



JURNAL INFORMATIKA RESONAT ALGORITHMUS

<https://jurnal.unipi.ac.id/index.php/ALGORITHMUS>

ISSN ONLINE : 3164-0230

RANCANG BANGUN APLIKASI LAYANAN BIRO JODOH “LOVESTREET” MENGUNAKAN ALGORITMA *JACCARD SIMILARITY* DAN MBTI BERBASIS WEB

Silmi Kamila Ihsan¹, Errysa Subekthi², Miftahul Sidiq³

¹ Informatika, Universitas Persatuan Islam

² Informatika, Universitas Persatuan Islam

³ Informatika, Universitas Persatuan Islam

Jl. Peta no. 154 Suka Asih, Bandung, Jawa Barat

ibnuhsan28@gmail.com, errysa@unipi.ac.id, sidiq.miftah@gmail.com

Abstract

In the digital era, the demand for an efficient, secure, and measurable matchmaking system continues to grow. This study aims to design and develop a web-based matchmaking service application called Lovestreet, which is intended to help users find life partners more accurately and personally. The application was developed using the CodeIgniter 4 framework and the Software Development Life Cycle (SDLC) with the Waterfall model as its development methodology. The matchmaking process integrates the Jaccard Similarity algorithm and the Myers-Briggs Type Indicator (MBTI) to determine compatibility between users. The Jaccard Similarity algorithm is used to calculate similarities based on user attributes and preferences, while the MBTI method is utilized to analyze complementary personality types. The system provides several main features, including partner recommendation, user profile management, biodata request, and meeting scheduling (ta'aruf). The results show that the developed application can deliver efficient and secure matchmaking services that recommend partners based on user preferences and personality characteristics. In conclusion, the Lovestreet system provides a matchmaking service that combines the Jaccard Similarity algorithm and MBTI personality analysis to generate an objective compatibility score (match score) between users based on shared characteristics and complementary personality factors.

Keywords: Partner recommendation, Matchmaking system, CodeIgniter 4, Web application, Jaccard Similarity, MBTI.

Abstrak

Dalam era digital, kebutuhan masyarakat terhadap sistem pencarian pasangan yang aman, efisien, dan terukur semakin meningkat. Penelitian ini bertujuan untuk merancang dan membangun sebuah aplikasi layanan biro jodoh berbasis web bernama Lovestreet, yang dirancang untuk membantu pengguna dalam menemukan pasangan hidup secara lebih akurat dan personal. Aplikasi ini dikembangkan menggunakan framework CodeIgniter 4 dengan metode pengembangan perangkat lunak Software Development Life Cycle (SDLC) model Waterfall. Proses pencocokan pasangan pada sistem ini menerapkan kombinasi algoritma Jaccard Similarity dan MBTI (Myers-Briggs Type Indicator) untuk menentukan tingkat kesesuaian antar pengguna. Algoritma Jaccard digunakan untuk menghitung kesamaan berdasarkan atribut dan preferensi pengguna, sedangkan MBTI digunakan untuk mengidentifikasi kecocokan kepribadian yang saling melengkapi. Fitur utama sistem meliputi rekomendasi pasangan, manajemen profil pengguna, permintaan biodata, serta penjadwalan pertemuan (ta'aruf). Hasil penelitian menunjukkan bahwa aplikasi yang dikembangkan mampu memberikan rekomendasi pasangan secara efisien, aman, dan sesuai dengan karakteristik kepribadian pengguna. Kesimpulannya, sistem Lovestreet menyediakan layanan pencarian pasangan berbasis algoritma Jaccard Similarity yang dikombinasikan dengan analisis kepribadian MBTI, sehingga dapat menghasilkan tingkat kecocokan (match score) antar pengguna secara objektif berdasarkan kesamaan karakteristik dan kepribadian yang saling melengkapi.

Kata Kunci: Rekomendasi pasangan, Biro jodoh, CodeIgniter 4, Aplikasi web, Jaccard Similarity, MBTI.

1. PENDAHULUAN

Problem memilih pasangan menjadi salah satu problem rumit yang dihadapi pada masa sekarang ini mewabahnya gejala penyimpangan yang menimpa komunitas masyarakat saat ini dan masuknya unsur-unsur psikologis, kultural, dan sosial dari ranah budaya lain ke dalam kehidupan Islam hingga merasuki keinginan, orientasi dan kecenderungannya dalam memilih pasangan hidup yang semakin menjauhkannya dari petunjuk Islam dalam masalah ini. Akhirnya aplikasi biro jodoh *online* memberikan dampak serta bermanfaat bagi masyarakat dengan kesibukan dan keseharian dengan segudang aktivitas padat sehingga tidak memiliki cukup waktu untuk mencari pasangan tetapi memiliki niat untuk menjalin hubungan serius untuk menuju ke jenjang pernikahan [1].

Perkembangan dan kemajuan teknologi saat ini yang semakin canggih membuat menjadi kebutuhan yang sangat penting bagi setiap orang. Oleh karena itu, para pengguna teknologi merasa sangat nyaman dan memudahkan pekerjaan mereka, dan memudahkan setiap individu untuk berkomunikasi dengan orang lain. Pengguna teknologi juga banyak yang memanfaatkan teknologi informasi untuk mencari hal-hal yang bersifat pribadi, seperti mencari jodoh atau pasangan. Salah satu yang memberikan pengaruh terbesar terhadap perkembangan teknologi tersebut adalah dunia hiburan yang menggunakan teknologi sebagai media hiburannya. Pasalnya, manusia tidak bisa lepas dari teknologi dan lebih memilih melakukan segala hal melalui perangkat seluler atau komputer. Media hiburan yang dimaksud bisa berupa berbagai macam aplikasi, antara lain aplikasi toko *online*, game, penjualan tiket pesawat, biro jodoh *online*, dan lain sebagainya [3].

Biro jodoh adalah sebuah layanan yang dapat membantu seseorang untuk menemukan pasangan hidup. Terdapat banyak aturan bagi klien yang ditetapkan oleh beberapa penyedia layanan perjodohan untuk menemukan pasangan yang diinginkan. Biro jodoh yang banyak ditemui adalah bertipe *offline* (konvensional). Biro jodoh konvensional biasanya mempunyai beberapa kekurangan, yaitu klien harus dating ke lokasi kantor biro jodoh untuk mendaftar, yang sangat memakan waktu dan minimnya privasi [1].

Masalah diatas ini yang kemudian ingin diangkat untuk mencari solusi bagaimana menerapkan teknologi terhadap layanan biro jodoh sehingga dapat diakses dimana saja, kapan saja tanpa harus mendatangi kantor tersebut hanya dengan menggunakan teknologi *internet*. Biro jodoh *online* memberikan fasilitas bagi masyarakat yang sedang mencari pasangan hidup ketika terlalu sibuk dengan pekerjaannya sehingga menyerahkan pencarian pasangan hidup kepada pihak lain. Layanan biro jodoh *online* lebih populer dan banyak diminati karena dianggap lebih rahasia dan privasi, karena proses Pendaftaran dapat dilakukan dengan mengisi formulir yang tersedia di website biro jodoh sehingga privasi tetap terjaga dan klien tidak perlu lagi mendatangi kantornya.

Sistem ini juga dapat memberikan kontribusi positif terhadap pengembangan hubungan interpersonal dalam masyarakat, merangsang pertumbuhan positif dalam hal

membina hubungan yang sehat dan berkelanjutan. Selain itu, sistem ini juga memberikan alternatif bagi individu yang mungkin sulit menemukan pasangan hidup secara tradisional, seperti mereka yang memiliki kesibukan tinggi atau berada di lingkungan yang minim kesempatan untuk bertemu dengan calon pasangan [3]. Dengan mempertimbangkan aspek-aspek tersebut, rancang bangun sistem informasi biro jodoh berbasis *website* menjadi sebuah langkah yang signifikan dan relevan dalam mengatasi tantangan di *era digital*.

Penelitian ini bertujuan untuk merancang dan membangun sistem yang bisa memfasilitasi seseorang dalam proses pencarian pasangan hidup dengan lebih efisien dan efektif melalui platform *online*. Dengan demikian, rancang bangun sistem informasi biro jodoh berbasis website dapat menjadi solusi yang tepat untuk menjawab kebutuhan masyarakat dalam perihal mencari pasangan hidup. Penelitian ini diharapkan dapat memberikan andil positif dalam meningkatkan kualitas layanan di bidang biro jodoh serta memperkaya literatur penelitian di bidang sistem informasi dan teknologi aplikasi berbasis *web*.

2. TINJAUAN PUSTAKA

2.1. Aplikasi Berbasis Web

Aplikasi berbasis web (*Web Based Application*) adalah aplikasi yang dirancang khusus agar dapat berjalan pada web *browser* [10]. Pengguna yang menggunakan web *browser* seperti Chrome, Firefox, Safari, atau Microsoft Edge dapat mengakses dan menggunakan aplikasi web melalui alamat URL atau tautan yang diberikan.

Aplikasi berbasis web memiliki keunggulan yaitu dapat digunakan melalui *internet* tanpa memerlukan instalasi tambahan. Selain itu, aplikasi ini dapat diakses melalui berbagai perangkat seperti komputer, laptop, tablet, dan *smartphone*. Aplikasi berbasis web berjalan melalui peramban (*browser*) dan membutuhkan koneksi *internet* yang stabil, sementara aplikasi *desktop* atau *mobile (offline)* dapat berjalan secara langsung pada perangkat pengguna tanpa memerlukan koneksi *internet* [11].

2.2. Biro Jodoh

Menurut KBBI, biro jodoh merupakan badan usaha jasa untuk menjodohkan pria atau wanita. mengatakan biro jodoh adalah sebuah layanan yang membantu seseorang pencari pasangan dalam menemukan pasangan hidup, banyak hal atau aturan-aturan yang dibuat oleh layanan biro jodoh tertentu untuk mempertemukan seorang member layanan tersebut pada pasangan yang didambakannya (Kemas Faerussy Nizar, dkk. 2010).

Sebelum adanya *internet*, pencarian dan pemilihan jodoh diatur dan diawasi oleh orang tua. Kalaupun ada biro jodoh yang tersedia, yaitu bersifat konvensional, artinya mereka tidak perlu berkomunikasi melalui media sosial, dan sebaliknya harus melakukan pertemuan secara langsung. Salah satu kekurangan biro jodoh konvensional adalah mengharuskan pendaftaran di tempat penyedia biro jodoh *offline*. Oleh karena itu, waktu dan prosedur yang dibutuhkan dinilai kurang efisien dan efektif di seluruh area. Namun, keuntungan dari biro jodoh *offline* adalah bahwa tidak ada kejahatan *online* (*cyber crime*) dan setiap pengguna tidak dapat membohongi data mereka [1].

2.3. Bahasa Pemrograman

1) HTML

HTML (*Hyper Text Markup Language*) adalah sebuah bahasa *markup* yang digunakan untuk membuat halaman web dan menampilkan berbagai informasi dalam *browser*. Ini berasal dari bahasa yang lebih dulu populer. Di industri penerbitan dan percetakan yang dikenal sebagai SGML (*Standard Generalized Markup Language*). HTML merupakan sebuah standar umum untuk menampilkan halaman web. Saat ini, HTML adalah standar *internet* yang ditetapkan dan dikelola oleh *World Wide Web Consortium* (W3C).

2) CSS

Cascading Style Sheet (CSS) bukanlah bahasa pemrograman, tetapi merupakan aturan untuk mengatur beberapa komponen web sehingga lebih seragam dan terstruktur. Parameter seperti ukuran gambar, warna bagian tubuh teks, warna tabel, ukuran *border*, warna *border*, warna *hyperlink*, warna *mouse over*, spasi antar paragraf dan teks, dan margin kiri, kanan, atas, dan bawah semuanya dapat diatur oleh CSS [12].

3) Javascript

Javascript adalah Bahasa pemrograman tingkat tinggi yang digunakan dalam pengembangan website agar lebih dinamis dan interaktif. *Website* dinamis berarti konten didalamnya dapat bergerak atau mengubah tampilan layar tanpa harus dimuat ulang secara manual. Contoh konten dinamis termasuk gambar animasi, *maps*, *slideshow*, *polling*, dan lainnya [13].

4) PHP

PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman *open source* sisi server (*server-side*) yang biasa digunakan dalam pengembangan *website*. Karena PHP bekerja (membuat, membuka, membaca, menulis, dan membatasi *file*) di *server-side*, maka skrip PHP diproses di *server* sehingga tidak akan muncul di *sisi client* (*browser*) [14].

2.4. CodeIgniter 4

CodeIgniter 4 merupakan kerangka kerja (*framework*) berbasis PHP yang dirancang untuk membangun aplikasi web dengan cepat dan efisien. Versi 4 menonjolkan struktur aplikasi yang lebih modern, termasuk pemisahan direktori yang lebih baik serta dukungan terhadap standar-autoloading dan arsitektur MVC (*Model-View-Controller*). Dengan demikian, pengembang dapat mengorganisasikan komponen aplikasi secara sistematis dan meminimalkan konfigurasi manual.

Salah satu keunggulan utama CI4 terletak pada performanya yang ringan dan *footprint* yang kecil. Dokumentasi resminya menyebutkan bahwa CI4 mengutamakan kemudahan penggunaan tanpa mengorbankan fleksibilitas, serta menyediakan berbagai pustaka (*library*) bawaan yang siap pakai guna mempercepat pembuatan modul aplikasi. CI4 juga memperkuat aspek keamanan dengan menyertakan mekanisme proteksi terhadap serangan umum seperti CSRF (*Cross-Site Request Forgery*) dan XSS (*Cross-Site Scripting*).

2.5. MySQL

MySQL adalah salah satu jenis DBMS atau *Database Management System* (DBMS). DBMS adalah aplikasi yang digunakan untuk mengolah dan *manage* data dalam jumlah besar data atau disebut dengan big data. Aplikasi DBMS bekerja sama dengan aplikasi *server* dalam hal pengelolaan data, jadi kita hanya perlu memberikan perintah kepada sistem DBMS untuk melakukan proses seperti menambah, mengubah, menghapus, atau mengambil data. Beberapa jenis DBMS yang sangat populer dan banyak digunakan saat ini adalah MySQL, PostgreSQL, MongoDB, dan Oracle [15].

2.6. Penelitian Terdahulu

Untuk penelitian ini, data pendukung sangat penting untuk digunakan sebagai acuan atau tolak ukur dalam penyusunan penelitian. Oleh karena itu, penelitian ini juga menggunakan teori dari penelitian sebelumnya. Teori-teori yang digunakan dalam penelitian ini dijelaskan dalam rincian berikut berikut.

- 1) Devi Azwinda, judul “Analisis Terhadap Biro Jodoh Online: Kebutuhan Atau Tuntutan?” [17].
- 2) Regita Amelia, Rizqa Febry Ayu, judul “Biro Jodoh Online: Kegunaan Dan Dampak” [3].
- 3) Annisa Rahmawati Nento, judul “Sistem Informasi Administrasi Ta’aruf” [7].
- 4) Kemas Faerusy Nizar, Setiawardhana, Idris Winarno, judul “Aplikasi Biro Jodoh Berbasis Web Dan J2ME” [2].
- 5) Ekky Bayu Pramana, Aris Sunandar, judul “Analisa Rancang Bangun Aplikasi Biro Jodoh Islami Menggunakan Algoritma *Profile Matching* Berdasarkan Platform Android” [4].

3. METODE PENELITIAN

3.1. Metode Pengumpulan Data

Pada penelitian ini menggunakan 3 metode pengumpulan data, yaitu observasi, studi literatur, wawancara.

1) Observasi

Menurut Patton Patton (2017), observasi merupakan metode pengumpulan data yang bersifat akurat dan spesifik untuk mengumpulkan data dan mencari informasi mengenai segala kegiatan yang dijadikan objek penelitian [16].

Dalam hal ini dilakukan observasi dengan cara pengamatan pada aplikasi atau situs web penyedia layanan biro jodoh *online*.

2) Studi literatur

Menurut M. Nazir (2013) dalam bukunya yang berjudul Metode Penelitian, studi kepustakaan atau studi literatur adalah teknik pengumpulan data dengan melakukan penelaahan terhadap literatur, buku, catatan, dan laporan yang terkait dengan masalah yang dibahas [9].

Studi literatur sejenis dalam penelitian dilakukan untuk mendapatkan pemahaman yang menyeluruh tentang metode dan hasil penelitian sebelumnya, dan untuk menentukan seberapa berbeda penelitian yang akan dilakukan. Ini penting karena kita dapat menghindari usaha yang sebenarnya sudah dilakukan orang lain dan dapat digunakan pada penelitian kita untuk menghemat waktu, tenaga, dan biaya. Selain itu, penting untuk menetapkan langkah-langkah penelitian tambahan yang diperlukan untuk melanjutkan misi penelitian.

3) Wawancara

Wawancara merupakan kegiatan tanya jawab secara langsung untuk memperoleh berbagai informasi dari narasumber. Informasi yang diperoleh dapat disajikan dalam bentuk tulisan, rekaman suara, *video*, ataupun gabungan keduanya. Dalam kegiatan penelitian, wawancara menjadi salah satu metode penting untuk mendukung proses pengamatan.

Pelaksanaan wawancara dapat dilakukan secara langsung, yaitu dengan menemui dan berbicara langsung kepada orang yang memiliki informasi yang dibutuhkan, maupun tidak langsung, yaitu melalui pihak lain yang dianggap memahami atau mengetahui kondisi dari subjek penelitian. Melalui proses tanya jawab ini, terjadi pertukaran gagasan dan informasi yang membantu peneliti memahami makna suatu topik secara lebih mendalam.

Pada tanggal 21 Agustus 2024, penulis mengunjungi kantor Rumaysho yang bertempat di Yogyakarta sebagai Lembaga Layanan Biro Bodoh yang telah berpengalaman sejak tahun 2017 dengan tujuan tiada lain untuk menggali informasi lebih dalam dan juga sebagai pedoman atau acuan mengenai penelitian ini agar lebih terarah.



Gambar 3. 1 Proses wawancara di kantor Rumaysho, Yogyakarta

3.2. Metode Pengembangan Sistem

Pada penelitian ini penulis menggunakan metode pengembangan sistem *waterfall*. Metode *waterfall* adalah metode kerja SDLC (*Software Development Life Cycle*) yang menekankan fase-fase yang berurutan dan sistematis. Disebut *waterfall* karena proses mengalir satu arah "ke bawah", mirip dengan air terjun [8]. Tahap-tahap yang ada dalam metode ini harus dilakukan secara berurutan.

Penulis memilih pengembangan sistem menggunakan metode *waterfall* karena memiliki beberapa kelebihan, diantaranya sebagai berikut.

- 1) Memiliki proses yang teratur, sehingga pengerjaan dapat terjadwal dengan baik dan mudah.
- 2) Cocok untuk sistem dengan kompleksitas rendah (*predictable*) seperti aplikasi yang akan dikembangkan oleh penulis ini.
- 3) Setiap proses yang dilakukan tidak dapat saling tumpah tindih.

Dibawah ini merupakan tahapan dalam pengembangan metode *waterfall*.

1) Requirement gathering and analysis

Mengumpulkan semua kebutuhan untuk dianalisis dan menentukan apa yang harus dipenuhi program. Informasi dapat diperoleh melalui wawancara, diskusi, atau survey.

2) Design

Sebelum kode dibuat, lakukan perancangan desain perangkat lunak sebagai perkiraan. Bisa menggunakan *Flowchart*, *Mind Map*, atau *Entity Relationship Diagram* (ERD) untuk membuat desain sistem.

3) Implementasi

Seluruh desain yang telah dibuat sebelumnya diubah menjadi kode program. Kode yang dihasilkan masih berbentuk modul, yang harus digabungkan di tahap berikutnya.

4) Integration & Testing

Pada tahap ini, modul-modul yang telah dibuat sebelumnya digabungkan dan diuji untuk memastikan apakah perangkat lunak yang dibuat sesuai dengan desain dan fungsinya.

5) Verification

Pada tahap ini, pengguna atau klien langsung menguji sistem untuk melihat apakah ia sesuai dengan yang disetujui atau tidak.

6) Operation & Maintenance

Tahap terakhir dari model *waterfall*. Pada tahap ini, sistem telah dijalankan dan dilakukan *maintenance*. *Maintenance* adalah proses memperbaiki kesalahan yang tidak ditemukan pada tahap sebelumnya.

4. ANALISIS DAN PERANCANGAN SISTEM

4.1 Proses Sistem Berjalan

Layanan biro jodoh tradisional merupakan sistem yang berfungsi sebagai perantara antara individu yang sedang mencari pasangan. Seluruh prosesnya masih dilakukan secara langsung dan manual, mulai dari pendaftaran hingga tahap pertemuan antara calon pasangan.

Tahap awal biasanya dimulai ketika calon anggota datang ke kantor biro untuk mendaftar. Pada proses ini, petugas akan melakukan wawancara untuk memperoleh informasi dasar seperti identitas diri, latar belakang pendidikan, pekerjaan, dan kriteria pasangan yang diharapkan. Data tersebut kemudian dicatat dalam formulir pendaftaran dan disimpan dalam arsip fisik.

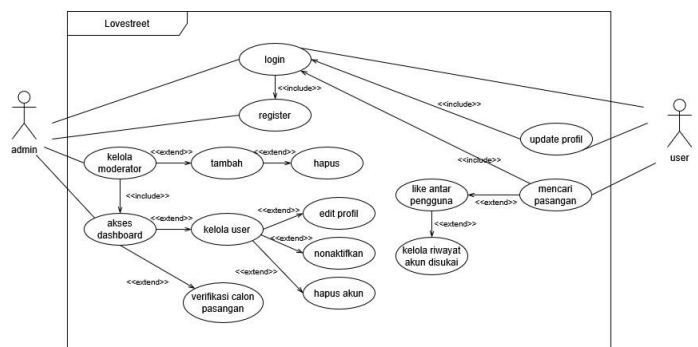
Setelah data terkumpul, pihak biro melakukan proses verifikasi dan pencocokan. Verifikasi dilakukan untuk memastikan keaslian data dan keseriusan calon anggota. Selanjutnya, pencocokan dilakukan dengan meninjau kesesuaian antara profil satu anggota dengan anggota lainnya berdasarkan kriteria yang telah disebutkan. Proses ini biasanya bergantung pada pengalaman dan pertimbangan pengelola biro, bukan menggunakan sistem digital atau algoritma tertentu.

Jika terdapat kecocokan, biro akan mengatur pertemuan awal (ta'aruf) antara dua calon anggota. Pertemuan ini dilakukan secara langsung dengan pengawasan pihak biro untuk menjaga kenyamanan dan etika selama proses berlangsung. Dari hasil pertemuan tersebut, kedua pihak dapat memutuskan apakah hubungan akan dilanjutkan ke tahap berikutnya atau tidak.

Meskipun sistem biro jodoh tradisional memiliki kelebihan dalam hal kepercayaan dan interaksi *personal*, masih terdapat beberapa keterbatasan. Proses administrasi yang bersifat manual sering kali menyebabkan keterlambatan layanan dan potensi kesalahan pencatatan. Selain itu,

jangkauan calon pasangan juga relatif terbatas karena bergantung pada lokasi dan jaringan biro yang bersangkutan. Namun, sistem ini tetap memiliki nilai positif, terutama dalam hal validasi data dan penilaian karakter calon anggota. Melalui interaksi tatap muka, pengelola dapat menilai kepribadian dan kesungguhan anggota secara langsung. Dengan demikian, biro jodoh tradisional masih dianggap relevan bagi sebagian masyarakat yang mengutamakan proses pencarian pasangan yang lebih aman, serius, dan terarah.

4.2. Perancangan Sistem



Gambar 4. 1 Use case diagram

Pada *use case diagram* ini, terdapat 2 aktor yaitu admin dan user yang mana masing-masing memiliki fitur dan perannya tersendiri. Fitur dan perannya adalah sebagai berikut.

1) User

- 1) Registrasi, user dapat melakukan registrasi untuk membuat akun baru dengan mengisi data yang diperlukan agar dapat terdaftar dan dikenali oleh sistem.
- 2) Login, user melakukan proses *login* sebagai mekanisme autentikasi untuk memperoleh akses ke seluruh fitur yang tersedia dalam sistem.
- 3) Memperbarui Profil, user dapat mengubah dan melengkapi data profil pribadi yang digunakan sebagai informasi utama dalam proses pencarian pasangan dan interaksi antar pengguna.
- 4) Mencari Pasangan, user dapat melakukan pencarian pasangan berdasarkan data dan kriteria yang tersedia dalam sistem sebagai bagian dari proses taaruf.
- 5) Interaksi Antar Pengguna, user dapat melakukan interaksi dengan pengguna lain yang relevan dalam rangka menjalankan proses taaruf sesuai dengan ketentuan sistem.
- 6) Mengelola Riwayat Like, user dapat mengakses dan mengelola riwayat aktivitas dan meminta biodata pasangan.

2) Admin

- 1) Mengakses Dashboard, admin dapat mengakses dashboard sebagai pusat kendali untuk memantau dan mengelola seluruh aktivitas sistem.
- 2) Mengelola Data User, admin memiliki kewenangan untuk menambah, mengubah, menonaktifkan, serta

menghapus data akun pengguna sesuai dengan kebutuhan pengelolaan sistem.

- 3) Mengelola Moderator, admin dapat melakukan pengelolaan akun moderator yang berperan dalam membantu pengawasan calon pasangan dan operasional sistem.
- 4) Verifikasi Calon Pasangan, admin bertugas melakukan verifikasi terhadap data calon pasangan untuk memastikan kesesuaian informasi yang digunakan dalam proses taaruf.

5. HASIL DAN PEMBAHASAN

5.1. Implementasi Tampilan (UI/UX)

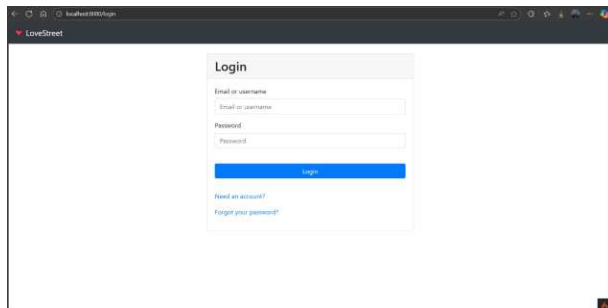
1) Halaman *Landing Page*



Gambar 5. 1 Antarmuka halaman landing page

Halaman ini berfungsi sebagai titik awal interaksi antara pengguna dan platform.

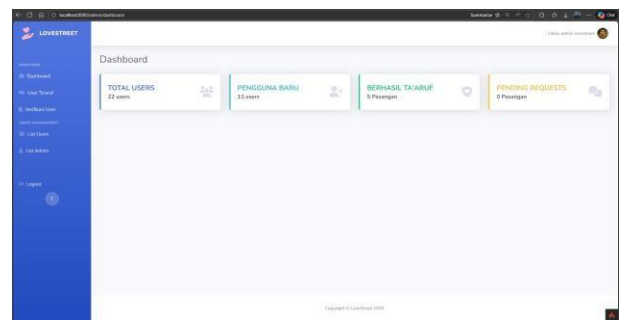
2) Halaman *Login*



Gambar 5. 2 Antarmuka halaman login

Halaman ini berfungsi sebagai gerbang autentikasi yang memungkinkan pengguna untuk mengakses fitur utama sistem sesuai hak aksesnya.

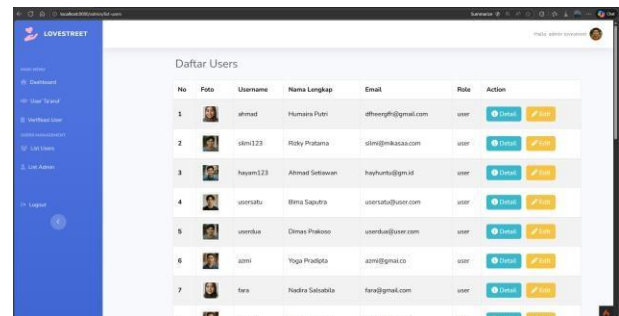
3) Halaman *Dashboard Admin*



Gambar 5. 3 Antarmuka halaman dashboard admin

Halaman *dashboard admin* ini berfungsi sebagai pusat kendali utama yang menampilkan rangkuman informasi penting terkait aktivitas dan data pengguna. Antarmuka ini terdiri atas panel navigasi di sisi kiri dengan menu seperti *Dashboard*, *User Management*, *List Users*, dan *User Ta'aruf*, yang memudahkan *admin* berpindah antarhalaman.

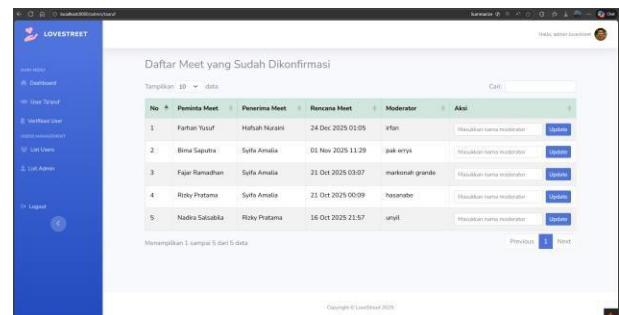
4) Halaman *List Users*



Gambar 5. 4 Antarmuka halaman list users

Halaman ini berfungsi sebagai area pengelolaan data pengguna yang digunakan oleh *admin* untuk memantau, serta memperbarui informasi akun yang telah terdaftar.

5) Halaman *List Confirmed Meet (Ta'aruf)*

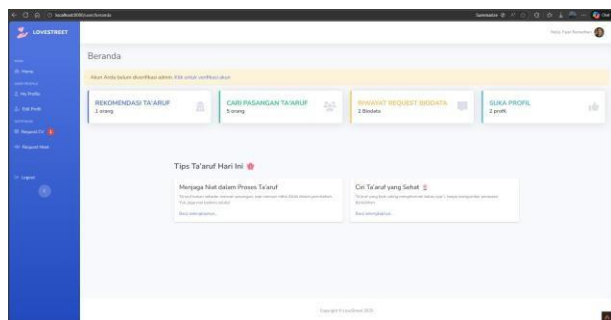


Gambar 5. 5 Antarmuka halaman list confirmed meet (ta'aruf)

Halaman ini berperan sebagai pusat pengelolaan aktivitas *ta'aruf* yang telah disetujui, serta menjadi penghubung

antara proses verifikasi data pengguna dengan pelaksanaan kegiatan pertemuan yang difasilitasi oleh *moderator*.

6) Halaman *Dashboard User (Home)*



Gambar 5. 6 Antarmuka halaman *dashboard user (home)*

Halaman beranda ini berfungsi sebagai tampilan utama yang memberikan pilihan menu serta akses cepat menuju fitur-fitur utama.

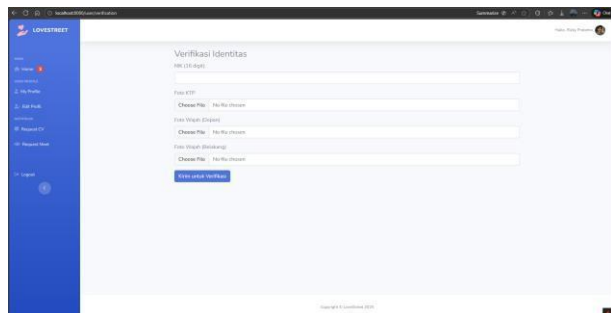
7) Halaman *Edit Profil*



Gambar 5. 7 Antarmuka halaman *edit profil*

Ini merupakan halaman yang sangat penting, karena sebelum mengakses menu cari atau rekomendasi pasangan harus melengkapi biodata terlebih dahulu pada halaman *edit profil*. Dimana, setiap *input* yang kita berikan pada masing-masing *form* akan berpengaruh terhadap hasil pencarian atau rekomendasi pasangan nantinya.

8) Halaman *User Approval*



Gambar 5. 8 Antarmuka halaman *user approval*

Halaman ini berfungsi untuk meng-upload data dan dokumen berupa NIK foto KTP, badan depan, belakang yang nantinya akan di-approve oleh admin.

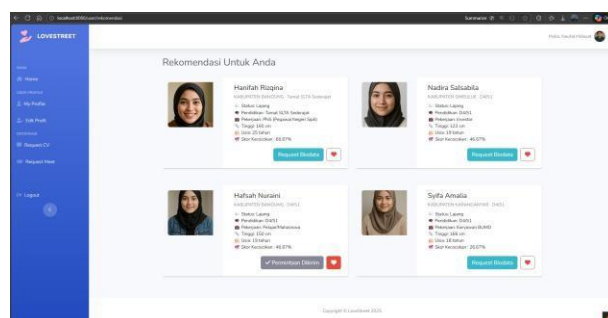
9) Halaman *User Approval (Admin)*



Gambar 5. 9 Antarmuka Halaman Admin *Approval*

Halaman ini merupakan dari sisi admin yang berfungsi untuk melakukan *approval* atau verifikasi data *user*.

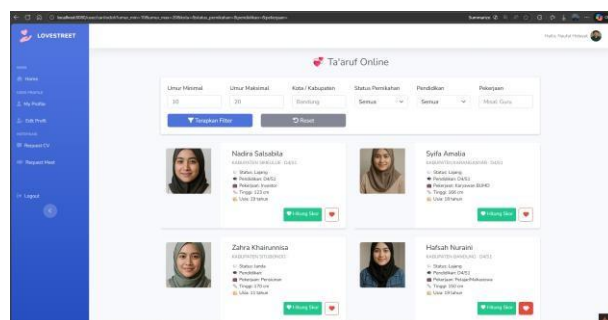
10) Halaman Rekomendasi Pasangan



Gambar 5. 10 Antarmuka halaman rekomendasi pasangan

Halaman rekomendasi pasangan ini dapat diakses melalui *dashboard user* atau beranda. Jika data pasangan pada *database* ada yang sesuai dengan *user*, maka pada halaman ini akan menampilkan beberapa *user* lain yang memiliki peluang kecocokkan berdasarkan algoritma *jaccard similarity* dan MBTI.

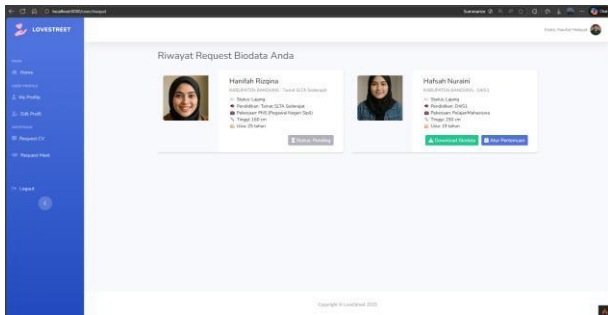
11) Halaman Cari Pasangan (Manual)



Gambar 5. 11 Antarmuka halaman cari pasangan (manual)

Halaman ini adalah untuk mencari pasangan secara manual berdasarkan kriteria yang dipilih. *Parameter* yang digunakan ialah umur, domisili, status pernikahan, pendidikan dan pekerjaan.

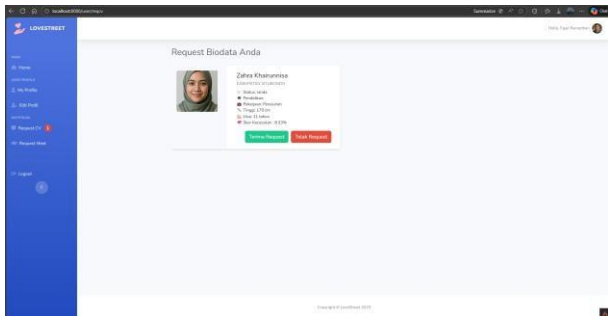
12) Halaman Riwayat Permintaan Biodata (Dikirim)



Gambar 5. 12 Antarmuka halaman riwayat permintaan biodata (dikirim)

Halaman ini akan menampilkan daftar *user* yang telah menerima permintaan biodata mereka untuk kita. Pada halaman ini kita bisa melihat biodata user lain dengan melakukan *download*, dan meminta jadwal pertemuan (*ta'aruf*) dengan *user* yang dipilih.

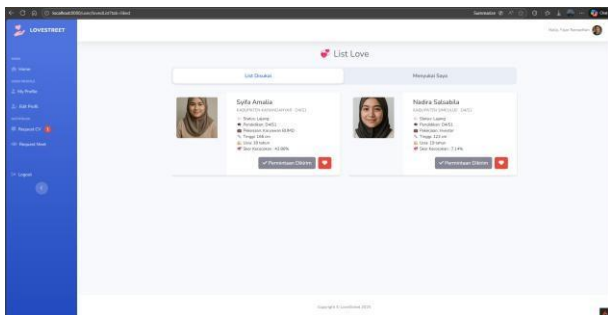
13) Halaman Riwayat Permintaan Biodata (Diterima)



Gambar 5. 13 Antarmuka halaman daftar permintaan biodata (diterima)

Halaman ini akan menampilkan riwayat daftar *user* yang telah mengirimkan permintaan biodata kita. Pada halaman ini kita bisa melakukan tindakan terima dan tolak permintaan.

14) Halaman Daftar Profil yang Disukai

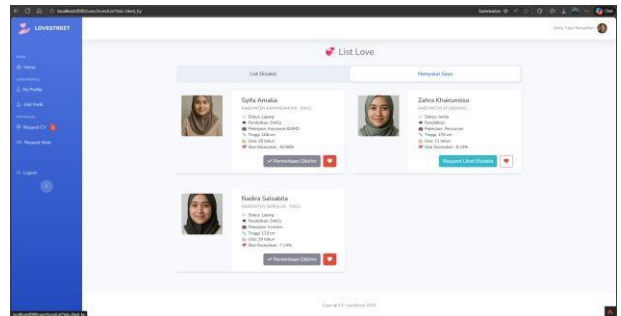


Gambar 5. 14 Antarmuka halaman daftar profil yang disukai

Setelah kita menekan tombol *love* atau *like* pada profil *user* lain, maka *user* tersebut akan masuk ke daftar profil yang disukai pada halaman ini. Tujuannya untuk mengelola *user*

tersebut sebelum kita melakukan permintaan (*request*) biodata kepada *user* yang dipilih.

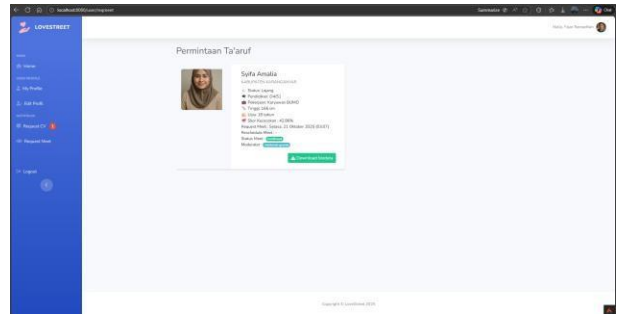
15) Halaman Daftar Profil yang Menyukai



Gambar 5. 15 Antarmuka halaman daftar profil yang menyukai

Halaman ini akan menampilkan daftar *user* yang menyukai kita, tujuannya sama seperti fitur daftar *user* yang disukai.

16) Halaman Riwayat Permintaan Ta'aruf



Gambar 5. 16 Antarmuka halaman daftar permintaan ta'aruf

Halaman ini akan menampilkan daftar *user* yang telah melakukan persetujuan permintaan *ta'aruf* dengan kita. Pada halaman ini kita bisa melihat beberapa data dan informasi dari *user* yang telah mempunyai jadwal *ta'aruf* dengan kita, serta kita juga dapat melakukan aksi *download* biodata.

5.2. Implementasi Kode (Source Code)

1) Kode Fitur Rekomendasi Pasangan

```

78 recommendations = [];
79
80 foreach ($allusers as $user) {
81     $prof = $this->profilmodel
82         ->where('user_id', $user->id)
83         ->first();
84     $pref = $this->preferencemodel
85         ->where('user_id', $user->id)
86         ->first();
87     if (!$prof) continue;
88
89     // ... Filter lawan jenis ...
90     if ($currentProf['jenis_kelamin'] === 'laki-laki' && $pref['jenis_kelamin'] !== 'perempuan') continue;
91     if ($currentProf['jenis_kelamin'] === 'perempuan' && $pref['jenis_kelamin'] !== 'laki-laki') continue;
92
93     $score = $this->calculatematchscore($currentProf, $currentPref, $pref, $pref);
94
95     if ($score >= 20) {
96         $recommendations[] = [
97             'user' => (object) $user,
98             'score' => $score
99         ];
100     }
101 }
102

```

Gambar 5. 17 Kode fitur rekomendasi pasangan

Sistem terlebih dahulu mengambil seluruh data pengguna lain yang berdasarkan perbedaan jenis kelamin agar hanya

lawan jenis saja yang dipertimbangkan. Untuk setiap kandidat yang lolos, fungsi *calculateMatchScore()* digunakan guna menghitung nilai kecocokan antara profil dan preferensi kedua pihak. Kandidat yang memperoleh skor minimal 20 dianggap relevan dan dimasukkan ke dalam daftar rekomendasi, dilengkapi dengan atribut tambahan seperti status permintaan berkas (*hasRequestedCV*) dan status ketertarikan (*isLoved*). Hasil akhir kemudian diurutkan berdasarkan skor tertinggi untuk ditampilkan pada antarmuka pengguna.

2) Kode Fitur Cari Pasangan

```

740     $filtered = [];
741
742     foreach ($allUsers as $user) {
743         $prof = $profileModel->where('user_id', $user->id)->first();
744         if (!$prof) continue;
745         if ($prof['jenis_kelamin'] != $targetGender) continue;
746
747         // --- filter manual ---
748         $umur = date_diff(date_create($prof['tanggal_lahir']), date_create('today'))->y;
749         if ($umur_min && $umur < $umur_min) continue;
750         if ($umur_max && $umur > $umur_max) continue;
751         if ($kota && strpos($prof['kota'], $kota) == false) continue;
752         if ($status && $prof['status_pernikahan'] != $status) continue;
753         if ($pendidikan && $prof['pendidikan_terakhir'] != $pendidikan) continue;
754         if ($pekerjaan && strpos($prof['pekerjaan'], $pekerjaan) == false) continue;
755
756         $filtered[] = [
757             'user' => (object) $user,
758             'profile' => $prof
759         ];
760     }

```

Gambar 5. 18 Kode fitur cari pasangan

Sistem melakukan iterasi terhadap seluruh pengguna lain, kemudian memeriksa data profil masing-masing calon. Setelah memastikan data profil tersedia, sistem menerapkan *filter manual* berdasarkan beberapa *parameter*, seperti rentang usia, kesesuaian lokasi, status pernikahan, tingkat pendidikan terakhir, serta kecocokan bidang pekerjaan. Hanya data pengguna yang memenuhi seluruh kriteria tersebut yang dimasukkan ke dalam variabel *\$filtered*, yang kemudian digunakan sebagai dasar untuk proses rekomendasi pasangan berikutnya.

3) Kode Fitur Hitung Skor Kecocokan

```

1139     public function hitungScore($targetUserId)
1140     {
1141         $currentUser = user()->id;
1142
1143         $currentProf = $this->profileModel->where('user_id', $currentUser)->first();
1144         $currentPref = $this->preferenceModel->where('user_id', $currentUser)->first();
1145
1146         $targetProf = $this->profileModel->where('user_id', $targetUserId)->first();
1147         $targetPref = $this->preferenceModel->where('user_id', $targetUserId)->first();
1148
1149         if (!$currentProf || !$currentPref || !$targetProf || !$targetPref) {
1150             return $this->response->setJSON(['error' => 'data belum lengkap']);
1151         }
1152
1153         $score = $this->calculateMatchScore($currentProf, $currentPref, $targetProf, $targetPref);
1154         return $this->response->setJSON(['score' => $score]);
1155     }

```

Gambar 5. 19 Kode fitur hitung kecocokan calon pasangan

Fungsi ini terlebih dahulu mengambil data profil dan preferensi kedua pengguna dari model *profileModel* dan *preferenceModel* berdasarkan identitas pengguna aktif dan *target*. Setelah data profil dan preferensi berhasil dimuat, sistem memeriksa kelengkapan data untuk memastikan proses perhitungan dapat dilakukan dengan benar.

4) Kode Fitur Like User

```

920     public function toggleLove($likedUserId)
921     {
922         $likeModel = new \App\Models\UserLikeModel();
923         $likerId = user()->id;
924
925         // Cek apakah sudah dilove
926         $existing = $likeModel->isLiked($likerId, $likedUserId);
927
928         if ($existing) {
929             // Jika sudah, hapus love (unlove)
930             $likeModel->delete($existing['id']);
931             return $this->response->setJSON(['status' => 'unliked']);
932         } else {
933             // Jika belum, tambahkan love
934             $likeModel->insert([
935                 'liker_id' => $likerId,
936                 'liked_user_id' => $likedUserId,
937             ]);
938             return $this->response->setJSON(['status' => 'liked']);
939         }
940     }

```

Gambar 5. 20 Kode controller fitur like user

Fungsi *toggleLove()* berperan untuk mengatur aksi pengguna dalam memberikan atau membatalkan tanda suka terhadap pengguna lain. Fungsi ini diawali dengan inisialisasi model *UserLikeModel* serta pengambilan identitas pengguna yang sedang aktif. Sistem kemudian memeriksa apakah relasi “like” antara pengguna aktif dan *target* sudah ada melalui pemanggilan metode *isLiked()*

5) Kode Fitur Kelola Liked User

```

942     public function lovedList()
943     {
944         $likerModel = new \App\Models\UserLikeModel();
945         $userModel = new \App\Models\UserModel();
946         $requestModel = new \App\Models\CvRequest();
947
948         $currentUserId = user()->id;
949
950         // Ambil data user login, profil, dan preferensinya
951         $currentUser = $userModel->find($currentUserId);
952         $currentProf = $this->profileModel->where('user_id', $currentUserId)->first();
953         $currentPref = $this->preferenceModel->where('user_id', $currentUserId)->first();
954
955         // Pastikan profil dan preferensi ada
956         if (!$currentProf || !$currentPref) {
957             return redirect()->to('user/profile')->with('error', 'lengkapi profil dan preferensi terlebih dahulu.');
```

Gambar 5. 21 Kode fitur kelola liked user

Proses dimulai dengan inisialisasi beberapa model yang digunakan. Sistem kemudian mengambil data pengguna aktif berupa profil dan preferensinya untuk memastikan kelengkapan data sebelum melanjutkan proses.

6) Kode Fitur *Request* Biodata

```

404 public function mintacv()
405 {
406     if ($this->request->getMethod() !== 'POST') {
407         return $this->response->setStatusCode(405, 'Method Not Allowed');
408     }
409
410     $requesterId = user_id(); // user login
411     $targetId = $this->request->getVar('target_user_id');
412
413     if (!$targetId) {
414         return $this->response->setJSON([
415             'status' => 'error',
416             'message' => 'Target user tidak ditemukan.'
417         ])->setStatusCode(400);
418     }
419
420     $cvModel = new \App\Models\CvRequest();
421
422     // cek apakah sudah ada request sebelumnya
423     $existing = $cvModel
424         ->where('requester_id', $requesterId)
425         ->where('target_user_id', $targetId)
426         ->whereIn('status', ['pending', 'accepted'])
427         ->first();
428
429     if ($existing) {
430         return $this->response->setJSON([
431             'status' => 'error',
432             'message' => 'Anda sudah merequest CV user ini.'
433         ]);
434     }

```

Gambar 5. 22 Kode fitur *request* biodata

Proses diawali dengan pemeriksaan metode permintaan agar hanya menerima permintaan dengan metode *POST*. Setelah itu, sistem mengambil identitas pengguna yang mengajukan *request* serta identitas pengguna *target* dari data permintaan yang diterima.

7) Kode Fitur Meng-*Update* Status Permintaan Biodata

```

596 // Update status request utama
597 $cvModel->update($requestId, ['status' => $status]);
598
599 if ($status === 'accepted') {
600     $userA = $request['requester_id']; // pengirim awal
601     $userB = $request['target_user_id']; // penerima (yang sekarang accepted)
602
603     // Cek apakah sudah ada request dari B ke A
604     $existing = $cvModel
605         ->where('requester_id', $userB)
606         ->where('target_user_id', $userA)
607         ->whereIn('status', ['accepted', 'pending'])
608         ->first();
609
610     if (!$existing) {
611         $cvModel->insert([
612             'requester_id' => $userB,
613             'target_user_id' => $userA,
614             'status' => 'accepted',
615             'created_at' => date('Y-m-d H:i:s'),
616         ]);
617     }
618 }
619
620 return $this->response->setJSON([
621     'status' => 'success',
622     'message' => 'Permintaan berhasil diubah!'
623 ]);
624

```

Gambar 5. 23 Kode fitur meng-*update* status permintaan biodata

Setelah status utama diperbarui menggunakan metode *update()*, sistem melakukan pengecekan khusus apabila status yang dikirim bernilai *accepted*. Dalam kondisi tersebut, sistem menetapkan dua variabel pengguna, yakni pengirim permintaan awal (*requester_id*) dan penerima permintaan (*target_user_id*) yang kini menyetujui permintaan tersebut.

8) Kode Fitur Mengirim Permintaan *Ta'aruf*

```

1090 $meetRequests = [];
1091
1092 foreach ($requests as $req) {
1093     // * Menentukan siapa lawan dari user login
1094     $partnerId = ($req['meet_id'] == $userId) ? $req['meet_user_id'] : $req['meet_id'];
1095     // * Kalau user login pengaju = ambil penerima
1096     // * Kalau user login penerima = ambil pengaju
1097
1098     $partner = $userModel->find($partnerId);
1099     $partnerProf = $profileModel->where('user_id', $partnerId)->first();
1100     $partnerPref = $prefModel->where('user_id', $partnerId)->first();
1101
1102     if (!$partner || !$partnerProf) continue;
1103
1104     // Optional filter lawan jenis
1105     if ($currentProf['jenis_kelamin'] === 'laki-laki' && $partnerProf['jenis_kelamin'] !== 'perempuan') continue;
1106     if ($currentProf['jenis_kelamin'] === 'perempuan' && $partnerProf['jenis_kelamin'] !== 'laki-laki') continue;
1107
1108     $score = $this->calculateMatchScore($currentProf, $currentPref, $partnerProf, $partnerPref);
1109
1110     $meetRequests[] = [
1111         'id_meet' => $req['id'],
1112         'tanggal_meet' => $req['tanggal_meet'],
1113         'moderator' => $req['moderator'],
1114         'reschedule' => $req['tanggal_hari'],
1115         'reschedule_by' => $req['reschedule_by'],
1116         'status' => $req['status'],
1117         'meet_id' => $req['meet_id'],
1118         'user' => (object)[
1119             'id' => $partner->id,
1120             'fullname' => $partner->fullname,
1121             'user_image' => $partner->user_image,
1122             'profile' => $partnerProf
1123         ],
1124         'score' => $score
1125     ];
1126 }

```

Gambar 5. 24 Kode fitur mengirim permintaan *ta'aruf*

Sistem terlebih dahulu menentukan identitas pengguna *sender* maupun *target*. Setelah itu, sistem mengambil data profil dan preferensi masing-masing pengguna untuk memastikan data yang diperlukan tersedia. Selanjutnya, fungsi *calculateMatchScore()* digunakan untuk menghitung nilai kecocokan antara profil pengguna dan calon pasangan. Hasil dari setiap iterasi kemudian disusun ke dalam array *\$meetRequests*, yang berisi informasi yang akan ditampilkan ke *user*.

9) Kode Fitur Mengelola Jadwal *Ta'aruf*

```

1013 public function schedule()
1014 {
1015     if ($this->request->getMethod() !== 'POST') {
1016         return $this->response->setStatusCode(405);
1017     }
1018
1019     $meetModel = new UserMeetModel();
1020     $meetUserId = $this->request->getPost('meet_user_id');
1021     $tanggalMeet = $this->request->getPost('tanggal_meet');
1022     $moderator = $this->request->getPost('moderator');
1023
1024     $existing = $meetModel
1025         ->where('meet_id', user_id())
1026         ->where('meet_user_id', $meetUserId)
1027         ->first();
1028
1029     if ($existing) {
1030         return $this->response->setJSON([
1031             'status' => 'error',
1032             'message' => 'Pertemuan sudah dijadwalkan sebelumnya.'
1033         ]);
1034     }
1035
1036     $meetModel->insert([
1037         'meet_id' => user_id(),
1038         'meet_user_id' => $meetUserId,
1039         'tanggal_meet' => $tanggalMeet,
1040         'moderator' => $moderator,
1041         'status' => 'pending',
1042         'created_at' => date('Y-m-d H:i:s'),
1043     ]);
1044
1045     return $this->response->setJSON([
1046         'status' => 'success',
1047         'message' => 'Jadwal pertemuan berhasil disimpan!'
1048     ]);
1049 }

```

Gambar 5. 25 Fitur mengelola jadwal *ta'aruf*

Proses dalam potongan kode ini diawali dengan validasi metode permintaan agar hanya menerima permintaan dengan metode *POST*. Setelah itu, sistem mengambil data dari *user* (pengirim), *target*, serta beberapa detail lain. Sebelum menyimpan data baru, sistem memeriksa apakah sudah terdapat jadwal pertemuan antara kedua pengguna untuk mencegah duplikasi. Jika ditemukan jadwal yang

sama, sistem mengembalikan respons kesalahan dalam format JSON. Jika tidak, data pertemuan baru akan disimpan ke dalam *database* dengan status baru.

6. KESIMPULAN

Berdasarkan uraian-uraian yang telah dibahas pada bab-bab sebelumnya. Penulis mencoba untuk menarik kesimpulan mengenai penulisan tugas akhir ini sebagai berikut.

- 1) *Web* layanan biro jodoh ini berperan dalam mempermudah para pengguna untuk menemukan calon pasangan hidup yang sesuai dengan kriteria yang diharapkan.
- 2) *Web* layanan biro jodoh menyediakan layanan pencarian pasangan berbasis algoritma algoritma *Jaccard Similarity* yang dikombinasikan dengan pembobotan atribut dan analisis kesesuaian kepribadian *MBTI (Myers-Briggs Type Indicator)* sehingga menghasilkan tingkat kecocokan (*match score*) antar dua pengguna secara objektif berdasarkan kesamaan karakteristik dan faktor kepribadian yang saling melengkapi.
- 3) *Web* layanan biro jodoh ini juga menyediakan fitur pelaporan yang menampilkan data pasangan yang telah melaksanakan proses ta'aruf.

DAFTAR PUSTAKA

- [1] Azzulfa, F. A. (2020). Biro Jodoh Online: Kebutuhan Atau Tuntutan. *Al Maqashidi: Jurnal Hukum Islam Nusantara*, 3(1), 35-49.
- [2] Setiawardhana, S. (2010). Aplikasi Biro Jodoh Berbasis Web dan J2ME. *eepis final project*.
- [3] Cahyani, R. A., & Ayu, R. F. (2020). Biro Jodoh Online: Kegunaan Dan Dampak. *JURIS (Jurnal Ilmiah Syariah)*, 19(2), 163-175.
- [4] Pramana, E. B., & Sunandar, A. (2022). Analisa Rancang Bangun Aplikasi Biro Jodoh Islami Menggunakan Algoritma Profile Matching Berdasarkan Platform Android. *Arcitech: Journal of Computer Science and Artificial Intelligence*, 2(2), 123-134.
- [5] Destianti, A. (2012). *Rancang Bangun Aplikasi Jejaring Sosial Berbasis Web Untuk Pendidikan Siswa Tingkat Sekolah Menengah Atas* (Bachelor's thesis, Fakultas Sains dan Teknologi UIN Syarif Hidayatullah Jakarta).
- [6] Rizki, M. M. (2021). RANCANG BANGUN JEJARING SOSIAL SAHABAT SOLID SEBAGAI WADAH KOMUNIKASI BERBASIS WEB.
- [7] Nento, A. R. (2019). Sistem Informasi Administrasi Ta'aruf.
- [8] Setiawan, R. (2021, July 27). *Metode SDLC Dalam Pengembangan Software*. Dicoding Blog. <https://www.dicoding.com/blog/metode-sdlc/>
- [9] Salmaa. (2023, March 17). *Studi Literatur: Pengertian, Ciri, Teknik Pengumpulan Datanya*. Penerbit Deepublish. https://penerbitdeepublish.com/studi-literatur/#Pengertian_Studi_Literatur_Menurut_Para_Ahli
- [10] *Aplikasi*. (2020, September 28). Wikipedia. <https://id.wikipedia.org/wiki/Aplikasi>
- [11] Ivan. *Aplikasi Berbasis Web: Definisi, Cara Kerja & Jenisnya*. (2025, April 30). Dewaweb. <https://blog.dewaweb.com/aplikasi-berbasis-web/>
- [12] *CSS Adalah: Pengertian, Fungsi, Jenis Dan Cara Kerjanya*. (2022, September 30). Coding Studio. <https://codingstudio.id/blog/css-pengertian-fungsi-jenis-dan-cara-kerjanya/>
- [13] Rendi, J. (2020, December 1). *Apa Itu JavaScript? Fungsi dan Contohnya*. Dicoding Blog. <https://www.dicoding.com/blog/apa-itu-javascript-fungsi-dan-contohnya/>
- [14] DomaiNesia. (2023, November 10). *PHP adalah: Pengertian, Fungsi dan Contohnya!* DomaiNesia. https://www.domainesia.com/berita/php-adalah/#Mengapa_Menggunakan_PHP
- [15] meilinaeka. (2022, December 27). *Salah Satu Jenis DBMS MySQL yang Wajib Anda Pahami*. Direktorat Pusat Teknologi Informasi. <https://it.telkomuniversity.ac.id/jenis-dbms-mysql-yang-wajib-anda-pahami/>
- [16] *Pengertian Metode Observasi dan Contohnya*. (2020). Gramedia.com. https://www.gramedia.com/literasi/pengertian-metode-observasi-dan-contohnya/#google_vignette.
- [17] Azwinda, D. (2022). Analisis terhadap biro jodoh online: Kebutuhan atau tuntutan. *Humanika*, 22(2), 107-16.