

BAB V PENGUJIAN

5.1. Pengujian

Pengujian sistem dilakukan dengan metode *white-box* dan *black-box*. Untuk pengujian *white-box* akan digunakan metode pengujian unit, sedangkan untuk pengujian *black-box* akan digunakan metode pengujian validasi. Pengujian tersebut dilakukan berdasarkan *use case* yang telah dirancang pada bab sebelumnya. Selain itu, dilakukan juga pengujian non-fungsional untuk menjamin kesesuaian sistem pada perangkat Android 4.0.3 ke atas dan menjamin sistem agar dapat diakses oleh pengguna yang telah terdaftar dan *login* terlebih dahulu.

5.1.1. Pengujian Unit

Pengujian unit dilakukan dengan metode *basis path testing*, dimana proses pengujian dilakukan dengan memodelkan algoritma pada suatu *flow graph*, menentukan jumlah *cyclomatic complexity*, menentukan jalur independen, dan menentukan kasus uji berdasarkan jalur independen yang telah ditentukan.

5.1.1.1. Pengujian Unit *Login*

Berikut ini merupakan algoritma untuk melakukan proses *login* yang akan dijelaskan pada Gambar 5.1.

| | |
|---|---|
| Nama Algoritma: <i>login</i> | |
| Deskripsi | |
| - Masukan: <i>email</i> dan <i>password</i> | |
| - Proses | |
| 1 | Melakukan pengecekan <i>email</i> dan <i>password</i> pada basis data. |
| 2 | Cek apakah posisi sebelumnya adalah halaman catatan. |
| 3 | Menampilkan halaman catatan |
| 4 | Cek apakah posisi sebelumnya adalah halaman profil. |
| 5 | Cek apakah id kerja sama dengan 0, jika iya maka menampilkan halaman pekerjaan. |
| 6 | Jika id kerja tidak sama dengan 0, maka menampilkan halaman profil. |
| 7 | endif |

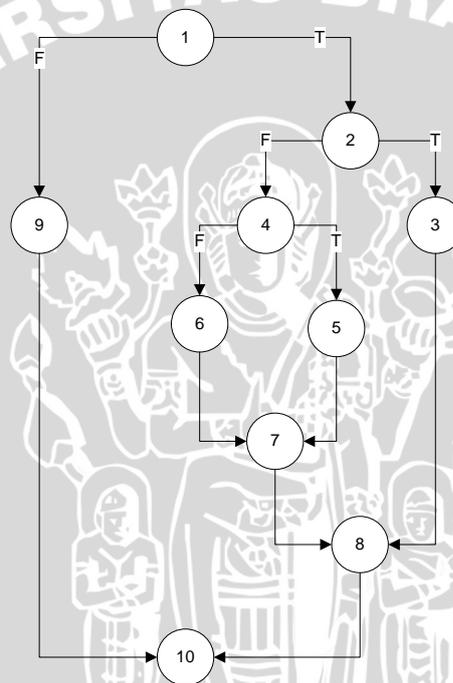


| | |
|----|--|
| 8 | endif |
| 9 | Jika <i>email</i> dan <i>password</i> tidak ada, maka akan menampilkan pesan kesalahan |
| 10 | endif |

- Keluaran: berhasil *login*

Gambar 5.1. Pengujian unit *login*

Berikut ini merupakan *flow graph* dari algoritma diatas yang akan dijelaskan pada Gambar 5.2.



Gambar 5.2. Flow graph *login*

Berdasarkan *flow graph* diatas akan diperoleh jumlah kompleksitas siklomatis (*cyclomatic complexity*) melalui persamaan $V = E - N + 2$, dimana V adalah jumlah kompleksitas siklomatis, E merupakan sisi atau *edge* (garis penghubung antar *node*) dan N merupakan jumlah simpul (*node*). Berikut ini merupakan perhitungan dari jumlah kompleksitas siklomatis:

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

Berdasarkan jumlah kompleksitas siklomatis yang telah didapatkan, maka akan ditentukan 2 jalur independen, yaitu:

Jalur 1: 1 – 2 – 3 – 8 – 10

Jalur 2: 1 – 2 – 4 – 5 – 7 – 8 – 10

Jalur 3: 1 – 2 – 4 – 6 – 7 – 8 – 10

Jalur 4: 1 – 9 – 10

Berdasarkan jalur independen yang telah ditentukan, maka dapat diperoleh kasus uji yang akan dijelaskan pada Tabel 5.1.

Tabel 5.1. Kasus uji *login*

| Jalur | Kasus Uji | Hasil Yang Diharapkan | Hasil Yang Didapatkan |
|-------|---|---|---|
| 1 | Pengguna memberikan masukan <i>email</i> dan/atau <i>password</i> yang ada pada basis data dan pengguna berada pada halaman catatan sebelum melakukan login | Sistem akan menampilkan halaman catatan | Sistem akan menampilkan halaman catatan |
| 2 | Pengguna memberikan masukan <i>email</i> dan/atau <i>password</i> yang ada pada basis data dan pengguna berada pada halaman profil dan belum memiliki pekerjaan sebelum melakukan login | Sistem akan menampilkan halaman pekerjaan | Sistem akan menampilkan halaman pekerjaan |
| 3 | Pengguna memberikan masukan <i>email</i> dan/atau <i>password</i> yang ada pada basis data dan pengguna berada pada halaman profil dan sudah memiliki pekerjaan sebelum melakukan login | Sistem akan menampilkan halaman profil | Sistem akan menampilkan halaman profil |
| 4 | Pengguna tidak memberikan masukan <i>email</i> dan/atau <i>password</i> atau memberikan masukan <i>email</i> dan/atau <i>password</i> yang tidak ada pada basis data | Pengguna tidak berhasil login | Pengguna tidak berhasil login |

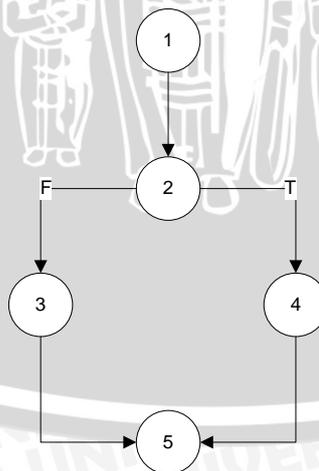
5.1.1.2. Pengujian Unit Register

Berikut ini merupakan algoritma untuk melakukan proses registrasi yang akan dijelaskan pada Gambar 5.3.

| | |
|---|--|
| Nama Algoritma: register | |
| Deskripsi | |
| - Masukan: nama, <i>email</i> , dan <i>password</i> | |
| - Proses | |
| 1 | Mengisi kolom <i>name</i> , <i>email</i> , dan <i>password</i> dengan nama, <i>email</i> , dan <i>password</i> yang ingin didaftarkan pengguna |
| 2 | Melakukan pengecekan apakah pengguna dengan nama, <i>email</i> , dan <i>password</i> ada pada basis data |
| 3 | Jika pengguna tersebut ada, maka akan menampilkan pesan kesalahan |
| 4 | Jika pengguna tersebut tidak ada, maka akan mendaftarkan akun pengguna pada basis data. |
| - Keluaran: halaman login | |

Gambar 5.3. Pengujian unit register

Berikut ini merupakan *flow graph* dari algoritma diatas yang akan dijelaskan pada Gambar 5.4.



Gambar 5.4. Flow graph register

Berdasarkan *flow graph* diatas akan diperoleh jumlah kompleksitas siklomatis (*cyclomatic complexity*) melalui persamaan $V = E - N + 2$, dimana V adalah jumlah kompleksitas siklomatis, E merupakan sisi atau *edge* (garis penghubung antar *node*) dan N merupakan jumlah simpul (*node*). Berikut ini merupakan perhitungan dari jumlah kompleksitas siklomatis:

$$\begin{aligned} V &= E - N + 2 \\ &= 5 - 5 + 2 \\ &= 2 \end{aligned}$$

Berdasarkan jumlah kompleksitas siklomatis yang telah didapatkan, maka akan ditentukan 2 jalur independen, yaitu:

Jalur 1: 1 – 2 – 3 – 5

Jalur 2: 1 – 2 – 4 – 5

Berdasarkan jalur independen yang telah ditentukan, maka dapat diperoleh kasus uji yang akan dijelaskan pada Tabel 5.2.

Tabel 5.2. Kasus uji *register*

| Jalur | Kasus Uji | Hasil Yang Diharapkan | Hasil Yang Didapatkan |
|-------|--|--|--|
| 1 | Pengguna tidak memberikan masukan nama, <i>email</i> , dan/atau <i>password</i> atau memberikan masukan nama, <i>email</i> , dan/atau <i>password</i> yang ada pada basis data | Sistem akan menampilkan pesan kesalahan | Sistem akan menampilkan pesan kesalahan |
| 2 | Pengguna memberikan masukan nama, <i>email</i> , dan/atau <i>password</i> yang tidak ada pada basis data | Sistem akan menampilkan halaman <i>login</i> | Sistem akan menampilkan halaman <i>login</i> |

5.1.1.3. Pengujian Unit Lihat Catatan

Berikut ini merupakan algoritma untuk melihat daftar catatan pengguna yang akan dijelaskan pada Gambar 5.5.

Nama Algoritma: lihat catatan

Deskripsi

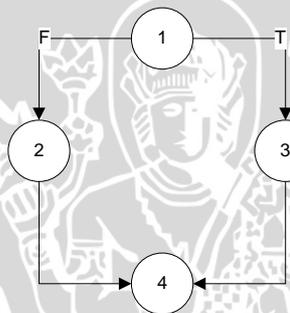
- Masukan: id pengguna dan id tempat bekerja
- Proses

| | |
|---|---|
| 1 | Melakukan pengecekan apakah catatan dengan id pengguna atau id tempat bekerja ada pada basis data |
| 2 | Jika catatan tersebut tidak ada, maka akan menampilkan daftar kosong |
| 3 | Jika catatan tersebut ada, maka akan menampilkan daftar catatan |

- Keluaran: daftar catatan pengguna

Gambar 5.5. Pengujian unit lihat catatan

Berikut ini merupakan *flow graph* dari algoritma diatas yang akan dijelaskan pada Gambar 5.6.

**Gambar 5.6.** *Flow graph* lihat catatan

Berdasarkan *flow graph* diatas akan diperoleh jumlah kompleksitas siklomatis (*cyclomatic complexity*) melalui persamaan $V = E - N + 2$, dimana V adalah jumlah kompleksitas siklomatis, E merupakan sisi atau *edge* (garis penghubung antar *node*) dan N merupakan jumlah simpul (*node*). Berikut ini merupakan perhitungan dari jumlah kompleksitas siklomatis:

$$\begin{aligned}
 V &= E - N + 2 \\
 &= 4 - 4 + 2 \\
 &= 2
 \end{aligned}$$

Berdasarkan jumlah kompleksitas siklomatis yang telah didapatkan, maka akan ditentukan 2 jalur independen, yaitu:

$$\text{Jalur 1: } 1 - 2 - 4$$

$$\text{Jalur 2: } 1 - 3 - 4$$

Berdasarkan jalur independen yang telah ditentukan, maka dapat diperoleh kasus uji yang akan dijelaskan pada Tabel 5.3.

Tabel 5.3. Kasus uji lihat catatan

| Jalur | Kasus Uji | Hasil Yang Diharapkan | Hasil Yang Didapatkan |
|-------|---|--|--|
| 1 | Pengguna tidak memiliki catatan | Sistem akan menampilkan daftar kosong | Sistem akan menampilkan daftar kosong |
| 2 | Pengguna memiliki catatan, baik catatan yang dibuat oleh pengguna sendiri maupun pengguna lain, tetapi memiliki tempat bekerja sama | Sistem akan menampilkan daftar catatan | Sistem akan menampilkan daftar catatan |

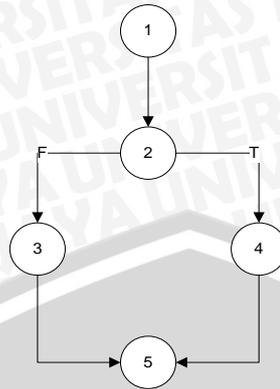
5.1.1.4. Pengujian Unit Tambah Catatan

Berikut ini merupakan algoritma untuk melakukan penambahan catatan yang akan dijelaskan pada Gambar 5.7.

| | |
|---|---|
| Nama Algoritma: tambah catatan | |
| Deskripsi | |
| - Masukan: judul, isi, dan status pembagian catatan | |
| - Proses | |
| 1 | Mengisi kolom judul dan isi catatan, serta mencentang atau tidak status pembagian catatan |
| 2 | Menambahkan data catatan pada basis data |
| 3 | Jika catatan gagal ditambahkan, maka akan menampilkan pesan kesalahan |
| 4 | Jika catatan berhasil ditambahkan, maka akan menampilkan daftar catatan |
| - Keluaran: daftar catatan | |

Gambar 5.7. Pengujian unit tambah catatan

Berikut ini merupakan *flow graph* dari algoritma diatas yang akan dijelaskan pada Gambar 5.8.



Gambar 5.8. Flow graph tambah catatan

Berdasarkan flow graph diatas akan diperoleh jumlah kompleksitas siklomatis (*cyclomatic complexity*) melalui persamaan $V = E - N + 2$, dimana V adalah jumlah kompleksitas siklomatis, E merupakan sisi atau *edge* (garis penghubung antar *node*) dan N merupakan jumlah simpul (*node*). Berikut ini merupakan perhitungan dari jumlah kompleksitas siklomatis:

$$\begin{aligned}
 V &= E - N + 2 \\
 &= 5 - 5 + 2 \\
 &= 2
 \end{aligned}$$

Berdasarkan jumlah kompleksitas siklomatis yang telah didapatkan, maka akan ditentukan 2 jalur independen, yaitu:

Jalur 1: 1 – 2 – 3 – 5

Jalur 2: 1 – 2 – 4 – 5

Berdasarkan jalur independen yang telah ditentukan, maka dapat diperoleh kasus uji yang akan dijelaskan pada Tabel 5.4.

Tabel 5.4. Kasus uji tambah catatan

| Jalur | Kasus Uji | Hasil Yang Diharapkan | Hasil Yang Didapatkan |
|-------|--|--|--|
| 1 | Pengguna tidak memberikan masukan judul dan/atau isi catatan | Sistem akan menampilkan pesan kesalahan | Sistem akan menampilkan pesan kesalahan |
| 2 | Pengguna memberikan masukan judul dan/atau isi catatan | Sistem akan memasukkan data catatan ke basis | Sistem akan memasukkan data catatan ke basis |



| Jalur | Kasus Uji | Hasil Yang Diharapkan | Hasil Yang Didapatkan |
|-------|-----------|--|--|
| | | data kemudian menampilkan daftar catatan | data kemudian menampilkan daftar catatan |

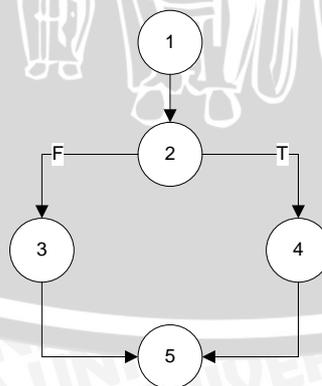
5.1.1.5. Pengujian Unit Lihat Detail Catatan

Berikut ini merupakan algoritma untuk melihat detail catatan pengguna yang akan dijelaskan pada Gambar 5.9.

| | |
|--|--|
| Nama Algoritma: lihat detail catatan pengguna | |
| Deskripsi | |
| - Masukan: id catatan | |
| - Proses | |
| 1 | Memilih catatan yang ingin dilihat detailnya |
| 2 | Mencari catatan berdasarkan id catatan pada basis data |
| 3 | Jika catatan tersebut tidak ada, maka akan menampilkan pesan kesalahan |
| 4 | Jika catatan tersebut ada, maka akan menampilkan detail catatan |
| - Keluaran: detail catatan | |

Gambar 5.9. Pengujian unit lihat detail catatan

Berikut ini merupakan *flow graph* dari algoritma diatas yang akan dijelaskan pada Gambar 5.10.



Gambar 5.10. *Flow graph* lihat detail catatan

Berdasarkan *flow graph* diatas akan diperoleh jumlah kompleksitas siklomatis (*cyclomatic complexity*) melalui persamaan $V = E - N + 2$, dimana V adalah jumlah kompleksitas siklomatis, E merupakan sisi atau *edge* (garis penghubung antar *node*) dan N merupakan jumlah simpul (*node*). Berikut ini merupakan perhitungan dari jumlah kompleksitas siklomatis:

$$\begin{aligned} V &= E - N + 2 \\ &= 5 - 5 + 2 \\ &= 2 \end{aligned}$$

Berdasarkan jumlah kompleksitas siklomatis yang telah didapatkan, maka akan ditentukan 2 jalur independen, yaitu:

Jalur 1: 1 – 2 – 3 – 5

Jalur 2: 1 – 2 – 4 – 5

Berdasarkan jalur independen yang telah ditentukan, maka dapat diperoleh kasus uji yang akan dijelaskan pada Tabel 5.5.

Tabel 5.5. Kasus uji detail catatan

| Jalur | Kasus Uji | Hasil Yang Diharapkan | Hasil Yang Didapatkan |
|-------|--|---|---|
| 1 | Sistem tidak dapat mendeteksi id catatan | Sistem akan menampilkan pesan kesalahan | Sistem akan menampilkan pesan kesalahan |
| 2 | Sistem dapat mendeteksi id catatan | Sistem akan menampilkan detail catatan | Sistem akan menampilkan detail catatan |

5.1.1.6. Pengujian Unit *Update* Catatan

Berikut ini merupakan algoritma untuk melakukan proses *update* catatan yang akan dijelaskan pada Gambar 5.11.

Nama Algoritma: *update* catatan

Deskripsi

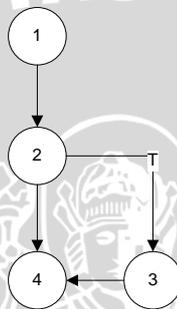
- Masukan: id catatan, judul, isi, dan status pembagian catatan
- Proses

| | |
|---|--|
| 1 | Menampilkan judul, isi, dan status pembagian catatan pada kolom yang |
|---|--|

| | |
|--|--|
| | tersedia berdasarkan id catatan dari basis data |
| 2 | Mengubah judul, isi, atau status pembagian catatan pada basis data |
| 3 | Jika proses <i>update</i> berhasil, maka akan menampilkan detail catatan yang telah diperbarui |
| - Keluaran: detail catatan yang telah diperbarui | |

Gambar 5.11. Pengujian unit *update* catatan

Berikut ini merupakan *flow graph* dari algoritma diatas yang akan dijelaskan pada Gambar 5.12.



Gambar 5.12. *Flow graph* update catatan

Berdasarkan *flow graph* diatas akan diperoleh jumlah kompleksitas siklomatis (*cyclomatic complexity*) melalui persamaan $V = E - N + 2$, dimana V adalah jumlah kompleksitas siklomatis, E merupakan sisi atau *edge* (garis penghubung antar *node*) dan N merupakan jumlah simpul (*node*). Berikut ini merupakan perhitungan dari jumlah kompleksitas siklomatis:

$$\begin{aligned}
 V &= E - N + 2 \\
 &= 4 - 4 + 2 \\
 &= 2
 \end{aligned}$$

Berdasarkan jumlah kompleksitas siklomatis yang telah didapatkan, maka akan ditentukan 2 jalur independen, yaitu:

Jalur 1: 1 – 2 – 3 – 4

Jalur 2: 1 – 2 – 4

Berdasarkan jalur independen yang telah ditentukan, maka dapat diperoleh kasus uji yang akan dijelaskan pada Tabel 5.6.

Tabel 5.6. Kasus uji *update* catatan

| Jalur | Kasus Uji | Hasil Yang Diharapkan | Hasil Yang Didapatkan |
|-------|---|--|--|
| 1 | Pengguna mengubah isi kolom judul, isi, atau status pembagian catatan | Sistem akan mengubah data catatan pada basis data dan akan menampilkan detail catatan | Sistem akan mengubah data catatan pada basis data dan akan menampilkan detail catatan |
| 2 | Pengguna tidak mengubah isi kolom judul, isi, atau status pembagian catatan | Sistem tidak mengubah data catatan pada basis data dan akan menampilkan detail catatan yang telah diperbarui | Sistem tidak mengubah data catatan pada basis data dan akan menampilkan detail catatan yang telah diperbarui |

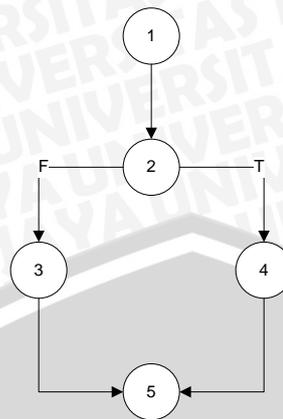
5.1.1.7. Pengujian Unit Hapus Catatan

Berikut ini merupakan algoritma untuk melakukan proses penghapusan catatan yang akan dijelaskan pada Gambar 5.13.

| | |
|--------------------------------------|---|
| Nama Algoritma: hapus catatan | |
| Deskripsi | |
| - Masukan: id catatan | |
| - Proses | |
| 1 | Melakukan konfirmasi penghapusan catatan |
| 2 | Menghapus catatan dari basis data |
| 3 | Jika catatan gagal dihapus, maka akan menampilkan pesan kesalahan |
| 4 | Jika catatan berhasil dihapus, maka akan menampilkan daftar catatan dimana catatan yang sudah dihapus tidak ditampilkan lagi dalam daftar |
| - Keluaran: daftar catatan | |

Gambar 5.13. Pengujian unit hapus catatan

Berikut ini merupakan *flow graph* dari algoritma diatas yang akan dijelaskan pada Gambar 5.14.



Gambar 5.14. Flow graph hapus catatan

Berdasarkan *flow graph* diatas akan diperoleh jumlah kompleksitas siklomatis (*cyclomatic complexity*) melalui persamaan $V = E - N + 2$, dimana V adalah jumlah kompleksitas siklomatis, E merupakan sisi atau *edge* (garis penghubung antar *node*) dan N merupakan jumlah simpul (*node*). Berikut ini merupakan perhitungan dari jumlah kompleksitas siklomatis:

$$\begin{aligned}
 V &= E - N + 2 \\
 &= 5 - 5 + 2 \\
 &= 2
 \end{aligned}$$

Berdasarkan jumlah kompleksitas siklomatis yang telah didapatkan, maka akan ditentukan 2 jalur independen, yaitu:

Jalur 1: 1 – 2 – 3 – 5

Jalur 2: 1 – 2 – 4 – 5

Berdasarkan jalur independen yang telah ditentukan, maka dapat diperoleh kasus uji yang akan dijelaskan pada Tabel 5.7.

Tabel 5.7. Kasus uji hapus catatan

| Jalur | Kasus Uji | Hasil Yang Diharapkan | Hasil Yang Didapatkan |
|-------|--|---|---|
| 1 | Sistem tidak dapat mendeteksi id catatan | Sistem akan menampilkan pesan kesalahan | Sistem akan menampilkan pesan kesalahan |
| 2 | Sistem dapat mendeteksi id catatan | Sistem menghapus catatan dari basis | Sistem menghapus catatan dari basis |



| Jalur | Kasus Uji | Hasil Yang Diharapkan | Hasil Yang Didapatkan |
|-------|-----------|--|--|
| | | data dan akan menampilkan daftar catatan dimana catatan yang sudah dihapus tidak ditampilkan lagi dalam daftar | data dan akan menampilkan daftar catatan dimana catatan yang sudah dihapus tidak ditampilkan lagi dalam daftar |

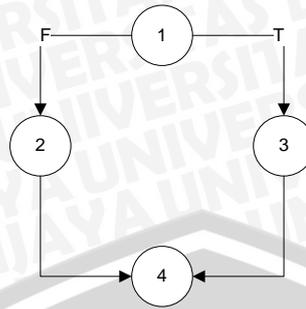
5.1.1.8. Pengujian Unit Lihat Tempat Bekerja

Berikut ini merupakan algoritma untuk melihat daftar lihat tempat bekerja yang akan dijelaskan pada Gambar 5.15.

| | |
|---|--|
| Nama Algoritma: lihat tempat bekerja | |
| Deskripsi | |
| - Masukan: id tempat bekerja | |
| - Proses | |
| 1 | Melakukan pengecekan apakah tempat bekerja dengan id tempat bekerja ada pada basis data |
| 2 | Jika tempat bekerja tersebut tidak ada, maka akan menampilkan daftar yang berisi belum bekerja dan lainnya |
| 3 | Jika tempat bekerja tersebut ada, maka akan menampilkan daftar tempat bekerja |
| - Keluaran: daftar tempat bekerja | |

Gambar 5.15. Pengujian unit lihat tempat bekerja

Berikut ini merupakan *flow graph* dari algoritma diatas yang akan dijelaskan pada Gambar 5.16.



Gambar 5.16. Flow graph lihat tempat bekerja

Berdasarkan *flow graph* diatas akan diperoleh jumlah kompleksitas siklomatis (*cyclomatic complexity*) melalui persamaan $V = E - N + 2$, dimana V adalah jumlah kompleksitas siklomatis, E merupakan sisi atau *edge* (garis penghubung antar *node*) dan N merupakan jumlah simpul (*node*). Berikut ini merupakan perhitungan dari jumlah kompleksitas siklomatis:

$$\begin{aligned}
 V &= E - N + 2 \\
 &= 4 - 4 + 2 \\
 &= 2
 \end{aligned}$$

Berdasarkan jumlah kompleksitas siklomatis yang telah didapatkan, maka akan ditentukan 2 jalur independen, yaitu:

Jalur 1: 1 – 2 – 4

Jalur 2: 1 – 3 – 4

Berdasarkan jalur independen yang telah ditentukan, maka dapat diperoleh kasus uji yang akan dijelaskan pada Tabel 5.8.

Tabel 5.8. Kasus uji lihat catatan

| Jalur | Kasus Uji | Hasil Yang Diharapkan | Hasil Yang Didapatkan |
|-------|---|--|--|
| 1 | Data tempat bekerja tidak ada pada basis data | Sistem akan menampilkan daftar yang berisi belum bekerja dan lainnya | Sistem akan menampilkan daftar yang berisi belum bekerja dan lainnya |
| 2 | Data tempat bekerja ada pada basis data | Sistem akan menampilkan daftar tempat | Sistem akan menampilkan daftar tempat |



| Jalur | Kasus Uji | Hasil Yang Diharapkan | Hasil Yang Didapatkan |
|-------|-----------|-----------------------|-----------------------|
| | | bekerja | bekerja |

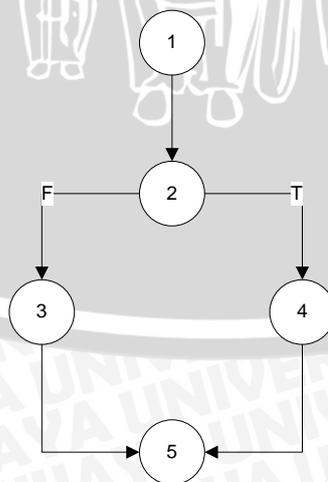
5.1.1.9. Pengujian Unit Tambah Tempat Bekerja

Berikut ini merupakan algoritma untuk melakukan proses penambahan tempat bekerja yang akan dijelaskan pada Gambar 5.17.

| | |
|--|---|
| Nama Algoritma: tambah tempat kerja | |
| Deskripsi | |
| - Masukan: tempat bekerja (Nama RS - Bagian) | |
| - Proses | |
| 1 | Mengisi kolom tempat bekerja |
| 2 | Menambahkan tempat bekerja pada basis data |
| 3 | Jika tempat bekerja gagal ditambahkan, maka akan menampilkan pesan kesalahan |
| 4 | Jika tempat bekerja berhasil ditambahkan, maka akan menampilkan daftar tempat bekerja |
| - Keluaran: daftar tempat bekerja | |

Gambar 5.17. Pengujian unit tambah tempat bekerja

Berikut ini merupakan *flow graph* dari algoritma diatas yang akan dijelaskan pada Gambar 5.18.



Gambar 5.18. *Flow graph* tambah tempat bekerja

Berdasarkan *flow graph* diatas akan diperoleh jumlah kompleksitas siklomatis (*cyclomatic complexity*) melalui persamaan $V = E - N + 2$, dimana V adalah jumlah kompleksitas siklomatis, E merupakan sisi atau *edge* (garis penghubung antar *node*) dan N merupakan jumlah simpul (*node*). Berikut ini merupakan perhitungan dari jumlah kompleksitas siklomatis:

$$\begin{aligned} V &= E - N + 2 \\ &= 5 - 5 + 2 \\ &= 2 \end{aligned}$$

Berdasarkan jumlah kompleksitas siklomatis yang telah didapatkan, maka akan ditentukan 2 jalur independen, yaitu:

Jalur 1: 1 – 2 – 3 – 5

Jalur 2: 1 – 2 – 4 – 5

Berdasarkan jalur independen yang telah ditentukan, maka dapat diperoleh kasus uji yang akan dijelaskan pada Tabel 5.9.

Tabel 5.9. Kasus uji tambah tempat bekerja

| Jalur | Kasus Uji | Hasil Yang Diharapkan | Hasil Yang Didapatkan |
|-------|--|--|--|
| 1 | Pengguna tidak memberikan masukan nama tempat bekerja | Sistem akan menampilkan pesan kesalahan | Sistem akan menampilkan pesan kesalahan |
| 2 | Pengguna memberikan masukan nama tempat bekerja yang sesuai format penulisan | Sistem akan memasukkan data tempat bekerja pada basis data dan menampilkan daftar tempat bekerja | Sistem akan memasukkan data tempat bekerja pada basis data dan menampilkan daftar tempat bekerja |

5.1.1.10. Pengujian Unit Pilih Tempat Bekerja

Berikut ini merupakan algoritma untuk memilih tempat bekerja yang akan dijelaskan pada Gambar 5.19.

Nama Algoritma: pilih tempat bekerja

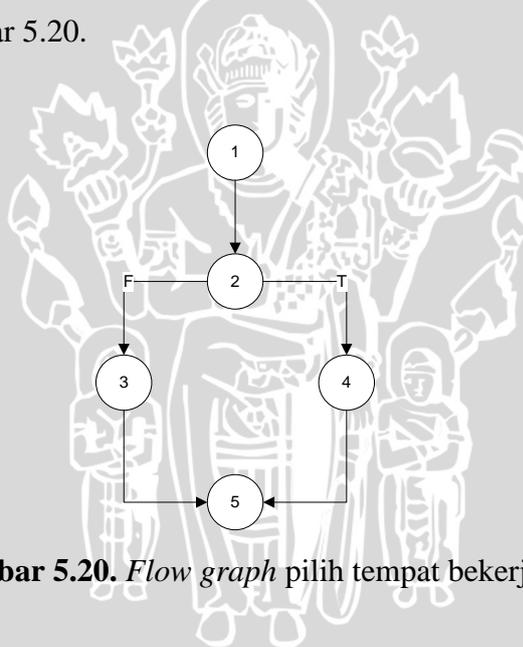
Deskripsi

- Masukan: id pengguna dan id tempat bekerja
- Proses

| | |
|----------------------------|--|
| 1 | Pilih tempat bekerja |
| 2 | Ubah data id tempat bekerja pengguna berdasarkan id pengguna pada basis data |
| 3 | Jika data pengguna gagal diubah, maka akan menampilkan pesan kesalahan |
| 4 | Jika data pengguna berhasil diubah, maka akan menampilkan halaman profil |
| - Keluaran: halaman profil | |

Gambar 5.19. Pengujian unit pilih tempat bekerja

Berikut ini merupakan *flow graph* dari algoritma diatas yang akan dijelaskan pada Gambar 5.20.

**Gambar 5.20.** *Flow graph* pilih tempat bekerja

Berdasarkan *flow graph* diatas akan diperoleh jumlah kompleksitas siklomatis (*cyclomatic complexity*) melalui persamaan $V = E - N + 2$, dimana V adalah jumlah kompleksitas siklomatis, E merupakan sisi atau *edge* (garis penghubung antar *node*) dan N merupakan jumlah simpul (*node*). Berikut ini merupakan perhitungan dari jumlah kompleksitas siklomatis:

$$\begin{aligned}
 V &= E - N + 2 \\
 &= 5 - 3 + 2 \\
 &= 4
 \end{aligned}$$

Berdasarkan jumlah kompleksitas siklomatis yang telah didapatkan, maka akan ditentukan 2 jalur independen, yaitu:

Jalur 1: 1 – 2 – 3 – 5

Jalur 2: 1 – 2 – 4 – 5

Berdasarkan jalur independen yang telah ditentukan, maka dapat diperoleh kasus uji yang akan dijelaskan pada Tabel 5.10.

Tabel 5.10. Kasus uji pilih tempat bekerja

| Jalur | Kasus Uji | Hasil Yang Diharapkan | Hasil Yang Didapatkan |
|-------|---|---|---|
| 1 | Sistem tidak dapat mendeteksi id pengguna dan id tempat bekerja | Sistem akan menampilkan pesan kesalahan | Sistem akan menampilkan pesan kesalahan |
| 2 | Sistem dapat mendeteksi id pengguna dan id tempat bekerja | Sistem akan mengubah data pengguna pada basis data dan menampilkan halaman profil | Sistem akan mengubah data pengguna pada basis data dan menampilkan halaman profil |

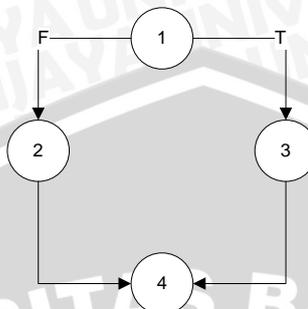
5.1.1.11. Pengujian Unit Lihat Profil

Berikut ini merupakan algoritma untuk melihat profil yang akan dijelaskan pada Gambar 5.21.

| | |
|--|---|
| Nama Algoritma: lihat profil | |
| Deskripsi | |
| - Masukan: id pengguna dan id tempat bekerja | |
| - Proses | |
| 1 | Melakukan pengecekan apakah pengguna dengan id pengguna dan id tempat bekerja ada pada basis data |
| 2 | Jika data pengguna tidak ditemukan, maka akan menampilkan pesan kesalahan |
| 3 | Jika data pengguna ditemukan, maka akan menampilkan profil pengguna |
| - Keluaran: profil pengguna | |

Gambar 5.21. Pengujian unit lihat profil

Berikut ini merupakan *flow graph* dari algoritma diatas yang akan dijelaskan pada Gambar 5.22.



Gambar 5.22. *Flow graph* lihat profil

Berdasarkan *flow graph* diatas akan diperoleh jumlah kompleksitas siklomatis (*cyclomatic complexity*) melalui persamaan $V = E - N + 2$, dimana V adalah jumlah kompleksitas siklomatis, E merupakan sisi atau *edge* (garis penghubung antar *node*) dan N merupakan jumlah simpul (*node*). Berikut ini merupakan perhitungan dari jumlah kompleksitas siklomatis:

$$\begin{aligned}
 V &= E - N + 2 \\
 &= 4 - 4 + 2 \\
 &= 2
 \end{aligned}$$

Berdasarkan jumlah kompleksitas siklomatis yang telah didapatkan, maka akan ditentukan 2 jalur independen, yaitu:

Jalur 1: 1 – 2 – 4

Jalur 2: 1 – 3 – 4

Berdasarkan jalur independen yang telah ditentukan, maka dapat diperoleh kasus uji yang akan dijelaskan pada Tabel 5.11.

Tabel 5.11. Kasus uji lihat profil

| Jalur | Kasus Uji | Hasil Yang Diharapkan | Hasil Yang Didapatkan |
|-------|---|---|---|
| 1 | Sistem tidak dapat mendeteksi id pengguna dan id tempat bekerja | Sistem akan menampilkan pesan kesalahan | Sistem akan menampilkan pesan kesalahan |
| 2 | Sistem dapat mendeteksi id | Sistem akan | Sistem akan |

| Jalur | Kasus Uji | Hasil Yang Diharapkan | Hasil Yang Didapatkan |
|-------|--------------------------------|----------------------------|----------------------------|
| | pengguna dan id tempat bekerja | menampilkan halaman profil | menampilkan halaman profil |

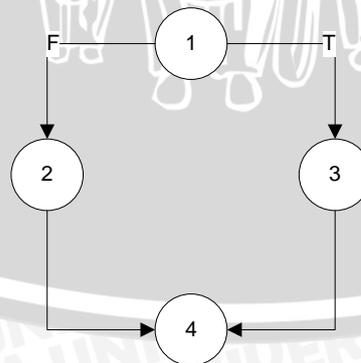
5.1.1.12. Pengujian Unit Lihat Konten Keperawatan

Berikut ini merupakan algoritma untuk melihat konten keperawatan seperti data farmakologi, patofisiologi, pemeriksaan diagnostik, dan anatomi fisiologi yang akan dijelaskan pada Gambar 5.23.

| | |
|---|---|
| Nama Algoritma: lihat konten keperawatan | |
| Deskripsi | |
| - Masukan: - | |
| - Proses | |
| 1 | Melakukan pengecekan apakah data konten keperawatan ada pada basis data |
| 2 | Jika data tidak ada, maka akan menampilkan daftar kosong |
| 3 | Jika data ada, maka akan menampilkan daftar konten keperawatan |
| - Keluaran: daftar konten keperawatan | |

Gambar 5.23. Pengujian unit lihat konten keperawatan

Berikut ini merupakan *flow graph* dari algoritma diatas yang akan dijelaskan pada Gambar 5.24.



Gambar 5.24. *Flow graph* lihat konten keperawatan

Berdasarkan *flow graph* diatas akan diperoleh jumlah kompleksitas siklomatis (*cyclomatic complexity*) melalui persamaan $V = E - N + 2$, dimana V adalah jumlah kompleksitas siklomatis, E merupakan sisi atau *edge* (garis penghubung antar *node*) dan N merupakan jumlah simpul (*node*). Berikut ini merupakan perhitungan dari jumlah kompleksitas siklomatis:

$$\begin{aligned} V &= E - N + 2 \\ &= 4 - 4 + 2 \\ &= 2 \end{aligned}$$

Berdasarkan jumlah kompleksitas siklomatis yang telah didapatkan, maka akan ditentukan 2 jalur independen, yaitu:

Jalur 1: 1 – 2 – 4

Jalur 2: 1 – 3 – 4

Berdasarkan jalur independen yang telah ditentukan, maka dapat diperoleh kasus uji yang akan dijelaskan pada Tabel 5.12.

Tabel 5.12. Kasus uji lihat konten keperawatan

| Jalur | Kasus Uji | Hasil Yang Diharapkan | Hasil Yang Didapatkan |
|-------|--|---|---|
| 1 | Konten keperawatan tidak ada pada basis data | Sistem akan menampilkan daftar kosong | Sistem akan menampilkan daftar kosong |
| 2 | Konten keperawatan ada pada basis data | Sistem akan menampilkan daftar konten keperawatan | Sistem akan menampilkan daftar konten keperawatan |

5.1.1.13. Pengujian Unit Lihat Detail Konten Keperawatan

Berikut ini merupakan algoritma untuk melihat detail konten keperawatan seperti detail data farmakologi, patofisiologi, pemeriksaan diagnostik, dan anatomi fisiologi yang akan dijelaskan pada Gambar 5.25.

Nama Algoritma: lihat detail konten keperawatan

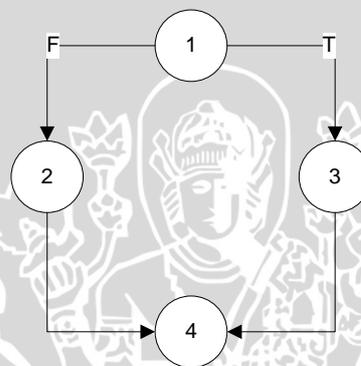
Deskripsi

- Masukan: id konten

| | |
|---------------------------------------|--|
| - Proses | |
| 1 | Melakukan pengecekan apakah data konten keperawatan dengan id konten ada pada basis data |
| 2 | Jika data tidak ada, maka akan menampilkan pesan kesalahan |
| 3 | Jika data ada, maka akan menampilkan detail konten keperawatan |
| - Keluaran: detail konten keperawatan | |

Gambar 5.25. Pengujian unit lihat detail konten keperawatan

Berikut ini merupakan *flow graph* dari algoritma diatas yang akan dijelaskan pada Gambar 5.26.



Gambar 5.26. *Flow graph* lihat detail konten keperawatan

Berdasarkan *flow graph* diatas akan diperoleh jumlah kompleksitas siklomatis (*cyclomatic complexity*) melalui persamaan $V = E - N + 2$, dimana V adalah jumlah kompleksitas siklomatis, E merupakan sisi atau *edge* (garis penghubung antar *node*) dan N merupakan jumlah simpul (*node*). Berikut ini merupakan perhitungan dari jumlah kompleksitas siklomatis:

$$\begin{aligned}
 V &= E - N + 2 \\
 &= 4 - 4 + 2 \\
 &= 2
 \end{aligned}$$

Berdasarkan jumlah kompleksitas siklomatis yang telah didapatkan, maka akan ditentukan 2 jalur independen, yaitu:

$$\text{Jalur 1: } 1 - 2 - 4$$

$$\text{Jalur 2: } 1 - 3 - 4$$

Berdasarkan jalur independen yang telah ditentukan, maka dapat diperoleh kasus uji yang akan dijelaskan pada Tabel 5.13.

Tabel 5.13. Kasus uji lihat detail konten keperawatan

| Jalur | Kasus Uji | Hasil Yang Diharapkan | Hasil Yang Didapatkan |
|-------|---|---|---|
| 1 | Sistem tidak dapat mendeteksi id konten keperawatan | Sistem akan menampilkan pesan kesalahan | Sistem akan menampilkan pesan kesalahan |
| 2 | Sistem dapat mendeteksi id konten keperawatan | Sistem akan menampilkan detail konten keperawatan | Sistem akan menampilkan detail konten keperawatan |

5.1.2. Pengujian Validasi

Pengujian validasi dilakukan untuk menguji seluruh skenario *use case* yang telah dibuat pada bab sebelumnya.

5.1.2.1. Kasus Uji Pengujian Validasi

Berikut ini merupakan kasus uji untuk melakukan pengujian validasi pada Sistem Buku Saku Perawat.

1. Kasus Uji *Login*

Berikut ini merupakan kasus uji untuk melakukan proses *login* yang akan dijelaskan pada Tabel 5.14., Tabel 5.15., dan Tabel 5.16.

Tabel 5.14. Kasus uji *login*

| | |
|-------------------------|--|
| Nama Kasus Uji | Kasus uji <i>login</i> |
| Objek Uji | Kebutuhan Fungsional (KF_01) |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan bahwa sistem dapat melakukan proses <i>login</i> |
| Data Masukan | <i>email</i> dan <i>password</i> |
| Prosedur Uji | <ol style="list-style-type: none"> 1. Membuka aplikasi dan memilih menu <i>login</i> 2. Memasukkan <i>email</i> dan <i>password</i> pada kolom yang tersedia dengan masukan valid (data ada pada basis data) |

| | |
|------------------------------|----------------------------------|
| | 3. Menekan tombol <i>sign in</i> |
| Hasil Yang Diharapkan | Pengguna berhasil login |

Tabel 5.15. Kasus uji *login* alur alternatif 1 (*email* atau *password* salah)

| | |
|------------------------------|--|
| Nama Kasus Uji | Kasus uji <i>login</i> alur alternatif 1 (<i>email</i> atau <i>password</i> salah) |
| Objek Uji | Kebutuhan Fungsional (KF_01) |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan bahwa sistem dapat menampilkan pesan kesalahan ketika pengguna memasukkan <i>email</i> atau <i>password</i> yang tidak valid |
| Data Masukan | <i>email</i> dan <i>password</i> |
| Prosedur Uji | <ol style="list-style-type: none"> 1. Membuka aplikasi dan memilih menu <i>login</i> 2. Memasukkan <i>email</i> dan <i>password</i> pada kolom yang tersedia dengan masukan tidak valid (data tidak ada pada basis data) 3. Menekan tombol <i>sign in</i> |
| Hasil Yang Diharapkan | Sistem akan menampilkan pesan kesalahan |

Tabel 5.16. Kasus uji *login* alur alternatif 2 (kolom *email* atau *password* tidak diisi)

| | |
|------------------------------|--|
| Nama Kasus Uji | Kasus uji <i>login</i> alur alternatif 2 (kolom <i>email</i> atau <i>password</i> tidak diisi) |
| Objek Uji | Kebutuhan Fungsional (KF_01) |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan bahwa sistem dapat menampilkan pesan kesalahan ketika pengguna tidak mengisi kolom <i>email</i> atau <i>password</i> |
| Data Masukan | <i>email</i> dan <i>password</i> |
| Prosedur Uji | <ol style="list-style-type: none"> 1. Membuka aplikasi dan memilih menu <i>login</i> 2. Pengguna tidak memasukkan <i>email</i> dan/atau <i>password</i> pada kolom yang tersedia 3. Menekan tombol <i>sign in</i> |
| Hasil Yang Diharapkan | Sistem akan menampilkan pesan kesalahan |

2. Kasus Uji Register

Berikut ini merupakan kasus uji untuk melakukan proses registrasi akun yang akan dijelaskan pada Tabel 5.17., Tabel 5.18., dan Tabel 5.19.

Tabel 5.17. Kasus uji *register*

| | |
|------------------------------|---|
| Nama Kasus Uji | Kasus uji <i>register</i> |
| Objek Uji | Kebutuhan Fungsional (KF_02) |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan bahwa sistem dapat melakukan proses registrasi akun |
| Data Masukan | nama, <i>email</i> dan <i>password</i> |
| Prosedur Uji | <ol style="list-style-type: none"> 1. Membuka aplikasi dan memilih menu <i>login</i> 2. Menekan tombol <i>create account</i> 3. Memasukkan nama, <i>email</i>, dan/atau <i>password</i> pada kolom yang tersedia dengan data valid (data belum ada pada basis data) 4. Menekan tombol <i>create account</i> |
| Hasil Yang Diharapkan | Sistem akan menampilkan halaman login |

Tabel 5.18. Kasus uji *register* alur alternatif 1 (pengguna sudah memiliki akun)

| | |
|------------------------------|---|
| Nama Kasus Uji | Kasus uji <i>register</i> alur alternatif 1 (pengguna sudah memiliki akun) |
| Objek Uji | Kebutuhan Fungsional (KF_02) |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan bahwa sistem dapat menampilkan pesan kesalahan ketika pengguna mendaftarkan akun yang sudah ada pada basis data |
| Data Masukan | nama, <i>email</i> dan <i>password</i> |
| Prosedur Uji | <ol style="list-style-type: none"> 1. Membuka aplikasi dan memilih menu <i>login</i> 2. Menekan tombol <i>create account</i> 3. Memasukkan nama, <i>email</i>, dan/atau <i>password</i> pada kolom yang tersedia dengan data tidak valid (data sudah ada pada basis data) 4. Menekan tombol <i>create login</i> |
| Hasil Yang Diharapkan | Sistem akan menampilkan pesan kesalahan |

Tabel 5.19. Kasus uji *register* alur alternatif 2 (kolom nama, *email*, atau *password* tidak diisi)

| | |
|-----------------------|---|
| Nama Kasus Uji | Kasus uji <i>register</i> alur alternatif 2 (kolom nama, <i>email</i> , atau <i>password</i> tidak diisi) |
|-----------------------|---|

| | |
|------------------------------|---|
| Objek Uji | Kebutuhan Fungsional (KF_02) |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan bahwa sistem dapat menampilkan pesan kesalahan ketika pengguna tidak mengisi kolom nama, <i>email</i> atau <i>password</i> |
| Data Masukan | nama, <i>email</i> dan <i>password</i> |
| Prosedur Uji | <ol style="list-style-type: none"> 1. Membuka aplikasi dan memilih menu <i>login</i> 2. Menekan tombol <i>create account</i> 3. Pengguna tidak memasukkan nama, <i>email</i>, dan/atau <i>password</i> pada kolom yang tersedia 4. Menekan tombol <i>create login</i> |
| Hasil Yang Diharapkan | Sistem akan menampilkan pesan kesalahan |

3. Kasus Uji Lihat Catatan

Berikut ini merupakan kasus uji untuk melihat daftar catatan yang dimiliki pengguna yang akan dijelaskan pada Tabel 5.20.

Tabel 5.20. Kasus uji lihat catatan

| | |
|------------------------------|---|
| Nama Kasus Uji | Kasus uji lihat catatan |
| Objek Uji | Kebutuhan Fungsional (KF_03) |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan bahwa sistem dapat menampilkan daftar catatan berdasarkan id pengguna atau id tempat bekerja |
| Data Masukan | - |
| Prosedur Uji | <ol style="list-style-type: none"> 1. Membuka aplikasi dan memilih menu catatan |
| Hasil Yang Diharapkan | Sistem akan menampilkan daftar catatan pengguna atau pengguna lain yang memiliki tempat bekerja sama |

4. Kasus Uji Tambah Catatan

Berikut ini merupakan kasus uji untuk melakukan proses penambahan catatan yang akan dijelaskan pada Tabel 5.21. dan Tabel 5.22.

Tabel 5.21. Kasus uji tambah catatan

| | |
|-------------------------|---|
| Nama Kasus Uji | Kasus uji tambah catatan |
| Objek Uji | Kebutuhan Fungsional (KF_04) |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan bahwa sistem dapat melakukan proses penambahan catatan pada basis data |

| | |
|------------------------------|---|
| Data Masukan | judul, isi, dan status pembagian catatan (opsional) |
| Prosedur Uji | <ol style="list-style-type: none"> 1. Membuka aplikasi dan memilih menu catatan 2. Menekan tombol <i>add note</i> 3. Mengisi kolom judul dan isi catatan, status pembagian catatan boleh tidak dicentang 4. Menekan tombol simpan |
| Hasil Yang Diharapkan | Sistem akan menampilkan daftar catatan pengguna atau pengguna lain yang memiliki tempat bekerja sama |

Tabel 5.22. Kasus uji tambah catatan alur alternatif 1 (kolom judul atau isi catatan tidak diisi)

| | |
|------------------------------|--|
| Nama Kasus Uji | Kasus uji tambah catatan alur alternatif 1 (kolom judul atau isi catatan tidak diisi) |
| Objek Uji | Kebutuhan Fungsional (KF_04) |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan bahwa sistem dapat menampilkan pesan kesalahan ketika pengguna tidak mengisi kolom judul atau isi catatan |
| Data Masukan | judul, isi, dan status pembagian catatan (opsional) |
| Prosedur Uji | <ol style="list-style-type: none"> 1. Membuka aplikasi dan memilih menu catatan 2. Menekan tombol <i>add note</i> 3. Pengguna tidak mengisi kolom judul dan isi catatan 4. Menekan tombol simpan |
| Hasil Yang Diharapkan | Sistem akan menampilkan pesan kesalahan |

5. Kasus Uji Lihat Detail Catatan

Berikut ini merupakan kasus uji untuk melihat detail catatan pengguna yang akan dijelaskan pada Tabel 5.23.

Tabel 5.23. Kasus uji lihat detail catatan

| | |
|-------------------------|---|
| Nama Kasus Uji | Kasus uji lihat detail catatan |
| Objek Uji | Kebutuhan Fungsional (KF_05) |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan bahwa sistem dapat menampilkan detail catatan berdasarkan id catatan |
| Data Masukan | - |
| Prosedur Uji | <ol style="list-style-type: none"> 1. Membuka aplikasi dan memilih menu catatan 2. Memilih salah satu catatan |
| Hasil Yang | Sistem akan menampilkan detail catatan pengguna atau |

| | |
|-------------------|---|
| Diharapkan | pengguna lain yang memiliki tempat bekerja sama |
|-------------------|---|

6. Kasus Uji Ubah Catatan

Berikut ini merupakan kasus uji untuk melakukan proses *update* catatan yang akan dijelaskan pada Tabel 5.24. dan Tabel 5.25.

Tabel 5.24. Kasus uji *update* catatan

| | |
|------------------------------|---|
| Nama Kasus Uji | Kasus uji <i>update</i> catatan |
| Objek Uji | Kebutuhan Fungsional (KF_06) |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan bahwa sistem dapat melakukan proses <i>update</i> catatan |
| Data Masukan | judul, isi, dan status pembagian catatan |
| Prosedur Uji | <ol style="list-style-type: none"> 1. Membuka aplikasi dan memilih menu catatan 2. Memilih salah satu catatan 3. Menekan tombol <i>update</i> 4. Mengubah judul, isi, atau status pembagian catatan 5. Menekan tombol simpan |
| Hasil Yang Diharapkan | Sistem akan menampilkan detail catatan pengguna yang telah diperbarui |

Tabel 5.25. Kasus uji *update* catatan alur alternatif 1 (catatan bukan milik pengguna)

| | |
|------------------------------|---|
| Nama Kasus Uji | Kasus uji <i>update</i> catatan alur alternatif 1 (catatan bukan milik pengguna) |
| Objek Uji | Kebutuhan Fungsional (KF_06) |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan bahwa pengguna tidak dapat melakukan <i>update</i> catatan jika pengguna bukan pemilik catatan |
| Data Masukan | judul, isi, dan status pembagian catatan |
| Prosedur Uji | <ol style="list-style-type: none"> 1. Membuka aplikasi dan memilih menu catatan 2. Memilih salah satu catatan |
| Hasil Yang Diharapkan | Sistem akan menyembunyikan tombol <i>update</i> |

7. Kasus Uji Hapus Catatan

Berikut ini merupakan kasus uji untuk melakukan proses hapus catatan yang akan dijelaskan pada Tabel 5.26., Tabel 5.27., dan Tabel 5.28.

Tabel 5.26. Kasus uji hapus catatan

| | |
|------------------------------|---|
| Nama Kasus Uji | Kasus uji hapus catatan |
| Objek Uji | Kebutuhan Fungsional (KF_07) |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan bahwa sistem dapat melakukan proses hapus catatan |
| Data Masukan | - |
| Prosedur Uji | <ol style="list-style-type: none"> 1. Membuka aplikasi dan memilih menu catatan 2. Memilih salah satu catatan 3. Menekan tombol <i>delete</i> 4. Melakukan konfirmasi hapus catatan |
| Hasil Yang Diharapkan | Sistem akan menampilkan daftar catatan dimana catatan yang sudah dihapus tidak ditampilkan lagi dalam daftar |

Tabel 5.27. Kasus uji hapus catatan alur alternatif 1 (pengguna tidak melakukan konfirmasi hapus catatan)

| | |
|------------------------------|---|
| Nama Kasus Uji | Kasus uji hapus catatan alur alternatif 1 (pengguna tidak melakukan konfirmasi hapus catatan) |
| Objek Uji | Kebutuhan Fungsional (KF_07) |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan bahwa sistem tidak melakukan proses hapus catatan |
| Data Masukan | - |
| Prosedur Uji | <ol style="list-style-type: none"> 1. Membuka aplikasi dan memilih menu catatan 2. Memilih salah satu catatan 3. Menekan tombol <i>delete</i> 4. Tidak melakukan konfirmasi hapus catatan |
| Hasil Yang Diharapkan | Sistem akan menampilkan detail catatan pengguna |

Tabel 5.28. Kasus uji hapus catatan alur alternatif 2 (catatan bukan milik pengguna)

| | |
|-------------------------|---|
| Nama Kasus Uji | Kasus uji hapus catatan alur alternatif 2 (catatan bukan milik pengguna) |
| Objek Uji | Kebutuhan Fungsional (KF_07) |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan bahwa pengguna tidak dapat melakukan hapus catatan jika pengguna bukan pemilik catatan |
| Data Masukan | - |

| | |
|------------------------------|---|
| Prosedur Uji | 1. Membuka aplikasi dan memilih menu catatan 2. Memilih salah satu catatan |
| Hasil Yang Diharapkan | Sistem akan menyembunyikan tombol <i>delete</i> |

8. Kasus Uji Lihat Tempat Bekerja

Berikut ini merupakan kasus uji untuk melihat daftar tempat bekerja yang dimiliki pengguna yang akan dijelaskan pada Tabel 5.29.

Tabel 5.29. Kasus uji lihat tempat bekerja

| | |
|------------------------------|---|
| Nama Kasus Uji | Kasus uji lihat tempat bekerja |
| Objek Uji | Kebutuhan Fungsional (KF_08) |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan bahwa sistem dapat menampilkan daftar tempat bekerja berdasarkan id tempat bekerja |
| Data Masukan | - |
| Prosedur Uji | 1. Masuk ke halaman tempat bekerja |
| Hasil Yang Diharapkan | Sistem akan menampilkan daftar tempat bekerja |

9. Kasus Uji Tambah Tempat Bekerja

Berikut ini merupakan kasus uji untuk melakukan proses penambahan tempat bekerja yang akan dijelaskan pada Tabel 5.30. dan Tabel 5.31.

Tabel 5.30. Kasus uji tambah tempat bekerja

| | |
|-------------------------|---|
| Nama Kasus Uji | Kasus uji tambah tempat bekerja |
| Objek Uji | Kebutuhan Fungsional (KF_09) |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan bahwa sistem dapat melakukan proses penambahan tempat bekerja pada basis data |
| Data Masukan | tempat bekerja (Nama RS - Bagian) |
| Prosedur Uji | 1. Masuk ke halaman tempat bekerja 2. Memilih Lainnya 3. Mengisi kolom tempat bekerja dengan format yang telah ditentukan 4. Menekan tombol simpan |
| Hasil Yang | Sistem akan menampilkan daftar tempat bekerja |

| | |
|-------------------|--|
| Diharapkan | |
|-------------------|--|

Tabel 5.31. Kasus uji tambah tempat bekerja alur alternatif 1 (kolom tempat bekerja tidak diisi)

| | |
|------------------------------|--|
| Nama Kasus Uji | Kasus uji tambah tempat bekerja alur alternatif 1 (kolom tempat bekerja tidak diisi) |
| Objek Uji | Kebutuhan Fungsional (KF_09) |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan bahwa sistem tidak dapat melakukan proses penambahan tempat bekerja pada basis data |
| Data Masukan | tempat bekerja (Nama RS - Bagian) |
| Prosedur Uji | <ol style="list-style-type: none"> 1. Masuk ke halaman tempat bekerja 2. Memilih Lainnya 3. Pengguna tidak mengisi kolom tempat bekerja 4. Menekan tombol simpan |
| Hasil Yang Diharapkan | Sistem akan menampilkan pesan kesalahan |

10. Kasus Uji Pilih Tempat Bekerja

Berikut ini merupakan kasus uji untuk melakukan pemilihan tempat bekerja yang akan dijelaskan pada Tabel 5.32.

Tabel 5.32. Kasus uji pilih tempat bekerja

| | |
|------------------------------|--|
| Nama Kasus Uji | Kasus uji pilih tempat bekerja |
| Objek Uji | Kebutuhan Fungsional (KF_10) |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan bahwa sistem dapat melakukan <i>update</i> tempat bekerja pengguna berdasarkan id pengguna |
| Data Masukan | - |
| Prosedur Uji | <ol style="list-style-type: none"> 1. Masuk ke halaman tempat bekerja 2. Memilih salah satu tempat bekerja 3. Menekan tombol simpan |
| Hasil Yang Diharapkan | Sistem akan menampilkan profil pengguna |

11. Kasus Uji Lihat Profil

Berikut ini merupakan kasus uji untuk melihat profil pengguna yang akan dijelaskan pada Tabel 5.33.

Tabel 5.33. Kasus uji lihat profil

| | |
|------------------------------|---|
| Nama Kasus Uji | Kasus uji lihat profil |
| Objek Uji | Kebutuhan Fungsional (KF_11) |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan bahwa sistem dapat menampilkan profil pengguna berdasarkan id pengguna dan id tempat bekerja |
| Data Masukan | - |
| Prosedur Uji | 1. Masuk ke halaman profil |
| Hasil Yang Diharapkan | Sistem akan menampilkan profil pengguna |

12. Kasus Uji Lihat Konten Keperawatan

Berikut ini merupakan kasus uji untuk melihat konten keperawatan yang akan dijelaskan pada Tabel 5.34.

Tabel 5.34. Kasus uji lihat konten keperawatan

| | |
|------------------------------|--|
| Nama Kasus Uji | Kasus uji lihat konten keperawatan |
| Objek Uji | Kebutuhan Fungsional (KF_12) |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan bahwa sistem dapat menampilkan daftar konten keperawatan berdasarkan menu yang dipilih pengguna |
| Data Masukan | - |
| Prosedur Uji | 1. Membuka aplikasi 2. Memilih salah satu menu konten keperawatan |
| Hasil Yang Diharapkan | Sistem akan menampilkan daftar konten keperawatan |

13. Kasus Uji Lihat Detail Konten Keperawatan

Berikut ini merupakan kasus uji untuk melihat konten keperawatan yang akan dijelaskan pada Tabel 5.35.

Tabel 5.35. Kasus uji lihat konten keperawatan

| | |
|------------------------------|---|
| Nama Kasus Uji | Kasus uji lihat konten keperawatan |
| Objek Uji | Kebutuhan Fungsional (KF_13) |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan bahwa sistem dapat menampilkan detail konten keperawatan berdasarkan id konten keperawatan |
| Data Masukan | - |
| Prosedur Uji | <ol style="list-style-type: none"> 1. Membuka aplikasi 2. Memilih salah satu menu konten keperawatan 3. Memilih salah satu konten keperawatan yang ada pada daftar |
| Hasil Yang Diharapkan | Sistem akan menampilkan detail konten keperawatan |

5.1.2.2. Hasil Pengujian Validasi

Berikut ini merupakan hasil pengujian validasi pada Sistem Buku Saku Perawat yang akan dijelaskan pada Tabel 5.36.

Tabel 5.36. Hasil pengujian validasi

| No | Nama Kasus Uji | Hasil Yang Diharapkan | Hasil Yang Didapatkan | Validitas |
|----|--|---|---|-----------|
| 1 | Kasus uji <i>login</i> | Pengguna berhasil login | Pengguna berhasil login | Valid |
| 2 | Kasus uji <i>login</i> alur alternatif 1 (<i>email</i> atau <i>password</i> salah) | Sistem akan menampilkan pesan kesalahan | Sistem akan menampilkan pesan kesalahan | Valid |
| 3 | Kasus uji <i>login</i> alur alternatif 2 (kolom <i>email</i> atau <i>password</i> tidak diisi) | Sistem akan menampilkan pesan kesalahan | Sistem akan menampilkan pesan kesalahan | Valid |
| 4 | Kasus uji <i>register</i> | Sistem akan menampilkan halaman login | Sistem akan menampilkan halaman login | Valid |
| 5 | Kasus uji <i>register</i> alur alternatif 1 (pengguna sudah memiliki akun) | Sistem akan menampilkan pesan kesalahan | Sistem akan menampilkan pesan kesalahan | Valid |
| 6 | Kasus uji <i>register</i> alur alternatif 2 (kolom | Sistem akan menampilkan | Sistem akan menampilkan | Valid |

| | nama, <i>email</i> , atau <i>password</i> tidak diisi) | pesan kesalahan | pesan kesalahan | |
|----|---|--|--|-------|
| 7 | Kasus uji lihat catatan | Sistem akan menampilkan daftar catatan pengguna atau pengguna lain yang memiliki tempat bekerja sama | Sistem akan menampilkan daftar catatan pengguna atau pengguna lain yang memiliki tempat bekerja sama | Valid |
| 8 | Kasus uji tambah catatan | Sistem akan menampilkan daftar catatan pengguna atau pengguna lain yang memiliki tempat bekerja sama | Sistem akan menampilkan daftar catatan pengguna atau pengguna lain yang memiliki tempat bekerja sama | Valid |
| 9 | Kasus uji tambah catatan alur alternatif 1 (kolom judul atau isi catatan tidak diisi) | Sistem akan menampilkan pesan kesalahan | Sistem akan menampilkan pesan kesalahan | Valid |
| 10 | Kasus uji lihat detail catatan | Sistem akan menampilkan detail catatan pengguna atau pengguna lain yang memiliki tempat bekerja sama | Sistem akan menampilkan detail catatan pengguna atau pengguna lain yang memiliki tempat bekerja sama | Valid |
| 11 | Kasus uji <i>update</i> catatan | Sistem akan menampilkan detail catatan pengguna yang telah diperbarui | Sistem akan menampilkan detail catatan pengguna yang telah diperbarui | Valid |
| 12 | Kasus uji <i>update</i> catatan alur alternatif 1 (catatan bukan milik pengguna) | Sistem akan menyembunyikan tombol <i>update</i> | Sistem akan menyembunyikan tombol <i>update</i> | Valid |
| 13 | Kasus uji hapus catatan | Sistem akan menampilkan | Sistem akan menampilkan | Valid |

| | | | | |
|----|---|--|--|-------|
| | | daftar catatan dimana catatan yang sudah dihapus tidak ditampilkan lagi dalam daftar | daftar catatan dimana catatan yang sudah dihapus tidak ditampilkan lagi dalam daftar | |
| 14 | Kasus uji hapus catatan alur alternatif 1 (pengguna tidak melakukan konfirmasi hapus catatan) | Sistem akan menampilkan detail catatan pengguna | Sistem akan menampilkan detail catatan pengguna | Valid |
| 15 | Kasus uji hapus catatan alur alternatif 2 (catatan bukan milik pengguna) | Sistem akan menyembunyikan tombol <i>delete</i> | Sistem akan menyembunyikan tombol <i>delete</i> | Valid |
| 16 | Kasus uji lihat tempat bekerja | Sistem akan menampilkan daftar tempat bekerja | Sistem akan menampilkan daftar tempat bekerja | Valid |
| 17 | Kasus uji tambah tempat bekerja | Sistem akan menampilkan daftar tempat bekerja | Sistem akan menampilkan daftar tempat bekerja | Valid |
| 18 | Kasus uji tambah tempat bekerja alur alternatif 1 (kolom tempat bekerja tidak diisi) | Sistem akan menampilkan pesan kesalahan | Sistem akan menampilkan pesan kesalahan | Valid |
| 19 | Kasus uji pilih tempat bekerja | Sistem akan menampilkan profil pengguna | Sistem akan menampilkan profil pengguna | Valid |
| 20 | Kasus uji lihat profil | Sistem akan menampilkan profil pengguna | Sistem akan menampilkan profil pengguna | Valid |
| 21 | Kasus uji lihat konten keperawatan | Sistem akan menampilkan daftar konten keperawatan | Sistem akan menampilkan daftar konten keperawatan | Valid |
| 22 | Kasus uji lihat detail konten keperawatan | Sistem akan menampilkan detail konten keperawatan | Sistem akan menampilkan detail konten keperawatan | Valid |

5.1.3. Pengujian Non-Fungsional

Pengujian non-fungsional dilakukan berdasarkan kebutuhan non-fungsional yang telah dijelaskan pada bab sebelumnya.

5.1.3.1. Pengujian Compatibility

Pengujian *compatibility* dilakukan untuk memastikan bahwa Sistem Buku Saku Perawat dapat berjalan sesuai dengan fungsinya.

1. Kasus Uji Pengujian *Compatibility*

Berikut ini merupakan kasus uji dari pengujian *compatibility* yang akan dijelaskan pada Tabel 5.37, Tabel 5.38, Tabel 5.39, Tabel 5.40, dan Tabel 5.41.

Tabel 5.37. Kasus uji *compatibility* Android 4.0.3 (Ice Cream Sandwich)

| | |
|------------------------------|---|
| Nama Kasus Uji | Kasus uji <i>compatibility</i> Android 4.0.3 |
| Objek Uji | Kebutuhan Non-Fungsional (KNF_01) |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan bahwa sistem dapat berjalan sesuai dengan fungsinya pada perangkat Android 4.0.3. |
| Prosedur Uji | <ol style="list-style-type: none"> 1. Menginstall aplikasi pada perangkat Android 4.0.3 2. Menjalankan setiap fitur yang disediakan aplikasi pada perangkat Android 4.0.3 |
| Hasil Yang Diharapkan | Sistem dapat berjalan sesuai dengan fungsinya pada perangkat Android 4.0.3 |

Tabel 5.38. Kasus uji *compatibility* Android 4.1.2 (Jelly Bean)

| | |
|------------------------------|---|
| Nama Kasus Uji | Kasus uji <i>compatibility</i> Android 4.1.2 |
| Objek Uji | Kebutuhan Non-Fungsional (KNF_01) |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan bahwa sistem dapat berjalan sesuai dengan fungsinya pada perangkat Android 4.1.2. |
| Prosedur Uji | <ol style="list-style-type: none"> 1. Menginstall aplikasi pada perangkat Android 4.1.2 2. Menjalankan setiap fitur yang disediakan aplikasi pada perangkat Android 4.1.2 |
| Hasil Yang Diharapkan | Sistem dapat berjalan sesuai dengan fungsinya pada perangkat Android 4.1.2 |

Tabel 5.39. Kasus uji *compatibility* Android 4.2.2 (Jelly Bean)

| | |
|------------------------------|---|
| Nama Kasus Uji | Kasus uji <i>compatibility</i> Android 4.2.2 |
| Objek Uji | Kebutuhan Non-Fungsional (KNF_01) |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan bahwa sistem dapat berjalan sesuai dengan fungsinya pada perangkat Android 4.2.2. |
| Prosedur Uji | <ol style="list-style-type: none"> 1. Menginstall aplikasi pada perangkat Android 4.2.2 2. Menjalankan setiap fitur yang disediakan aplikasi pada perangkat Android 4.2.2 |
| Hasil Yang Diharapkan | Sistem dapat berjalan sesuai dengan fungsinya pada perangkat Android 4.2.2 |

Tabel 5.40. Kasus uji *compatibility* Android 4.3 (Jelly Bean)

| | |
|------------------------------|---|
| Nama Kasus Uji | Kasus uji <i>compatibility</i> Android 4.3 |
| Objek Uji | Kebutuhan Non-Fungsional (KNF_01) |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan bahwa sistem dapat berjalan sesuai dengan fungsinya pada perangkat Android 4.3. |
| Prosedur Uji | <ol style="list-style-type: none"> 1. Menginstall aplikasi pada perangkat Android 4.3 2. Menjalankan setiap fitur yang disediakan aplikasi pada perangkat Android 4.3 |
| Hasil Yang Diharapkan | Sistem dapat berjalan sesuai dengan fungsinya pada perangkat Android 4.3 |

Tabel 5.41. Kasus uji *compatibility* Android 4.4.2 (KitKat)

| | |
|------------------------------|---|
| Nama Kasus Uji | Kasus uji <i>compatibility</i> Android 4.4.2 |
| Objek Uji | Kebutuhan Non-Fungsional (KNF_01) |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan bahwa sistem dapat berjalan sesuai dengan fungsinya pada perangkat Android 4.4.2. |
| Prosedur Uji | <ol style="list-style-type: none"> 1. Menginstall aplikasi pada perangkat Android 4.4.2 2. Menjalankan setiap fitur yang disediakan aplikasi pada perangkat Android 4.4.2 |
| Hasil Yang Diharapkan | Sistem dapat berjalan sesuai dengan fungsinya pada perangkat Android 4.4.2 |

2. Hasil Pengujian *Compatibility*

Berikut ini merupakan hasil pengujian *compatibility* pada Sistem Buku Saku Perawat yang akan dijelaskan pada Tabel 5.42.

Tabel 5.42. Hasil pengujian *compatibility*

| No | Nama Kasus Uji | Hasil Yang Diharapkan | Hasil Yang Didapatkan | Validitas |
|----|--|---|---|-----------|
| 1 | Kasus uji <i>compatibility</i> Android 4.0.3 | Sistem dapat bejalan sesuai dengan fungsinya pada perangkat Android 4.0.3 | Sistem dapat bejalan sesuai dengan fungsinya pada perangkat Android 4.0.3 | Valid |
| 2 | Kasus uji <i>compatibility</i> Android 4.1.2 | Sistem dapat bejalan sesuai dengan fungsinya pada perangkat Android 4.1.2 | Sistem dapat bejalan sesuai dengan fungsinya pada perangkat Android 4.1.2 | Valid |
| 3 | Kasus uji <i>compatibility</i> Android 4.2.2 | Sistem dapat bejalan sesuai dengan fungsinya pada perangkat Android 4.2.2 | Sistem dapat bejalan sesuai dengan fungsinya pada perangkat Android 4.2.2 | Valid |
| 4 | Kasus uji <i>compatibility</i> Android 4.3 | Sistem dapat bejalan sesuai dengan fungsinya pada perangkat Android 4.3 | Sistem dapat bejalan sesuai dengan fungsinya pada perangkat Android 4.3 | Valid |
| 5 | Kasus uji <i>compatibility</i> Android 4.4.2 | Sistem dapat bejalan sesuai dengan fungsinya pada perangkat Android 4.4.2 | Sistem dapat bejalan sesuai dengan fungsinya pada perangkat Android 4.4.2 | Valid |

5.1.3.2. Pengujian *Security*

Pengujian *security* dilakukan untuk menjamin bahwa beberapa fitur sistem, seperti menu catatan dan profil hanya dapat diakses oleh pengguna yang telah terdaftar dan *login* pada sistem.

1. Kasus Uji Pengujian *Security*

Berikut ini merupakan kasus uji dari pengujian *security* pada sistem yang akan dijelaskan pada Tabel 5.43. dan Tabel 5.44.

Tabel 5.43. Kasus uji *security* menu catatan

| | |
|------------------------------|--|
| Nama Kasus Uji | Kasus uji <i>security</i> menu catatan |
| Objek Uji | Kebutuhan Non-Fungsional (KNF_02) |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan bahwa menu catatan hanya dapat diakses oleh pengguna yang terdaftar dan <i>login</i> pada sistem. |
| Prosedur Uji | <ol style="list-style-type: none"> 1. Membuka aplikasi 2. Memilih menu catatan 3. Sistem melakukan pengecekan <i>session</i> pengguna 4. Jika <i>session</i> ada, maka sistem akan menampilkan daftar catatan pengguna. Jika <i>session</i> tidak ada, maka sistem akan menampilkan halaman <i>login</i> |
| Hasil Yang Diharapkan | Menu catatan hanya dapat diakses oleh pengguna yang terdaftar dan <i>login</i> pada sistem |

Tabel 5.44. Kasus uji *security* menu profil

| | |
|------------------------------|---|
| Nama Kasus Uji | Kasus uji <i>security</i> menu profil |
| Objek Uji | Kebutuhan Non-Fungsional (KNF_02) |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan bahwa menu profil hanya dapat diakses oleh pengguna yang terdaftar dan <i>login</i> pada sistem. |
| Prosedur Uji | <ol style="list-style-type: none"> 1. Membuka aplikasi 2. Memilih menu profil 3. Sistem melakukan pengecekan <i>session</i> pengguna 4. Jika <i>session</i> ada, maka sistem akan menampilkan profil pengguna. Jika <i>session</i> tidak ada, maka sistem akan menampilkan halaman <i>login</i> |
| Hasil Yang Diharapkan | Menu profil hanya dapat diakses oleh pengguna yang terdaftar dan <i>login</i> pada sistem |

2. Hasil Pengujian *Security*

Berikut ini merupakan hasil pengujian *security* yang akan dijelaskan pada Tabel 5.45.

Tabel 5.45. Hasil pengujian *security*

| No | Nama Kasus Uji | Hasil Yang Diharapkan | Hasil Yang Didapatkan | Validitas |
|----|--|--|--|-----------|
| 1 | Kasus uji <i>security</i> menu catatan | Menu catatan hanya dapat diakses oleh pengguna yang terdaftar dan <i>login</i> pada sistem | Menu catatan hanya dapat diakses oleh pengguna yang terdaftar dan <i>login</i> pada sistem | Valid |
| 2 | Kasus uji <i>security</i> menu profil | Menu profil hanya dapat diakses oleh pengguna yang terdaftar dan <i>login</i> pada sistem | Menu profil hanya dapat diakses oleh pengguna yang terdaftar dan <i>login</i> pada sistem | Valid |

5.2. Analisis Hasil Pengujian

Analisis hasil pengujian dilakukan untuk mendapatkan kesimpulan dari hasil pengujian sistem yang telah dilakukan. Analisis dilakukan untuk setiap pengujian yang telah dilakukan, yaitu analisis hasil pengujian unit, validasi, dan non-fungsional.

5.2.1. Analisis Hasil Pengujian Unit

Proses analisis hasil pengujian unit dilakukan dengan melihat kesesuaian fungsi dari implementasi sistem dengan hasil pengujian unit yang telah dilakukan. Berdasarkan hasil pengujian unit, implementasi sistem sudah sesuai dengan hasil yang diharapkan. Oleh karena itu, dapat diambil kesimpulan bahwa Sistem Buku Saku Perawat telah memenuhi setiap fungsionalitas yang telah dirancang pada bab sebelumnya.

5.2.2. Analisis Hasil Pengujian Validasi

Proses analisis hasil pengujian validasi dilakukan dengan melihat kesesuaian kinerja sistem dengan daftar kebutuhan fungsional sistem. Berdasarkan hasil pengujian validasi, kinerja sistem sudah sesuai dengan setiap kebutuhan fungsional sistem. Oleh karena itu, dapat diambil kesimpulan bahwa Sistem Buku

Saku Perawat telah memenuhi setiap kebutuhan fungsional sistem yang telah dijelaskan pada bab sebelumnya.

5.2.3. Analisis Hasil Pengujian Non-Fungsional

Proses analisis hasil pengujian non-fungsional dilakukan dengan mencoba menjalankan sistem pada perangkat bergerak berbasis Android 4.0.3 keatas dan menjamin pengguna yang tidak memiliki akun tidak dapat mengakses fitur sistem. Berdasarkan hasil pengujian non-fungsional, sistem dapat dijalankan pada perangkat bergerak berbasis Android 4.0.3 keatas dan beberapa fitur sistem, seperti menu catatan dan profil hanya dapat diakses oleh pengguna yang telah terdaftar dan *login*. Oleh karena itu, dapat disimpulkan bahwa Sistem Buku Saku Perawat telah memenuhi setiap kebutuhan non-fungsional yang telah dijelaskan pada bab sebelumnya.

