

BAB 7 PENGUJIAN

Bab ini membahas tahapan pengujian terhadap sistem yang telah dibangun pada penelitian ini. Pengujian sistem ini bertujuan untuk menguji bahwa setiap bagian dari sistem telah sesuai dengan rancangan dan persyaratan yang telah ditetapkan sebelumnya. Terdapat 2 jenis pengujian yang akan dilakukan yaitu pengujian *black-box* untuk menguji persyaratan fungsional sistem, serta pengujian komparabilitas sebagai bagian dari persyaratan non-fungsional sistem. Pada masing-masing jenis pengujian dijelaskan mengenai mekanisme pengujian, hasil pengujian, serta hasil analisis terhadap pengujian yang telah dilakukan.

7.1 Pengujian *White-Box*

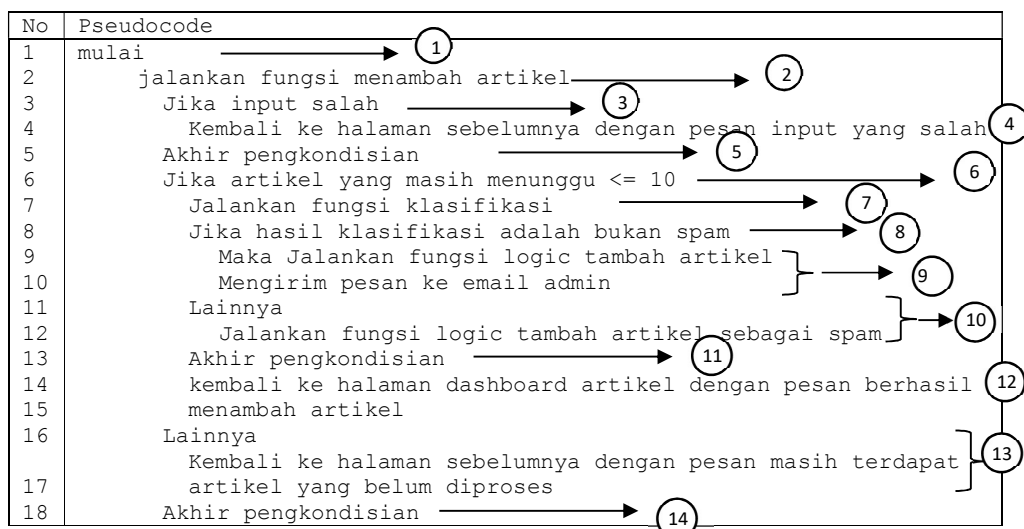
Pengujian *white-box* dilakukan dengan menggunakan pengujian *basis path testing*. Pengujian dilakukan dengan penggambaran *flow graph* yang berasal dari *pseudocode* atau *flowchart* dari algoritma yang diuji. Kemudian dilakukan penghitungan *cyclomatic complexity* dan jumlah *independent path* yang terdapat pada algoritma yang diuji. Setelah didapatkan jumlah *independent path*, pada pengujian ini dilakukan pemaparan *test case* sesuai dengan jumlah *independent path* yang diperoleh. Kemudian dilanjutkan dengan pengujian integrasi pada beberapa fungsionalitas

Dalam sub bab ini, hanya 3 kasus pengujian yang ditulis, yaitu menambah artikel, verifikasi laporan, verifikasi KIM.

7.1.1 Pengujian Basis Path Menambah Artikel

Pada Tabel 7.1 adalah *pseudocode* dari algoritma menambah artikel yang akan digunakan sebagai acuan untuk menentukan *cyclomatic complexity* dan *independent path*.

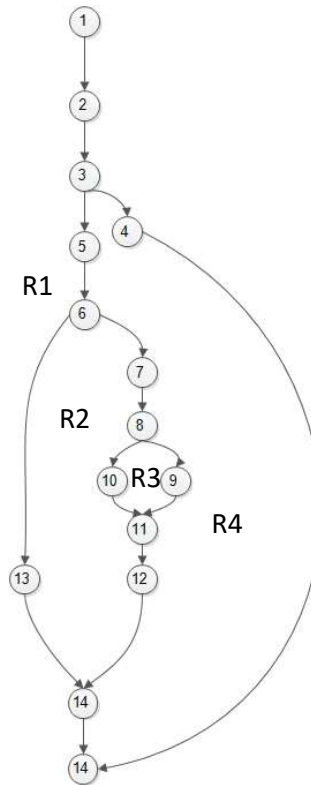
Tabel 7.1 Pseudocode menambah artikel



Tabel 7.1 Pseudocode menambah artikel (lanjutan)

No	Pseudocode
19	selesai → (15)

Flow Graph yang dihasilkan dari pseudocode yang terdapat dalam Tabel 7.1 dapat dilihat pada Gambar 7.1



Gambar 7.1 Flow graph menambah artikel

Dari flow graph pada Gambar 7.1 dapat dilakukan penghitungan *cyclometric complexity* dengan perhitungan sebagai berikut :

$$V(G) = \text{Jumlah edge} - \text{Jumlah Node} + 2 = 20 + 17 + 2 = 5$$

$$V(G) = \text{Jumlah predicate node} + 1 = 4 + 1 = 5$$

$$V(G) = \text{Jumlah Region} = 5$$

Dari *cyclometric complexity* yang diperoleh didapatkan jalur independent sejumlah 5. Jalur-jalur tersebut antara lain :

- 1) **1-2-3-4-15**
- 2) **1-2-3-5-6-13-14-15**
- 3) **1-2-3-5-6-7-8-9-11-12-14-15**
- 4) **1-2-3-5-6-7-8-10-11-12-14-15**

Berdasarkan 5 jalur tersebut didapatkan *test case* yang dapat dilihat pada Tabel 7.2.

Tabel 7.2 Test case menambah artikel

No	Input	Hasil yang diharapkan	Hasil yang diperoleh	Status
1	Input tidak lengkap/salah format	Menampilkan variable string dari input yang <i>error</i>	Menampilkan variable string dari input yang <i>error</i>	valid
2	Artikel status menunggu >10	Menampilkan variable status "over"	Menampilkan variable status "over"	valid
3	Input lengkap && artikel menunggu <= 10	Menampilkan variable status "continue"	Menampilkan variable status "continue"	valid
4	Hasil klasifikasi==spam	Menampilkan variable klasifikasi "spam"	Menampilkan variable klasifikasi "spam"	valid

7.1.2 Pengujian Basis Path Melihat Artikel dan Informasi KIM

Pada Tabel 7.2 adalah *pseudocode* dari algoritma melihat artikel dan informasi KIM artikel yang akan digunakan sebagai acuan untuk menentukan *cyclomatic complexity* dan *independent path*.

Flow Graph yang dihasilkan dari *pseudocode* yang terdapat dalam Tabel 7.3 dapat dilihat pada gambar 7.3

Dari flow graph pada gambar 7.2 dapat dilakukan penghitungan *cyclomatic complexity* dengan perhitungan sebagai berikut :

$$V(G) = \text{Jumlah edge} - \text{Jumlah Node} + 2 = 43 - 34 + 2 = 11$$

$$V(G) = \text{Jumlah predicate node} + 1 = 10 + 1 = 11$$

$$V(G) = \text{Jumlah Region} = 11$$

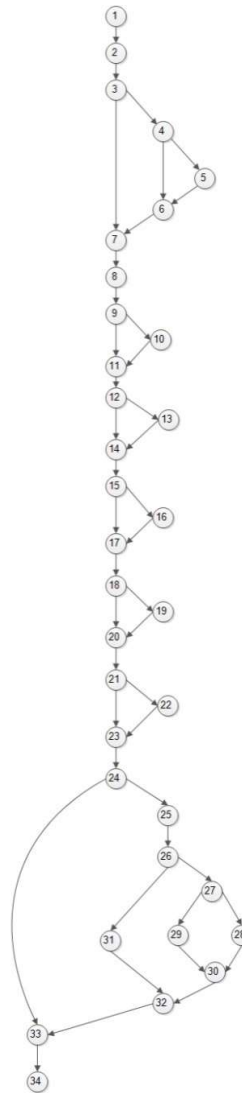
Dari *cyclomatic complexity* yang diperoleh didapatkan jalur independent sejumlah 5. Jalur-jalur tersebut antara lain :

- 1) 1-2-3-7-8-9-11-12-14-15-17-18-20-21-23-24-33-34
- 2) 1-2-3-4-6-7-8-9-11-12-14-15-17-18-20-21-23-24-33-34
- 3) 1-2-3-4-5-6-7-8-9-11-12-14-15-17-18-20-21-23-24-33-34
- 4) 1-2-3-7-8-9-10-11-12-14-15-17-18-20-21-23-24-33-34
- 5) 1-2-3-7-8-9-11-12-13-14-15-17-18-20-21-23-24-33-34
- 6) 1-2-3-7-8-9-11-12-15-16-17-18-20-21-23-24-33-34
- 7) 1-2-3-7-8-9-11-12-15-17-18-19-20-21-23-24-33-34

- 8) 1-2-3-4-5-6-7-8-9-11-12-14-15-17-18-20-21-22-23-24-33-34
- 9) 1-2-3-4-5-6-7-8-9-11-12-14-15-17-18-20-21-23-24-25-26-31-32-33-34
- 10) 1-2-3-4-5-6-7-8-9-11-12-14-15-17-18-20-21-23-24-25-26-27-29-30-32-33-34
- 11) 1-2-3-4-5-6-7-8-9-11-12-14-15-17-18-20-21-23-24-25-26-27-28-30-32-33-34

Tabel 7.3 Pseudocode melihat artikel dan informasi KIM

No	Pseudocode
1	mulai → ①
2	Jalankan fungsi getLatestArticle → ①
3	Jalankan fungsi getPopularArticle → ①
4	Jalankan fungsi cek getCategoryArticle → ②
5	Jalankan fungsi getKabupaten → ②
6	Jika pengguna memiliki otorisasi → ③
7	Jika otorisasi pengguna = kim → ④
8	Maka set photoprofile → ⑤
9	Akhiri pengkondisian → ⑥
10	Akhiri pengkondisian → ⑦
11	Inisialisasi data article → ⑧
12	Jika parameter kabupaten != "jatim" → ⑨
13	Maka tambah kondisi lokasi pada query database → ⑩
14	Akhiri pengkondisian → ⑪
15	Jika parameter category != "all" → ⑫
16	Maka tambah kondisi category pada query database → ⑬
17	Akhiri pengkondisian → ⑭
18	Jika parameter year tidak kosong → ⑮
19	Maka tambahkan kondisi tahun pada query database → ⑯
20	Akhiri pengkondisian → ⑰
21	Jika parameter month tidak kosong → ⑱
22	Maka tambahkan kondisi bulan pada query database → ⑲
23	Akhiri pengkondisian → ⑳
24	Jika category == "galeri kim" → ㉑
25	Maka tampilkan halaman list galeri KIM → ㉒
26	Akhiri pengkondisian → ㉓
27	Jika parameter id tidak kosong → ㉔
28	Maka jalankan fungsi mengambil data komentar terkait dari artikel dengan id sesuai parameter → ㉕
29	Jika status artikel != publish → ㉖
30	Maka Jika pengguna memiliki otorisasi admin atau pengguna adalah pemilik artikel terkait → ㉗
31	Maka tampilkan aritkel terkait dalam mode "preview" → ㉘
32	Lainnya redirect ke list artikel → ㉙
33	Akhiri pengkondisian → ㉚
34	Lainnya Tampilkan artikel terkait → ㉛
35	Akhiri pengkondisian → ㉜
36	Akhiri pengkondisian → ㉝
37	Selesai → ㉞



Gambar 7.2 Flow graph melihat artikel dan informasi KIM

Berdasarkan 11 jalur tersebut didapatkan *test case* yang dapat dilihat pada Tabel 7.4

Tabel 7.4 Test case melihat artikel dan informasi KIM

No	Input	Hasil yang diharapkan	Hasil yang diperoleh	Status
1	otorisasi pengguna = null	Mengembalikan avatar <i>default</i>	Mengembalikan avatar <i>default</i>	valid

Tabel 7.4 Test case melihat artikel dan informasi KIM

No	Input	Hasil yang diharapkan	Hasil yang diperoleh	Status
2	Otorisasi pengguna != kim	Mengembalikan avatar <i>default</i>	Mengembalikan avatar <i>default</i>	valid
3	Otorisasi pengguna == kim	Mengembalikan gambar avatar KIM terkait	Mengembalikan gambar avatar KIM terkait	valid
4	Parameter kabupaten != "jatim"	Mengembalikan daftar artikel berdasarkan lokasi dari input parameter kabupaten	Mengembalikan daftar artikel berdasarkan lokasi dari input parameter kabupaten	valid
5	Parameter category != "all"	Mengembalikan daftar artikel berdasarkan kategori dari input parameter category	Mengembalikan daftar artikel berdasarkan kategori dari input parameter category	valid
6	Parameter year != null	Mengembalikan daftar artikel berdasarkan tahun dari input parameter year	Mengembalikan daftar artikel berdasarkan tahun dari input parameter year	valid
7	Parameter month != null	Mengembalikan daftar artikel berdasarkan bulan dari input parameter month	Mengembalikan daftar artikel berdasarkan bulan dari input parameter month	valid
8	Parameter category == "galeri kim"	Mengembalikan daftar galeri KIM dengan tampilan galeri	Mengembalikan daftar galeri KIM dengan tampilan galeri	valid
9	Parameter id != null	Mengembalikan artikel dengan id terkait	Mengembalikan artikel dengan id terkait	valid
10	Status artikel != publish dan otorisasi sebagai	Mengembalikan artikel terkait dengan status <i>preview</i>	Mengembalikan artikel terkait dengan status <i>preview</i>	valid

Tabel 7.4 Test case melihat artikel dan informasi KIM (lanjutan)

No	Input	Hasil yang diharapkan	Hasil yang diperoleh	Status
	admin / pengguna adalah pemilik artikel terkait			
11	Status artikel != publish pengguna bukan pemilik artikel terkait	Mengembalikan daftar kosong	Mengembalikan daftar kosong	valid

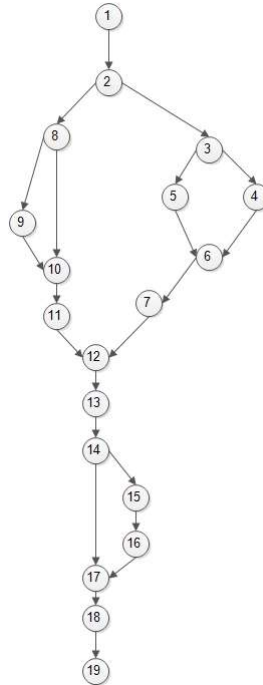
7.1.3 Pengujian Basis Path Menambah Komentar

Pada Tabel 7.2 adalah *pseudocode* dari algoritma menambah komentar yang akan digunakan sebagai acuan untuk menentukan *cyclomatic complexity* dan *independent path*.

Tabel 7.5 Pseudocode menambah komentar

No	Pseudocode
1	Mulai → ①
2	Jika pengguna memiliki otorisasi → ②
3	Maka Jika role otorisasi == admin → ③
4	Set nama = admin → ④
5	Lainnya → ⑤
6	Set nama = nama kim yang membuat komentar
7	Akhiri pengkondisian → ⑥
8	Set messages = komentar berhasil dibuat → ⑦
9	Lainya
10	Jika validasi input bernilai false → ⑧
11	Kembali ke tampilan sebelumnya dengan pesan input yang salah → ⑨
12	Akhiri pengkondisian → ⑩
13	Set messages = Komentar akan diverifikasi terlebih dahulu → ⑪
14	Akhiri pengkondisian → ⑫
15	Jalankan logic membuat komentar → ⑬
16	Jika pengguna tidak terotorisasi dan logic membuat komentar berhasil → ⑭
17	Set data notifikasi email → ⑮
18	Jalankan fungsi mengirim notifikasi komentar dari masyarakat → ⑯
19	Akhiri pengkondisian → ⑰
20	Kembali ke tampilan sebelumnya dengan pesan "sukses" → ⑱
21	selesai → ⑲

Flow Graph yang dihasilkan dari *pseudocode* yang terdapat dalam Tabel 7.5 dapat dilihat pada gambar 7.3



Gambar 7.3 Flow graph menambah komentar

Dari flow graph pada gambar 7.3 dapat dilakukan penghitungan *cyclometric complexity* dengan perhitungan sebagai berikut :

$$V(G) = \text{Jumlah edge} - \text{Jumlah Node} + 2 = 22 - 19 + 2 = 5$$

$$V(G) = \text{Jumlah predicate node} + 1 = 4 + 1 = 5$$

$$V(G) = \text{Jumlah Region} = 5$$

Dari *cyclometric complexity* yang diperoleh didapatkan jalur independent sejumlah 5. Jalur-jalur tersebut antara lain :

- 1) **1-2-8-10-11-12-13-14-17-18-19**
- 2) 1-2-8-9-10-11-12-13-14-17-18-19
- 3) 1-2-**3-5-6-7**-12-13-14-17-18-19
- 4) 1-2-3-**4**-6-7-12-13-14-17-18-19
- 5) 1-2-3-5-6-7-12-13-14-**15-16**-17-18-19

Berdasarkan 5 jalur tersebut didapatkan *test case* yang dapat dilihat pada Tabel 7.6

Tabel 7.6 Test case menambah komentar

No	Input	Hasil yang diharapkan	Hasil yang diperoleh	Status
1	otorisasi==null	Menyimpan komentar dan	Menyimpan komentar dan menampilkan	valid

Tabel 7.6 Test case menambah komentar (lanjutan)

		menampilkan pesan akan diverifikasi terlebih dahulu	pesan akan diverifikasi terlebih dahulu	
2	Form komentar untuk masyarakat tidak lengkap	Menampilkan pesan eror pada input yang salah	Menampilkan pesan eror pada input yang salah	valid
3	Role otorisasi ==kim	Komentar disimpan dengan nama sesuai nama KIM, menampilkan pesan komentar berhasil dibuat dan komentar ditampilkan pada artikel terkait	Komentar disimpan dengan nama sesuai nama KIM, menampilkan pesan komentar berhasil dibuat dan komentar ditampilkan pada artikel terkait	valid
4	Role otorisasi == admin	Komentar disimpan dengan nama admin, menampilkan pesan komentar berhasil dibuat dan komentar ditampilkan pada artikel terkait	Komentar disimpan dengan nama admin, menampilkan pesan komentar berhasil dibuat dan komentar ditampilkan pada artikel terkait	valid
5	Otorisasi == null && method membuat komentar berhasil	Mengirim pesan notifikasi ke <i>email</i> KOMINFO bahwa terdapat komentar yang perlu diverifikasi	Mengirim pesan notifikasi ke <i>email</i> KOMINFO bahwa terdapat komentar yang perlu diverifikasi	valid

7.1.4 Pengujian Integrasi Menambah Artikel

Tabel 7.7 Pengujian integrasi menambah artikel

Unit 1	Fungsi checkHasCreated() pada <i>class</i> ArticleRepository
Unit 2	Fungsi postCreateArticle() pada <i>class</i> ArticleController

Tabel 7.7 Pengujian integrasi menambah artikel (lanjutan)

Relation Unit	Dapat menentukan kelanjutan dari proses membuat artikel berdasarkan jumlah artikel yang sedang menunggu
Expected Result	Unit 1 mengembalikan value <10, sehingga fungsi postCreateArticle() melanjutkan proses pembuatan artikel.
Result	Dapat menentukan kelanjutan dari proses membuat artikel
Status	Valid

7.1.5 Pengujian Integrasi Melihat Artikel dan Informasi KIM

Tabel 7.8 Pengujian integrasi melihat artikel dan informasi KIM

Unit 1	Fungsi getCategoryArticle() pada class CategoryRepository
Unit 2	Fungsi getArticle() pada class ArticleController
Relation Unit	Dapat menentukan apakah category dari parameter getArticle memiliki sub kategori untuk digunakan sebagai variable kondisi dalam menentukan kembalian dari method getArticle ()
Expected Result	Unit 1 mengambil nilai "x" sehingga kembalian dari Unit 2 mengembalikan artikel yang parent kategorinya sama dengan parameter input pada unit 1
Result	Dapat menentukan kepemilikan sub kategori dari parameter category yang dimasukkan
Status	Valid

7.1.6 Pengujian Integrasi Menambah Komentar

Tabel 7.9 Pengujian integrasi menambah komentar

Unit 1	Fungsi create_comment() pada class CommentRepository
Unit 2	Fungsi postNewComment() pada class CommentsController

Tabel 7.9 Pengujian integrasi menambah komentar (lanjutan)

Relation Unit	Dapat menentukan pengiriman notifikasi menggunakan <i>email</i> bahwa komentar baru telah berhasil dibuat dari masyarakat
Expected Result	Unit 1 mengembalikan nilai true setelah berhasil melakukan operasi penambahan komentar, sehingga Unit 2 menjalankan fungsi pengiriman notifikasi
Result	Dapat menentukan pengiriman notifikasi menggunakan <i>email</i> bahwa komentar baru telah berhasil dibuat dari masyarakat
Status	Valid

7.2 Pengujian *Black-Box*

Pengujian *black-box* ini bertujuan untuk memastikan bahwa setiap persyaratan fungsional yang telah didefinisikan sebelumnya mampu dipenuhi oleh sistem yang dibangun. Setiap persyaratan fungsional yang telah didefinisikan pada bagian analisis persyaratan akan diuji pada pengujian ini. Pengujian akan dilakukan dengan mendefinisikan kasus uji (*test case*) yang didasarkan pada skenario untuk setiap *use case* yang telah didefinisikan pada bagian analisis persyaratan. Kemudian setiap hasil pengujian yang dilakukan akan dibandingkan dengan hasil yang telah didefinisikan sebelumnya pada skenario *use case* (hasil yang diharapkan).

7.2.1 Pengujian Validasi Menambah Artikel

Tabel 7.7, 7.8, 7.9 menjelaskan kasus uji dari pengujian validasi yang merujuk pada *use case* dengan kode UC-1 (menambah artikel).

Tabel 7.10 Kasus uji validasi menambah artikel

Nomor Kasus Uji	PV-001
Nama Kasus Uji	Kasus uji menambah artikel
Kode Persyaratan	P-KF-1
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat menambah artikel
Test Case	Pengujian dengan memasukan data sesuai format yang ditentukan pada <i>form</i> tambah artikel
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji memilih menu <i>dashboard</i> 2. Penguji memilih menu tambah artikel 3. Penguji mengisi <i>form</i> tambah artikel

Tabel 7.10 Kasus uji validasi menambah artikel (lanjutan)

	4. Penguji menekan tombol simpan
Hasil yang diharapkan	Artikel tersimpan di <i>database</i> dan dapat dilihat di <i>dashboard</i> KOMINFO pada menu artikel
Hasil Pengujian	Artikel tersimpan di <i>database</i> dan dapat dilihat di <i>dashboard</i> KOMINFO pada menu artikel
Status Validasi	Valid

Tabel 7.11 Kasus uji validasi menambah artikel:Exception 6a

Nomor Kasus Uji	PV-002
Nama Kasus Uji	Kasus uji menambah artikel exception 6a
Kode Persyaratan	P-KF-1
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem menampilkan pesan <i>error</i> ketika <i>form</i> belum lengkap
Test Case	Pengujian dengan mengosongkan salah <i>form</i> menambah artikel
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji memilih menu <i>dashboard</i> 2. Penguji memilih menu tambah artikel 3. Penguji hanya mengisi sebagian <i>form</i> tambah artikel 4. Penguji menekan tombol simpan
Hasil yang diharapkan	Sistem menampilkan pesan <i>error</i> pada <i>form</i> yang belum diisi.
Hasil Pengujian	Sistem menampilkan pesan <i>error</i> pada <i>form</i> yang belum diisi.
Status Validasi	valid

Tabel 7.12 Kasus uji validasi menambah artikel: exception 6b

Nomor Kasus Uji	PV-003
Nama Kasus Uji	Kasus uji menambah artikel exception 6b
Kode Persyaratan	P-KF-1
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat memfilter artikel yang mengandung <i>spam</i>
Test Case	Pengujian dengan mengisi konten artikel dengan konten yang mengandung <i>spam</i>
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji memilih menu <i>dashboard</i> 2. Penguji memilih menu tambah artikel

Tabel 7.12 Kasus uji validasi menambah artikel: exception 6b (lanjutan)

	<ol style="list-style-type: none"> 3. Penguji hanya mengisi konten artikel yang mengandung <i>spam</i> 4. Penguji menekan tombol simpan
Hasil yang diharapkan	Artikel dikenali sebagai <i>spam</i> dan artikel terkait disimpan sebagai <i>spam</i> di <i>database</i> dan dapat dilihat di <i>dashboard</i> KOMINFO pada menu <i>spam</i>
Hasil Pengujian	Artikel dikenali sebagai <i>spam</i> dan artikel terkait disimpan sebagai <i>spam</i> di <i>database</i> dan dapat dilihat di <i>dashboard</i> KOMINFO pada menu <i>spam</i>
Status Validasi	valid

7.2.2 Pengujian Validasi Mengelola Artikel

Tabel 7.10, 7.11, 7.12, 7.13, 7.14 menjelaskan kasus uji dari pengujian validasi yang merujuk pada *use case* dengan kode UC-2 (menambah artikel).

Tabel 7.13 Kasus uji validasi mengelola artikel *variant 2a*: fungsi *filter*

Nomor Kasus Uji	PV-004
Nama Kasus Uji	Kasus uji mengelola artikel <i>variant 2a</i> : fungsi <i>filter</i> artikel
Kode Persyaratan	P-KF-2
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat memfilter artikel sesuai <i>filter</i> yang dipilih
Test Case	Pengujian dengan memilih <i>filter</i> artikel
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji memilih menu <i>dashboard</i> 2. Penguji memilih salah satu <i>filter</i> artikel
Hasil yang diharapkan	Artikel yang ditampilkan di daftar sesuai <i>filter</i> yang dipilih
Hasil Pengujian	Artikel yang ditampilkan di daftar sesuai <i>filter</i> yang dipilih
Status Validasi	valid

Tabel 7.14 Kasus uji validasi mengelola artikel *variant 2b*: fungsi *search*

Nomor Kasus Uji	PV-005
Nama Kasus Uji	Kasus uji mengelola artikel <i>variant 2b</i> : fungsi <i>search</i> artikel
Kode Persyaratan	P-KF-2

**Tabel 7.14 Kasus uji validasi mengelola artikel
variant 2b: fungsi *search*(lanjutan)**

Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat mencari suatu artikel
Test Case	Pengujian dengan memasukkan kata kunci pada <i>form search</i>
Prosedur Uji	1. Penguji memilih menu <i>dashboard</i> 2. Penguji memasukkan kata kunci ke <i>form search</i>
Hasil yang diharapkan	Artikel yang ditampilkan di daftar sesuai kata kunci yang dimasukkan
Hasil Pengujian	Artikel yang ditampilkan di daftar sesuai kata kunci yang dimasukkan
Status Validasi	valid

Tabel 7.15 Kasus uji validasi mengelola artikel variant 3a: mengubah artikel

Nomor Kasus Uji	PV-006
Nama Kasus Uji	Kasus uji mengelola artikel <i>variant 3a</i> : mengubah artikel
Kode Persyaratan	P-KF-2.1
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat mengubah data artikel
Test Case	Pengujian dengan merubah data dengan memasukan data yang baru sesuai format yang ditentukan pada <i>form</i> tambah artikel
Prosedur Uji	1. Penguji memilih menu <i>dashboard</i> 2. Penguji memilih menu <i>edit</i> artikel dari salah satu artikel 3. Penguji mengisi <i>form</i> dengan data yang baru 4. Penguji menekan tombol simpan
Hasil yang diharapkan	Data artikel yang baru menggantikan yang lama
Hasil Pengujian	Data artikel yang baru menggantikan yang lama
Status Validasi	Valid

Tabel 7.16 Kasus uji validasi mengelola artikel *variant 3b*: menghapus artikel

Nomor Kasus Uji	PV-007
Nama Kasus Uji	Kasus uji mengelola artikel <i>variant 3b</i> : menghapus artikel
Kode Persyaratan	P-KF-2.2
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat menghapus artikel
Test Case	Pengujian dengan memilih salah artikel untuk dihapus
Prosedur Uji	<ol style="list-style-type: none">1. Penguji memilih menu <i>dashboard</i>2. Penguji memilih menu hapus artikel dari salah satu artikel3. Penguji mengkonfirmasi operasi hapus
Hasil yang diharapkan	Artikel terkait terhapus
Hasil Pengujian	Artikel terkait terhapus
Status Validasi	Valid

Tabel 7.17 Kasus uji validasi mengelola artikel *variant 3c*: verifikasi artikel

Nomor Kasus Uji	PV-008
Nama Kasus Uji	Kasus uji mengelola artikel <i>variant 3c</i> : verifikasi artikel
Kode Persyaratan	P-KF-2.3
Tujuan Pengujian	Pengujian dilakukan untuk memastikan artikel yang telah diverifikasi terpublikasi
Test Case	Pengujian dengan memilih salah artikel untuk diverifikasi
Prosedur Uji	<ol style="list-style-type: none">1. Penguji memilih menu <i>dashboard</i>2. Penguji memilih menu verifikasi artikel dari salah satu artikel3. Penguji mengkonfirmasi operasi verifikasi
Hasil yang diharapkan	Artikel terkait terpublikasi
Hasil Pengujian	Artikel terkait terpublikasi
Status Validasi	Valid

7.2.3 Pengujian Validasi Mengelola *Spam Variant 1a*: Menambah *Data Train*

Tabel 7.18 menjelaskan kasus uji dari pengujian validasi yang merujuk pada *use case* dengan kode UC-6 (mengelola *spam*).

Tabel 7.18 Kasus uji validasi mengelola *spam variant 1a*: menambah *data train*

Nomor Kasus Uji	PV-009
Nama Kasus Uji	Kasus uji mengelola <i>spam variant 1a</i> : menambah <i>data train</i>
Kode Persyaratan	P-KF -6.1
Tujuan Pengujian	Pengujian dilakukan untuk penambahan <i>data train</i> menambah pengetahuan dari klasifikasi <i>naïve bayes</i>
Test Case	Pengujian dengan menambah data train pada kategori <i>spam</i>
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji memilih menu <i>spam</i> 2. Penguji memilih kategori <i>spam</i> 3. Penguji mengisi <i>form</i> penambahan <i>data train</i> 4. Penguji memilih simpan <i>data train</i> 5. Penguji menambah artikel dengan konten yang memiliki makna sama dengan konten <i>spam</i> yang sudah di tambahkan sebelumnya
Hasil yang diharapkan	Artikel yang mengandung <i>spam</i> akan dimasukkan ke dalam <i>spam</i>
Hasil Pengujian	Artikel yang mengandung <i>spam</i> akan dimasukkan ke dalam <i>spam</i>
Status Validasi	Valid

7.3 Pengujian Akurasi

Pengujian akurasi pada fungsi klasifikasi menggunakan perhitungan *precision recall*. Pada Tabel 7.19 dapat dilihat hasil klasifikasi dari beberapa pengujian.

Total berita yang diuji = 15

Tabel 7.19 Coincidence matrix

		Actual Classification of Classes in the Dataset		
		Spam	Bukan Spam	
Model Classification	Spam	10	0	
	Bukan Spam	1	4	
Sum		11	4	
Probability		0.73	0.27	
Accuracy		1	0.8	0.93

Dari hasil klasifikasi yang terdapat pada Tabel 7.19 didapatkan TPR 90%, TNR 100%, precision 100%, recall 90%, dan akurasi 93%.

7.4 Pengujian *Compability*

Pengujian kompabilitas dilakukan dengan menggunakan SortSite versi 5.22.1766. Dari pengujian ini dihasilkan versi *browser* apa saja yang mendukung berjalannya sistem informasi KIM dengan baik. Pada Tabel 7.20 dapat dilihat *browser* beserta versinya yang digunakan dalam uji kompabilitas.

Tabel 7.20 *Browser* yang digunakan pada *compatibility testing*

No	Nama <i>Browser</i>	Versi <i>Browser</i>
1	Microsoft Edge	14
2	Firefox	51
3	Safari	<= 9.0, dan 10
4	Opera	41
5	Chrome	55
6	iOS	<= 8.0 , 9, 10
7	Android	<= 3.0 dan 4.0
8	Internet Explorer	8, 9, 10, 11

Browser Version	Internet Explorer	Edge	Firefox	Safari	Opera	Chrome	iOS	Android
	8 9 10 11	14	51	≤ 9 10	41	55	≤ 8 9 10	≤ 3 4*
Critical Issues	✓ ✓ ✓ ✓	✓	✓	✓ ✓	✓	✓	✓ ✓ ✓	✓ ✓
Major Issues	⚠ ✓ ✓ ✓	✓	✓	✓ ✓	✓	✓	✓ ✓ ✓	✓ ✓
Minor Issues	⚠ ⚠ ⚠ ⚠	✓	⚠	⚠ ⚠	✓	⚠	✓ ✓ ✓	✓ ✓

Gambar 7.4 Hasil *compatibility testing*

Terdapat 3 kategori masalah yang dapat diketahui dari hasil pengujian kompabilitas menggunakan SortSite, yakni *critical Issues*, *major issues*, dan *minor issues*. Penejelasan dari ketiga kategori masalah dapat dilihat pada Gambar 7.5.

✓	Tidak ada kendala pada browser
●	Critical Issues : Terdapat konten atau fungsi yang hilang pada beberapa browser
⚠	Major Issues : Terdapat masalah utama pada konten atau performa pada beberapa browser
⚠	Minor Issues : Terdapat masalah minor pada konten atau performa pada beberapa browser

Gambar 7.5 Kategori *compatibility issues* pada SortSite

Sumber : Diadaptasi Dari Power Mapper (2016)

Dari hasil pengujian kompabilitas pada Gambar 7.4 didapatkan bahwa *website* KIM yang dibuat dapat berjalan hampir pada semua *browser* yang di uji kecuali di internet explorer versi 8 yang didapatkan masalah utama.

7.5 *User Acceptance Testing (UAT)*

Pada pengujian UAT digunakan Tabel 7.21 untuk digunakan sebagai acuan pengujiannya

Tabel 7.21 UAT

No	Kebutuhan Penerimaan	Hasil	Komentar
1	proses penambahan artikel dalam sistem berjalan sesuai dengan kebutuhan		
2	Informasi dari pengolahan data sesuai dengan kebutuhan		
3	Informasi untuk dicetak sesuai dengan kebutuhan		
4	Kinerja KIM dapat dipantau		
5	<i>Role</i> dalam sistem berjalan sesuai kebutuhan		