

BAB V

HASIL DAN PEMBAHASAN

5.1. Hasil

Sistem Seleksi Calon Penerima Bantuan Sosial untuk panti jompo Menggunakan Metode SAW berhasil dirancang, dengan penggunaan data panti jompo sebanyak 23 set data, 5 set data kriteria, dan 25 set data crips. Sistem dapat mengolah data dan memberikan data rekomendasi calon panti jompo untuk diberikan bantuan sosial berdasarkan perhitungan prioritas dari nilai kriteria dan besaran bobotnya. Selain itu, aplikasi memiliki beberapa fitur untuk memudahkan penggunaanya dalam melihat maupun mengelola data panti jompo.

5.2. Pembahasan

Berdasarkan perhitungan menggunakan metode SAW, berikut data alternatif panti jompo yang telah diurutkan berdasarkan *ranking* prioritas dari bobot kriteria.

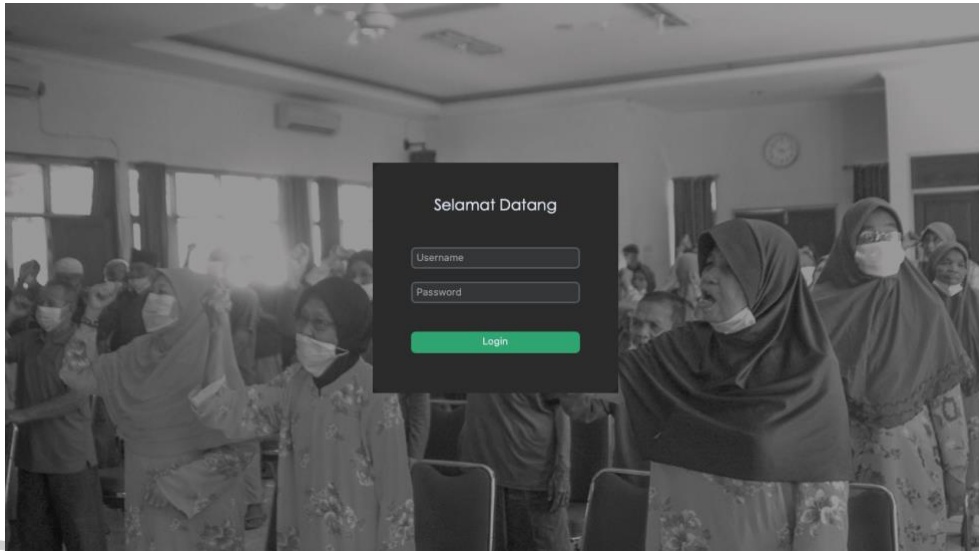
Tabel 5.1. Data Rekomendasi Alternatif

Kode Alternatif	Nama Panti Jompo	Jumlah Lansia	Perawatan Kesehatan / Tahun	Program Rutin / Minggu	Biaya Bulanan	Rata-Rata Bantuan Diterima / Tahun
A4	Panti Jompo D	358	0 (Kerjasama dengan Puskesmas)	6 - 10 kali	Rp 0,- (Gratis)	12 – 23 kali
A6	Panti Jompo F	52	0 (Kerjasama dengan Puskesmas)	11 - 15 kali	Rp 0,- (Gratis)	< 12 kali
A17	Panti Jompo Q	35	≤ 12 kali	≥ 5 kali	Rp 0,- (Gratis)	0 – 11 kali
A16	Panti Jompo P	30	0 (Kerjasama dengan Puskesmas)	6 – 10 kali	Rp 0,- (Sukarela)	0 – 11 kali
A1	Panti Jompo A	250	0 (Kerjasama dengan Puskesmas)	11 – 15 kali	Rp 0,- (Gratis)	11 – 23 kali

A21	Panti Jompo U	71	>12 kali	≤ 5 kali	Rp 1.500.000,-	0 – 11 kali
A18	Panti Jompo R	15	Memiliki staff medis menetap	6 – 10 kali	Rp 0,- (Sukarela)	0 – 11 kali
A7	Panti Jompo G	200	0 (Kerjasama dengan Puskesmas)	11 – 15 kali	Rp 0,- (Gratis)	12 – 23 kali
A3	Panti Jompo C	339	>12 kali	11- 15 kali	Rp 0,- (Gratis)	12 – 23 kali
A12	Panti Jompo L	53	≤ 12 kali	6 – 10 kali	Rp 2.000.000,-	0 – 11 kali
A2	Panti Jompo B	156	12 kali	11 - 15 kali	Rp 0,- (Gratis)	12 – 23 kali
A13	Panti Jompo M	27	0 (Tidak ada kerjasama)	6 – 10 kali	Rp 2.000.000,-	12 – 23 kali
A20	Panti Jompo T	380	≥ 12 kali	16 – 20 kali	Rp 0,- (Gratis)	≥ 48 kali
A5	Panti Jompo E	360	Memiliki staff medis menetap	11 - 15 kali	Rp 0,- (Gratis)	≥ 48 kali
A19	Panti Jompo S	50	≥ 12 kali	11- 15 kali	Rp 0,- (Gratis)	11 – 23 kali
A8	Panti Jompo H	25	12 kali	16 + 20 kali	Rp 0,- (Gratis)	12 – 23 kali
A10	Panti Jompo J	30	0 (Kerjasama dengan Puskesmas)	6 – 10 kali	Rp 0,- (Gratis)	36 – 47 kali
A15	Panti Jompo O	106	Memiliki staff medis menetap	11 – 15 kali	Rp 1.500.000,-	12 – 23 kali
A9	Panti Jompo I	36	≤ 12 kali	6 – 10 kali	Rp 0,- (Gratis)	24 – 35 kali
A23	Panti Jompo W	16	>12 kali	6 – 10 kali	Rp 0,- (Sukarela)	≥ 48 kali
A22	Panti Jompo V	52	≥ 12 kali, memiliki staff medis	>21 kali	Rp 4.000.000,-	11 – 23 kali
A11	Panti Jompo K	33	Memiliki staff medis menetap	11 – 15 kali	Rp 6.000.000,-	12 – 23 kali
A14	Panti Jompo N	8	>12 kali	>21 kali	Rp 3.500.000,-	12 – 23 kali

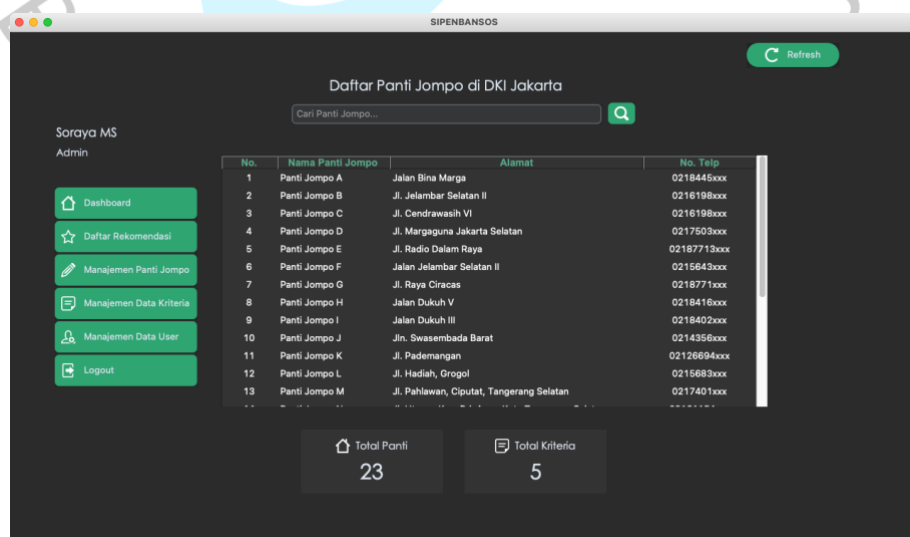
5.2.1. Hasil Perancangan Tampilan

Tampilan aplikasi *desktop* dari sistem yang dirancang turut mempertimbangkan prinsip-prinsip *usability* dan *user experience*, dengan begitu diharapkan tampilan antarmuka menjadi lebih intuitif dan mudah digunakan.



Gambar 5.1. Tampilan Halaman Login

Gambar 5.1. merupakan tampilan halaman *login* untuk pengguna dapat mengakses aplikasi. Terdapat dua kotak yang dapat dilakukan *input*, yakni '*username*' dan '*password*' serta tombol untuk melanjutkan proses *login*.



Gambar 5.2. Tampilan Aplikasi Halaman Dashboard

Ketika pengguna berhasil melakukan *login*, maka user akan mengakses tampilan ‘Dashboard’. Nama pengguna dan *role* pengguna maka akan ditampilkan pada bagian sisi kiri halaman. Selain itu, pengguna dapat melakukan navigasi ke menu lainnya dengan mengklik tombol yang tersedia. Fokus utama halaman ini adalah pengguna dapat melihat data panti jompo yang sudah terintegrasi dengan *database* dalam sebuah tabel. Pada bagian sisi bawah halaman, terdapat total jumlah angka data panti jompo dan kriteria yang tercatat dalam sistem. Fitur lainnya yang tersedia pada halaman ini adalah kotak pencarian nama panti jompo yang terletak pada bagian atas halaman dan tombol *refresh* untuk memperbaharui halaman.

Rank	Nama Panti Jompo	Alamat	No. Telp	Jumlah Lansia	
1	Panti Jompo D	Jl. Margaguna Jakarta Selatan	0217503xxx	> 200 orang	Tidak ada kun
2	Panti Jompo F	Jalan Jelambar Selatan II	0215643xxx	51 - 100 orang	Tidak ada kun
3	Panti Jompo Q	Jl. Rumah Sakit Ancol, kel. Sunter Agung, kec. T.	081350206xxx	11 - 50 orang	Kunjungan do
4	Panti Jompo P	Jl. Kincir Raya, Kel. Cengkareng Tim., Kecamatan	081294214xxx	11 - 50 orang	Tidak ada kun
5	Panti Jompo A	Jalan Bina Marga	0218445xxx	> 200 orang	Tidak ada kun
6	Panti Jompo U	Jl. Masda, kel. Pejagalan, kec. Penjaringan, Jaka	0216616xxx	51 - 100 orang	Kunjungan do
7	Panti Jompo R	Jl. Swadaya I, Kel. Semper Timur, Kec. Cilincing,	0818975xxx	11 - 50 orang	Memiliki staff
8	Panti Jompo G	Jl. Raya Ciracas	0218771xxx	101 - 200 orang	Tidak ada kun
9	Panti Jompo C	Jl. Cendrawasih VI	0216198xxx	> 200 orang	Kunjungan do
10	Panti Jompo L	Jl. Hadiah, Grogol	0215683xxx	51 - 100 orang	Kunjungan do
11	Panti Jompo B	Jl. Jelambar Selatan II	0216198xxx	101 - 200 orang	Kunjungan do
12	Panti Jompo M	Jl. Pahlawan, Ciputat, Tangerang Selatan	0217401xxx	11 - 50 orang	Tidak ada kun
13	Panti Jompo T	Jl. Puri Kembangan, Kel. Kembangan Selatan, Ke	0215406xxx	> 200 orang	Kunjungan do
14	Panti Jompo E	Jl. Radio Dalam Raya	02187713xxx	> 200 orang	Memiliki staff
15	Panti Jompo S	Jl. Puri Kembangan, kel. Kedoya Selatan, Kec. KI	0215814xxx	11 - 50 orang	Kunjungan do
16	Panti Jompo H	Jalan Dukuh V	0218416xxx	11 - 50 orang	Kunjungan do
17	Panti Jompo J	Jln. Swasembada Barat	0214356xxx	11 - 50 orang	Tidak ada kun

Gambar 5.3. Tampilan Aplikasi Halaman Daftar Rekomendasi

Ketika pengguna memilih menu ‘Daftar Rekomendasi’, maka *user* akan diarahkan ke menu tersebut, seperti yang terlihat pada Gambar 5.3. Pada halaman ini, pengguna dapat melihat seluruh data panti jompo yang sudah dilakukan pengurutan berdasarkan *ranking* prioritas penerima bantuan dengan menggunakan algoritma *Simple Additive Weighting* (SAW) yang diolah oleh sistem dan juga informasi lengkap masing-masing kriteria pada tiap data panti jompo.

SIPENBANSOS

Refresh

Manajemen Data Panti Jompo

Cari Panti Jompo...

Soraya MS
Admin

- Dashboard
- Daftar Rekomendasi
- Manajemen Panti Jompo
- Manajemen Data Kriteria
- Manajemen Data User
- Logout

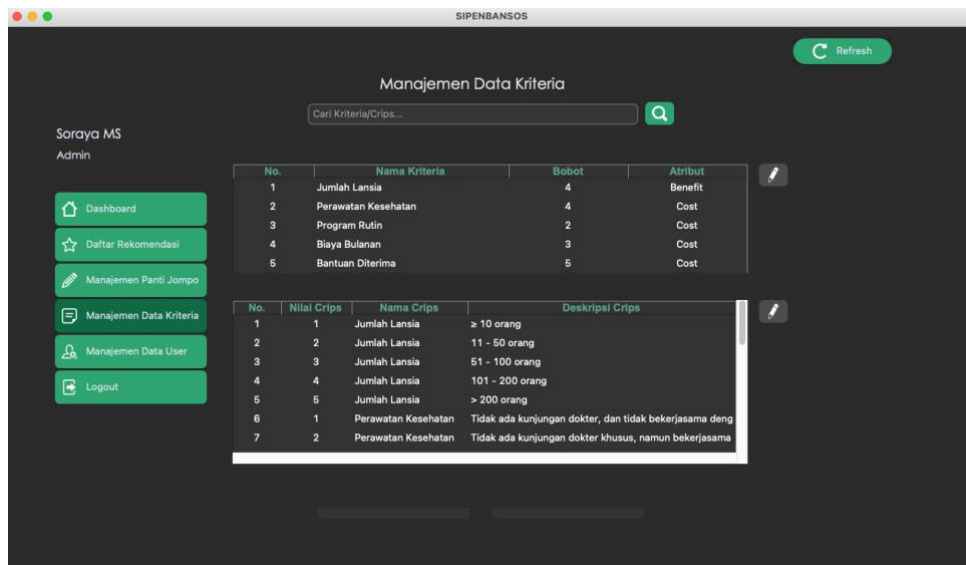
No.	Nama Panti Jompo	Alamat	No. Telp	Jumlah Lansia	
1	Panti Jompo A	Jalan Bina Marga	0218445xxx	> 200 orang	Ti
2	Panti Jompo B	Jl. Jelambar Selatan II	0216198xxx	101 - 200 orang	Ku
3	Panti Jompo C	Jl. Cendrawasih VI	0216198xxx	> 200 orang	Ku
4	Panti Jompo D	Jl. Margaguna Jakarta Selatan	0217503xxx	> 200 orang	Ti
5	Panti Jompo E	Jl. Radio Dalam Raya	02187713xxx	> 200 orang	Me
6	Panti Jompo F	Jalan Jelambar Selatan II	0215643xxx	51 - 100 orang	Ti
7	Panti Jompo G	Jl. Raya Ciracas	0218771xxx	101 - 200 orang	Ti
8	Panti Jompo H	Jalan Dukuh V	0218416xxx	11 - 50 orang	Ku
9	Panti Jompo I	Jalan Dukuh III	0218402xxx	11 - 50 orang	Ku
10	Panti Jompo J	Jln. Swasembada Barat	0214356xxx	11 - 50 orang	Ti
11	Panti Jompo K	Jl. Pademangan	02126694xxx	11 - 50 orang	Me
12	Panti Jompo L	Jl. Hadiah, Grogol	0215683xxx	51 - 100 orang	Ku
13	Panti Jompo M	Jl. Pahlawan, Ciputat, Tangerang Selatan	0217401xxx	11 - 50 orang	Ti
14	Panti Jompo N	Jl. Utama, Kec. Pd. Aren, Kota Tangerang Selata	08161154xxx	≥ 10 orang	Ku
15	Panti Jompo O	Jl. Kramat Kwitang, Kec. Senen, Kota Jakarta Pu	0217423xxx	101 - 200 orang	Me
16	Panti Jompo P	Jl. Kincir Raya, Kel. Cengkareng Tim., Kecamatan	081294214xxx	11 - 50 orang	Ti
17	Panti Jompo Q	Jl. Rumah Sakit Ancol, kel. Sunter Agung, kec. T.	081350206xxx	11 - 50 orang	Ku

Gambar 5.4. Tampilan Halaman Manajemen Panti Jompo

Pada Gambar 5.4. adalah tampilan ketika seorang admin memilih menu ‘Manajemen Panti Jompo’. Pada halaman ini, sistem akan menampilkan tabel data panti jompo sebanyak data yang tercatat pada basis data. Data yang ditampilkan mencakup informasi lengkap, yakni nilai masing-masing kategori kriteria untuk setiap data panti jompo. Selain tabel, halaman ini memiliki tiga opsi yang digambarkan melalui *icons* yang terletak pada bagian sebelah kanan tabel. Opsi ini berfungsi untuk seorang admin dalam melakukan pengelolaan data panti jompo. Admin dapat melakukan penambahan, edit, ataupun menghapus data. Apabila admin melakukan perubahan pada data panti jompo, maka sistem akan memperbaharui data yang diubah secara langsung pada basis data. Selain pengelolaan data, terdapat pula pilihan *refresh* halaman dengan menggunakan tombol yang terletak pada sisi kanan atas halaman. *Refresh* dipergunakan untuk menampilkan data yang terbaru pada halaman apabila ada perubahan yang dilakukan.

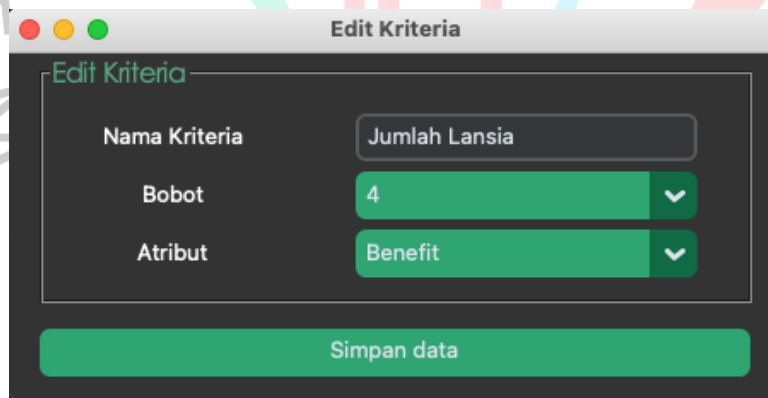
Gambar 5.5. Tampilan Penambahan Data Panti Jompo

Gambar 5.5. adalah tampilan ketika admin melakukan klik fitur tambah data panti jompo. Sistem akan menampilkan jendela baru untuk mengarahkan admin mengisi data lengkap panti jompo. Data yang dibutuhkan oleh sistem meliputi: nama panti jompo, alamat, no. telp, dan data informasi kriteria dengan opsi crips yang telah terintegrasi dengan basis data. Apabila data panti jompo belum lengkap, maka sistem akan menampilkan pesan notifikasi untuk melengkapi data. Namun apabila seluruh data sudah terisi, maka data akan disimpan oleh sistem. Data tersebut kemudian akan secara otomatis masuk ke dalam perhitungan SAW dan masuk ke dalam tabel daftar rekomendasi panti jompo sesuai dengan *ranking* prioritasnya.



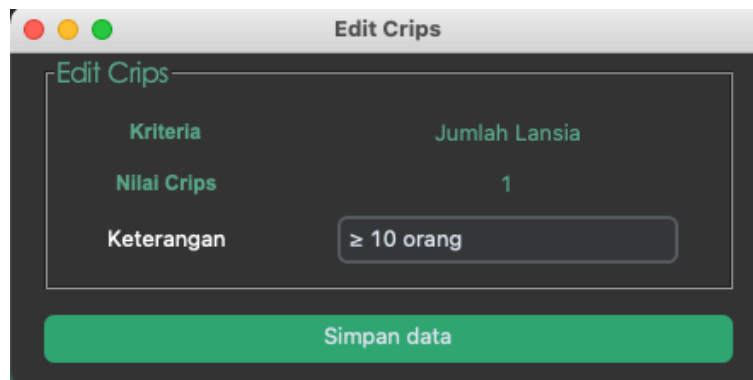
Gambar 5.6. Tampilan Aplikasi Halaman Manajemen Data Kriteria

Pada halaman 'Data Kriteria' sistem akan menampilkan dua tabel seperti pada Gambar 5.6., yaitu tabel kriteria, dan tabel crips. Admin dapat melakukan perubahan data dengan mengklik tombol edit yang terletak di samping kanan tabel.



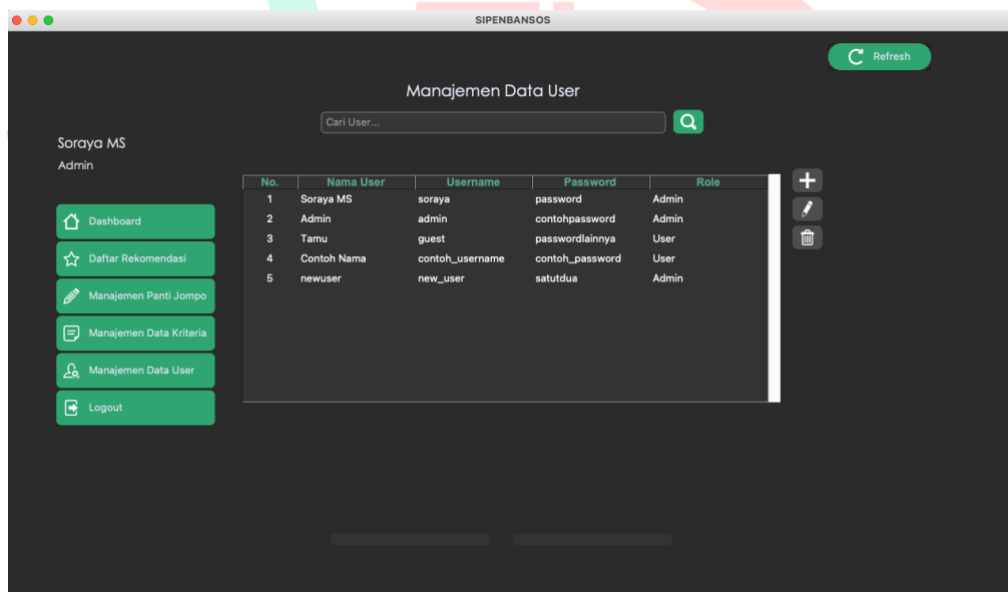
Gambar 5.7. Tampilan Aplikasi Edit Data Kriteria

Gambar 5.7. merupakan tampilan ketika seorang admin ingin melakukan perubahan terhadap kriteria yang dipilih. Ketika ingin melakukan perubahan, admin hanya dapat melakukan input nama, pemilihan opsi bobot dari 1-5, dan pemilihan jenis atribut 'cost' atau 'benefit'.



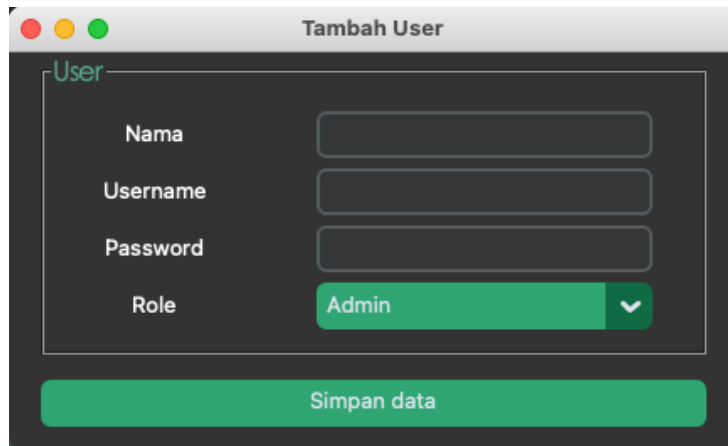
Gambar 5.8. Tampilan Aplikasi Edit Crips

Gambar 5.8. adalah tampilan ketika seorang admin ingin melakukan perubahan terhadap deskripsi yang dipilih. Maka sistem akan menampilkan nama kriteria dan nilai crips yang dipilih untuk dilakukan perubahan, serta teks deskripsi yang dapat diedit.



Gambar 5.9. Tampilan Aplikasi Halaman Manajemen User

Pada halaman 'Manajemen User', admin dapat melihat tabel yang berisikan nama, username, password, dan role pengguna. Selain itu admin dapat mengelola data dengan melakukan penambahan, edit, atau delete dengan melakukan interaksi pada icon yang terdapat di sisi kanan tabel.



Gambar 5.10. Tampilan Aplikasi Tambah User

Gambar 5.10. adalah tampilan ketika seorang admin memilih tombol tambah, maka admin dapat menginput data user baru yang memiliki hak untuk mengakses aplikasi.

5.2.2. Hasil Pengujian

Setelah aplikasi berhasil dibangun, maka dibutuhkan pengujian kembali untuk memastikan bahwa baik fungsionalitas aplikasi maupun logika internal aplikasi berjalan dengan baik dan sesuai dengan kebutuhan serta spesifikasi yang telah ditetapkan.

Pengujian Kotak Putih

Pengujian kotak putih dilakukan untuk memastikan implementasi kode dan logika pada aplikasi sudah berjalan sesuai dengan desain yang telah ditentukan di awal.

Tabel 3.2. Hasil Pengujian Kotak Putih

No	Algoritma	Perancangan Code
1	Normalisasi bobot kriteria	<pre>def normalisasiBobot(bobot): tBobot = 0 nBobot = [] for i in range(0, len(bobot)): tBobot = tBobot + bobot[i] for i in range(0, len(bobot)): nBobot.insert(i, bobot[i]/tBobot) return nBobot nBobot = normalisasiBobot(bobot)</pre>

```
print("bobot: ", bobot)
print("bobot ternormalisasi: ", nBobot)
```

Hasil

```
/usr/local/bin/python3.8 /Users/sorayams/AndroidStudioProjects/UAS/TA/a.py
bobot: [4, 4, 2, 3, 5]
bobot ternormalisasi: [0.2222222222222222, 0.2222222222222222, 0.1111111111111111, 0.1666666666666666, 0.2777777777777778]

Process finished with exit code 0
```

Pembahasan

Hasil perhitungan normalisasi bobot sudah sesuai dengan perhitungan secara manual. Total nilai normalisasi bobot apabila dijumlahkan harus sama dengan 1.

Tabel 5.3. Nilai Bobot Kriteria

Kode Kriteria	Nama Kriteria	Bobot	Bobot Ternormalisasi
C1	Jumlah Lansia	4	4/18 = 0,222
C2	Perawatan Kesehatan	4	4/18 = 0,222
C3	Program Rutin	2	2/18 = 0,111
C4	Biaya Bulanan	3	3/18 = 0,167
C5	Bantuan Diterima	5	5/18 = 0,278

2	Normalisasi matriks keputusan	<pre>def getNormalizeScore(criteria_type, scores, pembagi): result = [] print(scores) for j, score in enumerate(scores): result_point = [] for i in range(len(score)): normalized_point = None if criteria_type[i]=='benefit': normalized_point = score[i]/pembagi[i] else: normalized_point = pembagi[i]/score[i] result_point.append(normalized_point) result.append(result_point) print("Normalize Score",result) return result pembagi = getPembagi(criteria_type, scores) normalized_scores = getNormalizeScore(criteria_type, scores, pembagi) print("\n\nvektor ternormalisasi: ", normalized_scores)</pre>
---	-------------------------------	--

Hasil

Vektor ternormalisasi: $[[1.0, 0.5, 0.3333333333333333, 1.0, 0.5], [0.8, 0.3333333333333333, 0.3333333333333333, 1.0, 0.5],$

$, [1.0, 0.5, 1.0, 1.0, 0.5], [1.0, 0.2, 0.3333333333333333, 1.0, 0.2], [0.6, 0.5, 0.3333333333333333, 1.0, 1.0], [0.8, 0.5, 0.3333333333333333, 1.0, 0.5],$

dan seterusnya.

Pembahasan

Hasil perhitungan sistem sesuai dengan perhitungan manual dalam melakukan normalisasi matriks keputusan, di mana C1 adalah kriteria *benefit*, maka perhitungannya menggunakan rumusan:

$$R_{ii} = \left(\frac{x_{ij}}{\max\{x_{ij}\}} \right) \quad (5.1)$$

Karena nilai maksimal dari kolom C1 adalah 5, maka nilai dari tiap baris kolom C1 dilakukan pembagian oleh nilai maksimal tersebut, seperti berikut:

$$R_{11} = \frac{5}{\max\{5,4,5,5,5,3,4,2,2,2,2,3,2,1,4,2,2,2,2,5,3,3,2\}} = \frac{5}{5} = 1$$

$$R_{21} = \frac{4}{\max\{5,4,5,5,5,3,4,2,2,2,2,3,2,1,4,2,2,2,2,5,3,3,2\}} = \frac{4}{5} = 0,8$$

$$R_{31} = \frac{5}{\max\{5,4,5,5,5,3,4,2,2,2,2,3,2,1,4,2,2,2,2,5,3,3,2\}} = \frac{5}{5} = 1$$

$$R_{41} = \frac{5}{\max\{5,4,5,5,5,3,4,2,2,2,2,3,2,1,4,2,2,2,2,5,3,3,2\}} = \frac{5}{5} = 1$$

$$R_{51} = \frac{5}{\max\{5,4,5,5,5,3,4,2,2,2,2,3,2,1,4,2,2,2,2,5,3,3,2\}} = \frac{5}{5} = 1$$

Dan seterusnya sampai dengan $R_{23\ 1}$. Sedangkan pada kriteria C2, C3, C4, dan C5 merupakan kriteria *cost*, maka digunakan nilai minimum sebagai pembagi dari tiap baris kolom kriteria. Berikut perhitungan pada kolom C2:

$$R_{ii} = \left(\frac{\min_i x_{ij}}{x_{ij}} \right) \quad (5.2)$$

$$R_{12} = \frac{\min\{2,3,4,2,5,2,2,3,3,2,5,3,1,4,5,2,3,5,3,3,4,5,4\}}{2} = \frac{1}{2} = 0,5$$

$$R_{22} = \frac{\min\{2,3,4,2,5,2,2,3,3,2,5,3,1,4,5,2,3,5,3,3,4,5,4\}}{3} = \frac{1}{3} = 0,33$$

$$R_{32} = \frac{\min\{2,3,4,2,5,2,2,3,3,2,5,3,1,4,5,2,3,5,3,3,4,5,4\}}{4} = \frac{1}{4} = 0,25$$

$$R_{42} = \frac{\min\{2,3,4,2,5,2,2,3,3,2,5,3,1,4,5,2,3,5,3,3,4,5,4\}}{2} = \frac{1}{2} = 0,5$$

$$R_{52} = \frac{\min\{2,3,4,2,5,2,2,3,3,2,5,3,1,4,5,2,3,5,3,3,4,5,4\}}{5} = \frac{1}{5} = 0,2$$

Dan seterusnya sampai dengan $R_{23\ 2}$. Pada kriteria C3, dilakukan perhitungan dengan rumusan yang sama:

$$R_{13} = \frac{\min\{3,3,3,1,3,3,3,4,2,2,3,2,2,5,3,2,1,2,3,4,1,5,2\}}{3} = \frac{1}{3} = 0,33$$

$$R_{23} = \frac{\min\{3,3,3,1,3,3,3,4,2,2,3,2,2,5,3,2,1,2,3,4,1,5,2\}}{3} = \frac{1}{3} = 0,33$$

$$R_{33} = \frac{\min\{3,3,3,1,3,3,3,4,2,2,3,2,2,5,3,2,1,2,3,4,1,5,2\}}{3} = \frac{1}{3} = 0,33$$

$$R_{43} = \frac{\min\{3,3,3,1,3,3,3,4,2,2,3,2,2,5,3,2,1,2,3,4,1,5,2\}}{1} = \frac{1}{1} = 1$$

$$R_{53} = \frac{\min\{3,3,3,1,3,3,3,4,2,2,3,2,2,5,3,2,1,2,3,4,1,5,2\}}{3} = \frac{1}{3} = 0,33$$

Dan seterusnya sampai dengan $R_{23\ 3}$. Pada kriteria C4, masih dilakukan perhitungan dengan rumusan yang sama:

$$R_{14} = \frac{\min\{1,1,1,1,1,1,1,1,1,5,3,3,3,2,1,1,1,1,1,2,4,1\}}{1} = \frac{1}{1} = 1$$

$$R_{24} = \frac{\min\{1,1,1,1,1,1,1,1,1,5,3,3,3,2,1,1,1,1,1,2,4,1\}}{1} = \frac{1}{1} = 1$$

$$R_{34} = \frac{\min\{1,1,1,1,1,1,1,1,1,5,3,3,3,2,1,1,1,1,1,2,4,1\}}{1} = \frac{1}{1} = 1$$

$$R_{44} = \frac{\min\{1,1,1,1,1,1,1,1,1,5,3,3,3,2,1,1,1,1,1,2,4,1\}}{1} = \frac{1}{1} = 1$$

$$R_{54} = \frac{\min\{1,1,1,1,1,1,1,1,1,5,3,3,3,2,1,1,1,1,1,2,4,1\}}{1} = \frac{1}{1} = 1$$

Dan seterusnya sampai dengan $R_{23\ 4}$. Dan demikian juga pada kriteria C5:

$$R_{15} = \frac{\min\{2,2,2,2,5,1,2,2,3,4,2,1,2,2,2,1,1,1,2,5,1,2,5\}}{2} = \frac{1}{2} = 0,5$$

$$R_{25} = \frac{\min\{2,2,2,2,5,1,2,2,3,4,2,1,2,2,2,1,1,1,2,5,1,2,5\}}{2} = \frac{1}{2} = 0,5$$

$$R_{35} = \frac{\min\{2,2,2,2,5,1,2,2,3,4,2,1,2,2,2,1,1,1,2,5,1,2,5\}}{2} = \frac{1}{2} = 0,5$$

$$R_{45} = \frac{\min\{2,2,2,2,5,1,2,2,3,4,2,1,2,2,2,1,1,1,2,5,1,2,5\}}{2} = \frac{1}{2} = 0,5$$

$$R_{55} = \frac{\min\{2,2,2,2,5,1,2,2,3,4,2,1,2,2,2,1,1,1,2,5,1,2,5\}}{5} = \frac{1}{5} = 0,2$$

Dan seterusnya sampai dengan $R_{23\ 5}$. Maka, hasil dari seluruh perhitungan tersebut dimasukkan ke dalam tabel vektor yang telah dinormalisasi seperti tabel berikut:

Tabel 5.4. Vektor Ternormalisasi

Alternatif	C1	C2	C3	C4	C5
A1	1	0,5	0,33	1	0,5
A2	0,8	0,33	0,33	1	0,5
A3	1	0,25	0,33	1	0,5
A4	1	0,5	1	1	0,5
A5	1	0,2	0,33	1	0,2
A6	0,6	0,5	0,33	1	1
A7	0,8	0,5	0,33	1	0,5
A8	0,4	0,33	0,25	1	0,5
A9	0,4	0,33	0,5	1	0,33
A10	0,4	0,5	0,5	1	0,25
A11	0,4	0,2	0,33	0,2	0,5
A12	0,6	0,33	0,5	0,33	1
A13	0,4	1	0,5	0,33	0,5
A14	0,2	0,25	0,2	0,33	0,5
A15	0,8	0,2	0,33	0,5	0,5
A16	0,4	0,5	0,5	1	1
A17	0,4	0,33	1	1	1
A18	0,4	0,2	0,5	1	1
A19	0,4	0,33	0,33	1	0,5
A20	1	0,33	0,25	1	0,2
A21	0,6	0,25	1	0,5	1
A22	0,6	0,2	0,2	0,25	0,5
A23	0,4	0,25	0,5	1	0,2

3

Nilai preferensi

```
def getNormalizeScore(criteria_type, scores,
pembagi):
    result = []
    print(scores)
    for j, score in enumerate(scores):
        result_point = []
        for i in range(len(score)):
            normalized_point = None
            if criteria_type[i]=='benefit':
                normalized_point =
score[i]/pembagi[i]
            else:
                normalized_point =
pembagi[i]/score[i]
        result_point.append(normalized_point)
    result.append(result_point)
    print("Normalize Score", result)
    return result

criteria_type = connectDatabase("select case
atribut when '1' then 'benefit' else 'cost' end
criteria_type from kriteria order by
kriteria_id asc;")
normalized_scores =
getNormalizeScore(criteria_type, scores,
pembagi)
preference_scores = np.dot(normalized_scores,
weights)

print("Skor Preferensi: ", preference_scores)
```

Hasil

```
Skor Preferensi: [0.67592593 0.59444444 0.62037037 0.75          0.52592593 0.72592593
0.63148148 0.4962963 0.47777778 0.49166667 0.34259259 0.5962963
0.56111111 0.31666667 0.48148148 0.7          0.71851852 0.63333333
0.50555556 0.5462963 0.66111111 0.38055556 0.42222222]
```

Pembahasan

Hasil perhitungan nilai preferensi sesuai dengan perhitungan manual. Hasil nilai preferensi didapatkan setelah dilakukan pengalihan bobot kriteria yang telah ditetapkan dengan masing-masing kolom.

$$V_i = \sum_{j=1}^n W_j r_{ij} \quad (5.3)$$

$$V_1 = (0,22*1) + (0,22*0,5) + (0,11*0,33) + (0,16*1) + (0,27*0,5) = 0,68$$

$$V_2 = (0,22*0,8) + (0,22*0,33) + (0,11*0,33) + (0,16*1) + (0,27*0,5) = 0,59$$

$$V_3 = (0,22*1) + (0,22*0,25) + (0,11*0,33) + (0,16*1) + (0,27*0,5) = 0,62$$

$$V_4 = (0,22*1) + (0,22*0,5) + (0,11*1) + (0,16*1) + (0,27*0,5) = 0,75$$

$$V_5 = (0,22*1) + (0,22*0,2) + (0,11*0,33) + (0,16*1) + (0,27*0,2) = 0,53$$

$$V_6 = (0,22*0,6) + (0,22*0,5) + (0,11*0,33) + (0,16*1) + (0,27*1) = 0,73$$

$$V_7 = (0,22*0,8) + (0,22*0,5) + (0,11*0,33) + (0,16*1) + (0,27*0,5) = 0,63$$

$$V_8 = (0,22*0,4) + (0,22*0,33) + (0,11*0,25) + (0,16*1) + (0,27*0,5) = 0,50$$

$$V_9 = (0,22*0,4) + (0,22*0,33) + (0,11*0,5) + (0,16*1) + (0,27*0,33) = 0,48$$

$$V_{10} = (0,22*0,4) + (0,22*0,5) + (0,11*0,5) + (0,16*1) + (0,27*0,25) = 0,49$$

$$V_{11} = (0,22*0,4) + (0,22*0,2) + (0,11*0,33) + (0,16*0,2) + (0,27*0,5) = 0,34$$

$$V_{12} = (0,22*0,6) + (0,22*0,33) + (0,11*0,5) + (0,16*0,33) + (0,27*1) = 0,60$$

$$V_{13} = (0,22*0,4) + (0,22*1) + (0,11*0,5) + (0,16*0,33) + (0,27*2) = 0,56$$

$$V_{14} = (0,22*0,2) + (0,22*0,25) + (0,11*0,2) + (0,16*0,33) + (0,27*0,5) = 0,32$$

$$V_{15} = (0,22*0,8) + (0,22*0,2) + (0,11*0,33) + (0,16*0,5) + (0,27*0,5) = 0,48$$

$$V_{16} = (0,22*0,4) + (0,22*0,5) + (0,11*0,5) + (0,16*1) + (0,27*1) = 0,70$$

$$V_{17} = (0,22*0,4) + (0,22*0,33) + (0,11*1) + (0,16*1) + (0,27*1) = 0,72$$

$$V_{18} = (0,22*0,4) + (0,22*0,2) + (0,11*0,5) + (0,16*1) + (0,27*1) = 0,63$$

$$V_{19} = (0,22*0,4) + (0,22*0,33) + (0,11*0,33) + (0,16*1) + (0,27*0,5) = 0,51$$

$$V_{20} = (0,22*1) + (0,22*0,33) + (0,11*0,25) + (0,16*1) + (0,27*0,2) = 0,55$$

$$V_{21} = (0,22*0,6) + (0,22*0,25) + (0,11*1) + (0,16*0,5) + (0,27*1) = 0,66$$

$$V_{22} = (0,22*0,6) + (0,22*0,2) + (0,11*0,2) + (0,16*0,25) + (0,27*0,5) = 0,38$$

$$V_{23} = (0,22*0,4) + (0,22*0,25) + (0,11*0,5) + (0,16*1) + (0,27*0,2) = 0,42$$

4	Pengurutan data alternatif	<pre> query = ("SELECT panti_jompo_id, " "nama, " "alamat, " "nomor_telp, " "(SELECT deskripsi_crips FROM crips WHERE kriteria_id = 1 AND nilai_crips = jumlah_lansia) AS jumlah_lansia, " "(SELECT deskripsi_crips FROM crips WHERE kriteria_id = 2 AND nilai_crips = perawatan_kesehatan) AS perawatan_kesehatan, " "(SELECT deskripsi_crips FROM crips WHERE kriteria_id = 3 AND nilai_crips = program_rutin) AS program_rutin, " "(SELECT deskripsi_crips FROM crips WHERE kriteria_id = 4 AND nilai_crips = biaya_bulanan) AS biaya_bulanan, " "(SELECT deskripsi_crips FROM crips WHERE kriteria_id = 5 AND nilai_crips = bantuan_diterima) AS bantuan_diterima " "FROM master_panti " "ORDER BY panti_jompo_id ASC;") alternatives = connectDatabaseGetList(query) sorting_rank = np.argsort(preferance_scores)[::-1][:len(scores)] for i, top in enumerate(sorting_rank): print("Rank ", i + 1, alternatives[top][1], "Dengan Skor Preferensi : ", preferance_scores[top], "Serta Nilai Kriteria", scores[top]) return sorting_rank </pre>
---	----------------------------	---

Hasil

```

Rank 1 Panti Jompo D Dengan Skor Preferensi : 0.75 Serta Nilai Kriteria [5, 2, 1, 1, 2]
Rank 2 Panti Jompo F Dengan Skor Preferensi : 0.725925925925926 Serta Nilai Kriteria [3, 2, 3, 1, 1]
Rank 3 Panti Jompo Q Dengan Skor Preferensi : 0.7185185185185186 Serta Nilai Kriteria [2, 3, 1, 1, 1]
Rank 4 Panti Jompo P Dengan Skor Preferensi : 0.7 Serta Nilai Kriteria [2, 2, 2, 1, 1]
Rank 5 Panti Jompo A Dengan Skor Preferensi : 0.6759259259259258 Serta Nilai Kriteria [5, 2, 3, 1, 2]
Rank 6 Panti Jompo U Dengan Skor Preferensi : 0.6611111111111111 Serta Nilai Kriteria [3, 4, 1, 2, 1]
Rank 7 Panti Jompo R Dengan Skor Preferensi : 0.6333333333333333 Serta Nilai Kriteria [2, 5, 2, 1, 1]
Rank 8 Panti Jompo G Dengan Skor Preferensi : 0.6314814814814815 Serta Nilai Kriteria [4, 2, 3, 1, 2]
Rank 9 Panti Jompo C Dengan Skor Preferensi : 0.6203703703703703 Serta Nilai Kriteria [5, 4, 3, 1, 2]
Rank 10 Panti Jompo L Dengan Skor Preferensi : 0.5962962962962963 Serta Nilai Kriteria [3, 3, 2, 3, 1]
Rank 11 Panti Jompo B Dengan Skor Preferensi : 0.5944444444444444 Serta Nilai Kriteria [4, 3, 3, 1, 2]
Rank 12 Panti Jompo M Dengan Skor Preferensi : 0.5611111111111111 Serta Nilai Kriteria [2, 1, 2, 3, 2]
Rank 13 Panti Jompo T Dengan Skor Preferensi : 0.5462962962962963 Serta Nilai Kriteria [5, 3, 4, 1, 5]
Rank 14 Panti Jompo E Dengan Skor Preferensi : 0.5259259259259259 Serta Nilai Kriteria [5, 5, 3, 1, 5]
Rank 15 Panti Jompo S Dengan Skor Preferensi : 0.5055555555555555 Serta Nilai Kriteria [2, 3, 3, 1, 2]
Rank 16 Panti Jompo H Dengan Skor Preferensi : 0.4962962962962963 Serta Nilai Kriteria [2, 3, 4, 1, 2]
Rank 17 Panti Jompo J Dengan Skor Preferensi : 0.4916666666666667 Serta Nilai Kriteria [2, 2, 2, 1, 4]
Rank 18 Panti Jompo O Dengan Skor Preferensi : 0.48148148148148145 Serta Nilai Kriteria [4, 5, 3, 2, 2]
Rank 19 Panti Jompo I Dengan Skor Preferensi : 0.47777777777777775 Serta Nilai Kriteria [2, 3, 2, 1, 3]
Rank 20 Panti Jompo W Dengan Skor Preferensi : 0.4222222222222222 Serta Nilai Kriteria [2, 4, 2, 1, 5]
Rank 21 Panti Jompo V Dengan Skor Preferensi : 0.38055555555555554 Serta Nilai Kriteria [3, 5, 5, 4, 2]
Rank 22 Panti Jompo K Dengan Skor Preferensi : 0.3425925925925926 Serta Nilai Kriteria [2, 5, 3, 5, 2]
Rank 23 Panti Jompo N Dengan Skor Preferensi : 0.31666666666666665 Serta Nilai Kriteria [1, 4, 5, 3, 2]

```

Pembahasan

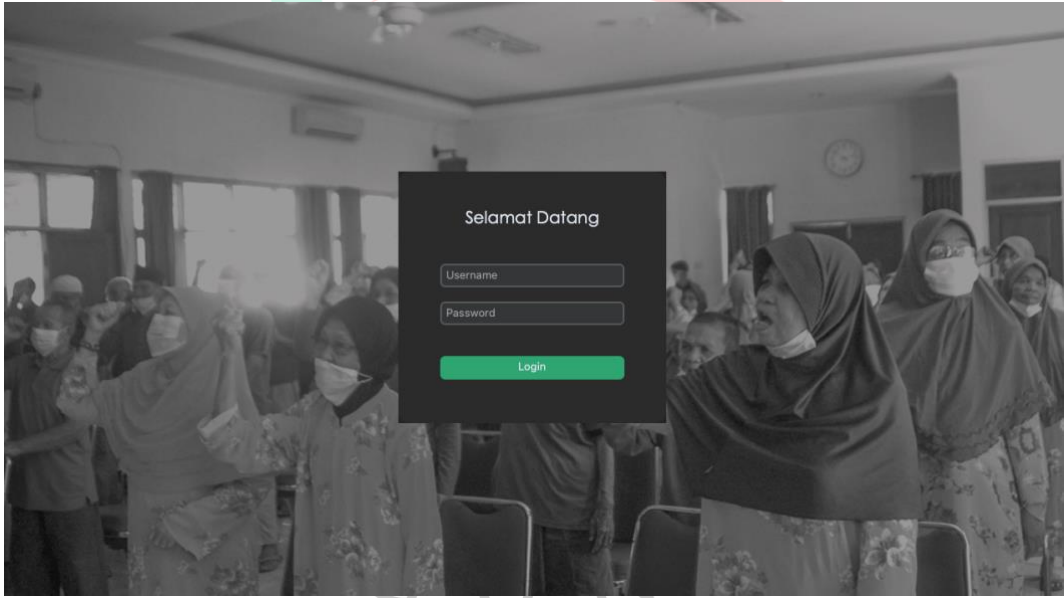
Hasil sesuai dengan urutan *ranking* perhitungan secara manual yang diketahui dari hasil penjumlahan nilai preferensi sebelumnya, yakni 5 alternatif panti jompo dengan nilai tertinggi yang dapat dijadikan pilihan sebagai calon penerima bantuan sosial, antara lain:

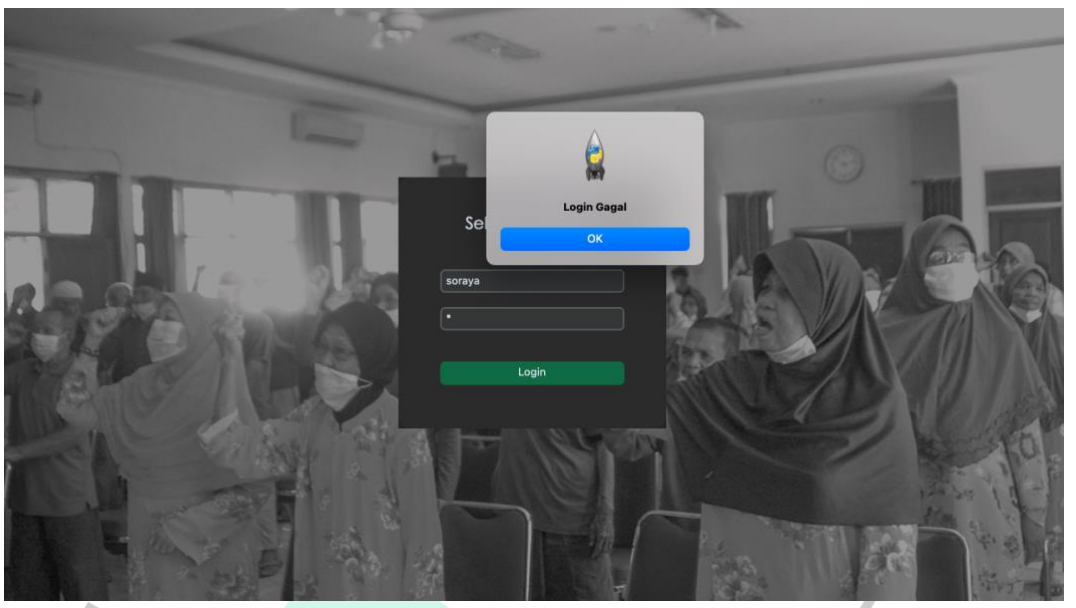
1. Alternatif A4 dengan nilai 0,75 (Panti Jompo D)
2. Alternatif A6 dengan nilai 0,73 (Panti Jompo F)
3. Alternatif A17 dengan nilai 0,72 (Panti Jompo Q)
4. alternatif A16 dengan nilai 0,70 (Panti Jompo P)
5. alternatif A1 dengan nilai 0,68 (Panti Jompo A)

Pengujian Kotak Hitam

Selain pengujian kotak putih, pengujian kotak hitam juga merupakan tahapan penting untuk memastikan fungsionalitas aplikasi berjalan sesuai dengan kebutuhan. Pengujian dilakukan dengan menguji aplikasi secara keseluruhan dan berfokus pada skenario yang telah dirancang sebelumnya.

Tabel 5.3. Hasil Pengujian Kotak Hitam

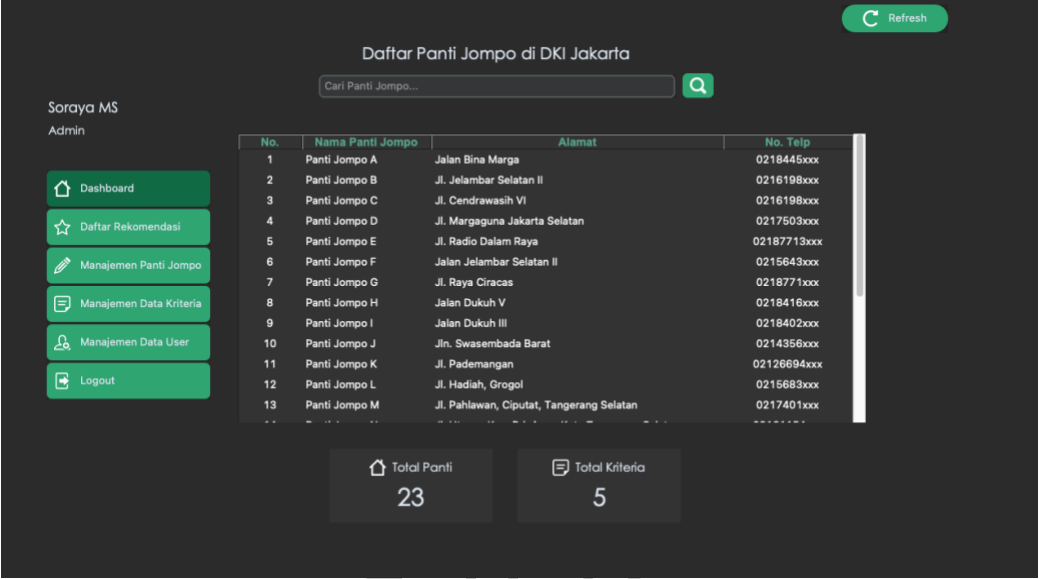
No	Skenario Pengujian	Hasil yang Diharapkan
1	User membuka aplikasi	Menampilkan halaman <i>login</i>
<p>Hasil</p> 		
<p>Kesimpulan: User berhasil mengakses halaman <i>login</i></p>		
2	User menginput ' <i>username</i> ' atau ' <i>password</i> ' yang salah atau tidak terdaftar dalam sistem	Menampilkan pesan notifikasi <i>login</i> gagal.
<p>Hasil</p>		



Kesimpulan: Sistem berhasil menampilkan pesan notifikasi *login gagal*

3	User login dengan akun yang terdaftar.	Sistem menampilkan menu <i>dashboard</i> .
---	--	--

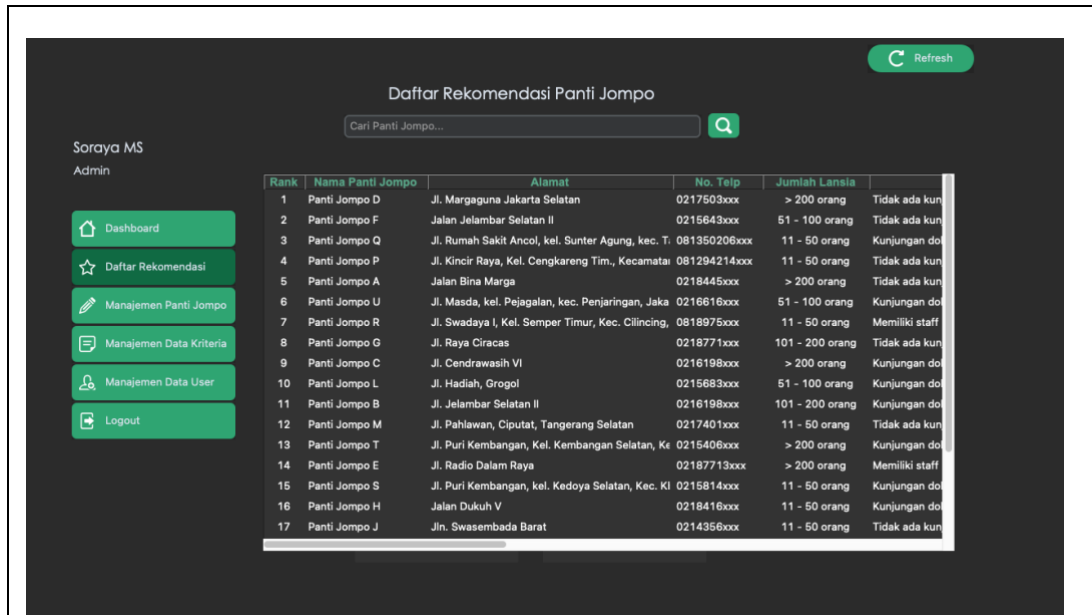
Hasil



Kesimpulan: Sistem berhasil menampilkan menu dashboard ketika *user* login dengan akun yang terdaftar.

4	User mengakses menu Daftar Rekomendasi	Menampilkan halaman Daftar Rekomendasi
---	--	--

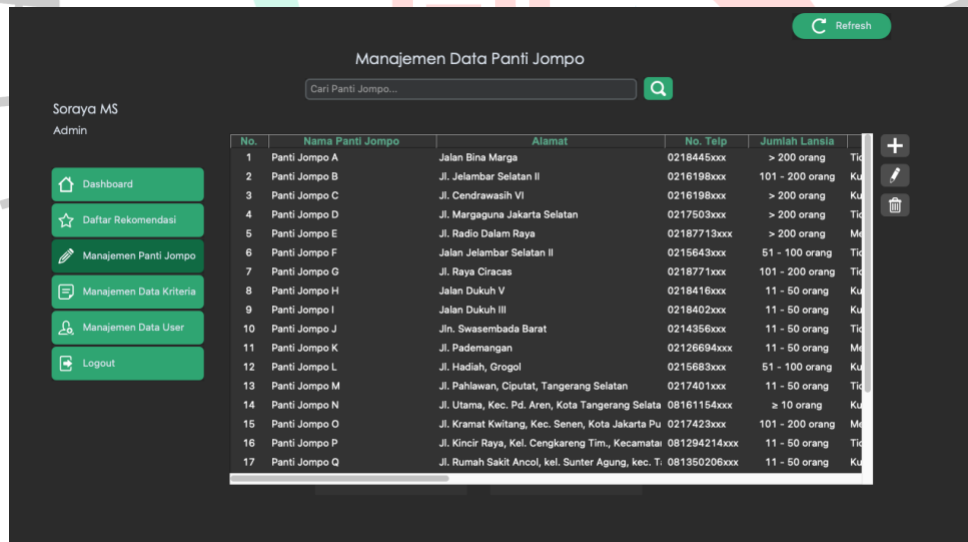
Hasil



Kesimpulan: sistem berhasil menampilkan halaman Daftar Rekomendasi.

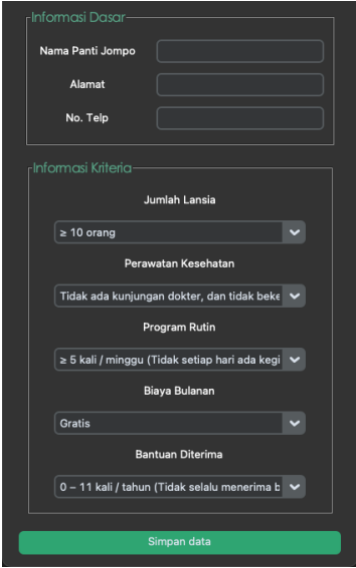
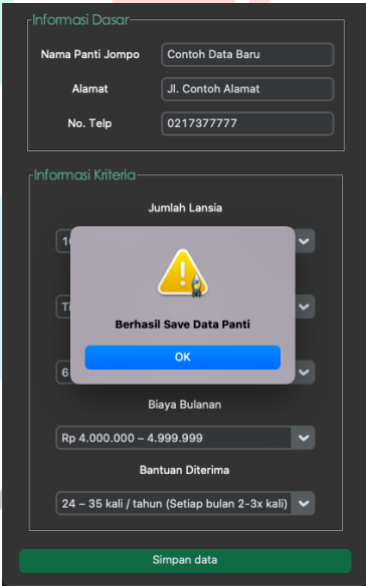
5	User (admin) mengakses menu 'Manajemen Panti Jompo'	Menampilkan halaman Manajemen Panti Jompo
---	---	---

Hasil

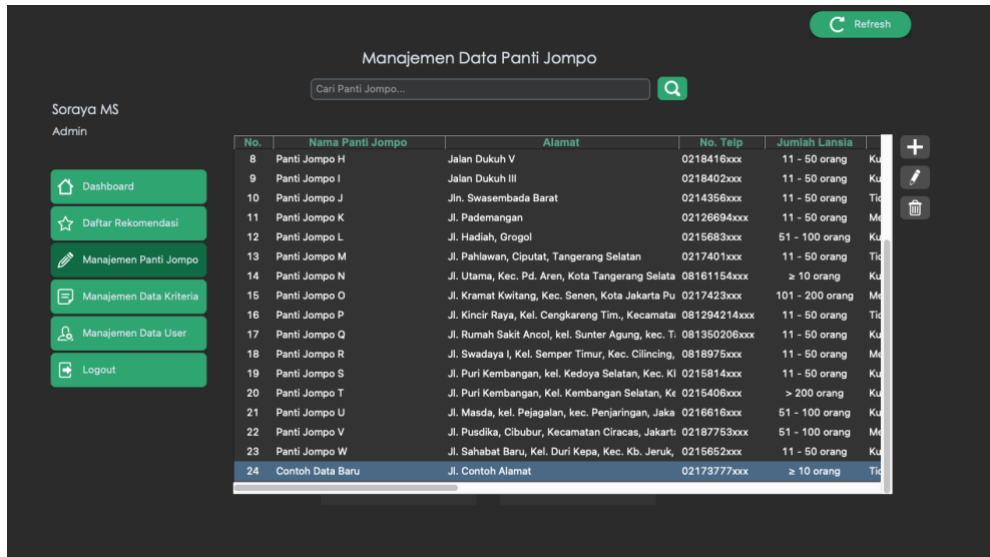


Kesimpulan: Sistem berhasil menampilkan halaman Manajemen Panti Jompo

6	User (admin) menekan tombol tambah untuk menambahkan data panti jompo	Menampilkan formulir penambahan data panti jompo
---	---	--

<p>Hasil</p> 		
<p>Kesimpulan: Sistem berhasil menampilkan formulir penambahan data Panti Jompo</p>		
7	<p>User (admin) memilih 'Simpan data' untuk menyimpan data baru panti jompo</p>	<p>Menyimpan data panti jompo yang baru ke dalam database dan menampilkan pesan data berhasil disimpan</p>
<p>Hasil</p> 		
<p>Kesimpulan: Sistem berhasil menyimpan data ke dalam database dan menampilkan pesan 'Berhasil save data panti'.</p>		
8	<p>User (admin) meng-klik tombol 'Refresh'</p>	<p>Halaman terbarukan dan menampilkan data yang paling baru setelah ada pembaharuan data</p>

Hasil

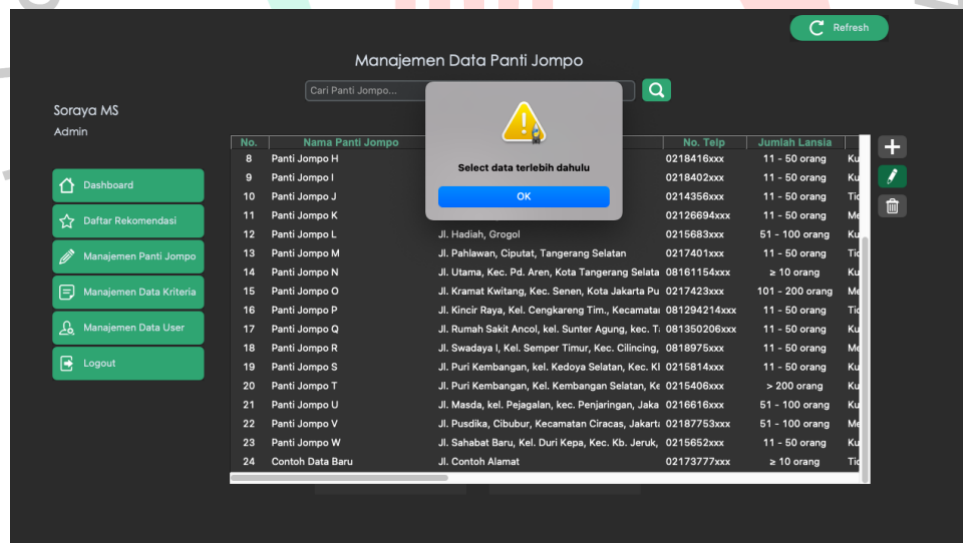


Kesimpulan: Sistem berhasil menampilkan data terbaru yang telah dilakukan penambahan atau edit setelah dilakukan *refresh*.

9 *User* (admin) meng-klik tombol edit saat belum ada data yang terpilih

Gagal edit dan menampilkan pesan harus ada data yang dipilih

Hasil

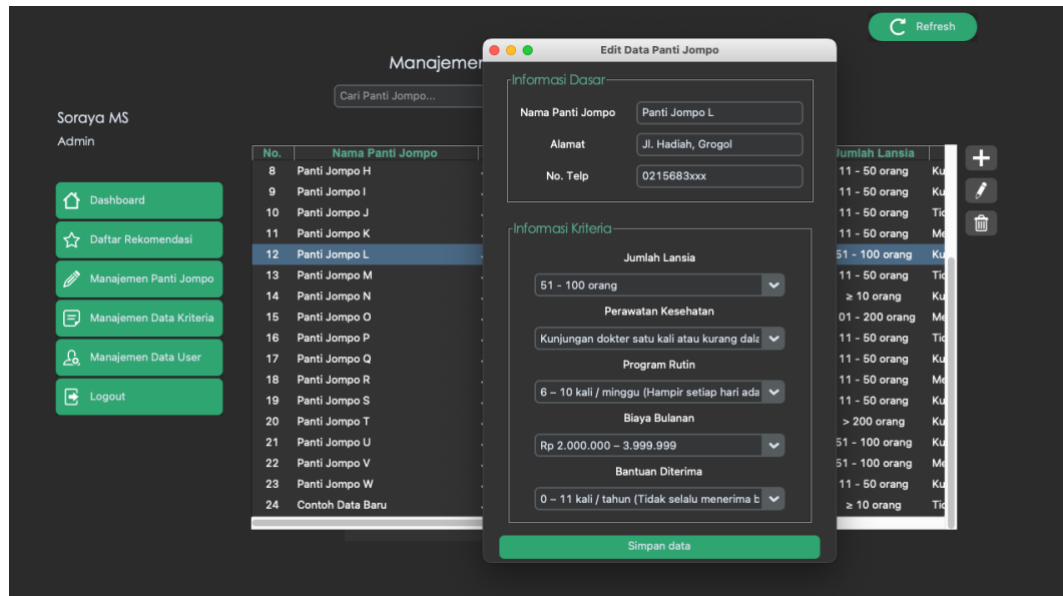


Kesimpulan: Sistem berhasil menampilkan pesan 'select data terlebih dahulu'.

10 *User* (admin) memilih data untuk dilakukan edit

Menampilkan formulir edit dengan data yang dipilih

Hasil



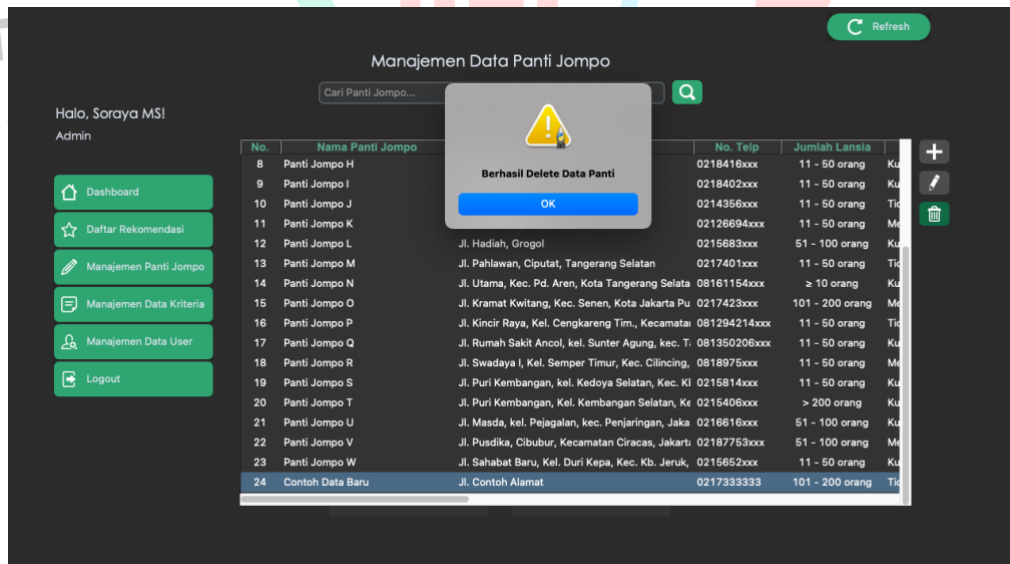
Kesimpulan: Sistem berhasil menampilkan formulir data sesuai dengan data yang dipilih dan user (admin) dapat melakukan perubahan data dan menyimpan data.

11

User memilih untuk menghapus data panti jompo

Menghapus data panti jompo yang dipilih dan menampilkan pesan data berhasil dihapus

Hasil



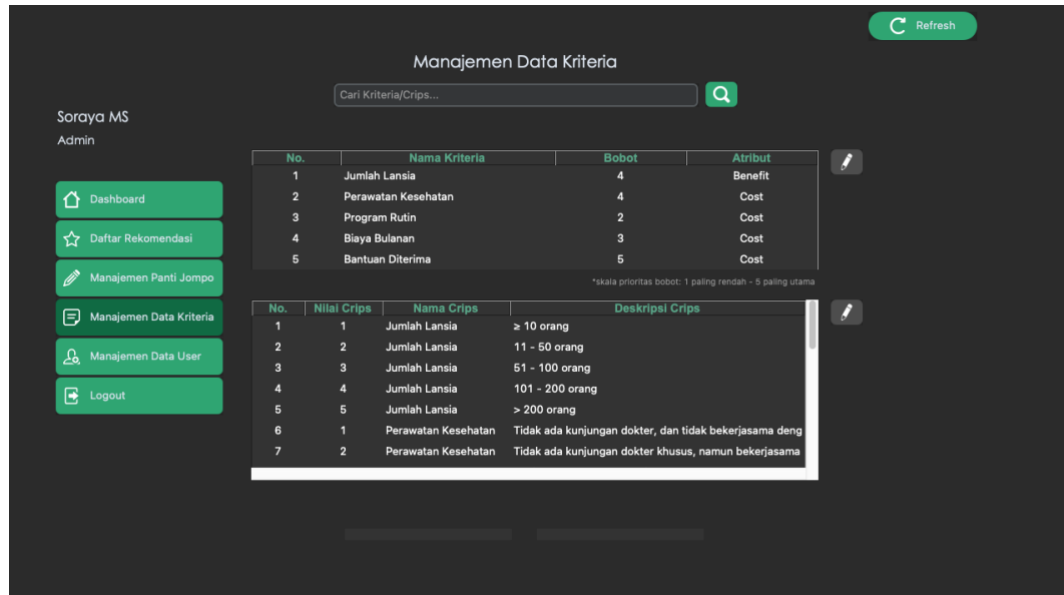
Kesimpulan: Sistem berhasil menghapus data yang dipilih dan menampilkan pesan 'berhasil delete data panti'

12

User (admin) memilih menu Manajemen Data Kriteria

Menampilkan halaman Manajemen Data Kriteria

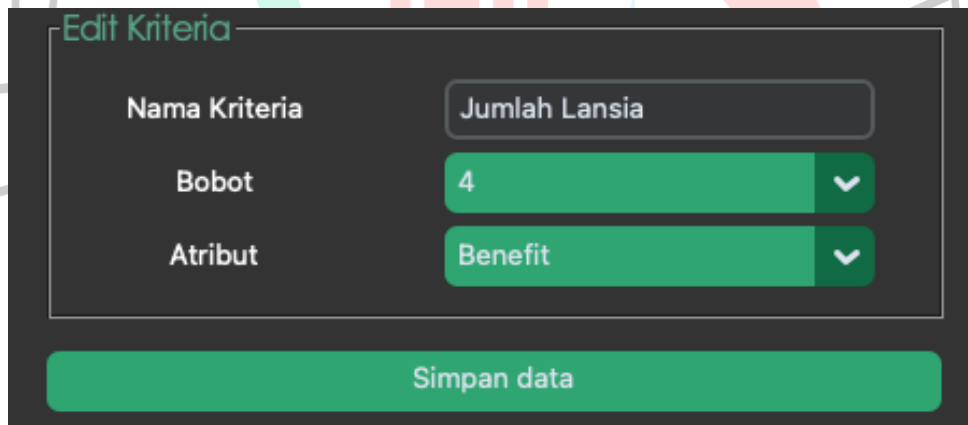
Hasil



Kesimpulan: Sistem berhasil menampilkan halaman Manajemen Data Kriteria

13	User (admin) memilih data kriteria untuk dilakukan edit	Menampilkan formulir edit kriteria sesuai data yang dipilih
----	---	---

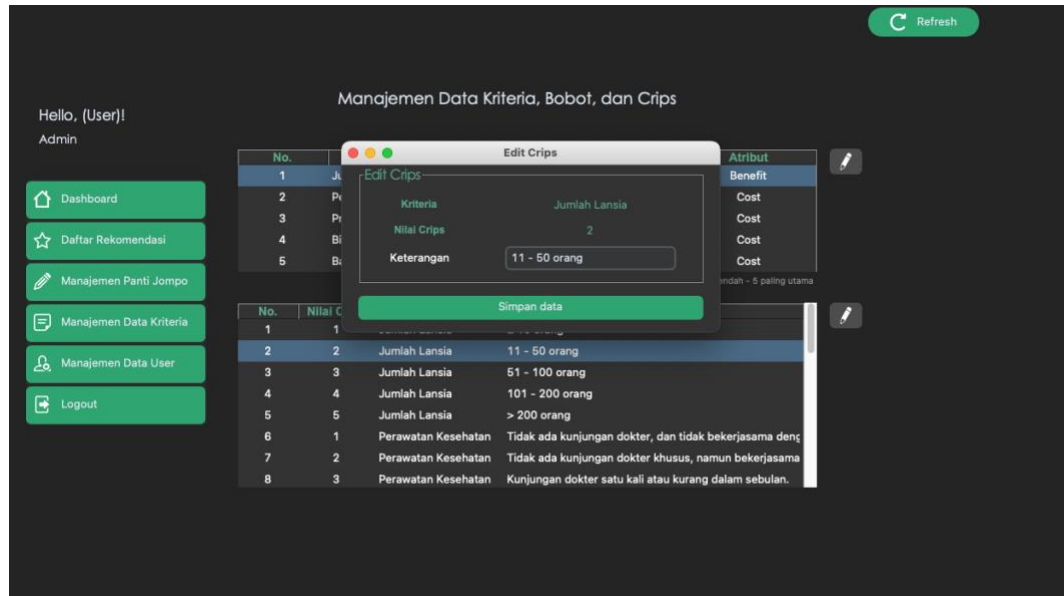
Hasil



Kesimpulan: Sistem berhasil menampilkan formulir edit kriteria yang dipilih.

14	User (admin) memilih data crips untuk dilakukan edit	Menampilkan formulir edit crips sesuai data yang dipilih
----	--	--

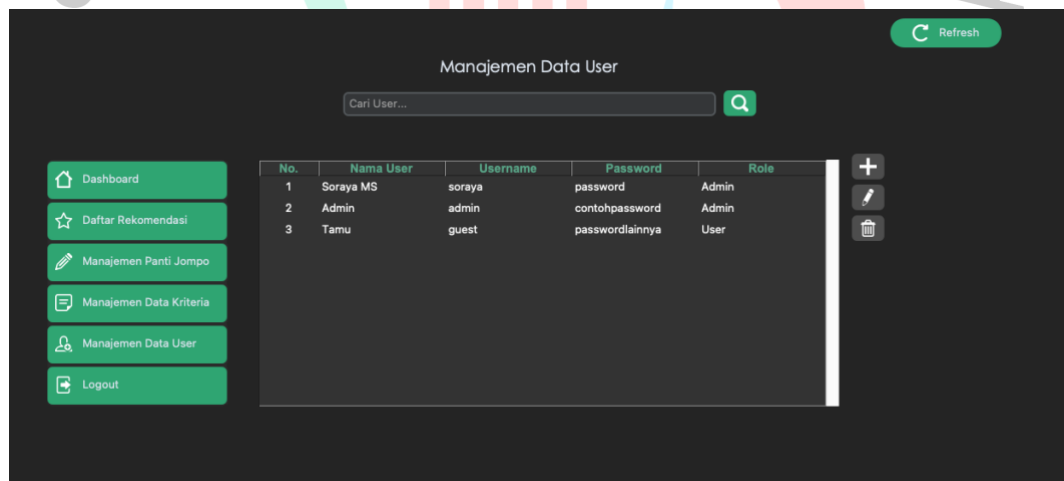
Hasil



Kesimpulan: Sistem berhasil menampilkan formulir edit crips yang dipilih.

15	User (admin) memilih menu Manajemen User	Menampilkan halaman Manajemen User
----	--	------------------------------------

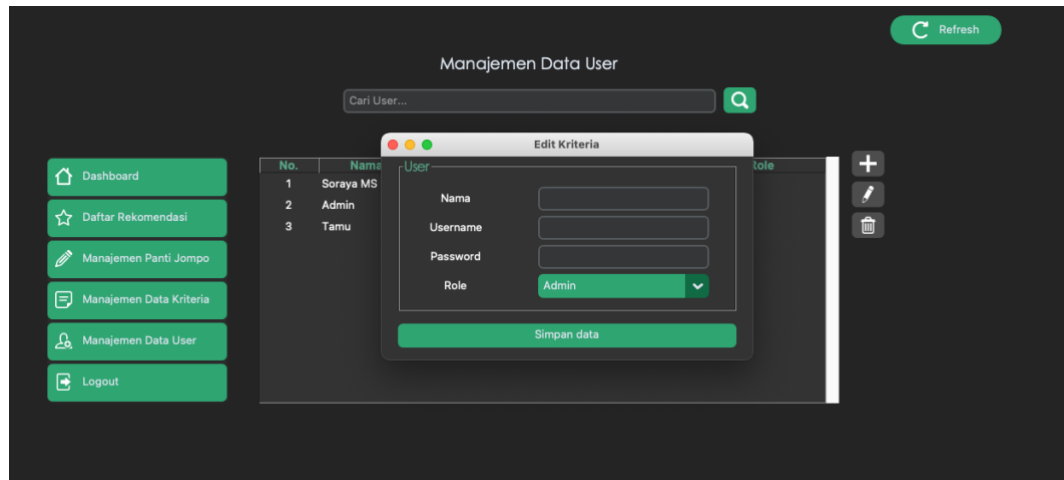
Hasil



Kesimpulan: Sistem berhasil menampilkan halaman Manajemen User

16	User (admin) memilih tambah untuk melakukan penambahan data user	Menampilkan formulir penambahan user baru
----	--	---

Hasil

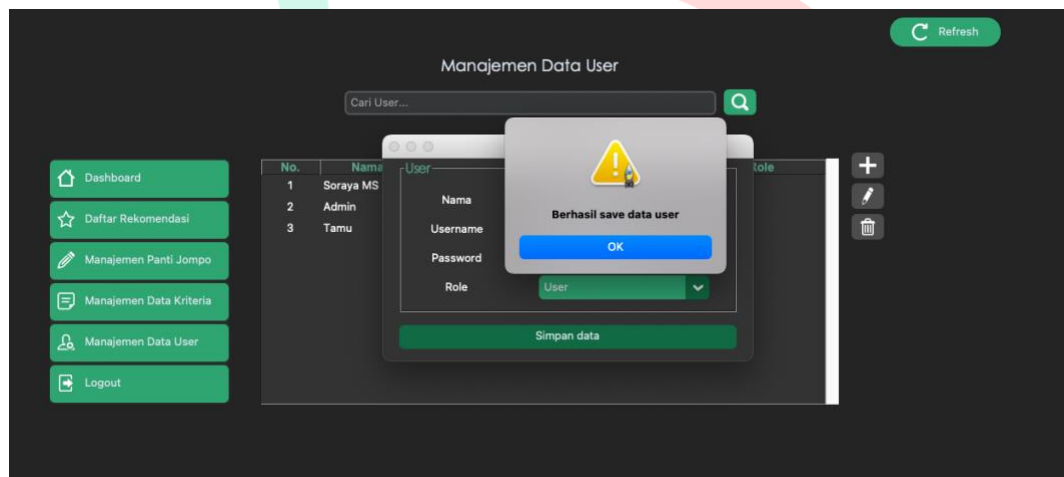


Kesimpulan: Sistem berhasil menampilkan formulir penambahan *user* baru.

17 *User* (admin) menyimpan data *user* baru.

Menyimpan data *user* baru dalam database dan menampilkan pesan data berhasil disimpan.

Hasil

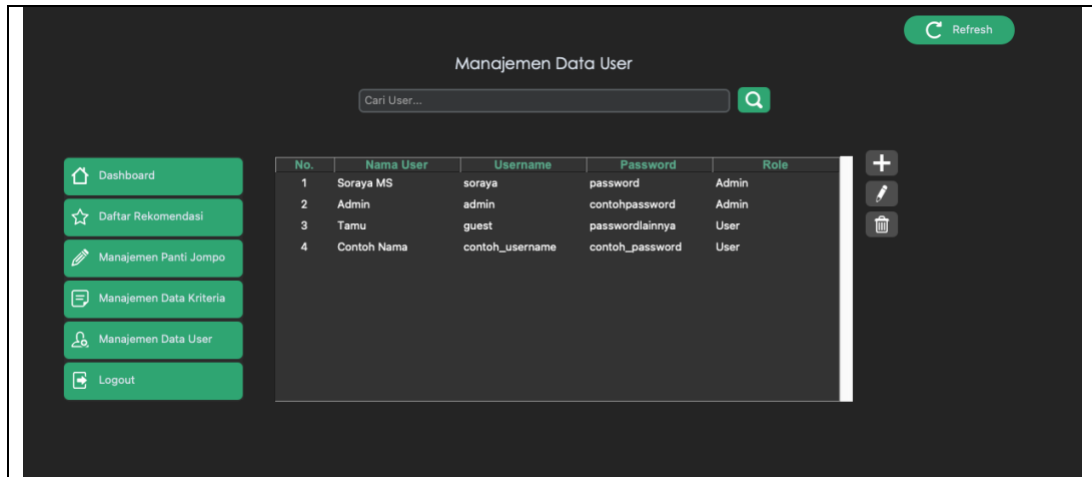


Kesimpulan: Sistem berhasil menyimpan data baru dan menampilkan pesan 'Berhasil save data *user*'.

18 *User* (admin) memilih tombol *refreh* di halaman Manajemen Data User

Menampilkan tabel data *user* yang terbaru

Hasil

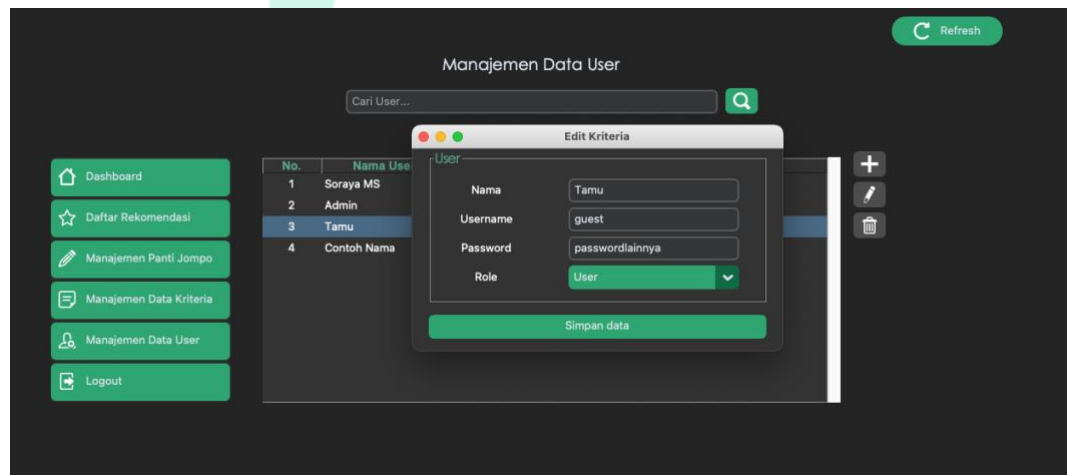


Kesimpulan: Sistem berhasil menampilkan data *user* terbaru yang telah dilakukan penambahan atau edit setelah dilakukan *refresh*.

19 *User* (admin) memilih menu edit pada data *user*

Menampilkan formulir edit sesuai dengan data yang dipilih

Hasil



Kesimpulan: Sistem berhasil menampilkan formulir edit sesuai dengan data *user* yang dipilih.

20 *User* memilih menu *logout*

Pengguna dari aplikasi

Hasil

Keluar dari aplikasi

Kesimpulan: Sistem berhasil menutup jendela aplikasi setelah *user* memilih menu *logout*.