

## BAB VI

### PENGUJIAN DAN ANALISIS

Pada bab ini dilakukan proses pengujian dan analisis terhadap Sistem Pendukung Keputusan Pemilihan Konsentrasi di Program Studi Teknik Informatika Universitas Brawijaya. Proses pengujian dilakukan melalui empat tahapan, yaitu pengujian unit, pengujian integrasi, pengujian validasi dan pengujian validitas sistem. Pada pengujian unit dan pengujian integrasi akan digunakan teknik pengujian *White Box*. Pada pengujian validasi akan digunakan teknik pengujian *Black Box*.

#### 6.1 Pengujian

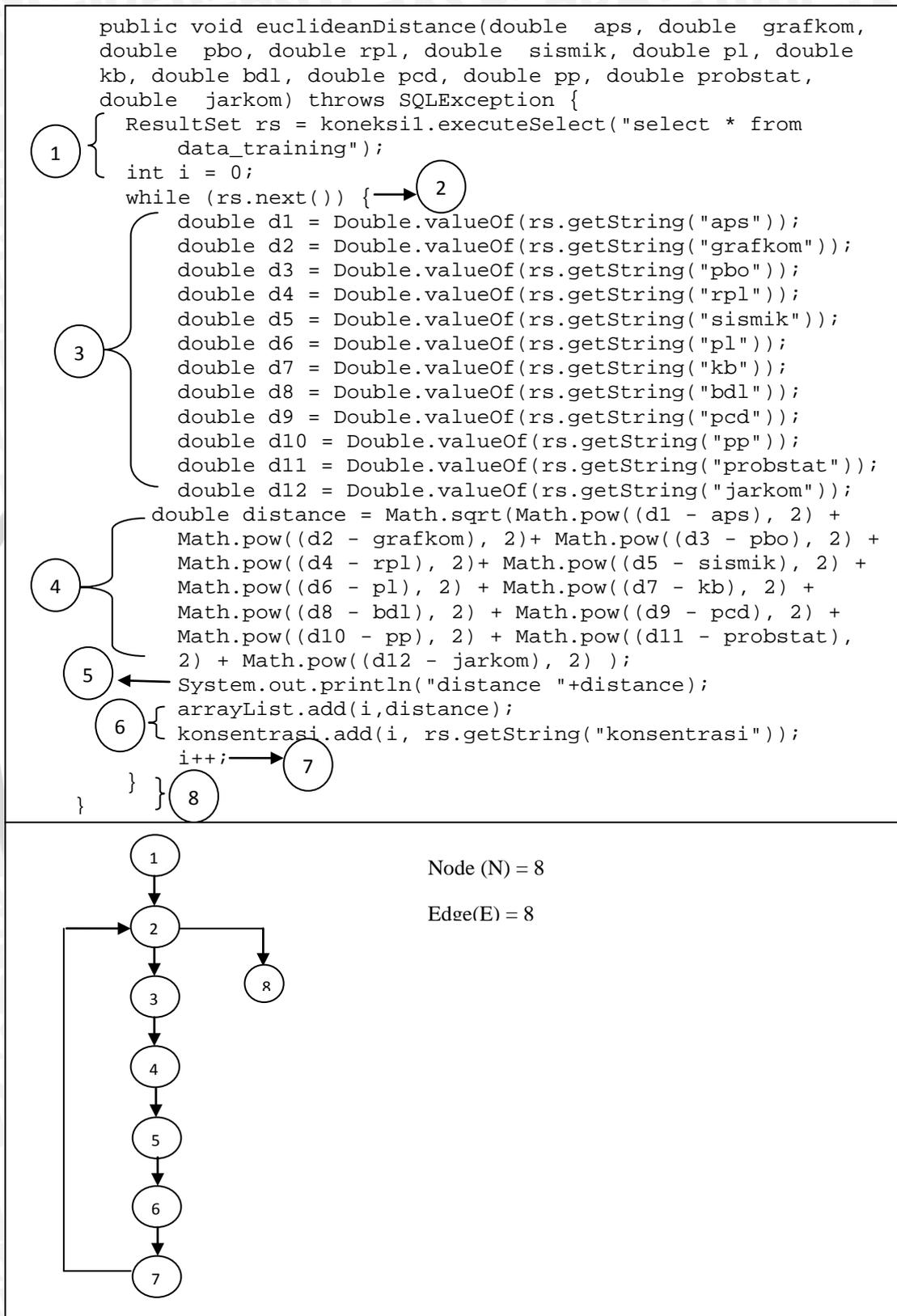
Proses pengujian dilakukan melalui tiga tahapan yaitu pengujian unit, pengujian integrasi, dan pengujian validasi.

##### 6.1.1 Pengujian Unit

Pada pengujian unit Sistem Pendukung Keputusan Pemilihan Konsentrasi PS.Teknik Informatika Universitas Brawijaya digunakan teknik pengujian *White Box* dengan teknik *Basis Path Testing*. Pada teknik *Basis Path Testing*, proses pengujian dilakukan dengan memodelkan algoritma pada suatu *flow graph*, menentukan jumlah kompleksitas siklomatis (*cyclomatic complexity*), menentukan sebuah basis set dari jalur independen dan memberikan kasus uji (*test case*) pada setiap basis set yang telah ditentukan. Penulisan laporan penelitian ini hanya dicantumkan hasil pengujian unit untuk algoritma dari beberapa metode (operasi) saja (tidak untuk keseluruhan metode).

##### a. Pengujian Unit untuk Operasi `euclideanDistance()` pada Kelas `Knn`

Operasi `euclideanDistance()` merupakan operasi yang digunakan untuk menghitung jarak antara data tes dengan data pelatihan. Gambar 6.1 menunjukkan proses pemodelan dalam *flow graph* pada operasi `euclideanDistance()`.



**Gambar 6.1** Pemodelan operasi euclideanDistance() ke dalam flow graph

Sumber: Pengujia



Pemodelan ke dalam *flow graph* yang telah dilakukan terhadap operasi `euclideanDistance()` menghasilkan jumlah kompleksitas siklomatis (*cyclometric complexity*) melalui persamaan  $V(G) = E - N + 2$ , dimana  $V(G)$  merupakan jumlah kompleksitas siklomatis,  $E$  merupakan sisi garis (garis penghubung antar *node*), dan  $N$  merupakan jumlah simpul (*node*).

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 8 - 8 + 2 \\ &= 2 \end{aligned}$$

Dari nilai *cyclometric complexity* yang telah dihasilkan dari perhitungan yaitu ditentukan lima buah basis set dari jalur *independent* yaitu:

Jalur 1 : 1 - 2 - 8

Jalur 2: 1 - 2 - 3 - 4 - 5 - 6 - 7 - ...

Penentuan kasus uji untuk masing-masing jalur dan hasil eksekusi untuk masing-masing kasus uji dijelaskan pada Tabel 6.1.

**Tabel 6.1** Test Case untuk Pengujian Unit Operasi `euclideanDistance()`

Jalur	Kasus Uji	Hasil yang Diharapkan	Hasil yang Didapatkan
1	Tidak ada data nilai mata kuliah yang dimasukkan	Sistem akan menampilkan nilai jarak <i>euclidean</i> masing-masing data pelatihan	Sistem menampilkan nilai jarak <i>euclidean</i> masing-masing data pelatihan
2	Memasukkan semua data nilai mata kuliah	Sistem akan menampilkan nilai jarak <i>euclidean</i> masing-masing data pelatihan	Sistem menampilkan nilai jarak <i>euclidean</i> masing-masing data pelatihan

**Sumber:** Pengujian

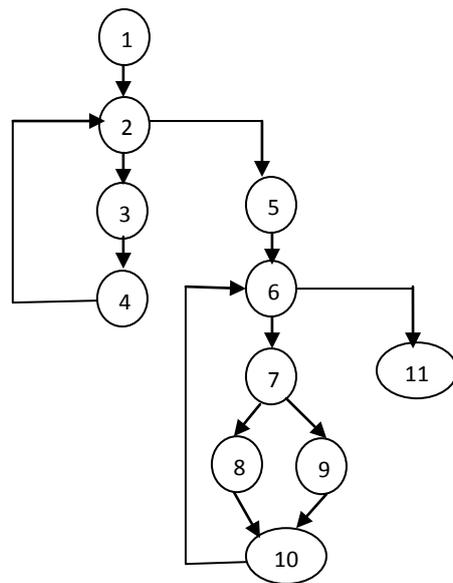
#### b. Pengujian Unit untuk Operasi `nearestNeighbor()` pada Kelas `Knn`

Operasi `nearestNeighbor()` merupakan operasi yang digunakan untuk menentukan tetangga terdekat berdasarkan jarak *euclidean* yang telah dihitung sebelumnya. Gambar 6.2 menunjukkan proses pemodelan dalam *flow graph* pada operasi `nearestNeighbor()`.

```

public void nearestNeighbor() {
1  ← arrayList=insertSort(arrayList) → 2
    for (int i = 0; i < k; i++) { → 2
        voting.add(i, (String)konsentrasi.get(i));
        int x = i + 1;
        fp.getOutputStream().append("\n Tetangga " + x + " \n jarak = "
3  {
        + arrayList.get(i) + " \n konsentrasi = " +
        konsentrasi.get(i));
        System.out.println(arrayList.get(i));
4  ← }
    String hasil = "\nhasil \t= MAX {"; → 5
6  ← for (int i = 0; i < voting.size(); i++) { → 5
    if (i == voting.size() - 1) { → 7
8  ← hasil += voting.get(i);
        } else {
        hasil += voting.get(i) + ", "; → 9
        }
10 ← }
    hasil += "}";
    System.out.println(hasil);
    fp.getOutputStream().append(hasil);
    }
11

```



Node(N) = 11

Edge(E) = 13

**Gambar 6.2** Pemodelan operasi nearestNeighbor() ke dalam *flow graph*  
**Sumber:** Pengujian

Pemodelan ke dalam *flow graph* yang telah dilakukan terhadap operasi nearestNeighbor() menghasilkan jumlah kompleksitas siklomatis (*cyclomatic complexity*) melalui persamaan  $V(G) = E - N + 2$ , dimana  $V(G)$

merupakan jumlah kompleksitas siklomatis, E merupakan sisi garis (garis penghubung antar *node*), dan N merupakan jumlah simpul (*node*).

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 13 - 11 + 2 \\ &= 4 \end{aligned}$$

Dari nilai *cyclometric complexity* yang telah dihasilkan dari perhitungan yaitu ditentukan lima buah basis set dari jalur *independent* yaitu:

Jalur 1 : 1 – 2 – 5 – 6 – 11

Jalur 2 : 1 – 2 – 3 – 4 – ...

Jalur 3 : 1 – 2 – 5 – 6 – 7 – 8 – 10 – ...

Jalur 4 : 1 – 2 – 5 – 6 – 7 – 9 – 10 – ...

Penentuan kasus uji untuk masing-masing jalur dan hasil eksekusi untuk masing-masing kasus uji dijelaskan pada Tabel 6.2

**Tabel 6.2** Test Case untuk Pengujian Unit Operasi `nearestNeighbor()`

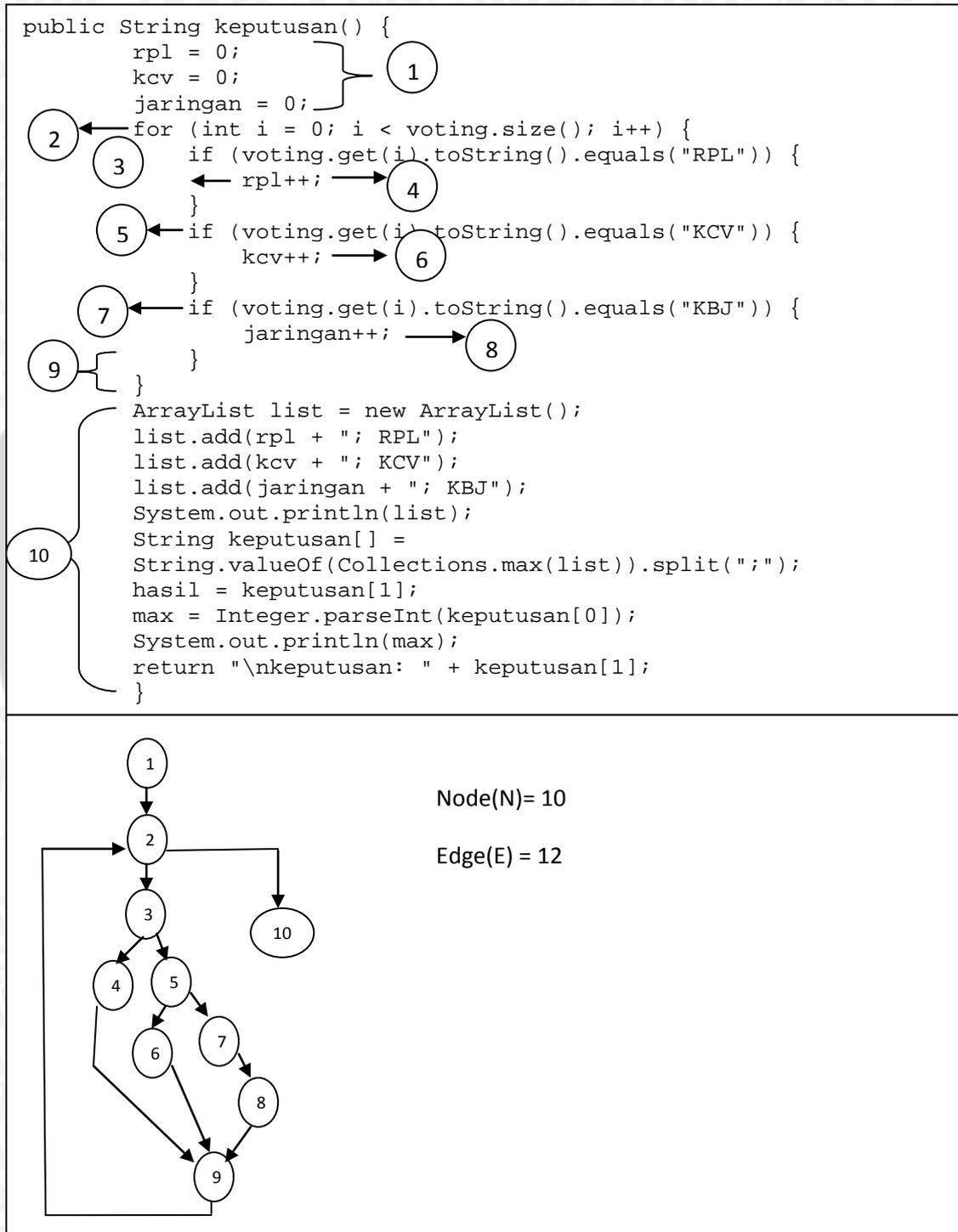
Jalur	Kasus Uji	Hasil yang Diharapkan	Hasil yang Didapatkan
1	Tidak ada nilai :”k” dan tidak ada nilai yang dihitung.	Sistem tidak menampilkan hasil perhitungan dan menampilkan pesan <i>error</i>	Sistem tidak menampilkan hasil perhitungan dan menampilkan pesan <i>error</i>
2	Nilai “k” ditentukan	Sistem menampilkan hasil perhitungan	Sistem menampilkan hasil perhitungan
3	Nilai “k=1”	Sistem menampilkan 1 tetangga terdekat	Sistem menampilkan 1 tetangga terdekat
4	Nilai “k>1”	Sistem akan menampilkan tetangga terdekat sejumlah “k”	Sistem akan menampilkan tetangga terdekat sejumlah “k”

Sumber: Pengujian

### c. Pengujian Unit untuk Operasi `keputusan()` pada Kelas `Knn`

Operasi `keputusan()` merupakan operasi yang digunakan untuk menentukan keputusan dari sistem berdasarkan jarak *euclidean* dan tetangga

terdekat yang telah dihitung sebelumnya. Gambar 6.3 menunjukkan proses pemodelan dalam *flow graph* pada operasi `keputusan()`.



**Gambar 6.3** Pemodelan operasi `keputusan()` ke dalam *flow graph*  
**Sumber:** Pengujian

Pemodelan ke dalam *flow graph* yang telah dilakukan terhadap operasi keputusan() menghasilkan jumlah kompleksitas siklomatis (*cyclometric complexity*) melalui persamaan  $V(G) = E - N + 2$ , dimana  $V(G)$  merupakan jumlah kompleksitas siklomatis,  $E$  merupakan sisi garis (garis penghubung antar *node*), dan  $N$  merupakan jumlah simpul (*node*).

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 12 - 10 + 2 \\ &= 4 \end{aligned}$$

Dari nilai *cyclometric complexity* yang telah dihasilkan dari perhitungan yaitu ditentukan lima buah basis set dari jalur *independent* yaitu:

Jalur 1 : 1 - 2 - 10

Jalur 2 : 1 - 2 - 3 - 4 - 9 - ...

Jalur 3 : 1 - 2 - 3 - 5 - 6 - 9 - ...

Jalur 4 : 1 - 2 - 3 - 5 - 7 - 8 - 9 - ...

Penentuan kasus uji untuk masing-masing jalur dan hasil eksekusi untuk masing-masing kasus uji dijelaskan pada Tabel 6.3

**Tabel 6.3** Test Case untuk Pengujian Unit Operasi keputusan ( )

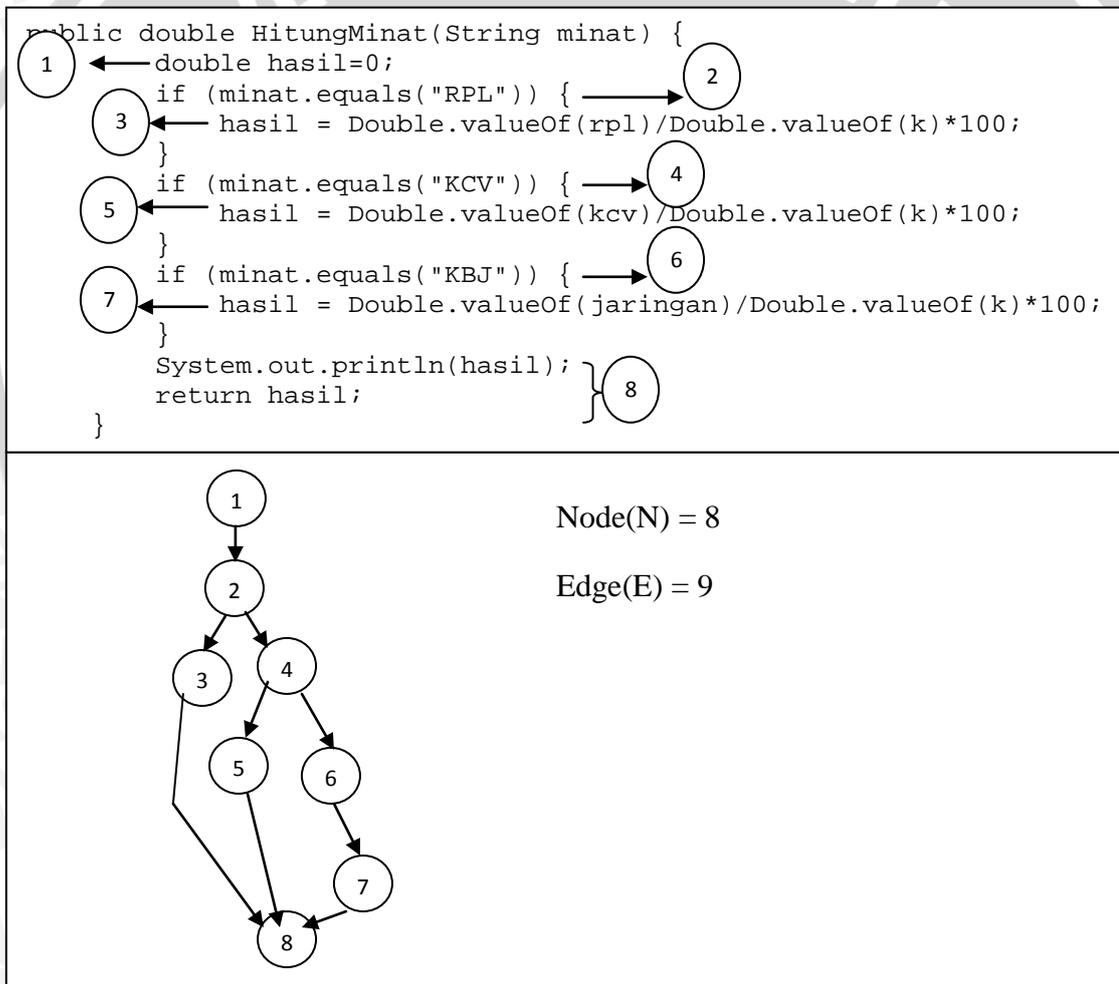
Jalur	Kasus Uji	Hasil yang Diharapkan	Hasil yang Didapatkan
1	Tidak ada Pelatihan yang dimasukkan	Sistem tidak menampilkan hasil keputusan	Sistem tidak menampilkan hasil keputusan
2	Data pelatihan yang dimasukkan dengan kategori RPL	Sistem akan menampilkan hasil keputusan	Sistem menampilkan hasil keputusan
3	Data pelatihan yang dimasukkan dengan kategori KCV	Sistem akan menampilkan hasil keputusan	Sistem menampilkan hasil keputusan
4	Data pelatihan yang	Sistem akan menampilkan hasil	Sistem menampilkan hasil keputusan

dimasukkan keputusan dengan kategori KBJ

Sumber: Pengujian

**d. Pengujian Unit untuk Operasi hitungMinat ( ) pada Kelas Knn**

Operasi hitungMinat() merupakan operasi yang digunakan untuk menghitung prosentase kecocokan antara konsentrasi yang diminati dengan hasil analisis sistem.. Gambar 6.4 menunjukkan proses pemodelan dalam *flow graph* pada operasi hitungMinat().



**Gambar 6.4** Pemodelan operasi hitungMinat ( ) ke dalam *flow graph*  
Sumber: Pengujian

Pemodelan ke dalam *flow graph* yang telah dilakukan terhadap operasi hitungMinat() menghasilkan jumlah kompleksitas siklomatis (*cyclometric*

*complexity*) melalui persamaan  $V(G) = E - N + 2$ , dimana  $V(G)$  merupakan jumlah kompleksitas siklomatis,  $E$  merupakan sisi garis (garis penghubung antar *node*), dan  $N$  merupakan jumlah simpul (*node*).

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 9 - 8 + 2 \\ &= 3 \end{aligned}$$

Dari nilai *cyclometric complexity* yang telah dihasilkan dari perhitungan yaitu ditentukan lima buah basis set dari jalur *independent* yaitu:

Jalur 1 : 1 – 2 – 3 – 8

Jalur 2 : 1 – 2 – 4 – 5 – 8 - ...

Jalur 3 : 1 – 2 – 4 – 6 – 7 – 8 - ...

Penentuan kasus uji untuk masing-masing jalur dan hasil eksekusi untuk masing-masing kasus uji dijelaskan pada Tabel 6.4

**Tabel 6.4** *Test Case* untuk Pengujian Unit Operasi `hitungMinat()`

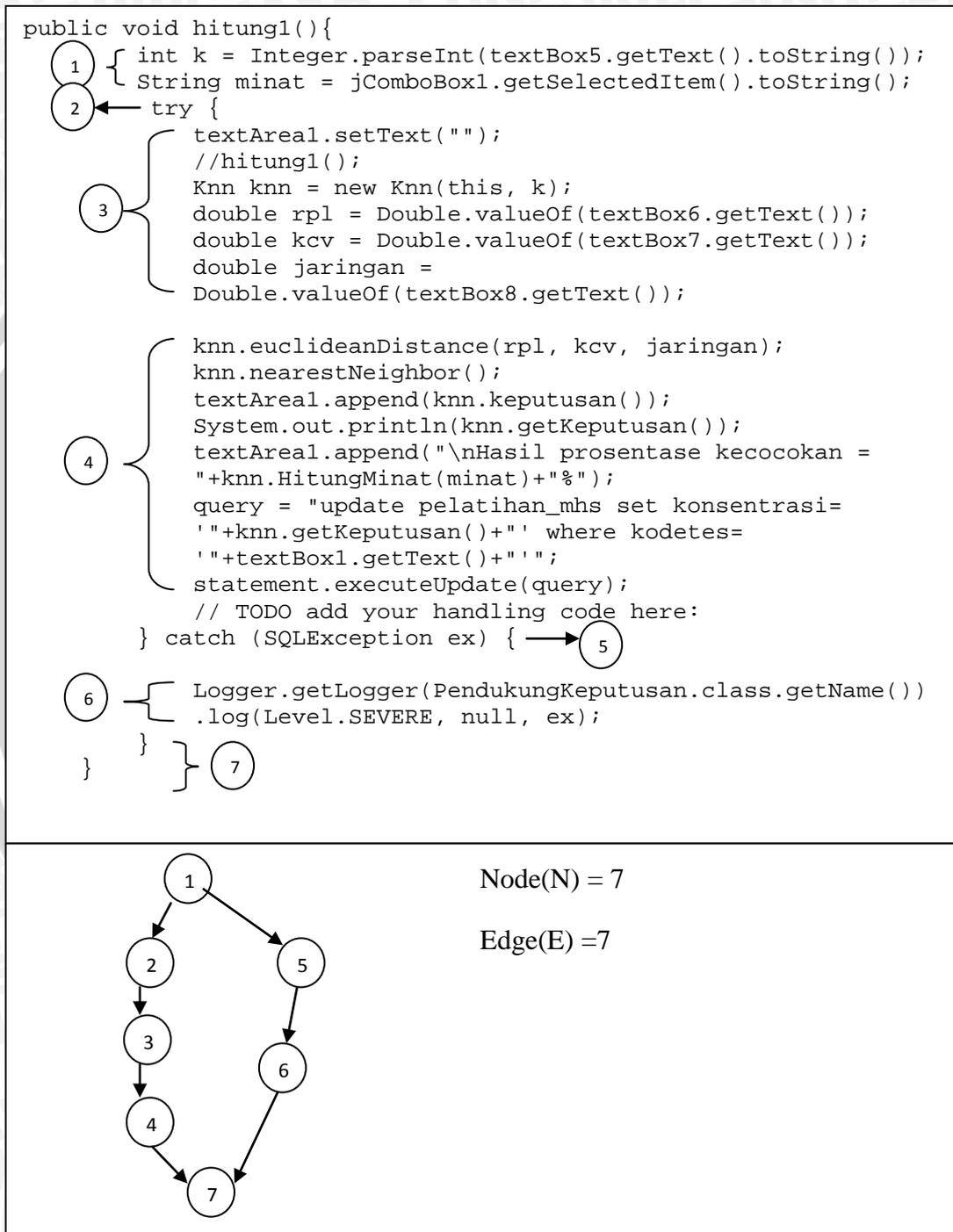
Jalur	Kasus Uji	Hasil yang Diharapkan	Hasil yang Didapatkan
1	Konsentrasi minat yang dimasukkan RPL	Sistem akan menampilkan hasil prosentase kecocokan	Sistem menampilkan hasil prosentase kecocokan
2	Konsentrasi minat yang dimasukkan KCV	Sistem akan menampilkan hasil prosentase kecocokan	Sistem menampilkan hasil prosentase kecocokan
3	Konsentrasi minat yang dimasukkan KBJ	Sistem akan menampilkan hasil prosentase kecocokan	Sistem menampilkan hasil prosentase kecocokan

Sumber: Pengujian

### 6.1.2 Pengujian Integrasi

Pengujian integrasi diterapkan pada proses yang mengintegrasikan fungsionalitas dari beberapa kelas untuk melakukan sebuah operasi tertentu. Pada pengujian integrasi yang dijadikan sebagai obyek uji adalah klas-klas yang menggabungkan kinerja dari kelas-kelas yang lain. Pengujian integrasi yang diterapkan dalam sistem ini menggunakan strategi *bottom-up*. Pengujian integrasi

pada sistem ini dilakukan pada operasi `hitung1()` yang ada pada kelas `PendukungKeputusan()`. Gambar 6.5 menunjukkan proses pemodelan dalam *flow graph* pada operasi `hitung1()`.



**Gambar 6.5** Pemodelan operasi `hitung1()` ke dalam *flow graph*  
**Sumber:** Pengujian



Pemodelan ke dalam *flow graph* yang telah dilakukan terhadap operasi `hitung1()` menghasilkan jumlah kompleksitas siklomatis (*cyclometric complexity*) melalui persamaan  $V(G) = E - N + 2$ , dimana  $V(G)$  merupakan jumlah kompleksitas siklomatis,  $E$  merupakan sisi garis (garis penghubung antar *node*), dan  $N$  merupakan jumlah simpul (*node*).

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 7 - 7 + 2 \\ &= 2 \end{aligned}$$

Dari nilai *cyclometric complexity* yang telah dihasilkan dari perhitungan yaitu ditentukan lima buah basis set dari jalur *independent* yaitu:

Jalur 1 : 1 – 2 – 3 – 4 – 7

Jalur 2 : 1 – 5 – 6 – 7

Penentuan kasus uji untuk masing-masing jalur dan hasil eksekusi untuk masing-masing kasus uji dijelaskan pada Tabel 6.5

**Tabel 6.5** Test Case untuk Pengujian Integrasi Operasi `hitung1()`

Jalur	Kasus Uji	Hasil yang Diharapkan	Hasil yang Didapatkan
1	Nilai k ditentukan dan ada nilai yang dihitung	Sistem akan menampilkan hasil perhitungan	Sistem menampilkan hasil perhitungan
2	Tidak melakukan koneksi ke basis data	Sistem akan menampilkan standar <code>SQLException</code>	Sistem akan menampilkan standar <code>SQLException</code>

Sumber: Pengujian

### 6.1.3 Pengujian Validasi

Pengujian validasi digunakan untuk mengetahui apakah sistem yang dibangun sudah benar sesuai dengan yang dibutuhkan. Item-item yang telah dirumuskan dalam daftar kebutuhan dan merupakan hasil analisis kebutuhan akan menjadi acuan untuk melakukan pengujian validasi. Pengujian validasi menggunakan metode pengujian *Black Box*, karena tidak memerlukan untuk berkonsentrasi

terhadap alur jalannya algoritma program dan lebih ditekankan untuk menemukan konformitas antara kinerja sistem dengan daftar kebutuhan.

### 6.1.3.1 Kasus Uji Validasi

#### a. Kasus Uji Melakukan *Log In* Sebagai Admin

Nama Kasus Uji : Kasus Uji Melakukan *Log In* Sebagai Admin

Objek Uji : Kebutuhan Fungsional *Log In* (SRS\_001\_01)

Tujuan Pengujian : Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional untuk melakukan *log in* sehingga sistem terjaga dari akses luar.

Prosedur Uji : Memasukkan *username* dan *password* admin.

Hasil yang Diharapkan: Aplikasi dapat menampilkan halaman admin.

#### b. Kasus Uji Melakukan *Log In* Sebagai Operator

Nama Kasus Uji : Kasus Uji Melakukan *Log In* Sebagai Operator

Objek Uji : Kebutuhan Fungsional *Log In* (SRS\_001\_01)

Tujuan Pengujian : Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional untuk melakukan *log in* sehingga sistem terjaga dari akses luar.

Prosedur Uji : Memasukkan *username* dan *password* operator.

Hasil yang Diharapkan: Aplikasi dapat menampilkan halaman operator.

**c. Kasus Uji Memasukkan Data Pelatihan**

- Nama Kasus Uji : Kasus Uji Memasukkan Data Pelatihan
- Objek Uji : Kebutuhan Fungsional Mengolah Data Pelatihan  
(SRS\_001\_05)
- Tujuan Pengujian : Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional untuk pengolahan data Pelatihan.
- Prosedur Uji : Memasukkan data Pelatihan, kemudian menekan tombol “Simpan”
- Hasil yang Diharapkan: Aplikasi dapat melakukan penyimpanan data pelatihan ke basis data.

**d. Kasus Uji Mengubah Data Pelatihan**

- Nama Kasus Uji : Kasus Uji Mengubah Data Pelatihan
- Objek Uji : Kebutuhan Fungsional Mengubah Data Pelatihan  
(SRS\_001\_05)
- Tujuan Pengujian : Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional untuk pengolahan data Pelatihan.
- Prosedur Uji : Mengubah data Pelatihan, kemudian menekan tombol “Simpan”
- Hasil yang Diharapkan: Aplikasi dapat melakukan perubahan dan

menyimpan data Pelatihan kembali ke basis data.

**e. Kasus Uji Mencari Data Pelatihan**

Nama Kasus Uji : Kasus Uji Mencari Data Pelatihan

Objek Uji : Kebutuhan Fungsional Mencari Data Pelatihan  
(SRS\_001\_05)

Tujuan Pengujian : Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional untuk pengolahan data Pelatihan.

Prosedur Uji : Memasukkan kode Pelatihan yang dicari

Hasil yang Diharapkan: Aplikasi dapat menampilkan Pelatihan yang dicari.

**f. Kasus Uji Menghapus Data Pelatihan**

Nama Kasus Uji : Kasus Uji Menghapus Data Pelatihan

Objek Uji : Kebutuhan Fungsional Menghapus Data Pelatihan  
(SRS\_001\_05)

Tujuan Pengujian : Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional untuk pengolahan data Pelatihan.

Prosedur Uji :Menghapus Pelatihan yang dipilih dengan menekan tombol “Hapus”

Hasil yang Diharapkan: Aplikasi dapat menghapus data pelatihan dari

basis data.

**g. Kasus Uji Melakukan Tes**

Nama Kasus Uji : Kasus Uji Melakukan Tes

Objek Uji : Kebutuhan Fungsional (SRS\_001\_07)

Tujuan Pengujian : Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional yaitu melakukan tes bagi mahasiswa.

Prosedur Uji : Memasukkan nilai mata kuliah sesuai dengan nama mata kuliah yang disediakan oleh sistem, kemudian menekan tombol “Proses”.

Hasil yang Diharapkan: Aplikasi dapat melakukan proses penyimpanan nilai yang dimasukkan.

**h. Kasus Uji Mendapatkan Hasil**

Nama Kasus Uji : Kasus Uji Mendapatkan Hasil

Objek Uji : Kebutuhan Fungsional (SRS\_001\_08)

Tujuan Pengujian : Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional yaitu mendapatkan hasil bagi mahasiswa.

Prosedur Uji : Memilih salah satu konsentrasi yang diminati, kemudian menekan tombol “Proses”.

Hasil yang Diharapkan: Aplikasi dapat melakukan proses perhitungan dan

menampilkan hasil analisis sistem..

### 6.1.3.2 Hasil Pengujian Validasi

Dari kasus uji yang telah dilaksanakan sesuai dengan prosedur pengujian pada sub pokok bahasan 6.1.3.1, didapatkan hasil seperti ditunjukkan pada Tabel 6.6.

**Tabel 6.6** *Test Case* untuk Pengujian Validasi

No.	Kasus Uji	Hasil yang Didapatkan	Status
1.	Melakukan Log In sebagai Admin	Aplikasi dapat menerima input dari user dan menampilkan halaman admin	Valid
2.	Melakukan Log In sebagai Operator	Aplikasi dapat menerima input dari user dan menampilkan halaman operator	Valid
3.	Memasukkan data Pelatihan	Aplikasi dapat menampilkan dan menyimpan data Pelatihan yang dimasukkan dalam basis data	Valid
4.	Pengubahan data Pelatihan	Aplikasi dapat menampilkan dan menyimpan data Pelatihan yang diubah dalam basis data	Valid
5.	Pencarian data Pelatihan	Aplikasi dapat menampilkan data Pelatihan yang dicari	Valid
6.	Penghapusan data Pelatihan	Aplikasi dapat menghapus data Pelatihan dari basis data	Valid
7.	Melakukan tes	Aplikasi dapat melakukan penyimpanan dan penghitungan nilai mata kuliah	Valid
8.	Mendapatkan Hasil	Aplikasi dapat melakukan perhitungan dan menampilkan hasil analisis	Valid

**Sumber:**Pengujian

## 6.2 Pengujian Validitas Sistem

Pengujian validitas sistem dilakukan untuk mengetahui seberapa besar keakuratan dan tingkat *error* pengklasifikasian yang dilakukan oleh sistem. Berikut ini 2 skenario pengujian yang akan digunakan:

1. Pengujian menggunakan data valid
2. Pengujian menggunakan *real data*

Masing-masing pengujian data dilakukan dengan menghitung besarnya *accuracy* dan *error-rate* pada nilai  $K=5$ ,  $K=9$ ,  $K=11$ , dan  $K=20$ .

### 6.2.1 Pengujian Menggunakan Data Valid

Pengujian dengan menggunakan data valid diambil dari *Machine Learning Database Repository*. Data yang digunakan yaitu *iris dataset* yang berjumlah 150, memiliki 4 atribut dan 3 *class*. Data *iris* tersebut dibagi menjadi dua secara acak, yaitu 90 data digunakan sebagai data pelatihan dan 60 data digunakan sebagai data tes. Pengujian validitas sistem menggunakan data *iris* pada  $K=5$  diperlihatkan pada Tabel 6.7.

**Tabel 6.7** Pengujian Validitas Sistem Menggunakan Data *Iris* pada  $K=5$

Data Ke-	Class	SPK	Keterangan
1.	Setosa	Setosa	T
2.	Setosa	Setosa	T
3.	Setosa	Setosa	T
4.	Setosa	Setosa	T
5.	Setosa	Setosa	T
6.	Setosa	Setosa	T
7.	Setosa	Setosa	T
8.	Setosa	Setosa	T
9.	Setosa	Setosa	T
10.	Setosa	Setosa	T
11.	Setosa	Setosa	T
12.	Setosa	Setosa	T
13.	Setosa	Setosa	T
14.	Setosa	Setosa	T
15.	Setosa	Setosa	T
16.	Setosa	Setosa	T
17.	Setosa	Setosa	T
18.	Setosa	Setosa	T
19.	Setosa	Setosa	T
20.	Setosa	Setosa	T
21.	Versicolor	Versicolor	T
22.	Versicolor	Versicolor	T
23.	Versicolor	Versicolor	T
24.	Versicolor	Virginica	F
25.	Versicolor	Versicolor	T
26.	Versicolor	Versicolor	T
27.	Versicolor	Versicolor	T
28.	Versicolor	Versicolor	T

29.	Versicolor	Versicolor	T
30.	Versicolor	Versicolor	T
31.	Versicolor	Versicolor	T
32.	Versicolor	Versicolor	T
33.	Versicolor	Versicolor	T
34.	Versicolor	Versicolor	T
35.	Versicolor	Versicolor	T
36.	Versicolor	Versicolor	T
37.	Versicolor	Versicolor	T
38.	Versicolor	Versicolor	T
39.	Versicolor	Versicolor	T
40.	Versicolor	Versicolor	T
41.	Virginica	Virginica	T
42.	Virginica	Virginica	T
43.	Virginica	Virginica	T
44.	Virginica	Virginica	T
45.	Virginica	Virginica	T
46.	Virginica	Virginica	T
47.	Virginica	Virginica	T
48.	Virginica	Virginica	T
49.	Virginica	Virginica	T
50.	Virginica	Virginica	T
51.	Virginica	Virginica	T
52.	Virginica	Virginica	T
53.	Virginica	Virginica	T
54.	Virginica	Virginica	T
55.	Virginica	Virginica	T
56.	Virginica	Virginica	T
57.	Virginica	Virginica	T
58.	Virginica	Virginica	T
59.	Virginica	Virginica	T
60.	Virginica	Virginica	T

**Sumber:**Pengujian

Keterangan:

T = *True*. Terjadi apabila hasil sistem sama dengan kenyataan

F = *False*. Terjadi apabila hasil sistem tidak sama dengan kenyataan

Berdasarkan pengujian pada nilai K=5, akan dilakukan perhitungan validitas sistem dengan menggunakan persamaan (2.2) dan (2.3):

$$Accuracy = \frac{59}{60} = 0,9833333 \quad (2.2)$$

$$error - rate = \frac{1}{60} = 0,0166667 \quad (2.3)$$

Pada nilai  $K=5$ , nilai *accuracy* yang diperoleh yaitu 0,9833333. Sedangkan nilai *error-rate* yang diperoleh yaitu 0,0166667.

Pengujian validitas sistem menggunakan data *iris* juga dilakukan pada nilai  $K=9$ , yang diperlihatkan pada Tabel 6.8.

**Tabel 6.8** Pengujian Validitas Sistem Menggunakan Data *Iris* pada  $K=9$

Data Ke-	Class	SPK	Keterangan
1.	Setosa	Setosa	T
2.	Setosa	Setosa	T
3.	Setosa	Setosa	T
4.	Setosa	Setosa	T
5.	Setosa	Setosa	T
6.	Setosa	Setosa	T
7.	Setosa	Setosa	T
8.	Setosa	Setosa	T
9.	Setosa	Setosa	T
10.	Setosa	Setosa	T
11.	Setosa	Setosa	T
12.	Setosa	Setosa	T
13.	Setosa	Setosa	T
14.	Setosa	Setosa	T
15.	Setosa	Setosa	T
16.	Setosa	Setosa	T
17.	Setosa	Setosa	T
18.	Setosa	Setosa	T
19.	Setosa	Setosa	T
20.	Setosa	Setosa	T
21.	Versicolor	Versicolor	T
22.	Versicolor	Versicolor	T
23.	Versicolor	Versicolor	T
24.	Versicolor	Virginica	F
25.	Versicolor	Versicolor	T
26.	Versicolor	Versicolor	T
27.	Versicolor	Versicolor	T
28.	Versicolor	Versicolor	T
29.	Versicolor	Versicolor	T

30.	Versicolor	Versicolor	T
31.	Versicolor	Versicolor	T
32.	Versicolor	Versicolor	T
33.	Versicolor	Versicolor	T
34.	Versicolor	Versicolor	T
35.	Versicolor	Versicolor	T
36.	Versicolor	Versicolor	T
37.	Versicolor	Versicolor	T
38.	Versicolor	Versicolor	T
39.	Versicolor	Versicolor	T
40.	Versicolor	Versicolor	T
41.	Virginica	Virginica	T
42.	Virginica	Virginica	T
43.	Virginica	Virginica	T
44.	Virginica	Virginica	T
45.	Virginica	Virginica	T
46.	Virginica	Virginica	T
47.	Virginica	Virginica	T
48.	Virginica	Virginica	T
49.	Virginica	Versicolor	F
50.	Virginica	Virginica	T
51.	Virginica	Virginica	T
52.	Virginica	Virginica	T
53.	Virginica	Virginica	T
54.	Virginica	Virginica	T
55.	Virginica	Virginica	T
56.	Virginica	Virginica	T
57.	Virginica	Virginica	T
58.	Virginica	Virginica	T
59.	Virginica	Virginica	T
60.	Virginica	Virginica	T

**Sumber:**Pengujian

Berdasarkan pengujian pada nilai K=9, akan dilakukan perhitungan validitas sistem dengan menggunakan persamaan (2.2) dan (2.3):

$$Accuracy = \frac{58}{60} = 0,9666667 \quad (2.2)$$

$$error - rate = \frac{2}{60} = 0,0333333 \quad (2.3)$$

Pada nilai  $K=9$ , nilai *accuracy* yang diperoleh yaitu 0,9666667. Sedangkan nilai *error-rate* yang diperoleh yaitu 0,0333333

Pengujian validitas sistem menggunakan data *iris* juga dilakukan pada nilai  $K=11$ , yang diperlihatkan pada Tabel 6.9.

**Tabel 6.9** Pengujian Validitas Sistem Menggunakan Data *Iris* pada  $K=11$

Data Ke-	Class	SPK	Keterangan
1.	Setosa	Setosa	T
2.	Setosa	Setosa	T
3.	Setosa	Setosa	T
4.	Setosa	Setosa	T
5.	Setosa	Setosa	T
6.	Setosa	Setosa	T
7.	Setosa	Setosa	T
8.	Setosa	Setosa	T
9.	Setosa	Setosa	T
10.	Setosa	Setosa	T
11.	Setosa	Setosa	T
12.	Setosa	Setosa	T
13.	Setosa	Setosa	T
14.	Setosa	Setosa	T
15.	Setosa	Setosa	T
16.	Setosa	Setosa	T
17.	Setosa	Setosa	T
18.	Setosa	Setosa	T
19.	Setosa	Setosa	T
20.	Setosa	Setosa	T
21.	Versicolor	Versicolor	T
22.	Versicolor	Versicolor	T
23.	Versicolor	Versicolor	T
24.	Versicolor	Virginica	F
25.	Versicolor	Versicolor	T
26.	Versicolor	Versicolor	T
27.	Versicolor	Versicolor	T
28.	Versicolor	Virginica	F
29.	Versicolor	Versicolor	T
30.	Versicolor	Versicolor	T
31.	Versicolor	Versicolor	T
32.	Versicolor	Versicolor	T
33.	Versicolor	Versicolor	T
34.	Versicolor	Virginica	F
35.	Versicolor	Versicolor	T

36.	Versicolor	Versicolor	T
37.	Versicolor	Versicolor	T
38.	Versicolor	Versicolor	T
39.	Versicolor	Versicolor	T
40.	Versicolor	Versicolor	T
41.	Virginica	Virginica	T
42.	Virginica	Virginica	T
43.	Virginica	Virginica	T
44.	Virginica	Versicolor	F
45.	Virginica	Versicolor	F
46.	Virginica	Virginica	T
47.	Virginica	Virginica	T
48.	Virginica	Versicolor	F
49.	Virginica	Versicolor	F
50.	Virginica	Versicolor	F
51.	Virginica	Virginica	T
52.	Virginica	Virginica	T
53.	Virginica	Virginica	T
54.	Virginica	Virginica	T
55.	Virginica	Virginica	T
56.	Virginica	Versicolor	F
57.	Virginica	Virginica	T
58.	Virginica	Versicolor	F
59.	Virginica	Virginica	T
60.	Virginica	Virginica	T

**Sumber:**Pengujian

Berdasarkan pengujian pada nilai  $K=11$ , akan dilakukan perhitungan validitas sistem dengan menggunakan persamaan (2.2) dan (2.3):

$$Accuracy = \frac{50}{60} = 0,8333333 \quad (2.2)$$

$$error - rate = \frac{10}{60} = 0,1666667 \quad (2.3)$$

Pada nilai  $K=11$ , nilai *accuracy* yang diperoleh yaitu 0,8333333. Sedangkan nilai *error-rate* yang diperoleh yaitu 0,01666667.

Pengujian validitas sistem menggunakan data *iris* juga dilakukan pada nilai  $K=20$ , yang diperlihatkan pada Tabel 6.10.

Tabel 6.10 Pengujian Validitas Sistem Menggunakan Data *Iris* pada K=20

Data Ke-	Class	SPK	Keterangan
1.	Setosa	Setosa	T
2.	Setosa	Setosa	T
3.	Setosa	Setosa	T
4.	Setosa	Setosa	T
5.	Setosa	Setosa	T
6.	Setosa	Setosa	T
7.	Setosa	Setosa	T
8.	Setosa	Setosa	T
9.	Setosa	Setosa	T
10.	Setosa	Setosa	T
11.	Setosa	Setosa	T
12.	Setosa	Setosa	T
13.	Setosa	Setosa	T
14.	Setosa	Setosa	T
15.	Setosa	Setosa	T
16.	Setosa	Setosa	T
17.	Setosa	Setosa	T
18.	Setosa	Setosa	T
19.	Setosa	Setosa	T
20.	Setosa	Setosa	T
21.	Versicolor	Virginica	F
22.	Versicolor	Virginica	F
23.	Versicolor	Versicolor	T
24.	Versicolor	Virginica	F
25.	Versicolor	Virginica	F
26.	Versicolor	Virginica	F
27.	Versicolor	Virginica	F
28.	Versicolor	Virginica	F
29.	Versicolor	Virginica	F
30.	Versicolor	Virginica	F
31.	Versicolor	Virginica	F
32.	Versicolor	Virginica	F
33.	Versicolor	Versicolor	T
34.	Versicolor	Virginica	F
35.	Versicolor	Virginica	F
36.	Versicolor	Versicolor	T
37.	Versicolor	Versicolor	T
38.	Versicolor	Virginica	F
39.	Versicolor	Setosa	F
40.	Versicolor	Versicolor	T
41.	Virginica	Versicolor	F
42.	Virginica	Versicolor	F

43.	Virginica	Versicolor	F
44.	Virginica	Virginica	T
45.	Virginica	Versicolor	F
46.	Virginica	Versicolor	F
47.	Virginica	Versicolor	F
48.	Virginica	Versicolor	F
49.	Virginica	Virginica	T
50.	Virginica	Versicolor	F
51.	Virginica	Versicolor	F
52.	Virginica	Versicolor	F
53.	Virginica	Versicolor	F
54.	Virginica	Versicolor	F
55.	Virginica	Versicolor	F
56.	Virginica	Versicolor	F
57.	Virginica	Versicolor	F
58.	Virginica	Versicolor	F
59.	Virginica	Versicolor	F
60.	Virginica	Versicolor	F

Sumber: Pengujian

Berdasarkan pengujian pada nilai K=20, akan dilakukan perhitungan validitas sistem dengan menggunakan persamaan (2.2) dan (2.3):

$$Accuracy = \frac{33}{60} = 0,55 \tag{2.2}$$

$$error - rate = \frac{27}{60} = 0,45 \tag{2.3}$$

Pada nilai K=20, nilai *accuracy* yang diperoleh yaitu 0,55. Sedangkan nilai *error-rate* yang diperoleh yaitu 0,45.

Hasil pengujian validitas sistem menggunakan data *iris* diperlihatkan pada Tabel 6.11.

Tabel 6.11 Hasil Pengujian Validitas Sistem Menggunakan Data *Iris*

Jumlah Tetangga(K)	Accuracy	Error-rate	Keterangan
K=5	0,9833333	0,0166667	Excellent Classification
K=9	0,9666667	0,0333333	Excellent Classification
K=11	0,8333333	0,1666667	Good Classification
K=20	0,55	0,45	Failure

Sumber: Pengujian

Berdasarkan perhitungan yang dilakukan menggunakan data *iris*, diperoleh nilai *accuracy* terbesar dan *error-rate* terkecil yaitu pada  $K=5$ . Sedangkan untuk  $K=9$ ,  $K=11$ , dan  $K=20$  nilai *accuracy* cenderung turun, dan nilai *error-rate* cenderung naik. Namun, untuk  $K=9$  dapat dikategorikan sebagai *excellent classification*, dan untuk  $K=11$  dapat dikategorikan sebagai *good classification*. Sedangkan untuk  $K=20$  nilai *accuracy* dan *error-rate* menurun drastis, sehingga dikategorikan menjadi *failure classification*. Jadi dapat disimpulkan bahwa semakin besar nilai  $K$  yang dipilih, maka akan terjadi kesalahan dalam pengklasifikasian.

### 6.2.2 Pengujian Menggunakan *Real Data*

Setelah dilakukan pengujian menggunakan data *iris*, pengujian juga dilakukan dengan menggunakan 10 data nilai matakuliah mahasiswa angkatan 2007-2008. Pengujian validitas sistem menggunakan data matakuliah pada  $K=5$  diperlihatkan pada Tabel 6.12.

**Tabel 6.12** Pengujian Validitas Sistem Menggunakan *Real Data* pada  $K=5$

Data Ke-	SPK	Konsentrasi	Keterangan
1.	RPL	RPL	T
2.	RPL	RPL	T
3.	RPL	RPL	T
4.	RPL	KBJ	F
5.	KBJ	KBJ	T
6.	KBJ	KBJ	T
7.	KCV	KCV	T
8.	KCV	KCV	T
9.	RPL	KCV	F
10.	KCV	KCV	T

**Sumber:** Pengujian

Keterangan:

T = *True*. Terjadi apabila hasil sistem sama dengan kenyataan

F = *False*. Terjadi apabila hasil sistem tidak sama dengan kenyataan

Berdasarkan pengujian pada nilai  $K=5$ , akan dilakukan perhitungan validitas sistem dengan menggunakan persamaan (2.2) dan (2.3):

$$Accuracy = \frac{8}{10} = 0,8 \quad (2.2)$$

$$error - rate = \frac{2}{10} = 0,2 \quad (2.3)$$

Pada nilai K=5, nilai *accuracy* yang diperoleh yaitu 0,8. Sedangkan nilai *error-rate* yang diperoleh yaitu 0,2.

Pengujian validitas sistem juga dilakukan pada nilai K=9, yang diperlihatkan pada Tabel 6.13.

**Tabel 6.13** Pengujian Validitas Sistem Menggunakan *Real Data* pada K=9

Data Ke-	SPK	Konsentrasi	Keterangan
1.	KBJ	RPL	F
2.	RPL	RPL	T
3.	RPL	RPL	T
4.	RPL	KBJ	F
5.	KBJ	KBJ	T
6.	KBJ	KBJ	T
7.	KCV	KCV	T
8.	KCV	KCV	T
9.	KCV	KCV	T
10.	KCV	KCV	T

**Sumber:** Pengujian

Berdasarkan pengujian pada nilai K=9, akan dilakukan perhitungan validitas sistem dengan menggunakan persamaan (2.2) dan (2.3):

$$Accuracy = \frac{8}{10} = 0,8 \quad (2.2)$$

$$error - rate = \frac{2}{10} = 0,2 \quad (2.3)$$

Pada nilai K=9, nilai *accuracy* yang diperoleh yaitu 0,8. Sedangkan nilai *error-rate* yang diperoleh yaitu 0,2.

Pengujian validitas sistem juga dilakukan pada nilai K=11, yang diperlihatkan pada Tabel 6.14.

**Tabel 6.14** Pengujian Validitas Sistem Menggunakan *Real Data* pada K=11

Data Ke-	SPK	Konsentrasi	Keterangan
1.	KBJ	RPL	F
2.	RPL	RPL	T
3.	RPL	RPL	T
4.	RPL	KBJ	F
5.	RPL	KBJ	F
6.	RPL	KBJ	F
7.	KCV	KCV	T
8.	KCV	KCV	T
9.	KCV	KCV	T
10.	KCV	KCV	T

**Sumber:** Pengujian

Berdasarkan pengujian pada nilai K=11, akan dilakukan perhitungan validitas sistem dengan menggunakan persamaan (2.2) dan (2.3):

$$Accuracy = \frac{6}{10} = 0,6 \quad (2.2)$$

$$error - rate = \frac{4}{10} = 0,4 \quad (2.3)$$

Pada nilai K=11, nilai *accuracy* yang diperoleh yaitu 0,6. Sedangkan nilai *error-rate* yang diperoleh yaitu 0,4.

Pengujian validitas sistem juga dilakukan pada nilai K=20, yang diperlihatkan pada Tabel 6.15.

**Tabel 6.15** Pengujian Validitas Sistem Menggunakan *Real Data* pada K=20

Data Ke-	SPK	Konsentrasi	Keterangan
1.	KCV	RPL	F
2.	RPL	RPL	T
3.	RPL	RPL	T
4.	RPL	KBJ	F
5.	RPL	KBJ	F
6.	RPL	KBJ	F
7.	KCV	KCV	T
8.	RPL	KCV	F
9.	KCV	KCV	T
10.	RPL	KCV	F

**Sumber:** Pengujian

Berdasarkan pengujian pada nilai  $K=20$ , akan dilakukan perhitungan validitas sistem dengan menggunakan persamaan (2.2) dan (2.3):

$$Accuracy = \frac{4}{10} = 0,4 \quad (2.2)$$

$$error - rate = \frac{6}{10} = 0,6 \quad (2.3)$$

Pada nilai  $K=20$ , nilai *accuracy* yang diperoleh yaitu 0,4. Sedangkan nilai *error-rate* yang diperoleh yaitu 0,6.

Hasil pengujian validitas sistem menggunakan *real data* diperlihatkan pada Tabel 6.16.

**Tabel 6.16** Hasil Pengujian Validitas Sistem Menggunakan *Real Data*

Jumlah Tetangga(K)	<i>Accuracy</i>	<i>Error-rate</i>	Keterangan
K=5	0,8	0,2	<i>Good Classification</i>
K=9	0,8	0,2	<i>Good Classification</i>
K=11	0,6	0,4	<i>Poor Classification</i>
K=20	0,4	0,6	<i>Failure</i>

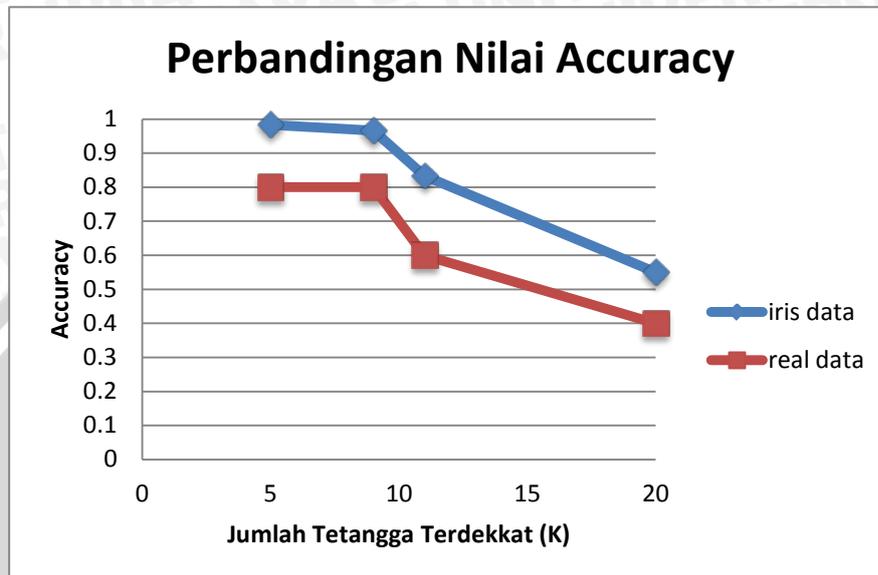
**Sumber:** Pengujian

Berdasarkan perhitungan yang dilakukan menggunakan *real data*, diperoleh nilai *accuracy* terbesar dan *error-rate* terkecil yaitu pada  $K=5$ . Sedangkan untuk  $K=9$ ,  $K=11$ , dan  $K=20$  nilai *accuracy* cenderung turun, dan nilai *error-rate* cenderung naik. Namun, untuk  $K=9$  masih dapat dikategorikan sebagai *good clasification*. Sedangkan untuk  $K=11$  dan  $K=20$  nilai *accuracy* dan *error-rate* menurun drastis, sehingga masing-masing dikategorikan menjadi *poor clasification* dan *failure classification*. Jadi dapat disimpulkan bahwa semakin besar nilai  $K$  yang dipilih, maka akan terjadi kesalahan dalam pengklasifikasian.

### 6.2.3 Perbandingan Hasil Pengujian Data Valid dengan *Real Data*

Setelah dilakukan pengujian menggunakan data *iris* dan *real data*, kemudian dilakukan analisis kevalidan sistem terhadap nilai *accuracy* dan *error-rate* dari data *iris* dan *real data*. Perbandingan nilai *accuracy* antara data *iris*

dengan *real data* diperlihatkan pada Gambar 6.6 dan Tabel 6.17. Sedangkan perbandingan nilai *error-rate* antara data *iris* dengan *real data* diperlihatkan pada Gambar 6.7 dan Tabel 6.18.



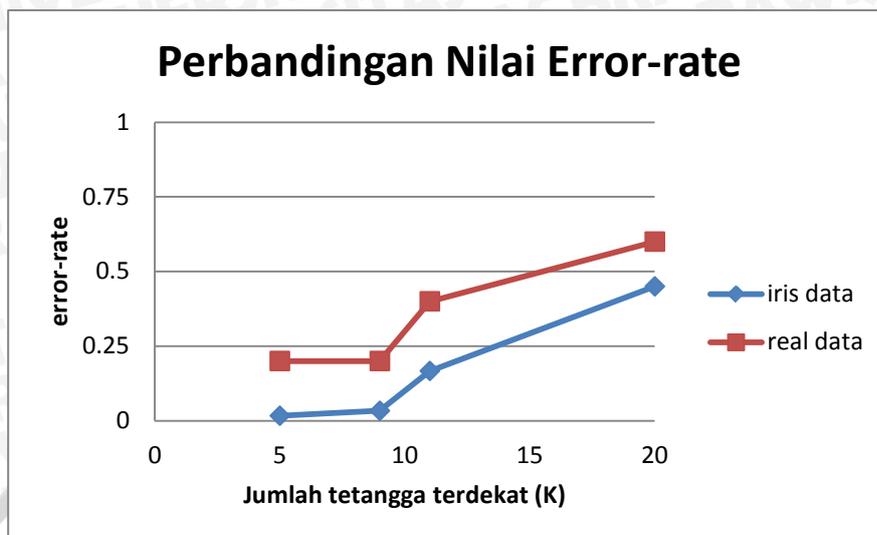
**Gambar 6.6** Grafik Perbandingan Nilai Accuracy antara Data Iris dengan Real Data

Sumber: Pengujian

**Tabel 6.17** Perbandingan Nilai Accuracy antara Data Iris dengan Real Data

Data	Jumlah Tetangga Terdekat (K)			
	K=5	K=9	K=11	K=20
<i>iris data</i>	0,9833333	0,9666667	0,8333333	0,55
<i>Keterangan</i>	<i>Excellent classification</i>	<i>Excellent classification</i>	<i>Good classification</i>	<i>Failure</i>
<i>real data</i>	0,8	0,8	0,6	0,4
<i>Keterangan</i>	<i>Good classification</i>	<i>Good classification</i>	<i>Poor classification</i>	<i>Failure</i>

Sumber: Pengujian



**Gambar 6.7** Grafik Perbandingan Nilai *Error-rate* antara Data *Iris* dengan *Real Data*

Sumber: Pengujian

**Tabel 6.18** Perbandingan Nilai *Error-rate* antara Data *Iris* dengan *Real Data*

Data	Jumlah Tetangga Terdekat (K)			
	K=5	K=9	K=11	K=20
<i>iris data</i>	0,0166667	0,0333333	0,1666667	0,45
<i>real data</i>	0,2	0,2	0,4	0,6

Sumber: Pengujian

Dari Tabel 6.17 diketahui kategori *excellent classification* diperoleh ketika sistem diuji menggunakan data *iris*. Sedangkan pengujian dengan menggunakan *real data*, kategori klasifikasi terbaik yang diperoleh yaitu *good classification*. Berdasarkan kategori *accuracy* yang dihasilkan pada pengujian menggunakan data *iris*, maka dapat dinyatakan bahwa sistem pendukung keputusan ini memiliki kinerja tinggi. Disisi lain, perlu dilakukan pengkajian ulang terhadap pemilihan atribut atau parameter yang digunakan dalam *real data*, karena kategori *accuracy* terbaik yang dihasilkan oleh *real data* hanya sampai pada *good classification* dan nilai *error-rate* yang dihasilkan jauh lebih besar dibandingkan nilai *error-rate* yang dihasilkan oleh data *iris*.