. In line 76, the system choose in the source from array in line 75. line 77-79, if the array value doesn't suit to any database, that value is skipped and continue to the next value. In line 80-96, if the value is exist, the sound initialize its format and initialize clip. The clip is to play or stop the sound and format is to determine what file is to play. The line 89-90 is to open the file and play the sound. In line 91-94, the thread must be sleep so that sound can come out. Line 95 is to stop the sound so the sound not repeated.

5.2 Testing

This test is to show the different between small text amount and big text amount. Word that used is AKU (2 syllables 3 letters), SAYA (2 syllables 4 letter), PANCI (2 syllables 5 letters), SUPAYA (3 syllables 6 letters), PRAKATA (3 syllables 7 letters), MATEMATIKA (5 syllables 10 letters).

Table 5.1: tabel waktu

	2S 3L	2S 4L	2S 5L	3S 6L	3S 7L	5S 10L
100	20	25	40	42	37	54
500	126	133	152	192	261	562
1000	198	240	240	350	380	1395
5000	1429	1371	1266	2238	2755	6552

After the number of letter getting larger, it can be seen that the amount of time is getting longer too. But it appears that words with the same number of syllables do not have a significant time difference even though the number of letters is getting longer. If described in chart will show like this:

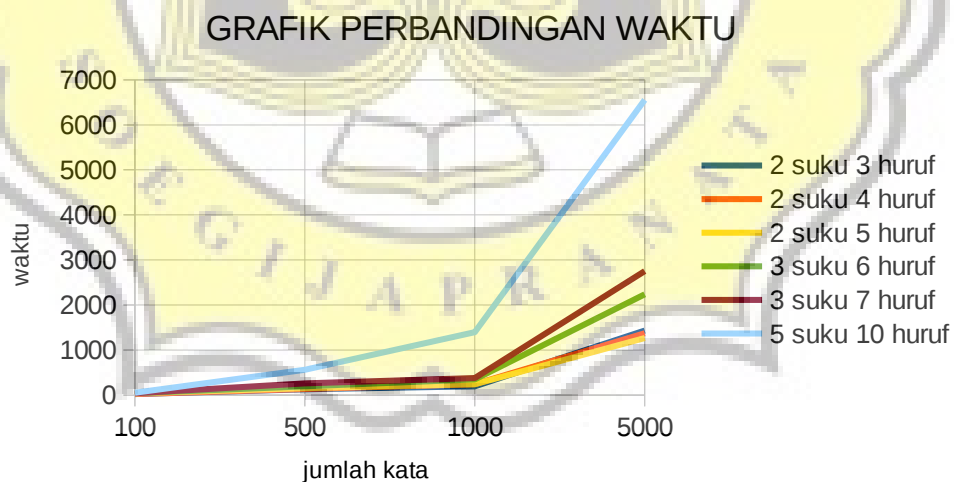


Illustration 5.3: Grafik Perbandingan