

BAB 4

HASIL PENELITIAN

4.1 RINGKASAN HASIL PENELITIAN

Pengembangan sistem informasi di Klinik Kecantikan XYZ telah menghasilkan peningkatan signifikan dalam efisiensi layanan. Sistem baru memungkinkan pendaftaran online, yang mengurangi kesalahan dan keterlambatan dalam pengelolaan data pelanggan. Proses reservasi menjadi lebih terstruktur, memungkinkan pelanggan memilih perawatan yang sesuai dengan kondisi kulit mereka melalui metode *Case-Based Reasoning* (CBR). Jadwal praktek dokter yang dapat diakses online mengurangi waktu tunggu dan penumpukan pelanggan di klinik. Konsultasi kini ditangani langsung oleh dokter melalui platform terintegrasi, mempercepat respon dan pengambilan keputusan. Implementasi sistem ini berhasil meningkatkan efisiensi operasional, mengurangi waktu tunggu, dan meningkatkan kepuasan pelanggan secara keseluruhan.

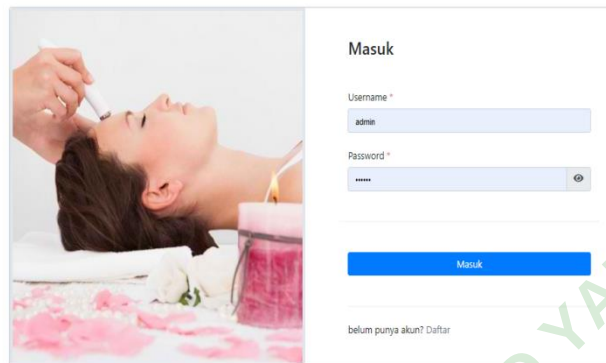
4.2 IMPLEMENTASI DESAIN ANTARMUKA

Antarmuka yang digunakan oleh pengguna untuk berinteraksi secara langsung yaitu desain interface. Dalam penelitian transformasi efisiensi layanan pengembangan sistem informasi terintegrasi untuk klinik XYZ, menggunakan bahasa pemrograman Python dan untuk tampilan dan warna website menggunakan beberapa bantuan bahasa pemrograman seperti *css*, *javascript*, *html* dan *bootstrap*. Memanfaatkan *framework Flask* dengan *SqlAlchemy* untuk pengelolaan *database*, dan *MySQL* sebagai sistem manajemen basis data. Berikut ini adalah beberapa kode dan hasil implementasi halaman yang terdapat dalam sistem informasi terintegrasi untuk klinik XYZ.

4.2.1 Implementasi Halaman Log In

Halaman ini digunakan user untuk mengakses sistem informasi klinik XYZ pada tahap awal. Disini user akan mengisi form username dan password, untuk user sendiri terdiri dua user yaitu pelanggan dan admin. Untuk pelanggan

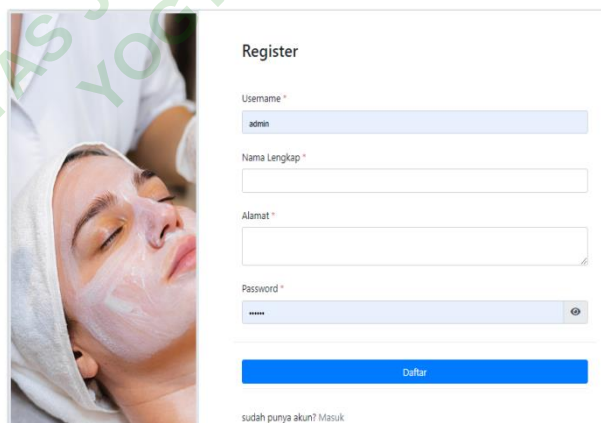
terlebih dahulu mendaftar jika tidak memiliki akun sebelumnya. Halaman login ini bisa dilihat pada gambar 4.1.



Gambar 4.1 halaman *login*

4.2.2 Implementasi Halaman Registrasi

Pada gambar 4.2 memperlihatkan form daftar bagi pelanggan yang belum memiliki akun. Pelanggan dapat mengisi form registrasi dengan memasukkan username, nama lengkap, alamat dan password. Untuk password sendiri diusahakan menggunakan unsur huruf kapital, angka dan simbol.

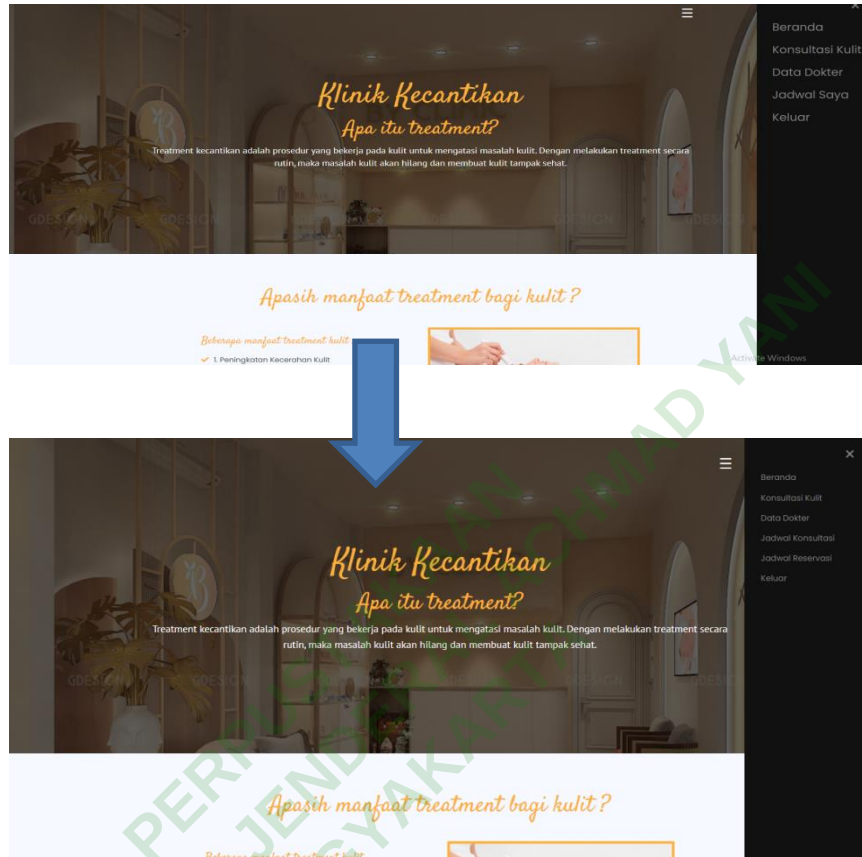


Gambar 4.2 halaman registrasi

4.2.3 Tampilan Dashboard Pelanggan

Tampilan dashboard pelanggan yang dapat dilihat pada gambar 4.3 memperlihatkan sidebar yang akan menampilkan navbar yang berisi beberapa fitur yang bisa dipilih oleh pelanggan seperti konsultasi kulit, data dokter, jadwal

saya, dan keluar. Dashboard juga menampilkan beberapa penjelasan terkait treatment manfaat dan pengertiannya.

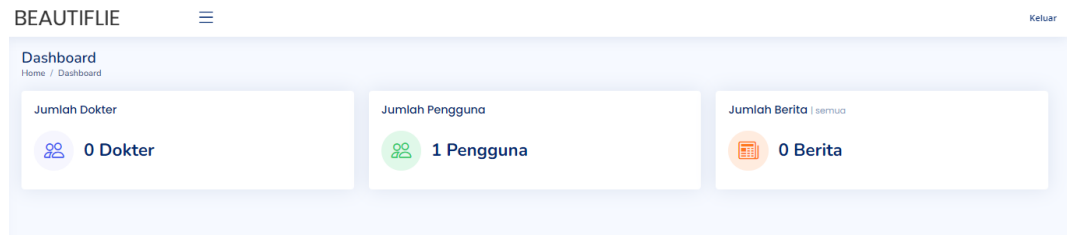


Gambar 4 3 dashboard pelanggan

Pada tampilan dashboard ini mendapatkan evaluasi setelah melakukan test, dimana isi navbar yang sebelumnya hanya 3 fitur menjadi 4 fitur dengan menambahkan fitur jadwal reservasi.

4.2.4 Tampilan Dashboard Admin

Tampilan dashboard admin pada gambar 4.4 menunjukkan jumlah dokter yang praktek diklinik XYZ, jumlah pengguna dan jumlah data berita. pada dashboard admin juga terdapat sidebar yang nantinya berisi data konsultasi dan data treatment.



Gambar 4.4 Gambar Dashboard Admin

4.2.5 Tampilan Konsultasi kulit

Pada gambar 4.5 tampilan konsultasi kulit pelanggan untuk mendapatkan rekomendasi treatment pada klinik. Pelanggan akan diarahkan untuk mengisi gejala kulit dan memasukan jenis kulit, kemudian pelanggan bisa klik mendapatkan rekomendasi.

Gambar 4.5 Konsultasi kulit

Setelah mendapatkan rekomendasi maka tampilan selanjutnya akan muncul seperti gambar 4.6

Gambar 4.6 halaman hasil rekomendasi

Pada fitur ini mengalami perubahan, sebelumnya tidak ada tombol reservasi setelah pencarian rekomendasi. Pada gambar 4.6 menunjukan reservasi yang digunakan pelanggan untuk melakukan treatment.

4.2.6 Tampilan Reservasi

Pada halaman jadwal saya pelanggan dapat melihat jadwal treatment di klinik XYZ dan melihat keterangan apakah sudah bisa terjadwal atau masih menunggu konfirmasi admin untuk memastikan jam tersebut kosong. Tampilan dapat dilihat pada gambar 4.7.

Jadwal Konsultasi

10 entries per page Search...

No	Nama Dokter	Tanggal/Hari Konsultasi	Jam Konsultasi	Telp	Keterangan
1	dr.Febiola Fujioko	Kamis, 27-06-2024	11.00 s.d 12.00 WIB	083825520613	Menunggu Konfirmasi

Showing 1 to 1 of 1 entries

Gambar 4.7 Halaman Jadwal Saya

4.2.7 Tampilan Jadwal Reservasi

Pada halaman admin jadwal reservasi akan menampilkan data pelanggan yang ingin melakukan treatment. Pada gambar 4.8 menampilkan data pelanggan berupa nama pasien, nama dokter yang dipilih, nomor telpon, tanggal, jadwal dan status.

Jadwal Konsultasi ke Dokter

Tanggal Mulai: Tanggal Selesai: Search Export to Excel

10 entries per page Search...

Nama Pasien	Nama Dokter	Telp Pasien	Tanggal Konsultasi	Jadwal Konsultasi	Status
nata tuti	dr.Febiola Fujioko	083825520613	Kamis, 27-06-2024	11.00 s.d 12.00 WIB	Pending

Konfirmasi Status Konsultasi

Konfirmasi status konsultasi pasien?

* Konfirmasi hanya bisa dilakukan sekali.

Tolak Konfirmasi

Status

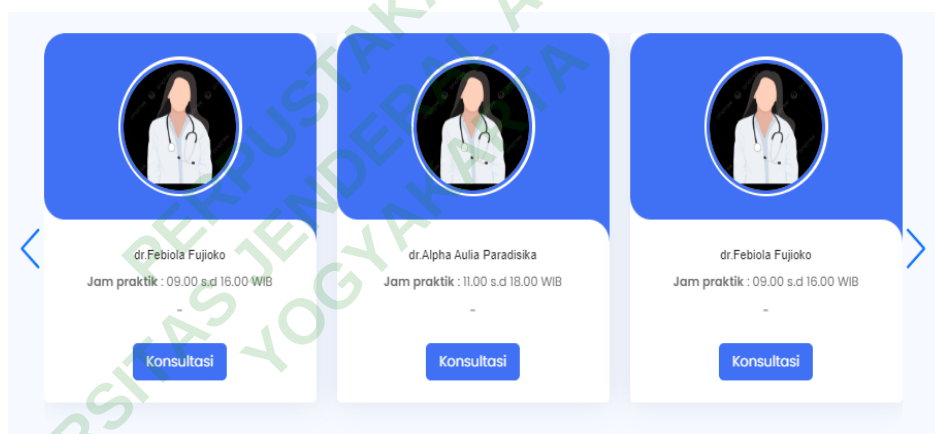
Terjadwal

Gambar 4 8 Halaman jadwal reservasi

Pada gambar 4.8 juga menunjukan admin dapat menerima jadwal reservasi pelanggan dengan klik status pending dan konfirmasi untuk menjadwalkan treatment pelanggan.

4.2.8 Tampilan Jadwal Konsultasi dan Data Dokter

Pada halaman ini pelanggan akan memilih dokter untuk konsultasi langsung secara *face to face*. Dimana fitur ini digunakan pelanggan untuk konsultasi kulit dengan dokter secara langsung tanpa menggunakan sistem rekomendasi. Pelanggan juga dapat melihat data dokter nama, jam kerja dan keterangan lainnya. Untuk konsultasi ini juga digunakan bagi pelanggan ketika gejala tidak ada di rekomendasi bisa terlebih dahulu konsultasi atau merupakan pelanggan baru.



Gambar 4.9 Halaman Data Dokter dan Konsultasi

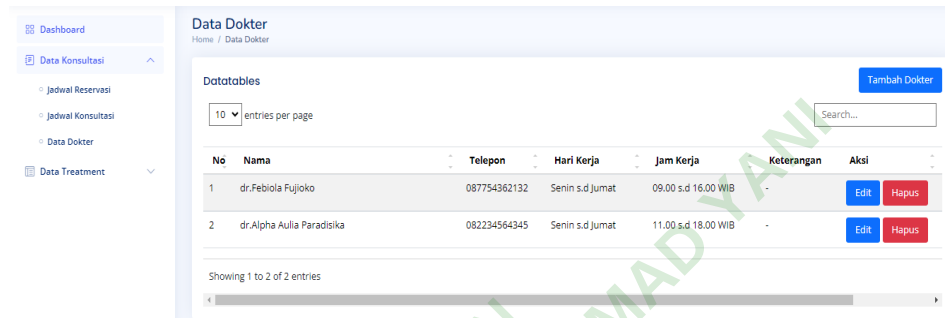
Selanjutnya setelah konsultasi pelanggan akan mengisi form untuk menjadwalkan konsultasi dengan dokter secara langsung. Tampilan halaman form konsultasi terdapat pada gambar 4.10.

 The image shows a web form titled 'Form Jadwal Konsultasi'. It contains several input fields: 'Nama & Gelar Dokter*' (filled with 'dr. Alpha Aulia Paradisika'), 'Telepon *' (filled with '083825520583'), 'Tanggal Konsultasi *' (filled with '06/09/2024'), and 'Jam Konsultasi *' (filled with '15.00 s.d 18.00 WIB'). There is a blue button at the bottom right labeled 'Buat Jadwal Sekarang'.

Gambar 4 10 Halaman Form Jadwal Konsultasi

4.2.9 Data Dokter

Pada gambar 4.11 menampilkan halaman input data dokter oleh admin, dimana admin akan menambahkan dokter yang ada di klinik XYZ. Admin juga dapat menghapus atau mengedit data jika ada perubahan atau kesalahan pada penginputan.



Gambar 4 11 Halaman Data Dokter

Admin menambahkan dokter dengan mengisi form data-data dokter, dengan mengisi nama dan gelar, telpon, hari kerja, upload foto, dan keterangan lalu simpan. Tampilan halaman dapat dilihat pada gambar 4.12.

Gambar 4 12 Halaman Form Data Dokter

4.2.10 Halaman Konsultasi

Pada halaman ini menampilkan jadwal konsultasi pelanggan di admin, dimana admin nanti akan mengkonfirmasi jadwal tersebut sama halnya dengan gambar halaman jadwal reservasi. Admin juga bisa import data untuk arsip perbulannya di klinik XYZ, digambar 4.13 ini menampilkan pilihan tanggal mulai dan tanggal selesai data yang akan dicari. Dan pada pojok kanan terdapat *search* untuk mencari data pelanggan dengan menginputkan nama pasien.

No	Nama Pasien	Nama Dokter	Telp Pasien	Tanggal Konsultasi	Jadwal Konsult	Status
1	afnisa sondari	dr.Febiola Fujioko	082234564345	Jumat, 28-06-2024	11.00 s.d 12.00 W	Pending
2	nata tuti	dr.Febiola Fujioko	083825520613	Kamis, 27-06-2024	11.00 s.d 12.00 W	Terjadwal

Gambar 4 13 Halaman Jadwal Konsultasi

4.2.11 Tampilan type Treatment

Pada halaman `type_treatment` admin dapat menambahkan `type treatment` yang tersedia di klinik XYZ untuk nantinya bisa dipakai sebagai data disistem rekomendasi.

No	Nama	Aksi
1	FLEK SERIES	Edit Hapus
2	SLIMMING TREATMENT	Edit Hapus
3	BODY SERIES	Edit Hapus
4	ACNE SERIES	Edit Hapus
5	SUPER GLOW SERIES	Edit Hapus

Gambar 4.14 Halaman Type_treatment

4.2.12 Data Treatment

Pada halaman `data treatment` ini berisi data-data menu *treatment* yang bisa diambil oleh pelanggan, data pada halaman ini akan diinput oleh admin dengan berupa file Excel. Pada halaman ini juga admin dapat mengedit, menghapus data dan menambahkan data yang belum terinput pada data ketika dokter menemukan permasalahan dengan gejala lain yang tidak ada sebelumnya.



No	Jenis	Treatment	Manfaat	Gejala	Jenis I	Aksi
1	BODY SERIES	Glow C Inject	melembabkan kulit tubuh dan imun booster	kulit tubuh kusam kasar dan susah putih	norma	Edit Hapus
2	SLIMMING TREATMENT	Inject Slimming For Body	menghilangkan lemak pada tubuh di area tertentu	bagian tubuh yang terasa membengkak akibat penumpukan lemak	tubuh dan pe	Edit Hapus
3	SUPER GLOW SERIES	Superglow Facial	mencerahkan, menghilangkan komedo, menghaluskan	warna kulit yang tidak merata, kurangnya kilau atau cahaya pada kulit, dan tekstur kulit yang kasar. Selain itu, kulit kusam	kulit ku, berkon	Edit Hapus

Gambar 4.15 Halaman Data *Treatment*

Admin akan menambahkan data ketika ada data baru yang akan di input, tampilan tambah bisa dilihat pada gambar 4.16. dengan mengisi *form* yang berisi jenis treatment, treatment, manfaat, gejala, jenis kulit dan informasi lalu admin bisa simpan data maka data otomatis bertambah.



Form Treatment

Jenis Treatment *
FLEK SERIES

Treatment *

Manfaat *

Gejala *

Jenis Kulit *

Informasi *

[Simpan](#)

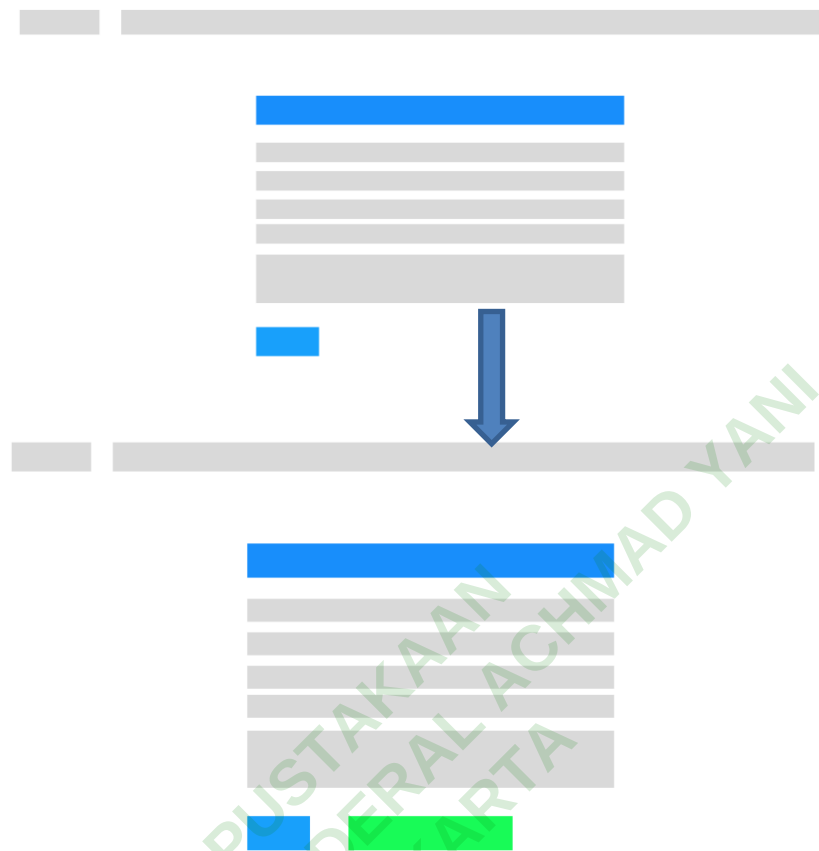
Gambar 4.16 Form tambah data *treatment*

4.3 IMPELEMNTASI METODE PENGEMBANGAN

Pada sistem informasi ini menggunakan metode paralel inkremental, dimana sistem akan melakukan perulangan sesuai dengan evaluasi dari test UAT sebelumnya. Pada implementasi ini terdapat beberapa perubahan tampilan dari segi pelanggan dan admin.

4.3.1 Implementasi Metode Pengembangan pada pelanggan

Perulangan dari segi pelanggan terdapat pada fitur konsultasi kulit, dimana fitur konsultasi kulit merupakan fitur sistem rekomendasi perawatan untuk pelanggan. Pada desain mockup sebelumnya bisa dilihat pada gambar 4.17.



Gambar 4.17 Desain hasil sistem rekomendasi

Setelah mengalami perubahan pada desain, tampilan perubahan implementasi sistemnya terdapat pada gambar 4.18. Pada implementasi tampilan sebelumnya tidak terdapat reservasi dokter. Pelanggan akan mendapatkan rekomendasi tetapi hanya mendapatkan rekomendasi dan kembali ke dashboard. Sedangkan pada pengujian test terhadap pelanggan terdapat saran untuk menambahkan fitur reservasi setelah konsultasi kulit menggunakan sistem rekomendasi. Reservasi dokter pada tampilan gambar 4.18 ini akan mengarahkan pelanggan untuk membuat jadwal treatment bersama dokter kemudian menunggu konfirmasi treatment oleh admin. Dimana nanti akan dilihat pada fitur jadwal saya.

Gambar 4.18 Implementasi Konsultasi Kulit

Implementasi code pada tampilan sebelumnya bisa dilihat code dibawah.

```

if rekomendasi_vote_count > 1:
    rekomendasi = rekomendasi_voting
else:
    rekomendasi = rekomendasi_bobot
combined_scores = {}
for treatment in vote_counts:
    combined_scores[treatment] = vote_counts[treatment] +
    weighted_vote_scores.get(treatment, 0)

rekomendasi_terbaik=max(combined_scores,key=combined_scores
.get)

total_votes = sum(vote_counts.values())
vote_percentages = {k: (v / total_votes) * 100 for k, v in
vote_counts.items()}

print(f"Vote Percentages: {vote_percentages}")
print(f"Rekomendasi Voting: {rekomendasi_voting}dengan
{rekomendasi_vote_count} votes")
print(f"Rekomendasi Bobot: {rekomendasi_bobot} dengan score
{rekomendasi_bobot_score}")
print(f"Rekomendasi Terbaik: {rekomendasi_terbaik} dengan
score {combined_scores[rekomendasi_terbaik]}")

if rekomendasi:

    final_rekomendasi=Treatment.query.filter_by(treatment=re
komendasi).first()
    Returnrender_template('user/rekomendasi.html',rekomendas
i=final_rekomendasi)
else:
    flash('Tidak ada rekomendasi perawatan yang ditemukan
untuk gejala dan jenis kulit ini.', 'error')

```

Selanjutnya hasil perubahan code sebelumnya menambahkan code yang bisa dilihat pada code dibawah, dimana menambahkan *def reservasi* untuk mrnampilkan fitur reservasi yang nantinya akan dipesan oleh pelanggan.

```
def reservasi():
    doctors = Doctor.query.all()

    gejala = request.args.get('gejala', '')
    jenis_kulit = request.args.get('jenis_kulit', '')
    treatment = request.args.get('treatment', '')
    types = request.args.get('types', '')
    benefit = request.args.get('benefit', '')
    information = request.args.get('information', '')
```

4.3.2 Implementasi Metode Pengembangan Pada Admin

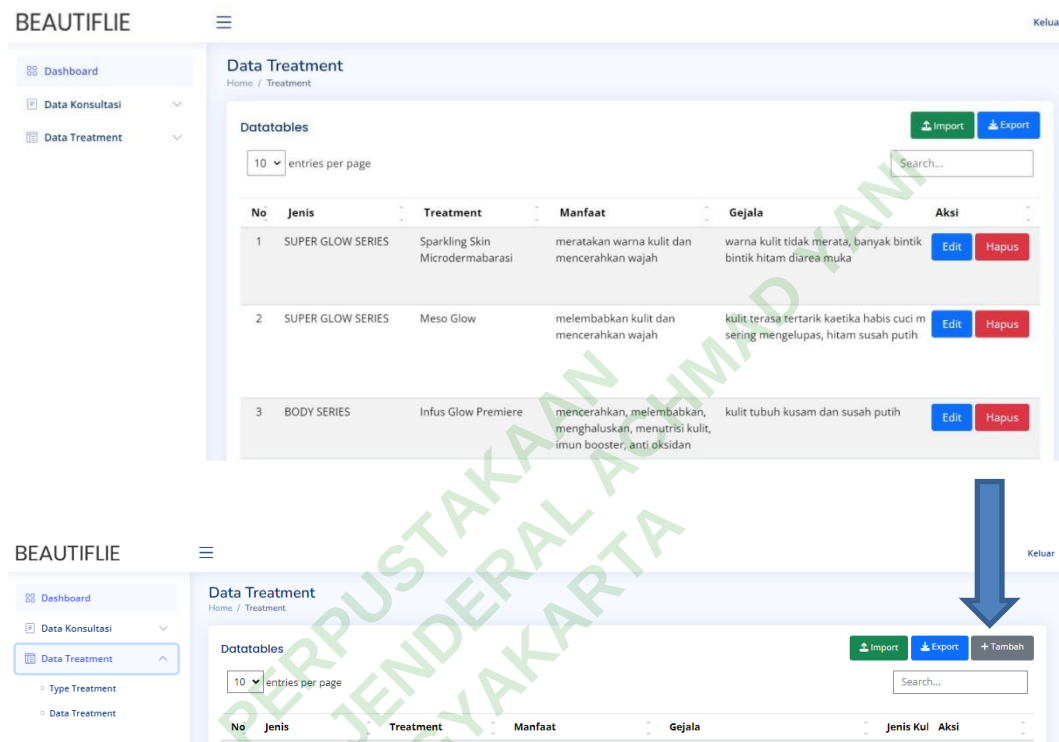
Perulangan yang terjadi dari sisi admin terdapat pada fitur data *treatment* yang mana data *treatment* klinik kecantikan XYZ di *import* dan *export* dengan file Excel, kemudian berubah dengan menambahkan tambah disamping *export*. Tampilan desain data treatment ini bisa dilihat pada gambar 4.19.



Gambar 4.19 Implementasi Desain Data Treatment

Setelah mengalami perubahan sesuai saran dengan menambahkan tombol tambah untuk memudahkan admin menambahkan data treatment baru. Tampilan

implementasi desainnya dapat dilihat pada gambar 4.20. Sebelumnya admin tidak dapat menambahkan gejala atau jenis kulit baru untuk pelanggan hanya bisa ditambahkan difile Excel kemudian di import ulang ke data treatment.



Gambar 4.20 Implementasi Fitur Data Treatment

Selanjutnya implementasi tampilan data treatment terdapat pada code dibawah yang merupakan penambahan code untuk fitur data treatment.

```
if typesData:
    add = Treatment(treatment=treatment, benefit=benefit,
                    symptom=symptom, skin=skin, information=information,
                    type_id=type)
    db.session.add(add)
    db.session.commit()
    flash('Data berhasil ditambahkan.', 'success')
    return redirect(url_for('DBhome.treatment'))
```

4.4 IMPLEMENTASI BASIS DATA

Database untuk sistem ini dinamakan appku dan terdiri dari 78 tabel. Setiap tabel memiliki `id` sebagai *primary key* dan menggunakan *foreign key* untuk menjalin hubungan antar tabel. Struktur database `appku` ditunjukkan pada gambar 4.17.

Tabel	Tindakan	Baris	Jenis	Penyortiran	Ukuran	Beban
bookings	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	0	InnoDB	utf8mb4_general_ci	48.0 KB	-
doctors	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	0	InnoDB	utf8mb4_general_ci	16.0 KB	-
profiles	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	1	InnoDB	utf8mb4_general_ci	32.0 KB	-
reservasi	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	0	InnoDB	utf8mb4_general_ci	48.0 KB	-
treatments	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	0	InnoDB	utf8mb4_general_ci	32.0 KB	-
type_treatments	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	0	InnoDB	utf8mb4_general_ci	16.0 KB	-
users	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	1	InnoDB	utf8mb4_general_ci	32.0 KB	-
7 tabel	Jumlah	2	InnoDB	utf8mb4_general_ci	224.0 KB	0.8

Gambar 4.21 Database Tabel appku

4.4.1 Implementasi tabel booking

Implementasi booking pengguna pada gambar 4.18. tabel ini memiliki *Primary key* yaitu id dan *Foreign Key* customer_id dan doctors_id dimana berelasi dengan tabel doctors dan user.

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
☐ 1	id	varchar(150)	utf8mb4_general_ci	Tidak	Tidak ada				Ubah Hapus Lainnya
☐ 2	telp	varchar(1000)	utf8mb4_general_ci	Ya	NULL				Ubah Hapus Lainnya
☐ 3	time	varchar(1000)	utf8mb4_general_ci	Ya	NULL				Ubah Hapus Lainnya
☐ 4	date	date		Ya	NULL				Ubah Hapus Lainnya
☐ 5	status	varchar(1000)	utf8mb4_general_ci	Ya	NULL				Ubah Hapus Lainnya
☐ 6	customer_id	varchar(150)	utf8mb4_general_ci	Tidak	Tidak ada				Ubah Hapus Lainnya
☐ 7	doctor_id	varchar(150)	utf8mb4_general_ci	Tidak	Tidak ada				Ubah Hapus Lainnya

Gambar 4.22 Implementasi Tabel Booking

4.4.2 Implementasi tabel doctors

Pada gambar 4.19 menunjukkan tabel data dokter dengan nama tabel doctors, dimana id sebagai *Primary Key* yang berelasi dengan tabel booking dan reservasi. Tabel doctors ini memiliki *Foreign Key* yaitu user_id.

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
1	id	varchar(150)	utf8mb4_general_ci	Tidak	Tidak ada				Ubah Hapus Lainnya
2	name	varchar(255)	utf8mb4_general_ci	Tidak	Tidak ada				Ubah Hapus Lainnya
3	photo	varchar(255)	utf8mb4_general_ci	Ya	NULL				Ubah Hapus Lainnya
4	telp	varchar(255)	utf8mb4_general_ci	Ya	NULL				Ubah Hapus Lainnya
5	time	varchar(255)	utf8mb4_general_ci	Ya	NULL				Ubah Hapus Lainnya
6	date	varchar(255)	utf8mb4_general_ci	Ya	NULL				Ubah Hapus Lainnya
7	information	varchar(255)	utf8mb4_general_ci	Ya	NULL				Ubah Hapus Lainnya

Gambar 4.23 Implementasi Tabel Doctors

4.4.3 Implementasi Profile

Gambar 4.20 menunjukkan tabel *profiles* yang memiliki *Primary key* id dan *Foreign Key* user_id yang berelasi dengan tabel *users*.

1	id	varchar(150)	utf8mb4_general_ci	Tidak	Tidak ada	Ubah	Hapus	Lainnya
2	name	varchar(255)	utf8mb4_general_ci	Tidak	Tidak ada	Ubah	Hapus	Lainnya
3	address	varchar(255)	utf8mb4_general_ci	Ya	NULL	Ubah	Hapus	Lainnya
4	photo	varchar(255)	utf8mb4_general_ci	Ya	NULL	Ubah	Hapus	Lainnya
5	user_id	varchar(150)	utf8mb4_general_ci	Tidak	Tidak ada	Ubah	Hapus	Lainnya

Gambar 4.24 Implementasi Tabel *Profiles*

4.4.4 Implementasi Treatment

Implementasi dari data treatment pada pelanggan dimana id sebagai *Primary Key* dan type_id sebagai *Foreign Key*. Tabel ini berelasi dengan tabel type_treatment. Gambar tabel treatment bisa dilihat pada gambar 4.25.

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
1	id	varchar(150)	utf8mb4_general_ci	Tidak	Tidak ada				Ubah Hapus Lainnya
2	treatment	varchar(255)	utf8mb4_general_ci	Ya	NULL				Ubah Hapus Lainnya
3	benefit	varchar(255)	utf8mb4_general_ci	Ya	NULL				Ubah Hapus Lainnya
4	symptom	varchar(255)	utf8mb4_general_ci	Ya	NULL				Ubah Hapus Lainnya
5	skin	varchar(2550)	utf8mb4_general_ci	Ya	NULL				Ubah Hapus Lainnya
6	information	varchar(255)	utf8mb4_general_ci	Ya	NULL				Ubah Hapus Lainnya
7	type_id	varchar(150)	utf8mb4_general_ci	Tidak	Tidak ada				Ubah Hapus Lainnya

Gambar 4.25 Implementasi Tabel Treatment

4.4.5 Implementasi Type_treatment

Gambar 4.26 implementasi dari data treatment yang diberi tabel type_treatment, dimana id sebagai *Primary Key*.

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
1	id	varchar(150)	utf8mb4_general_ci	Tidak	Tidak ada				Ubah Hapus Lainnya
2	type	varchar(255)	utf8mb4_general_ci	Ya	NULL				Ubah Hapus Lainnya

Gambar 4.26 Implementasi Tabel Type_treatment

4.4.6 Implementasi Tabel User

Gambar 4.27 implementasi dari tabel pengguna yang diberi nama tabel *users*, dengan *Primary key* id.

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
1	id	varchar(150)	utf8mb4_general_ci	Tidak	Tidak ada				Ubah Hapus Lainnya
2	username	varchar(255)	utf8mb4_general_ci	Tidak	Tidak ada				Ubah Hapus Lainnya
3	password	varchar(150)	utf8mb4_general_ci	Tidak	Tidak ada				Ubah Hapus Lainnya
4	role	varchar(150)	utf8mb4_general_ci	Tidak	Tidak ada				Ubah Hapus Lainnya

Gambar 4.27 Implementasi Tabel *Users*

4.4.7 Implementasi Tabel Reservasi

Pada gambar 4.28 menunjukkan tabel *reservasi* dengan *Primary Key* *id* dan *Foreign key* *customer_id* dan *doctor_id*.

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
☐ 1	id	varchar(150)	utf8mb4_general_ci	Tidak	Tidak ada				Ubah Hapus Lainnya
☐ 2	treatment	varchar(1000)	utf8mb4_general_ci	Ya	NULL				Ubah Hapus Lainnya
☐ 3	time	varchar(1000)	utf8mb4_general_ci	Ya	NULL				Ubah Hapus Lainnya
☐ 4	date	date		Ya	NULL				Ubah Hapus Lainnya
☐ 5	status	varchar(1000)	utf8mb4_general_ci	Ya	NULL				Ubah Hapus Lainnya
☐ 6	customer_id	varchar(150)	utf8mb4_general_ci	Tidak	Tidak ada				Ubah Hapus Lainnya
☐ 7	doctor_id	varchar(150)	utf8mb4_general_ci	Tidak	Tidak ada				Ubah Hapus Lainnya

Gambar 4.28 Implementasi Tabel *Reservasi*

4.5 PEMBAHASAN

Dari hasil penelitian tentang sistem informasi terintegrasi di klinik XYZ mendapatkan hasil sesuai pertanyaan dan manfaat penelitian. Dimana sistem informasi ini digunakan pelanggan dan admin sesuai dengan pertanyaan penelitian yang mencakup hasil tentang efisien, otomatis input data dan implelementasi CBR.

4.5.1 Fitur Jadwal Reservasi

Pada gambar 4.29 admin akan mendapatkan data reservasi pelanggan secara otomatis, kemudian admin hanya dapat mengkonfirmasi jadwal dengan mengubah status pending menjadi terjadwal.

No	Nama Pasien	Nama Dokter	Tanggal Konsultasi	Jadwal Konsultasi	Status
1	afnisa sondari	dr.Febiola Fujoko	Jumat, 28-06-2024	10.00 s.d 11.00 WIB	Pending

Showing 1 to 1 of 1 entries

Gambar 4.29 Fitur Jadwal Reservasi

Implementasi fitur jadwal reservasi pelanggan pada admin dengan code yang dapat dilihat pada code dibawah.

```
@DBhome.route('/jadwal-reservasi-dokter', methods=['GET',
'POST'])
@login_required
def datereservasi():
    viewDoctor = Reservasi.query.all()
```

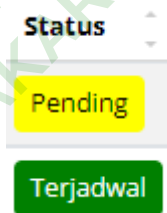
```

results = []
for booking in viewDoctor:
    results.append(
        {
            'id': booking.id,
            'time': booking.time,
            'date': format_date_with_day(booking.date),
            'status': booking.status,
            'spcustomer': {
                'name': booking.spcustomer.profile.name,
            },
            'spdoctor': {
                'profile': {
                    'name': booking.spdoctor.name
                }
            }
        }
    )

Return render_template('admin/reservasi/reservasi.html',
    results=results)

```

Perubahan status yang admin lakukan dari pending menjadi terjadwal dapat dilihat pada gambar 4.30.



Gambar 4.30 Perubahan Status

Dengan implementasi code perubahan status oleh admin dapat dilihat pada code dibawah.

```

@app.route('/update-reservasi-status/<booking_id>',
    methods=['POST'])
@login_required
def update_reservasi_status(booking_id):
    data = request.get_json()
    status = data.get('status')

    booking = Reservasi.query.filter_by(id=booking_id).first()
    if booking:
        booking.status = status
        db.session.commit()
        return jsonify({'success': True}), 200
    return jsonify({'success': False}), 400

```

4.5.2 Fitur Jadwal Konsultasi

Fitur jadwal konsultasi pada admin adalah data pelanggan yang melakukan jadwal konsultasi dengan dokter secara *face to face*, dimana untuk mengenali jenis kulit dan gejala pelanggan yang belum terdaftar di sistem rekomendasinya. Fitur dapat dilihat pada gambar 4.3.

No	Nama Pasien	Nama Dokter	Telp Pasien	Tanggal Konsultasi	Jadwal Konsult	Status
1	afnisa sondari	dr.Febiola Fujjoko	082234564345	Jumat, 28-06-2024	11.00 s.d 12.00 W	Pending
2	nata tuti	dr.Febiola Fujjoko	083825520613	Kamis, 27-06-2024	11.00 s.d 12.00 W	Terjadwal

Gambar 4.31 Fitur Konsultasi

Code tampilan dan update status onsultasi dokter bisa dilihat pada code yang ada dibawah.

```
viewDoctor = Booking.query.all()
results = []
for booking in viewDoctor:
    results.append(
        {
            'id': booking.id,
            'telp': booking.telp,
            'time': booking.time,
            'date': format_date_with_day(booking.date),
            'status': booking.status,
            'spcustomer': {
                'name': booking.spcustomer.profile.name,
            },
            'spdoctor': {
                'profile': {
                    'name': booking.spdoctor.name
                }
            }
        }
    )
return render_template('admin/konsultasi/konsultasi.html',
results=results)
```

4.5.3 Fitur Konsultasi Kulit

Pada fitur konsultasi kulit pelanggan akan mendapatkan rekomendasi terkait treatment apa yang harus di ambil dengan kecocokan data yang diinput

seperti gejala dan jenis kulit pelanggan. Tampilan fitur form pemilihan rekomendasi dapat dilihat pada gambar 4.32.

Gambar 4.32 Form Pemilihan Rekomendasi

Code pemilihan rekomendasi pelanggan pada fitur kosultasi kulit dapat dilihat dibawah.

```
def hitung_kesamaan(kasus_baru, kasus_lama):
    kesamaan = 0

    for gejala in kasus_baru['gejala'].split(","):
        gejala = gejala.strip().lower()
        if gejala in kasus_lama.symptom.lower():
            kesamaan += 0.6

    for jenis_kulit in kasus_baru['jenis_kulit'].split(","):
        jenis_kulit = jenis_kulit.strip().lower()
        if jenis_kulit in kasus_lama.skin.lower():
            kesamaan += 0.4

    return kesamaan
```

Setelah pelanggan berhasil menginput gejala dan jenis kulit, maka sistem akan mengirimkan rekomendasi kepada pelanggan dengan hasil ada pada gambar 4.33. dimana pada gambar 4.33 ini merupakan tampilan fitur sebelum dan sesudah di evaluasi. Pada gambar sebelumnya tampilan tidak ada fitur reservasi dokter untuk memesan treatment dari hasil rekomendasi.

Gambar 4.33 Hasil Rekomendasi

Pada tampilan gambar 4.33 implementasi dari code yang sudah melakukan proses perulangan sehingga tampilan hasil rekomendasi berubah. Code pertama sebelum perubahan terdapat pada code dibawah.

```

if rekomendasi_vote_count > 1:
    rekomendasi = rekomendasi_voting
else:
    rekomendasi = rekomendasi_bobot
combined_scores = {}
for treatment in vote_counts:
    combined_scores[treatment] = vote_counts[treatment] +
    weighted_vote_scores.get(treatment, 0)
    rekomendasi_terbaik=max(combined_scores,key=combined_scores
    .get)
total_votes = sum(vote_counts.values())
vote_percentages = {k: (v / total_votes) * 100 for k, v in
vote_counts.items()}
print(f"Vote Percentages: {vote_percentages}")
print(f"Rekomendasi Voting: {rekomendasi_voting}dengan
{rekomendasi_vote_count} votes")
print(f"Rekomendasi Bobot: {rekomendasi_bobot} dengan score
{rekomendasi_bobot_score}")
print(f"Rekomendasi Terbaik: {rekomendasi_terbaik} dengan
score {combined_scores[rekomendasi_terbaik]}")
if rekomendasi:

    final_rekomendasi=Treatment.query.filter_by(treatment=re
    komendasi).first()
    Returnrender_template('user/rekomendasi.html',rekomendas
    i=final_rekomendasi)
else:
    flash('Tidak ada rekomendasi perawatan yang ditemukan
    untuk gejala dan jenis kulit ini.', 'error')
return redirect(url_for('cbf_algoritma.input_gejala'))

```

untuk code penambahan tampilan reservasi dimana tampilan reservasi mengalami perubahan sebelum dan sesudah evaluasi tes UAT, code dapat dilihat dibawah.

```
def reservasi():
    doctors = Doctor.query.all()

    gejala = request.args.get('gejala', '')
    jenis_kulit = request.args.get('jenis_kulit', '')
    treatment = request.args.get('treatment', '')
    types = request.args.get('types', '')
    benefit = request.args.get('benefit', '')
    information = request.args.get('information', '')

    doctor_time_slots = {doctor.id: get_time_slots(doctor.time) for
doctor in doctors}

    if request.method == 'POST':
        doctor = request.form.get('doctor')
        treatmentName = request.form.get('treatmentName')
        date = request.form.get('date')
        time = request.form.get('time')

        if not all([treatmentName, doctor, time, date]):
            flash('Semua bidang harus diisi.', 'error')
            return redirect(url_for('home.formkonsultasi', id=id))

        new = Reservasi(
            treatment=treatmentName,
            time=time,
            date=date,
            status="waiting",
            customer_id=current_user.id,
            doctor_id=doctor,
        )
        db.session.add(new)
        db.session.commit()

        flash('Jadwal reservasi berhasil dibuat.', 'success')
        return redirect(url_for('cbf_algoritma.data_reservasi'))
```

4.6 PEMBOBOTAN CBR

Berikut adalah proses pembobotan dari CBR dalam system rekomendasi system informasi klinik XYZ, untuk mengetahui hasilnya terlebih dahulu melakukan data antara gejala dan jenis kulit pelanggan klinik XYZ, data dapat dilihat pada tabel 4.1 tentang kasus gejala baru, kasus lama gejala, jenis kulit baru dan jenis kulit lama..

Tabel 4. 1 Tabel kasus lama dan kasus baru gejala

Gejala baru	gejala lama
Jerawat, peradangan, Kusam, bertekstur	berjerawat
Peradangan	Kulit bertekstur
Kusam	Kusam
Bertekstur	Noda atau bekas jerawat
	warna kulit yang tidak merata, kurangnya kilau atau cahaya pada kulit, dan tekstur kulit yang kasar.

$$\text{Similarity} = \left(\frac{1*0.6+0*0.6+1*0.6+1*0.6}{0.6+0.6+0.6+0.6} \right) = \frac{1.8}{3} = 0,6$$

Dari perhitungan manual diatas untuk gejala 1 dengan kasus lama dan kasus baru menghasilkan persamaan 0,4. Kemudian menghitung jenis kulit baru dan lama dengan rumus yang sama. Tabel pembobotan jenis kulit terdapat pada tabel 4.2.

Tabel 4. 2 tabel jenis kulit baru dan jenis kulit lama

Jenis kulit baru	Jenis kulit lama
Kulit kering, normal, berjerawat	kulit berjerawat
Normal	kusam
berjerawat	bertekstur

$$\text{Similarity} = \left(\frac{0*0.4+0*0.4+1*0.4}{0.4+0.4+0.4} \right) = \frac{0.4}{1.2} = 0,33$$

Dari perhitungan manual untuk jenis kulit mengasilkan persamaan 0,33. dari pembobotan satu treatment pertama mendapatkan hasil :

$$S = \text{hasil gejala} + \text{hasil jenis kulit} = 0,6 + 0,33 = 0,93$$

Dari pengujian CBR pada sistem informasi klinik XYZ menghasilkan *similarity* 0,93. Dengan hasil pembobotan gejala 0,6 dan jenis kulit 0,33.

PERPUSTAKAAN
UNIVERSITAS JENDERAL ACHMAD YANI
YOGYAKARTA