

## BAB 5 IMPLEMENTASI

### 5.1 Spesifikasi Sistem

#### 5.1.1 Spesifikasi Perangkat Keras

Dalam implementasi *named entity recognition* pada *factoid question answering system* pada sistem ini diperlukan spesifikasi perangkat keras seperti pada Tabel 5.1.

**Tabel 5.1 Spesifikasi Perangkat keras**

Komponen	Spesifikasi
Processor	Intel(R) Core(TM) i5
Memori	2048 MB
Hardisk	500 GB

#### 5.1.2 Spesifikasi Perangkat Lunak

Dalam implementasi *Named Entity Recognition* pada *factoid question answering system* pada sistem ini diperlukan spesifikasi perangkat lunak seperti pada Tabel 5.2.

**Tabel 5.2 Spesifikasi perangkat lunak**

Nama	Ketentuan
Sistem Operasi	Windows 10
Bahasa Pemrograman	PHP
Tools Pemrograman	XAMPP dan Sublime

### 5.2 Implementasi Algoritme

Implementasi algoritme metode *Named Entity Recognition* dalam *Question Answering System* pada cerita rakyat Indonesia akan dijelaskan pada sub bab berikutnya.

#### 5.2.1 Implementasi Algoritme Penggabungan Dua Kata

Algoritme klasifikasi penggabungan dua kata merupakan proses klasifikasi menggunakan metode *Naive Bayes* yang dihitung berdasarkan kata yang ada dalam koleksi pustaka yang kemudian dihitung probabilitas kata tersebut dapat digabung dengan kata sebelumnya atau tidak dengan melibatkan sembilan parameter yang terdapat dalam data latih. Parameter yang digunakan dalam klasifikasi tersebut adalah *POS tagging* kata tersebut, *POS tagging* kata sebelumnya, *POS tagging* dua kata sebelumnya, *morphologi* kata tersebut, *morphologi* kata sebelumnya, *morphologi* dua kata sebelumnya, *sentence level* kata tersebut, *sentence level* kata sebelumnya dan *sentence level* dua kata sebelumnya. Kode program dari implementasi algoritme klasifikasi penggabungan dua kata dapat dilihat pada Kode Program 5.1.

1	Public function bayes_gb1(\$postag1,\$postag2,\$postag3,
2	\$m1,\$m2,\$m3,\$s1,\$s2,\$s3) {
3	ini set('max execution time', 0);

```

4  $data = $this->db->query("Select * from datalatih");
5  $jumlah_data = count($data->result_array());
6  $gabung = 'ya';
7  $tidak = 'tidak';
8  $Quantity='Quantity';
9  $Product='Product';
10 $Job = 'Job'; $Location = 'Location';
11 $None = 'None'; $Person='Person';
12 $jumlah_gb = count($this->select(" where gabung1 =
13 '$gabung'"));
14 $jumlah_td =count($this->select(" where gabung1 =
15 '$tidak'"));
16 $jumlah_gb2 = count($this->select(" where gabung2 =
17 '$gabung'"));
18 $jumlah_td2 = count($this->select(" where gabung2 =
19 '$tidak'"));
20
21 $p_ya=$jumlah_gb/$jumlah_data;
22 $p_tidak=$jumlah_td/$jumlah_data;
23
24 //Ya
25 $tag1_ya = count($this->select(" where postag1 = '$postag1'
26 AND gabung1 = '$gabung'"));
27 $tag2_ya = count($this->select(" where postag2 = '$postag2'
28 AND gabung1 = '$gabung'"));
29 $tag3_ya = count($this->select(" where postag3 = '$postag3'
30 AND gabung1 = '$gabung'"));
31 $ptag1_ya = $tag1_ya/$p_ya;$ptag2_ya =
32 $tag2_ya/$p_ya;$ptag3_ya = $tag3_ya/$p_ya;
33 $morp1_ya = count($this->select(" where morfologi1 = '$m1'
34 AND gabung1 = '$gabung'"));
35 $morp2_ya = count($this->select(" where morfologi2 = '$m2'
36 AND gabung1 = '$gabung'"));
37 $morp3_ya = count($this->select(" where morfologi3 = '$m3'
38 AND gabung1 = '$gabung'"));
39 $pmorp1_ya = $morp1_ya/$p_ya;$pmorp2_ya =
40 $morp2_ya/$p_ya;$pmorp3_ya = $morp3_ya/$p_ya;
41
42 $s1_ya = count($this->select(" where s_level1 = '$s1' AND
43 gabung1 = '$gabung'"));
44 $s2_ya = count($this->select(" where s_level2 = '$s2' AND
45 gabung1 = '$gabung'"));
46 $s3_ya = count($this->select(" where s_level3 = '$s3' AND
47 gabung1 = '$gabung'"));
48 $ps1_ya = $s1_ya/$p_ya;$ps2_ya = $s2_ya/$p_ya;$ps3_ya =
49 $s3_ya/$p_ya;
50
51 //Tidak
52 $tag1_tidak = count($this->select(" where postag1 =
53 '$postag1' AND gabung1 = '$tidak'"));
54 $tag2_tidak = count($this->select(" where postag2 =
55 '$postag2' AND gabung1 = '$tidak'"));
56 $tag3_tidak = count($this->select(" where postag3 =
57 '$postag3' AND gabung1 = '$tidak'"));
58 $ptag1_tidak = $tag1_tidak/$p_tidak;$ptag2_tidak =
59 $tag2_tidak/$p_tidak;$ptag3_tidak = $tag3_tidak/$p_tidak;
60 $morp1_tidak = count($this->select(" where morfologi1 =
61 '$m1' AND gabung1 = '$tidak'"));
62 $morp2_tidak = count($this->select(" where morfologi2 =

```

```

63 '$m2' AND gabung1 = '$tidak'));
64 $morp3_tidak = count($this->select(" where morfologi3 =
65 '$m3' AND gabung1 = '$tidak'));
66 $pmorp1_tidak = $morp1_tidak/$p_tidak;$pmorp2_tidak =
67 $morp2_tidak/$p_tidak;$pmorp3_tidak =
68 $morp3_tidak/$p_tidak;
69
70 $s1_tidak = count($this->select(" where s_level1 = '$s1'
71 AND gabung1 = '$tidak'));
72 $s2_tidak = count($this->select(" where s_level2 = '$s2'
73 AND gabung1 = '$tidak'));
74 $s3_tidak = count($this->select(" where s_level3 = '$s3'
75 AND gabung1 = '$tidak'));
76 $ps1_tidak = $s1_tidak/$p_tidak;$ps2_tidak =
77 $s2_tidak/$p_tidak;$ps3_tidak = $s3_tidak/$p_tidak;
78
79 $probabilitas_ya=($ptag1_ya*$ptag2_ya*$ptag3_ya*$pmorp1_ya*
80 $pmorp2_ya*$pmorp3_ya*$ps1_ya*$ps2_ya*$ps3_ya*$p_ya);
81 $probabilitas_tidak=($ptag1_tidak*$ptag2_tidak*$ptag3_tidak
82 *$pmorp1_tidak*$pmorp2_tidak*$pmorp3_tidak*$ps1_tidak*$ps2_
83 tidak*$ps3_tidak*$p_tidak);
84 $gabung = 'tidak';
85     if($probabilitas_ya>$probabilitas_tidak){
86         $gabung = 'ya';
87     }else if($probabilitas_tidak>$probabilitas_ya){
88         $gabung = 'tidak';
89     }
90     return $gabung;
91 }

```

**Kode Program 5.1 Kode program klasifikasi penggabungan dua kata**

Pada implementasi algoritme diatas masukan yang didapat merupakan sembilan parameter yang telah disebutkan sebelumnya. Kemudian sistem akan menghitung tingkat probabilitas apakah kedua kata tersebut akan digabung atau tidak dengan mengambil nilai probabilitas yang terbesar dari kedua kemungkinan tersebut. Penjelasan dari Kode Program 5.1 adalah sebagai berikut:

1. Baris 12-19 merupakan inialisasi variabel yang berisikan jumlah data yang sesuai dengan parameter dalam basis data.
2. Baris 21-22 merupakan perhitungan perbandingan data yang digabung dan tidak digabung dengan seluruh data.
3. Baris 25-49 merupakan perhitungan probabilitas parameter kata yang sesuai dengan parameter yang diberikan pada *function* dan digabung.
4. Baris 52-77 merupakan perhitungan probabilitas parameter kata yang sesuai dengan parameter yang diberikan pada *function* dan tidak digabung.
5. Baris 79-80 merupakan perhitungan probabilitas kata tersebut dengan kata sebelumnya digabung.
6. Baris 81-83 merupakan perhitungan probabilitas kata tersebut dengan kata sebelumnya tidak digabung.

7. Baris 85-90 merupakan pemilihan kondisi untuk menentukan kata tersebut digabung atau tidak dengan memilih nilai tertinggi diantara probabilitas digabung dan tidak digabung.

### 5.2.2 Implementasi Algoritme Penggabungan Tiga Kata

Algoritme klasifikasi penggabungan tiga kata merupakan proses klasifikasi menggunakan metode *Naive Bayes* yang dihitung berdasarkan kata yang terdapat pada koleksi pustaka yang kemudian dihitung probabilitas kata tersebut apakah dapat digabung dengan kata sebelumnya atau tidak dengan melibatkan sepuluh parameter yang terdapat dalam data latih. Parameter yang digunakan dalam klasifikasi tersebut adalah POS *tagging* kata tersebut, POS *tagging* kata sebelumnya, POS *tagging* dua kata sebelumnya, *morphologi* kata tersebut, *morphologi* kata sebelumnya, *morphologi* dua kata sebelumnya, *sentence level* kata tersebut, *sentence level* kata sebelumnya, *sentence level* dua kata sebelumnya dan apakah kata tersebut digabung dengan kata sebelumnya. Kode program dari implementasi algoritme klasifikasi penggabungan dua kata dapat dilihat pada Kode Program 5.2.

```

1 public function bayes_gb2($postag1,$postag2,$postag3,$m1,
2 $m2,$m3,$s1,$s2,$s3,$gb1){
3 ini_set('max_execution_time', 0);
4 $data = $this->db->query("Select * from datalatih");
5 $jumlah_data = count($data->result_array());
6 $gabung='ya';$tidak='tidak';
7 $Quantity='Quantity';$Product='Product';
8 $Job='Job';$Location='Location';
9 $None = 'None'; $Person='Person';
10 $jumlah_gb = count($this->select(" where gabung1 =
11 '$gabung'"));
12 $jumlah_td =count($this->select(" where gabung1 =
13 '$tidak'"));
14 $jumlah_gb2 = count($this->select(" where gabung2 =
15 '$gabung'"));
16 $jumlah_td2 = count($this->select(" where gabung2 =
17 '$tidak'"));
18
19 $p_ya=$jumlah_gb/$jumlah_data;
20 $p_tidak=$jumlah_td/$jumlah_data;
21
22 //Ya
23
24 $tag1_ya = count($this->select(" where postag1 =
25 '$postag1' AND gabung2 = '$gabung'"));
26 $tag2_ya = count($this->select(" where postag2 =
27 '$postag2' AND gabung2 = '$gabung'"));
28 $tag3_ya = count($this->select(" where postag3 =
29 '$postag3' AND gabung2 = '$gabung'"));
30 $ptag1_ya = $tag1_ya/$p_ya;$ptag2_ya =
31 $tag2_ya/$p_ya;$ptag3_ya = $tag3_ya/$p_ya;
32
33 $morp1_ya = count($this->select(" where morfologi1 =
34 '$m1' AND gabung2 = '$gabung'"));
35 $morp2_ya = count($this->select(" where morfologi2 =
36 '$m2' AND gabung2 = '$gabung'"));

```

```

37 $morp3_ya = count($this->select(" where morfologi3 =
38 'm3' AND gabung2 = '$gabung'"));
39 $pmorp1_ya = $morp1_ya/$p_ya;$pmorp2_ya =
40 $morp2_ya/$p_ya;$pmorp3_ya = $morp3_ya/$p_ya;
41 $s1_ya = count($this->select(" where s_level1 = '$s1' AND
42 gabung2 = '$gabung'"));
43 $s2_ya = count($this->select(" where s_level2 = '$s2' AND
44 gabung2 = '$gabung'"));
45 $s3_ya = count($this->select(" where s_level3 = '$s3' AND
46 gabung2 = '$gabung'"));
47 $ps1_ya = $s1_ya/$p_ya;$ps2_ya = $s2_ya/$p_ya;$ps3_ya =
48 $s3_ya/$p_ya;
49
50 $gb1_ya = count($this->select(" where gabung1 = '$gb1' AND
51 gabung2 = '$gabung'"));
52 $pgb1_ya = $gb1_ya*$p_ya;
53
54 //Tidak
55
56 $tag1_tidak = count($this->select(" where postag1 =
57 '$postag1' AND gabung2 = '$tidak'"));
58 $tag2_tidak = count($this->select(" where postag2 =
59 '$postag2' AND gabung2 = '$tidak'"));
60 $tag3_tidak = count($this->select(" where postag3 =
61 '$postag3' AND gabung2 = '$tidak'"));
62 $ptag1_tidak = $tag1_tidak/$p_tidak;$ptag2_tidak =
63 $tag2_tidak/$p_tidak;$ptag3_tidak = $tag3_tidak/$p_tidak;
64
65 $morp1_tidak = count($this->select(" where morfologi1 =
66 'm1' AND gabung2 = '$tidak'"));
67 $morp2_tidak = count($this->select(" where morfologi2 =
68 'm2' AND gabung2 = '$tidak'"));
69 $morp3_tidak = count($this->select(" where morfologi3 =
70 'm3' AND gabung2 = '$tidak'"));
71 $pmorp1_tidak = $morp1_tidak/$p_tidak;$pmorp2_tidak =
72 $morp2_tidak/$p_tidak;$pmorp3_tidak =
73 $morp3_tidak/$p_tidak;
74
75 $s1_tidak = count($this->select(" where s_level1 = '$s1'
76 AND gabung2 = '$tidak'"));
77 $s2_tidak = count($this->select(" where s_level2 = '$s2'
78 AND gabung2 = '$tidak'"));
79 $s3_tidak = count($this->select(" where s_level3 = '$s3'
80 AND gabung2 = '$tidak'"));
81 $ps1_tidak = $s1_tidak/$p_tidak;$ps2_tidak =
82 $s2_tidak/$p_tidak;$ps3_tidak = $s3_tidak/$p_tidak;
83 $gb1_tidak = count($this->select(" where gabung1 = '$gb1'
84 AND gabung2 = '$tidak'"));
85 $pgb1_tidak = $gb1_tidak*$p_tidak;
86
87 $probabilitas_ya=($ptag1_ya*$ptag2_ya*$ptag3_ya*$pmorp1_ya
88 *$pmorp2_ya*$pmorp3_ya*$ps1_ya*$ps2_ya*$ps3_ya*$p_ya*$pgb1
89 _ya);
90
91 $probabilitas_tidak=($ptag1_tidak*$ptag2_tidak*$ptag3_tida
92 k*$pmorp1_tidak*$pmorp2_tidak*$pmorp3_tidak*$ps1_tidak*$ps
93 2_tidak*$ps3_tidak*$p_tidak*$pgb1_tidak);
94
95 $gabung='tidak';

```

96	
97	if(\$probabilitas_ya>\$probabilitas_tidak){
98	\$gabung='ya';}else{
99	if(\$probabilitas_tidak>\$probabilitas_ya){
100	\$gabung = 'tidak';
101	}
102	return \$gabung;
103	}

**Kode Program 5.2 Klasifikasi penggabungan tiga kata**

Pada implementasi algoritme di atas masukan yang didapat merupakan sepuluh parameter yang telah disebutkan sebelumnya. Kemudian sistem akan menghitung tingkat probabilitas apakah kedua kata tersebut akan digabung atau tidak dengan mengambil nilai probabilitas yang terbesar dari kedua kemungkinan tersebut. Penjelasan dari Kode Program 5.2 adalah sebagai berikut:

1. Baris 10-17 merupakan inisialisasi variabel yang berisikan jumlah data yang sesuai dengan parameter dalam basis data.
2. Baris 19-20 merupakan perhitungan perbandingan data yang digabung dan tidak digabung dengan seluruh data.
3. Baris 24-52 merupakan perhitungan probabilitas kata tersebut akan digabung dengan syarat parameter yang telah dimasukkan dalam *function*.
4. Baris 56-85 merupakan perhitungan probabilitas kata tersebut tidak digabung dengan syarat parameter yang telah dimasukkan dalam *function*.
5. Baris 87-89 merupakan perhitungan probabilitas kata tersebut akan digabung.
6. Baris 90-92 merupakan perhitungan probabilitas kata tersebut tidak akan digabung.
7. Baris 97-103 merupakan pemilihan kondisi untuk menentukan kata tersebut akan digabung atau tidak dengan memilih nilai tertinggi dari kedua probabilitas tersebut.

### 5.2.3 Implementasi Algoritme Perhitungan Klasifikasi *Named Entity*

Algoritme perhitungan klasifikasi *Named Entity* pada sistem ini diimplementasikan dengan menggunakan metode *Naive Bayes*. Implementasi dari algoritme ini dilakukan dengan melakukan perhitungan klasifikasi sebelas parameter dari koleksi pustaka terhadap data latih yang ada. Parameter yang digunakan dalam klasifikasi ini adalah POS *tagging* kata tersebut, POS *tagging* kata sebelumnya, POS *tagging* dua kata sebelumnya, *morphologi* kata tersebut, *morphologi* kata sebelumnya, *morphologi* dua kata sebelumnya, *sentence level* kata tersebut, *sentence level* kata sebelumnya, *sentence level* dua kata sebelumnya, apakah kata tersebut digabung dengan kata sebelumnya dan apakah kata tersebut digabung dengan dua kata sebelumnya. Implementasi dari algoritme ini dapat dilihat pada Kode Program 5.3.

1	public function NE(\$postag1, \$postag2, \$postag3, \$m1, \$m2, \$m3,
2	\$s1, \$s2, \$s3, \$gb1, \$gb2) {
3	ini_set('max_execution_time', 0);
4	\$data = \$this->db->query("Select * from datalatih");

```

5 $jumlah_data = count($data->result_array());
6 $gabung = 'ya'; $tidak =
7 'tidak'; $Quantity='Quantity'; $Product='Product';
8 $Job = 'Job'; $Location = 'Location'; $None = 'None';
9 $Person='Person';
10 $jumlah_gb = count($this->select(" where gabung1 =
11 '$gabung'"));
12 $jumlah_td =count($this->select(" where gabung1 =
13 '$tidak'"));
14 $jumlah_gb2 = count($this->select(" where gabung2 =
15 '$gabung'"));
16 $jumlah_td2 = count($this->select(" where gabung2 =
17 '$tidak'"));
18
19 $NE_Person = count($this->select(" where NE =
20 '$Person'"));
21 $NE_None = count($this->select(" where NE = '$None'"));
22 $NE_Location = count($this->select(" where NE =
23 '$Location'"));
24 $NE_Job = count($this->select(" where NE = '$Job'"));
25 $NE_Product = count($this->select(" where NE =
26 '$Product'"));
27 $NE_Quantity = count($this->select(" where NE =
28 '$Quantity'"));
29
30 $p_Quantity=$NE_Quantity/$jumlah_data;
31 $p_Product=$NE_Product/$jumlah_data;
32 $p_Person=$NE_Person/$jumlah_data;
33 $p_Job=$NE_Job/$jumlah_data;
34 $p_Location=$NE_Location/$jumlah_data;
35 //Quantity
36 $tag1_Quantity = count($this->select(" where postag1 =
37 '$postag1' AND NE = '$Quantity'"));
38 $tag2_Quantity = count($this->select(" where postag2 =
39 '$postag2' AND NE = '$Quantity'"));
40 $tag3_Quantity = count($this->select(" where postag3 =
41 '$postag3' AND NE = '$Quantity'"));
42 $ptag1_Quantity =
43 $tag1_Quantity/$p_Quantity; $ptag2_Quantity =
44 $tag2_Quantity/$p_Quantity; $ptag3_Quantity =
45 $tag3_Quantity/$p_Quantity;
46
47 $morp1_Quantity = count($this->select(" where morfologi1
48 = '$m1' AND NE = '$Quantity'"));
49 $morp2_Quantity = count($this->select(" where morfologi2
50 = '$m2' AND NE = '$Quantity'"));
51 $morp3_Quantity = count($this->select(" where morfologi3
52 = '$m3' AND NE = '$Quantity'"));
53 $pmorp1_Quantity=$morp1_Quantity/$p_Quantity;
54 $pmorp2_Quantity=$morp2_Quantity/$p_Quantity;
55 $pmorp3_Quantity = $morp3_Quantity/$p_Quantity;
56
57 $s1_Quantity=count($this->select(" where s_level1 = '$s1'
58 AND NE = '$Quantity'"));
59 $s2_Quantity = count($this->select(" where s_level2 =
60 '$s2' AND NE = '$Quantity'"));
61 $s3_Quantity = count($this->select(" where s_level3 =
62 '$s3' AND NE = '$Quantity'"));
63 $ps1_Quantity = $s1_Quantity/$p_Quantity;

```

```

64 $ps2_Quantity = $s2_Quantity/$p_Quantity;
65 $ps3_Quantity = $s3_Quantity/$p_Quantity;
66
67 $gb1_Quantity = count($this->select(" where gabung1 =
68 '$gb1' AND NE = '$Quantity'"));
69 $pgb1_Quantity = $gb1_Quantity*$p_Quantity;
70 $gb2_Quantity = count($this->select(" where gabung2 =
71 '$gb2' AND NE = '$Quantity'"));
72 $pgb2_Quantity = $gb2_Quantity*$p_Quantity;
73
74 //produk
75 $tag1_Product = count($this->select(" where postag1 =
76 '$postag1' AND NE = '$Product'"));
77 $tag2_Product = count($this->select(" where postag2 =
78 '$postag2' AND NE = '$Product'"));
79 $tag3_Product = count($this->select(" where postag3 =
80 '$postag3' AND NE = '$Product'"));
81 $ptag1_Product = $tag1_Product/$p_Product;$ptag2_Product =
82 $tag2_Product/$p_Product;$ptag3_Product =
83 $tag3_Product/$p_Product;
84
85 $morp1_Product = count($this->select(" where morfologi1 =
86 '$m1' AND NE = '$Product'"));
87 $morp2_Product = count($this->select(" where morfologi2 =
88 '$m2' AND NE = '$Product'"));
89 $morp3_Product = count($this->select(" where morfologi3 =
90 '$m3' AND NE = '$Product'"));
91 $pmorp1_Product=$morp1_Product/$p_Product;
92 $pmorp2_Product= $morp2_Product/$p_Product;
93 $pmorp3_Product = $morp3_Product/$p_Product;
94
95 $s1_Product = count($this->select(" where s_level1 = '$s1'
96 AND NE = '$Product'"));
97 $s2_Product = count($this->select(" where s_level2 = '$s2'
98 AND NE = '$Product'"));
99 $s3_Product = count($this->select(" where s_level3 = '$s3'
100 AND NE = '$Product'"));
101 $ps1_Product = $s1_Product/$p_Product;
102 $ps2_Product = $s2_Product/$p_Product;
103 $ps3_Product = $s3_Product/$p_Product;
104
105 $gb1_Product = count($this->select(" where gabung1 =
106 '$gb1' AND NE = '$Product'"));
107 $pgb1_Product = $gb1_Product*$p_Product;
108 $gb2_Product = count($this->select(" where gabung2 =
109 '$gb2' AND NE = '$Product'"));
110 $pgb2_Product = $gb2_Product*$p_Product;
111
112 //Person
113 $tag1_Person = count($this->select(" where postag1 =
114 '$postag1' AND NE = '$Person'"));
115 $tag2_Person = count($this->select(" where postag2 =
116 '$postag2' AND NE = '$Person'"));
117 $tag3_Person = count($this->select(" where postag3 =
118 '$postag3' AND NE = '$Person'"));
119 $ptag1_Person = $tag1_Person/$p_Person;$ptag2_Person =
120 $tag2_Person/$p_Person;$ptag3_Person =
121 $tag3_Person/$p_Person;
122

```



```

123 $morp1_Person = count($this->select(" where morphologi1 =
124 '$m1' AND NE = '$Person'"));
125 $morp2_Person = count($this->select(" where morphologi2 =
126 '$m2' AND NE = '$Person'"));
127 $morp3_Person = count($this->select(" where morphologi3 =
128 '$m3' AND NE = '$Person'"));
129 $pmorp1_Person = $morp1_Person/$p_Person;$pmorp2_Person =
130 $morp2_Person/$p_Person;$pmorp3_Person =
131 $morp3_Person/$p_Person;
132
133 $s1_Person = count($this->select(" where s_level1 = '$s1'
134 AND NE = '$Person'"));
135 $s2_Person = count($this->select(" where s_level2 = '$s2'
136 AND NE = '$Person'"));
137 $s3_Person = count($this->select(" where s_level3 = '$s3'
138 AND NE = '$Person'"));
139 $ps1_Person = $s1_Person/$p_Person;$ps2_Person =
140 $s2_Person/$p_Person;$ps3_Person = $s3_Person/$p_Person;
141
142 $gb1_Person = count($this->select(" where gabung1 = '$gb1'
143 AND NE = '$Person'"));
144 $pgb1_Person = $gb1_Person*$p_Person;
145 $gb2_Person = count($this->select(" where gabung2 = '$gb2'
146 AND NE = '$Person'"));
147 $pgb2_Person = $gb2_Person*$p_Person;
148
149 //Location
150 $tag1_Location = count($this->select(" where postag1 =
151 '$postag1' AND NE = '$Location'"));
152 $tag2_Location = count($this->select(" where postag2 =
153 '$postag2' AND NE = '$Location'"));
154 $tag3_Location = count($this->select(" where postag3 =
155 '$postag3' AND NE = '$Location'"));
156 $ptag1_Location=$tag1_Location/$p_Location;
157 $ptag2_Location=$tag2_Location/$p_Location;
158 $ptag3_Location = $tag3_Location/$p_Location;
159
160 $morp1_Location = count($this->select(" where morphologi1
161 = '$m1' AND NE = '$Location'"));
162 $morp2_Location = count($this->select(" where morphologi2
163 = '$m2' AND NE = '$Location'"));
164 $morp3_Location = count($this->select(" where morphologi3
165 = '$m3' AND NE = '$Location'"));
166 $pmorp1_Location=$morp1_Location/$p_Location;
167 $pmorp2_Location=$morp2_Location/$p_Location;
168 $pmorp3_Location = $morp3_Location/$p_Location;
169
170 $s1_Location=count($this->select(" where s_level1 = '$s1'
171 AND NE = '$Location'"));
172 $s2_Location=count($this->select(" where s_level2 = '$s2'
173 AND NE = '$Location'"));
174 $s3_Location=count($this->select(" where s_level3 = '$s3'
175 AND NE = '$Location'"));
176 $ps1_Location = $s1_Location/$p_Location;
177 $ps2_Location = $s2_Location/$p_Location;
178 $ps3_Location = $s3_Location/$p_Location;
179
180 $gb1_Location=count($this->select(" where gabung1 = '$gb1'
181 AND NE = '$Location'"));

```

```

182 $pgb1_Location = $gb1_Location*$p_Location;
183 $gb2_Location = count($this->select(" where gabung2 =
184 '$gb2' AND NE = '$Location'"));
185 $pgb2_Location = $gb2_Location*$p_Location;
186
187 $probabilitas_Location=($ptag1_Location*$ptag2_Location*$p
188 tag3_Location*$pmorp1_Location*$pmorp2_Location*$pmorp3_Lo
189 cation*$ps1_Location*$ps2_Location*$ps3_Location*$p_Locati
190 on*$pgb1_Location*$pgb2_Location);
191
192
193 $probabilitas_Person=($ptag1_Person*$ptag2_Person*$ptag3_P
194 erson*$pmorp1_Person*$pmorp2_Person*$pmorp3_Person*$ps1_Pe
195 rson*$ps2_Person*$ps3_Person*$p_Person*$pgb1_Person*$pgb2_
196 Person);
197
198 $probabilitas_Quantity=($ptag1_Quantity*$ptag2_Quantity*$p
199 tag3_Quantity*$pmorp1_Quantity*$pmorp2_Quantity*$pmorp3_Qu
200 antity*$ps1_Quantity*$ps2_Quantity*$ps3_Quantity*$p_Quant
201 ity*$pgb1_Quantity*$pgb2_Quantity);
202
203 $probabilitas_Product=($ptag1_Product*$ptag2_Product*$ptag
204 3_Product*$pmorp1_Product*$pmorp2_Product*$pmorp3_Product*
205 $ps1_Product*$ps2_Product*$ps3_Product*$p_Product*$pgb1_Pr
206 oduct*$pgb2_Product);
207
208 $maksimal=max($probabilitas_Product,$probabilitas_Quantity
209 , $probabilitas_Location, $probabilitas_Person);
210
211 $NE='None';
212 if($maksimal=$probabilitas_Person){
213     $NE = 'Person';
214     //echo $NE;
215 }else if($maksimal=$probabilitas_Product){
216     $NE = 'Product';
217     //echo $NE;
218 }else if($maksimal=$probabilitas_Quantity){
219     $NE = 'Quantity';
220     //echo $NE;
221 }else if($maksimal=$probabilitas_Location){
222     $NE = 'Location';
223     //echo $NE;
224 }
return $NE;
}

```

### Kode Program 5.3 Klasifikasi named entity

Pada Kode Program 5.3 masukan yang didapat merupakan sebelas parameter yang telah disebutkan sebelumnya. Kemudian sistem akan menghitung tingkat probabilitas kata atau frasa tersebut termasuk dalam *named entity* apa berdasarkan jumlah probabilitas yang terbesar terhadap masing- masing NE. Penjelasan Kode Program 5.3 adalah sebagai berikut:

1. Baris 10-17 merupakan perhitungan untuk menghitung jumlah data yang terdapat pada basis data sesuai dengan parameter yang terdapat pada *function*.
2. Baris 19-28 merupakan perhitungan untuk menghitung jumlah data pada masing-masing NE.

3. Baris 30-34 merupakan perhitungan untuk menghitung probabilitas masing-masing NE terhadap jumlah data.
4. Baris 36-72 merupakan perhitungan untuk menghitung probabilitas NE *Quantiy* dengan syarat parameter yang dijadikan masukan pada *function*.
5. Baris 75-110 merupakan perhitungan probabilitas NE *Product* pada basis data dengan syarat parameter yang terdapat pada *function*.
6. Baris 113-147 merupakan perhitungan probabilitas NE *Person* pada basis data dengan syarat parameter yang terdapat pada *function*.
7. Baris 150-185 merupakan perhitungan probabilitas NE *Location* pada basis data dengan syarat parameter yang terdapat pada *function*.
8. Baris 187-191 merupakan perhitungan probabilitas kata tersebut termasuk dalam NE *Location*.
9. Baris 193-196 merupakan perhitungan probabilitas kata tersebut termasuk dalam NE *Person*.
10. Baris 197-200 merupakan perhitungan probabilitas kata tersebut termasuk dalam NE *Quantity*.
11. Baris 201-204 merupakan perhitungan probabilitas kata tersebut termasuk dalam NE *Product*.
12. Baris 206-207 merupakan perhitungan untuk mencari nilai maksimal dari seluruh probabilitas NE.
13. Baris 209-221 merupakan pemilihan kondisi untuk mencari probabilitas yang bernilai sama dengan hasil perhitungan nilai maksimal pada NE.

#### 5.2.4 Implementasi Algoritme *Answer Finder*

Implentasi algoritme *answer finder* dilakukan dengan menghitung jarak antara kandidat jawaban dengan kata kunci pertanyaan dengan menggunakan metode *Manhattan*. Implementasi algoritme *answer finder* dapat dilihat pada Kode Program 5.4.

```

1 public function answer_finder($keyword,$NE){
2   ini_set('max_execution_time', 0);
3   $data = $this->db->query("Select * from data_korpus");
4   $data_latih = $data->result_array();
5   $bukan = array('VBT', 'VBI', 'IN', 'CC', 'NNG', 'RB', 'CDI');
6   $i=0;$kw[$i]='';
7
8     foreach ($keyword as $key) {
9       $kw[$i]=$key;
10      $i++;
11    }
12
13  $i=0;$kata[$i]=0;
14    foreach ($data_latih as $key) {
15      $kata[$key['No']]=$key['frasa'];
16      $i++;
17    }
18
19  $i=0;$indeks_keyword[$i] = 0;
20  $j=0; $indeks_jawab[$j]=0;
21    foreach ($data_latih as $key) {
22      foreach ($keyword as $value) {

```

```

23     foreach ($NE as $key1) {
24         if($value==$key['dasar']){
25             $indeks_keyword[$i]= intval($key['No']);
26             $i++;
27         }
28
29         if(($key1==$key['NE'])&&($key['postag1']='NN')
30         &&(!in_array($key['dasar'],$kw)){
31             $indeks_jawab[$j]= intval($key['No']);$j++;
32         }}}
33
34     $i=0;$j=0;$jarak[$j]=0;$No[$j]=0;
35     foreach ($indeks_keyword as $key) {
36         foreach ($indeks_jawab as $value) {
37             $jarak[$j] = abs($key-$value);
38
39             $No[$j] = $value;
40             $j++;}
41             $i++;}
42     $terdekat = min($jarak); $i=0;$j=0;
43
44     foreach ($jarak as $key) {
45         if($key==$terdekat){
46             $indeks = $No[$j];
47             break;
48             }$j++;}
49
50     return $kata[$indeks];
51 }
52

```

#### **Kode Program 5.4 Answer finder**

Pada Kode Program 5.4 masukan terhadap sistem merupakan kata kunci pertanyaan dan NE kandidat jawaban dari pertanyaan yang ada. Sistem terlebih dahulu mencari NE yang sesuai dengan NE kandidat jawaban kemudian menghitung jarak antara NE tersebut dengan kata kunci pertanyaan. Selanjutnya diambil jarak terkecil yang merupakan jawaban dari sistem terhadap pertanyaan yang ada. Penjelasan Kode Program 5.4 adalah sebagai berikut:

1. Baris 3-4 merupakan pengaksesan seluruh data dari tabel data\_korpus dalam basis data.
2. Baris 5 merupakan inialisasi array bernama bukan.
3. Baris 8-11 merupakan perulangan yang berfungsi menampung keyword dari basis data ke dalam array dengan nama kw.
4. Baris 13-16 merupakan perulangan yang berfungsi menampung frasa dalam variabel kata
5. Baris 19-33 merupakan perulangan yang berfungsi untuk menampung frasa dari basis data dengan nama kata dengan indeks nomor data pada basis data.
6. Baris 35-41 merupakan perulangan yang berfungsi untuk mencari kandidat jawaban yang sesuai dengan NE yang terdapat pada parameter *function* beserta jaraknya terhadap kata kunci pertanyaan.

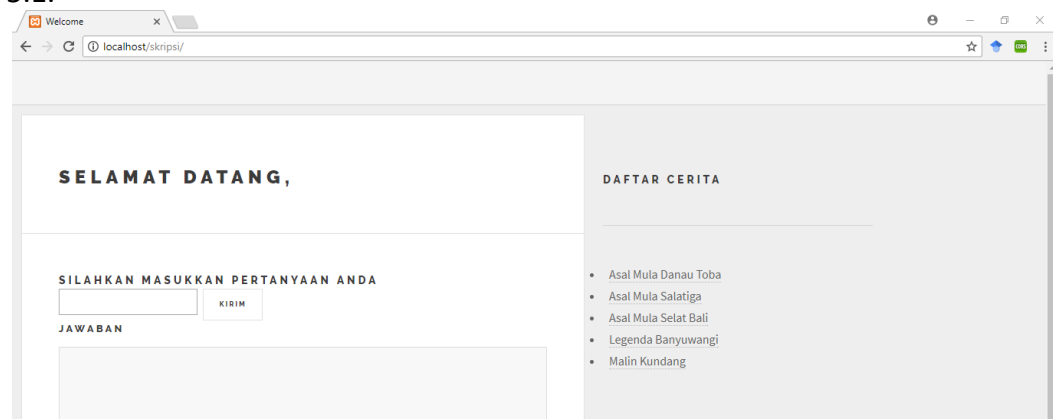
7. Baris 42 berfungsi untuk menghitung nilai minimal dari jarak NE terhadap kata kunci pertanyaan.
8. Baris 44-48 berfungsi untuk mencari indeks data dengan jarak terdekat dengan kata kunci pertanyaan.

### 5.3 Implementasi Antarmuka Sistem

Pada sistem *question answering* ini terdapat satu halaman utama yang digunakan user untuk memasukkan pertanyaan serta mendapatkan jawaban dari pertanyaan yang diberikan pengguna.

#### 5.3.1 Implementasi Halaman Utama Sistem

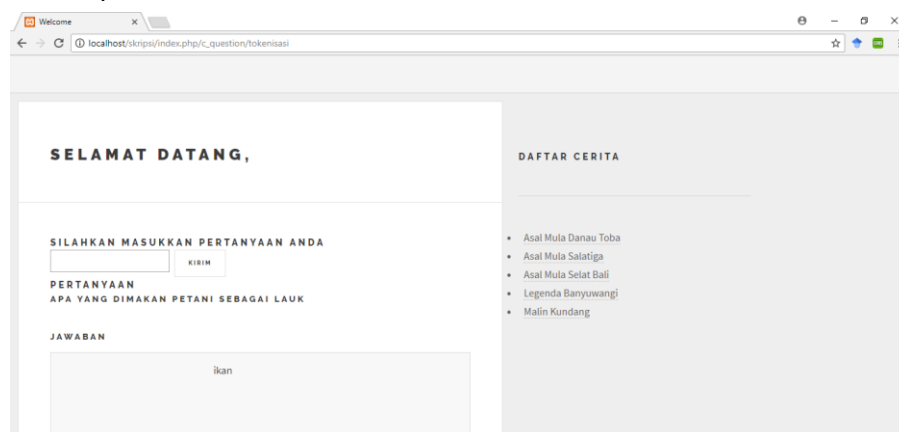
Halaman utama sistem berisi *textbox* yang digunakan oleh pengguna untuk memberikan pertanyaan kepada sistem dengan menekan tombol kirim. Selain itu di sebelah kanan halaman utama terdapat daftar cerita yang dijadikan koleksi pustaka dalam sistem dan merupakan batasan pertanyaan yang dapat diberikan kepada sistem. Implementasi halaman utama sistem dapat dilihat pada Gambar 5.1.



Gambar 5.1 Halaman utama

#### 5.3.2 Implementasi Halaman Utama Ketika Menampilkan Jawaban Sistem

Jawaban dari pertanyaan pengguna yang diberikan oleh sistem diberikan pada halaman utama yang sebelumnya merupakan tempat pengguna untuk memasukkan pertanyaan. Jawaban tersebut oleh sistem ditunjukkan pada kolom jawaban. Implementasi halaman utama ketika menampilkan jawaban dari sistem dapat dilihat pada Gambar 5.2.



Gambar 5.2 Halaman utama ketika menampilkan jawaban sistem