PENERAPAN *CONTENT-BASED FILTERING* PADA SISTEM REKOMENDASI *ANIME* BERBASIS SEMANTIK

Ikhsan Rifky Baihaqi
Pembimbing Utama: Wahyuni, S.Kom., M.Kom. Pembimbing Pendamping: Siti Lailiyah, S.Kom., M.Kom.

Program Studi Teknik Informatika, STMIK Widya Cipta Dharma
Jl. M. Yamin No.25, Samarinda, 75123
E-mail : ikhsanrifkyb@gmail.com

ABSTRAK

Ikhsan Rifky Baihaqi, 2021, Penerapan *Content-Based Filtering* Pada Sistem Rekomendasi *Anime* Berbasis Semantik. Skripsi Program Studi Teknik Informatika, Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK) Widya Cipta Dharma Samarinda, Ketua Pembimbing : Wahyuni, S.Kom., M.Kom, Anggota Pembimbing : Siti Lailiyah, S.Kom., M.Kom.

Anime begitu banyak digemari oleh masyarakat Indonesia, khususnya di kalangan remaja dan ini menjadi bukti bahwa pengaruh *anime* tidak bisa dipandang sebelah mata. Banyak berbagai macam situs *streaming anime* ber*subtitle* kan Bahasa Indonesia, namun belum ada sistem yang memberikan rekomendasi *anime* yang mendekati karakteristik pengguna atau penikmat *anime*.

Tujuan dari penelitian ini adalah untuk membantu pengguna atau penikmat *anime* dalam merekomendasikan *anime* kepada pengguna dengan menggunakan bantuan metode *Content-Based Filtering*. *Content-Based Filtering* adalah metode yang merekomendasikan suatu dokumen berdasarkan jumlah kemunculan term atau kata yang ada pada sisi pengguna kemudian dilakukan pembobotan akhir dari rumus TF-IDF hingga menghasilkan sebuah rekomendasi.

Kriteria yang digunakan untuk menetukan sebuah rekomendasi adalah Judul, Tipe, Studio, Genre, Tahun Rilis dan Seiyuu (Pengisi Suara Karakter). Sistem Rekomendasi *Anime* Berbasis Semantik yang dapat memperkaya informasi yang diberikan sehingga menjadi lebih baik dalam pendefinisiannya, agar memungkinkan komputer dapat memahami informasi yang telah diberikan sehingga komputer dan manusia dapat bekerja sama.

Hasil dari penelitian ini adalah sistem rekomendasi a*nime* berhasil menampilkan hasil rekomendasi dengan melewati beberapa pengujian seperti *beta testing* dan *white box*, namun keputusan akhir pemilihan *anime* tetap berada di pihak pengguna.

Kata Kunci : Penerapan, *Content-Based Filtering*, Sistem Rekomendasi, *Anime*, Semantik

---

1. PENDAHULUAN

Menikmati hiburan saat ini sudah menjadi kebutuhan dasar dari setiap orang, cara mendapatkan hiburan setiap orang pun beragam, namun ditengah kondisi terjadinya wabah virus yang diberi nama *Severe Acute Respiratory Syndrome Coronavirus* 2 (SARS-CoV-2), dan menyebabkan penyakit *Coronavirus Disease*-2019 (COVID-19), (INFEKSIEMERGING Kemenkes, 2020). Tidak butuh waktu lama hingga virus ini masuk di Negara Indonesia pada awal tahun 2020, Pemerintah Indonesia menanggapi masalah ini dengan memberlakukan kebijakan Pembatasan Sosial Berskala Besar (PSBB), (KEMENKO PMK, 2020). Secara tidak langsung kebijakan ini memberikan pengaruh diseluruh bidang pergerakan masyarakat dari pembatasan aktifitas sosial, hingga pembatasan aktifitas pribadi. Pembatasan aktifitas ini membuat sebagian masyarakat menjadi jenuh dan mencari kegiatan yang dapat dilakukan untuk mengurangi rasa jenuh dengan menghibur diri, cara menghibur diri dan menikmati hiburan setiap orang pun berbeda-beda misalnya saja dengan menonton *anime*, (LiniMuda, 2020).

Anime menurut Gilles Poltras, ada dua pengertian, yang pertama adalah kata yang digunakan oleh orang Jepang untuk menyebut film animasi apa pun tanpa memperhatikan dari mana asal *anime* tersebut. Kedua, penggunaan kata *anime* di luar Jepang adalah film animasi yang berasal dari Jepang, jadi pengertian *anime* terdapat dua pandangan, yaitu pandangan dari orang Jepang dan pandangan dari luar orang Jepang, orang

Jepang mengatakan segala jenis film animasi dari seluruh dunia dengan sebutan *anime*, kebalikannya dengan orang luar Jepang mengatakan bahwa *anime* merupakan film animasi yang hanya dibuat oleh Jepang saja.

Saat ini *anime* begitu banyak digemari oleh masyarakat Indonesia, khususnya di kalangan remaja dan ini menjadi bukti bahwa pengaruh *anime* tidak bisa dipandang sebelah mata (Venny, 2018). Di Indonesia sendiri menonton *anime* secara online tidaklah sulit, sudah banyak berbagai macam situs *streaming anime* ber*subtitle* kan Bahasa Indonesia. Namun belum ada situs maupun sistem yang memberikan rekomendasi *anime* yang mendekati karakteristik pengguna atau pun penikmat dari *anime* itu sendiri, hal ini membuat suatu keadaan dimana masyarakat kesulitan dalam mencari *anime* yang mereka inginkan, maka dari itu dibutuhkan sebuah rekomendasi *anime*. Pada website tertentu terdapat banyak judul anime yang dapat dimanfaatkan untuk merekomendaasikan *anime* kepada pengguna. Oleh karena itu, diperlukan suatu sistem yang dapat merekomendasikan judul *anime* kepada pengguna.

Dari permasalahan diatas diharapkan ada suatu sistem yang dapat menganalisis kesamaan profil pengguna dan menghasilkan rekomendasi *anime* berdasarkan kesamaan profil seorang pengguna. Oleh karena itu dimaksudkan untuk dibuat suatu sistem "Penerapan *Content-Based Filtering* Pada Sistem Rekomendasi *Anime* Berbasis Semantik" berbasis *website* yang dapat membantu dan mempermudah pengguna dalam menentukan pilihan mereka untuk menonton *anime* berdasarkan profile pengguna, serta diharapkan mampu membantu pengguna mengatasi dalam permasalahan menentukan pemilihan *anime* yang akan mereka tonton.

1.2 BATASAN MASALAH

Berdasarkan perumusan masalah diatas, ditetapkan suatu batasan masalah pada Penerapan *Content-Based Filtering* Pada Sistem Rekomendasi *Anime* Berbasis Semantik. Untuk menghindari meluasnya pokok pembahasan, maka dibuatlah suatu batasan masalah sebagai berikut :

1. Metode yang digunakan pada penelitian ini adalah *Content-Based Filtering*.
2. Semantik yang dimaksud merupakan Semantik Impresi.
3. Sistem rekomendasi yang dibuat merupakan alat bantu untuk menentukan pemilihan *anime*, sedangkan keputusan akhir tetap berada pada pihak pengguna.
4. Sistem rekomendasi yang dibuat hanya berfokus untuk memberikan rekomendasi berdasarkan konten atau *anime* yang sebelumnya pernah ditonton oleh pengguna.
5. Data yang akan digunakan dalam penelitian ini didapatkan dari *website myanimelist.net*.
6. Penentuan kriteria atau attribut berdasarkan hasil wawancara pada mahasiswa yang memiliki ketertarikan pada *anime* di STMIK Widya Cipta Dharma Samarinda.
7. Penentuan frekuensi kata / *term* berdasarkan data *anime* yang disukai pengguna.
8. Data yang akan digunakan dalam penelitian ini adalah judul *anime* yang rilis dalam kurun waktu 2014 sampai dengan tahun penelitian ini berlangsung.
9. Sistem ini hanya untuk merekomendasikan pemilihan *Anime.*
10. Sistem ini ditujukan oleh kalangan pengguna yang telah lama berkecimpung maupun yang masih awam terhadap *anime.*

2. BAHAN DAN METODE

Adapun bahan dan metode yang gunakan dalam membangun sitem pendukung keputusan ini yaitu:

2.1 Metode *Content-Based Filtering*

Menurut T Badriyah, dkk (2017), *Content-Based Filtering* merupakan sebuah metode yang akan memilih dan melakukan peringkat suatu item berdasarkan hasil analisa kesamaan profil pengguna dengan profil item pengguna, sehingga sistem dapat mengetahui ketertarikan dan minat pengguna berdasarkan profil pengguna.

Metode *Content-Based Filtering* menggunakan algoritma TF-IDF. TF (*Term Frequency*) adalah jumlah dari kemunculan sebuah term / kata dalam suatu dokumen terkait yang dikemukakan oleh Hans Peter Luhn (1957). IDF (*Invers Document Frequency*) merupakan perhitungan dari bagaimana *term* pada data koleksi dokumen yang terkait didistribusikan secara luas pada koleksi dokumen yang bersangkutan yang dikemukakan oleh Karen Spärck Jones (1972).

Metode ini pada awalnya diterapkan pada sistem rekomendasi dokumen. Variasi dari metode ditawarkan dengan dasar suatu analisis isi dokumen dan pencarian keteraturan dalam berbagai dokumen.

Maka dapat disimpulkan metode *Content-Based Filtering* adalah metode yang merekomendasikan suatu dokumen berdasarkan jumlah kemcunculan term/atau kata yang ada pada sisi pengguna kemudian dilakukan pembobotan akhir dari rumus TF-IDF hingga menghasilkan sebuah rekomendasi.

Algoritma metode Content-Based Filtering dijelaskan dalam tahap-tahap berikut ini :

1. Suatu item dokumen dipisahkan berdasarkan vector komponen pembentuknya. Misalnya pada penelitian kali ini dibagi atas komponen genre, studio produksi, produser, *seiyuu* (pengisi suara karakter).
2. Pengguna memberi penilaian suka atau tidak suka pada item.
3. Sistem akan membentuk profil pengguna berdasarkan bobot vektor komponen pembentuk suatu item. Pembuatan profil pengguna dapat menggunakan algoritma TF-IDF (term frequency - invers document frequency). TF (Term Frequency) adalah frekuensi dari kemunculan sebuah term / kata dalam suatu dokumen yang bersangkutan. Semakin besar jumlah kemunculan suatu term (TF tinggi) dalam dokumen, semakin besar pula bobotnya memberikan kemungkinan rekomendasi yang semakin besar. IDF (Invers Document

Frequency) merupakan perhitungan dari bagaimana term didistribusikan secara luas pada koleksi dokumen yang bersangkutan. Nilai IDF dapat dihitung menggunakan rumus:

$$idf_i = \log\left(\frac{n}{df_i}\right)$$

Dimana :
n = jumlah semua
df = jumlah dokumen yang memiliki *term* / kata *i*.

Sedangkan bobot term / kata pada suatu dokumen dihitung dengan rumus :

$$w_{i,d} = tf_{i,d} \times log\left(\frac{n}{df_i}\right)$$

Dimana :
tfi,d = frekuensi term / kata i pada dokumen d
n = jumlah semua
dfi = jumlah dokumen yang memiliki *term* / kata *i*

Dengan menggunakan bobot TF-IDF, sebuah dokumen dapat dimodelkan sebagai sebuah vektor seperti Gambar 2.1. Dokumen *Di* dapat dimodelkan atas komponen *Ti* sehingga jika seluruh dokumen dikumpulkan maka akan membentuk matriks term-dokumen dengan bobot term/TF-IDF sebagai nilainya.

	T_1	T_2	....	T_t
D_1	w_{11}	w_{21}	...	w_{t1}
D_2	w_{12}	w_{22}	...	w_{t2}
:	:	:		:
:	:	:		:
D_n	w_{1n}	w_{2n}	...	w_{tn}

Gambar 2.1 Model Vektor Dokumen

Sistem akan melakukan penilaian berdasarkan analisis kemiripan profil pengguna dengan vektor komponen pembentuk item. Jika item tersebut akan disukai pengguna, maka item tersebut akan direkomendasikan ke pengguna.

Content-based filtering memberikan suatu rekomendasi berdasarkan hasil analisa kemiripan item pada penggunanya. Metode ini pada awalnya diterapkan dalam sistem rekomendasi dokumen. Variasi metode ditawarkan berdasarkan suatu analisis isi dokumen dan pencarian keteraturan dalam berbagai dokumen. Pendekatan metode lain dilakukan menggunakan metode klasifikasi dengan tujuan memperkirakan golongan suatu item atau teks. Seperti pada gambar 2.2.

Gambar 2.2 *Content-Based Filtering*

Pada penelitian ini, penulis menggunakan sistem rekomendasi *Content-Based Filtering* untuk rekomendasi *Anime* dengan menggunakan atribut seperti genre, studio produksi, produser, *seiyuu* (pengisi suara karakter). Dengan ilustrasi seperti pada gambar 2.2.

2.2 Sistem Rekomendasi

Menurut Ricci, dkk (2011), Sistem rekomendasi merupakan sebuah perangkat lunak yang bertujuan untuk membantu pengguna dengan cara memberikan rekomendasi kepada pengguna ketika pengguna dihadapkan dengan jumlah informasi yang besar. Rekomendasi yang diberikan diharapkan dapat membantu pengguna dalam proses pengambilan keputusan, seperti barang apa yang akan dibeli, buku apa yang akan dibaca, atau musik apa yang akan didengar, dan lainnya.

Sistem rekomendasi dapat didefinisikan sebagai program yang mencoba untuk merekomendasikan item yang paling cocok (produk atau jasa) untuk pengguna tertentu (individu atau bisnis) dengan memprediksi minat pengguna di item berdasarkan informasi terkait tentang item, pengguna dan interaksi antara item dan pengguna (Fathurrahman, M., Nurjanah, D. and Rismala, R., 2017).

Maka dapat disimpulkan sistem rekomendasi adalah sebuah sistem yang dapat memprediksi ketertarikan pengguna berdasarkan data yang telah didapat dari sisi pengguna dan sistem bertujuan untuk mempermudah pengguna dalam penentuan atau pemilihan akhir.

2.3 *Anime*

Menurut Tim *Japanese Station* dalam bukunya *Japanese Station* (2015), *Anime* merupakan kependekan dari pengucapan *animation* dalam Bahasa Inggris, yakni *animeshon*. Di Jepang sendiri, istilah *anime* digunakan untuk mendefinisikan semua jenis karya animasi dari negara manapun. Sementara, diluar Jepang, istilah *anime* biasanya digunakan untuk mendefinisikan karya animasi yang berasal dari Jepang saja.

Anime adalah jenis film animasi dari Jepang yang digambar dengan tangan maupun menggunakan teknologi komputer. *Anime* merupakan singkatan dari kata “animation” dalam bahasa Inggris yang merujuk pada semua jenis animasi. Di luar Jepang, istilah ini digunakan secara spesifik untuk menyebutkan segala animasi yang diproduksi di Jepang (Hidayat, S.N. & Muhibbin, A., 2019).

Awal mula perkembangan *anime* di Jepang, dimulai sekitar tahun 1907-1911. Diawali dengan ditemukannya

sebuah gambar bergerak berdurasi tiga detik berjudul *Katsudao Shasin* yang digunakan untuk keperluan pribadi atau tidak ditayangkan secara komersil. Barulah ditahun 1917, *anime* berjudul *Imokawa Mukuzo Genkaban no Maki* karya Oten Shimokawa ditayangkan secara komersil untuk pertama kali di Jepang. Selain itu, masih ada dua karya yang dianggap sebagai pionir dalam perkembangan industri *anime* di Jepang, yaitu *Nakamura Gatana* (1917) karya Jun'ichi Kouchi dan *Urashima Tarou* karya Seitaro Kitayoma.

2.4 Semantik

Makna semantik merupakan salah satu cabang linguistik yang mempelajari makna yang terkandung dalam bahasa, kode, atau jenis representasi lainnya. Singkatnya, semantik adalah studi tentang makna. Istilah semantik mengacu pada berbagai ide yang sangat teknis dan populer. Biasanya sering digunakan dalam bahasa sehari-hari untuk mengungkapkan masalah pemahaman kata atau pilihan arti (Asmara, R., Mubtada'i, N.R. & Bimantara, V., 2021).

Pengertian *web* semantik yang sering dirujuk berasal dari T. Berners-Lee, J. Hendler, O. Lassila. (2001), dalam jurnalnya yaitu *The Semantic (*2001*)* "*The Semantic Web is not a separate Web but an extension of the current one, in which information is given well-defined meaning, better enabling computers and people to work in cooperation*." Semantik merupakan suatu pendekatan yang dikembangkan khusus pada teknologi *World Wide Web* (WWW) atau yang biasa disebut dengan istilah *web.*

2.5 *Website*

Menurut Yuhefizar (2013), *Website* adalah suatu metode untuk menampilkan informasi di *internet*, baik berupa teks, gambar, suara, maupun video yang interaktif dan mempunyai kelebihan untuk menghubungkan (*link*) satu dokumen dengan dokumen lainnya (*hypertext*) yang dapat diakses melalui sebuah *browser*. *Website* juga dapat diartikan sebagai kumpulan dari berbagai macam halaman situs, yang terangkum didalam sebuah domain atau juga subdomain, yang lebih tempatnya berada di dalam WWW (*World Wide Web*) yang tentunya terdapat di dalam *Internet*.

Halaman *website* biasanya berupa dokumen yang ditulis dalam format *Hyper Text Markup Language* (HTML), yang bisa diakses melalui HTTP, HTTP adalah suatu protokol yang menyampaikan berbagai informasi dari *server website* untuk ditampilkan kepada para *user* atau pemakai melalui *web browser*. Jenis-jenis *website* ada 3 (tiga) macam, yaitu:

1. *Website* Statis, adalah suatu *website* yang halamannya tidak berubah.
2. *Website* Dinamis, merupakan suatu *website* yang secara strukturnya diperuntukan untuk *update* sesering mungkin.
3. *Website* Interaktif, adalah *website* yang para penggunanya bisa berinteraksi dan juga beradu *argument* mengenai apa yang menjadi pemikiran mereka.

2.6 Kriteria website yang baik

Menurut Sa'ad (2020), Kriteria *webiste* yang baik adalah :

1. *Usability*, adalah sebagai suatu pengalaman penggunaan dalam berinteraksi dengan aplikasi atau *Website* sampai pengguna dapat mengoprasikannya dengan mudah dan cepat.
2. Navigasi, membantu pengunjung untuk menemukan jalan yang mudah ketika menjelajahi *Website*, memberitahukan dimana mereka berada, kemana mereka bisa pergi.
3. *Grafic* desain atau desain visual yang baik setidaknya memiliki komposisi warna yang baik dan konsisten, tata ruang grafik yang konsisten, teks yang mudah dibaca, pengguna grafik yang memperkuat isi teks, penggunaan animasi pada tempat yang tepat, isi animasi yang memperkuat isi teks dan secara keseluruhan membentuk suatu pola yang harmonis.
4. *Contents* yang baik akan menarik, relevan dan pantas untuk target audien *Website* tersebut. Gaya penulisan dan bahasa yang dipergunakan harus sesuai dengan *Website* dan target audien.
5. *Website* harus *compatibel* dengan berbagai perangkat tampilannya (*Browser*), harus memberikan alternatif bagi *browser* yang tidak dapat melihat situsnya.
6. *Loading Time*, sebuah *Website* yang tampil lebih cepat kemungkinan besar akan kembali dikunjungi, apalagi dengan konten dan tampilan yang menarik. Waktu mengunduh memang tidak hanya dipengaruhi desain tetapi juga dipengaruhi koneksi, *server* dan lain-lain.
7. *Funtionality*, seberapa baik sebuah *Website* bekerja dari aspek teknologi, ini bisa melibatkan *programmer* dengan *scriptnya*. Misalkan HTML (DHTML), PHP, dan lain-lain.
8. *Accesibility*, halaman *Website* harus bisa dipakai oleh setiap orang baik anak-anak, orang tua dan orang muda termasuk orang-orang cacat.
9. Interaktivitas adalah apa yang melibatkan pengguna *Website* sebagai *user experience* dengan *Website* itu sendiri. Gunakan *hyperlink* untuk membawa pengunjung ke sumber berita, topik lebih lanjut, topik terkait atau lainnya seperti link yang berbunyi *more info about this*, *glossary*, *related link* dan lain-lain.

2.7 *HTTP (Hyper Text Markup Language)*

HTML atau *HyperText Makup Language* merupakan salah satu format yang digunakan dalam pembuatan dokumen dan aplikasi yang berjalan dihalaman web . Halaman ini dikenal sebagai *web page*. Dokumen HTML merupakan dokumen yang disajikan pada *web browser* (Suhartanto, M., 2017).

Menurut Yuhefizar (2013), HTML (*Hyper Text Markup Language*) adalah sebuah *protocol* yang digunakan untuk membuat halaman dokumen *Web*. Sifat bahasa HTML adalah *Client scripting*, dimana dokumen yang ditulis dengan HTML dapat dibuka pada komputer *stand alone* yang tidak membutuhkan *server*. Dokumen HTML merupakan *file* yang pada umumnya berekstensi

.htm atau .HTML dan penulisan *script* HTML selalu diawali dengan <HTML> dan diakhiri </HTML>.

2.8 PHP (*Hypertext Preprocessor*)

Menurut Kustiyahningsih (2011) dalam Supono, dkk (2018), PHP (atau resminya PHP: *Hypertext Preprocessor*) adalah skrip bersifat *server-side* yang ditambahkan kedalam HTML. PHP sendiri merupakan singkatan dari *Personal Home Page Tools*. Skrip ini akan membuat suatu aplikasi dapat diintegrasikan ke dalam HTML sehingga suatu halaman *web* tidak lagi bersifat statis, namun menjadi bersifat dinamis. Sifat *server-side* berarti pengerjaan kode program dilakukan di *server*, baru kemudian hasilnya dikirimkan ke *browser*.

2.9 CSS (*Cascading Style Sheet*)

Menurut Rohi Abdulloh (2015), CSS merupakan singkatan dari *cascading style sheets*, yaitu skrip yang digunakan untuk mengatur desain website. Walaupun HTML mempunyai kemampuan untuk mengatur tampilan website, namun kemampuannya sangat terbatas. Fungsi CSS adalah memberikan pengaturan yang lebih lengkap agar struktur website yang dibuat dengan HTML terlihat lebih rapi dan indah.

CSS dapat mengendalikan ukuran gambar, warna bagian tubuh pada teks, warna *table*, ukuran *border*, warna *hyperlink*, warna *mouse over*, spasi antar *paragraph*, spasi antar teks, *margin* kiri, kanan, atas, bawah, dan *parameter* lainnya. CSS adalah bahasa yang digunakan untuk mengatur tampilan dokumen.

2.10 *JavaScript*

Menurut Sianipar (2015), *JavaScript* merupakan bahasa skrip populer yang dipakai untuk menciptakan halaman *web* yang dapat berinteraksi dengan pengguna dan dapat merespon *event* yang terjadi pada halaman. *JavaScript* merupakan perekat yang menyatukan halaman-halaman *web*. Akan sangat susah menjumpai halaman web komersial yang tidak memuat kode *JavaScript*.

JavaScript, awalnya dikenal sebagai *LiveScript*, dikembangkan oleh Brendan Eich di *Netscape* pada tahun 1995 yang menjadi bagian terintegrasi di dalam *Netscape Navigator 2.0*. *JavaScript* merupakan bahasa skrip yang menghidupkan halaman-halaman HTML. *JavaScript* dapat dijalankan pada hampir semua platform. *JavaScript* merupakan bahasa sisi-klien yang didesain pada browser komputer anda, bukan pada server. Ia dibangun secara langsung ke dalam browser. *Microsoft Internet Explorer*, *Mozilla Firefox*, *Google Chrome*, *Opera*, hampir semua *browser*. Dalam sintaksis, *JavaScript* mirip dengan *C*, *Perl*, dan *Java*.

2.11 *jQuery*

Menurut Wardana (2016), *jQuery* adalah *JavaScript library*. *jQuery* mempunyai semboyan *"write less, do more"*. *jQuery* dirancang untuk memperingkas kode-kode *JavaScript*. *jQuery* adalah JavaScript library yang cepat dan ringan untuk menangani dokumen HTML, *event*, membuat animasi, dan interaksi *ajax*. *jQuery* dirancang untuk mengubah cara kita menulis kode JavaScript. Dengan kata lain, *jQuery* ini meringkas koding-koding kita untuk animasi. Dengan *ajax* dari *jQuery*, website dapat langsung berkomunikasi dengan (melakukan transaksi data) dengan *server*, tanpa harus me-*reloading* halaman *web* tersebut secara keseluruhan.

2.12 PHPMyAdmin

Menurut Sidik (2012), PHPMyAdmin adalah suatu program *open source* yang berbasis *web* yang dibuat menggunakan aplikasi PHP. Program ini digunakan untuk mengakses *database* MySQL. Program ini mempermudah dan mempersingkat kerja penggunanya. Dengan kelebihannya, para pernggguna awam tidak harus paham sintak-sintak SQL dalam pembuatan *database* dan tabel.

2.13 MySQL

Menurut Fitri (2020), MySQL adalah pengembangan lanjutan dari proyek UNIREG yang dikerjakan oleh Michael Monty Widenius dan TcX (Perusahaan perangkat lunak asal Swedia).

MySQL adalah DBMS yang *open source* dengan dua bentuk lisensi, yaitu *Free Software* (perangkat lunak bebas) dan *Shareware* (perangkat lunak berpemilik yang penggunaannya terbatas). Jadi MySQL adalah *database server* yang gratis dengan lisensi GNU *General Public License* (GPL) sehingga dapat dipakai untuk keperluan pribadi atau komersial tanpa harus membayar lisensi yang ada.

MySQL masuk kedalam jenis RDBMS (*Relational Database Management System*). Maka dari itu, istilah semacam baris, kolom, tabel, dipakai pada MySQL. Contohnya di dalam MySQL sebuah *database* terdapat satu atau beberapa tabel.

2.14 Internet

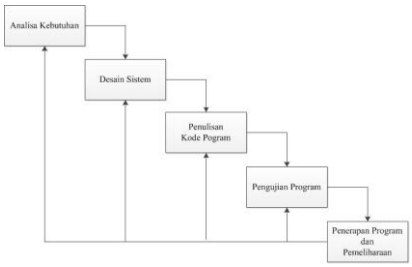
Menurut Ardhana (2012), *Internet* adalah kelompok atau kumpulan dari jutaan komputer yang terkait pada satu jaringan global. Penggunaan *internet* memungkinkan kita untuk mendapatkan informasi dari komputer yang ada dalam suatu jaringan *internet* selama memberikan izin akses tertentu.

2.15 *Web Browser*

Menurut Ardhana (2012), *Web Browser* adalah suatu program yang digunakan untuk membaca kode HTML yang kemudian menerjemahkannya ke dalam bentuk visual. Beberapa contoh *web browser* adalah *Internet Explorer*, *Mozilla Firefox*, *Opera*, *Google Chrome*, *Safari* dan lain-lain.

2.16 Metode Pengembangan Sistem

Menurut Ian Sommerville (2011), metode waterfall memiliki tahapan utama mencerminkan aktifitas pengembangan dasar. Terdapat 5 (lima) tahapan pada metode w*aterfall*, yaitu analisa kebutuhan (*requirement analysis*), desain sistem (*system and software design*), penulisan kode program (*implementation and unit testing*), pngujian program (*integration and system testing*), serta penerapan program dan pemeliharaan (*operation and maintenance*).



Gambar 2.3 Tahapan Metode *Waterfall*
Sumber : Ian Sommerville (2011), Tahapan Metode *Waterfall*

Penulis menggunakan Metode *Waterfall* (Air Terjun) karena dianggap cocok untuk pengembangan sistem yang penulis buat.

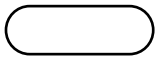
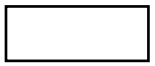
2.17 *Flowchart*

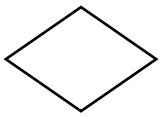
Menurut Rahayudi (2011), *Flowchart* dapat diartikan sebagai suatu simbol/bagan yang menggambarkan urutan-urutan penyelesaian permasalahan, dimana terjadi hubungan antara proses yang satu dengan yang lainnya. *Flowchart* dituangkan kedalam bentuk gambar-gambar atau simbol-simbol yang telah menjadi kesepakatan di dalam penyusunan program komputer.

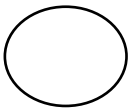
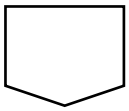
Ada 2 jenis *flowchart* yang dikenal secara umum oleh masyarakat, yaitu:

1. *System Flowchart*, adalah suatu *flowchart* yang digunakan untuk menyelesaikan permasalahan pada sistem pengolahan data, dimana melibatkan beberapa perangkat keras, seperti *magnetic disc*, *card* dll.
2. Progrem Flowchart, adalah suatu *flowchart* yang digunakan untuk menyelesaikan permasalahan pada pembatan program. Setiap penyelesaian permasalahan pemrograman dapat digambarkan dengan menggunakan simbol-simbol yang terdapat pada program *flowchart.*

Tabel 2.2 Daftar Simbol *Flowchart*

Simbol	Nama symbol	Keterangan
	*Terminal*	Menyatakan permulaan atau akhir suatu program.
	Input / Output	Menyatakan proses *input* atau *output* tanpa tergantung jenis peralatan.
	*Process*	Menyatakan suatu tindakan (proses) yang dilakukan oleh komputer.
	Decision	Menunjukkan

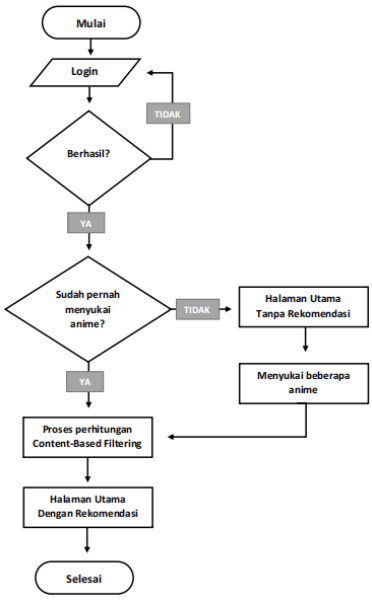


Simbol	Nama symbol	Keterangan
		suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban, ya atau tidak.
	*Connector*	Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama.
	*Offline Connector*	Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang berbeda
	Predefined Process	Menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal.
	Punched Card	Menyatakan *input* berasal dari kartu atau *output* ditulis ke kartu.
	Document	Mencetak keluaran dalam bentuk dokumen (melalui *printer*)
	Flow	Menyatakan jalannya arus suatu proses.

4. HASIL PENELITIAN DAN PEMBAHASAN

4.1 *Flowchart* Sistem

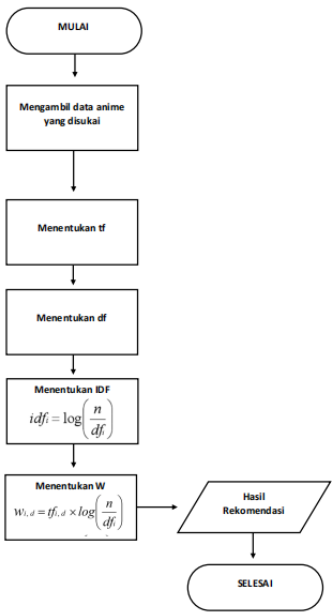
Pada Gambar 4.1 menggambarkan bagaimana sistem yang akan dibuat berjalan, sistem dimulai dengan login terlebih dahulu, pengguna harus memasukan data login yang benar agar dapat masuk ke dalam sistem. Setelah pengguna berhasil masuk sistem akan melihat apakah pengguna sebelumnya sudah memiliki data *anime* yang disukai, apabila belum ada data *anime* yang disukai sistem akan mengarahkan ke halaman utama tanpa hasil rekomendasi, jika pengguna sudah memilih *anime* yang disukai pada halaman utama barulah kemudian sistem dapat melakukan perhitungan metode content-based filtering dan sistem akan mengarahkan kembali ke halaman utama dengan hasil rekomendasi *anime*.



Gambar 4.1 *Flowchart* Sistem

4.2 *Flowchart* Algoritma

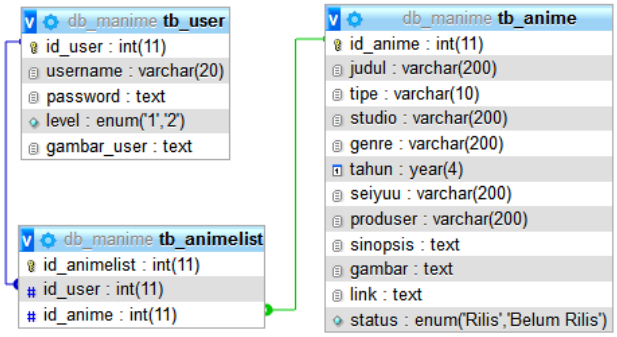
Pada Gambar 4.2 menggambarkan bagaimana sistem akan mendeklarasikan rumus metode *Content-Based Filtering*, sistem dimulai dengan membaca apakah *user* sudah memiliki data *anime* yang disukai, apabila tidak maka *user* diharuskan untuk memilih *anime* yang disukai terlebih dahulu. Berdasarkan data *anime* yang telah disukai user sistem akan menentukan tf, lalu nilai df, kemudian mencari nilai IDF, dan hasil IDF akan menentukan nilai W, yang digunakan untuk penentuan hasil akhir dari rekomendasi *anime* kepada *user*.



Gambar 4.2 *Flowchart* Algoritma

4.3 *Mapping Table System*

Adapun desain rancangan tabel dalam pembuatan sistem ini seperti pada yang ditunjukan pada Gambar 4.3.



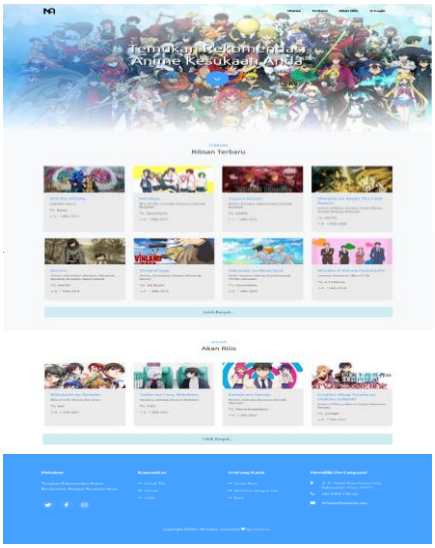
Gambar 4.3 *Mapping Table* Sistem

5. IMPLEMENTASI

Pada tahapan implementasi merupakan tahap penerapan dari bentuk sistem kedalam sebuah program.

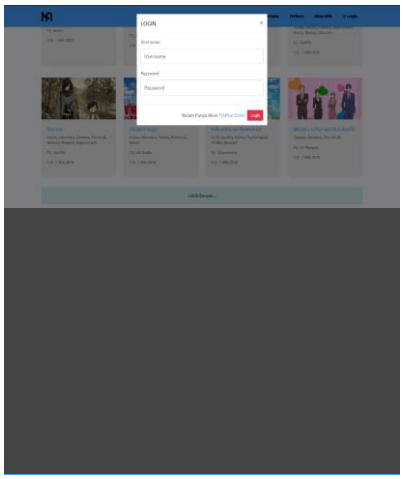
1) Halaman Utama

Pada Halaman Utama ini akan ditampilkan daftar *anime* rilisan terbaru dan *anime* yang akan rilis, untuk lebih jelas dapat dilihat pada Gambar 4.4.



Gambar 4.4 Halaman Utama

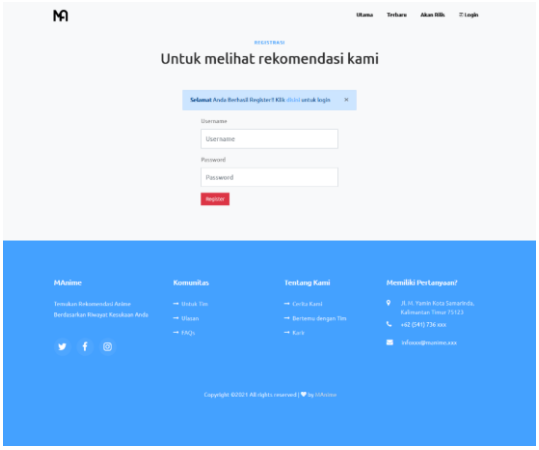
namun daftar *anime* tidak dapat diklik dan akan langsung dimunculkan jendela kecil untuk pengguna login terlebih dahulu sebelum melihat detail *anime*. Untuk lebih jelas dapat dilihat pada Gambar 4.5.



Gambar 4.5 Jendela kecil untuk *Login*

2) Halaman Registrasi / *Register*

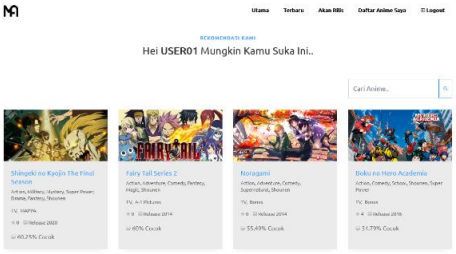
Pada halaman ini pengguna dapat melakukan registrasi untuk dapat mendaftar untuk dapat mengakses dan mendapatkan rekomendasi *anime*. Seperti pada gambar 4.6.



Gambar 4.6 Halaman Registrasi

3) Halaman Utama Pengguna

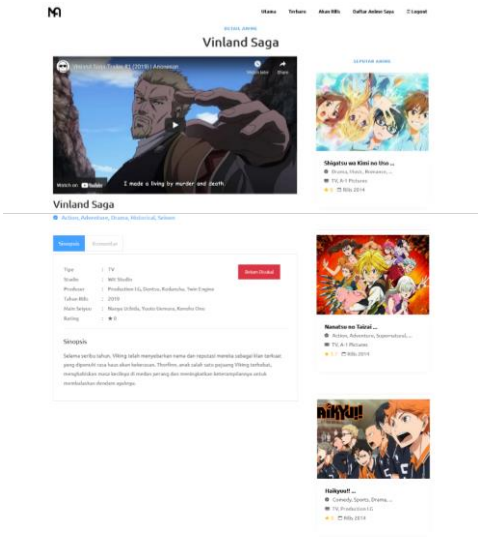
Pada Halaman Utama Pengguna ini akan ditampilkan daftar *anime* rekomendasi, rilisan terbaru, *anime* yang akan rilis, dan daftar *anime* yang disukai pengguna. Untuk lebih jelas dapat dilihat pada Gambar 4.7.



Gambar 4.7 Halaman Pengguna

4) Halaman Detail *Anime*

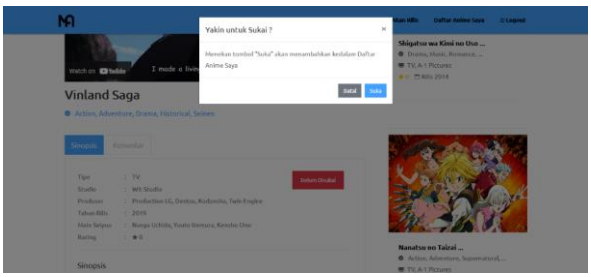
Pada Halaman Detail *Anime* pengguna dapat melihat cuplikan dari *anime*, beserta beberapa data seperti genre, tipe, studio, produser, tahun rilis, main seiyuu atau pengisi suara karakter, rating serta sinpsopsis. Dan ditampilkan beberapa seputar *anime* pada sisi kanan. Pengguna juga dapat meng-klik tombol berwarna merah "Belum Disukai" apabila pengguna ingin menambahkan kedalam daftar *anime* saya, seperti pada gambar 4.8.



Gambar 4.8 Halaman Detail *Anime*

5) Tampilan untuk menyukai *anime*

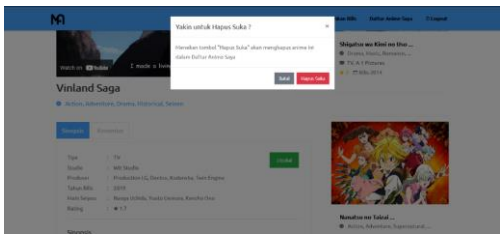
Pada Halaman Detail *Anime* apabila pengguna meng-klik tombol berwarna merah "Belum Disukai" maka akan muncul jendela kecil untuk mengkonfirmasi pengguna dalam menambahkan *anime* kedalam daftar *anime* saya seperti gambar 4.9.



Gambar 4.9 Jendela Menyukai *Anime*

6) Tampilan untuk hapus suka *anime*

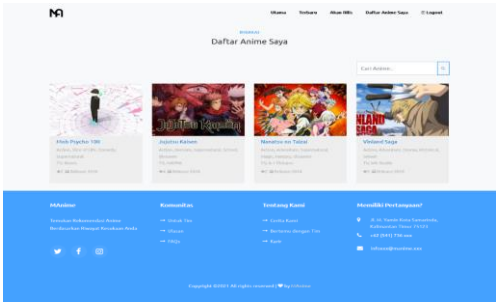
Pada Halaman Detail *Anime* apabila pengguna sudah menyukai *anime* dan meng-klik tombol berwarna hijau "Disukai" maka akan muncul jendela kecil untuk mengkonfirmasi pengguna dalam menghapus *anime* dari daftar *anime* saya seperti gambar 4.10.



Gambar 4.10 Jendela Menghapus Suka *Anime*

7) Halaman Daftar Anime Saya

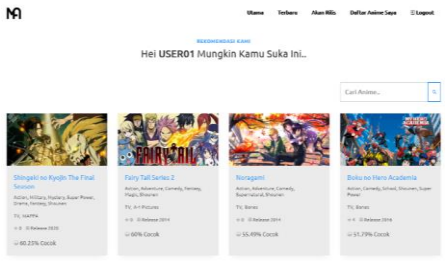
Pada Halaman Daftar *Anime* Saya adalah halaman yang menampilkan daftar *anime* yang sudah diklik suka oleh pengguna , untuk lebih jelas ditampilkan seperti gambar 4.11.



Gambar 4.11 Halaman Daftar *Anime* Saya

8) Halaman Rekomendasi

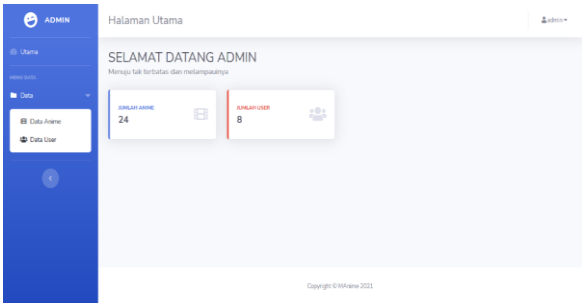
Pada Halaman Utama Pengguna terdapat tombol berwarna biru “Lebih Banyak” tombol ini akan mengarahkan pengguna kepada halaman rekomendasi seluruh *anime* yang akan direkomendasikan kepada pengguna, untuk lebih jelas ditampilkan seperti gambar 4.12.



Gambar 4.12 Halaman Rekomendasi

9) Halaman Utama Admin

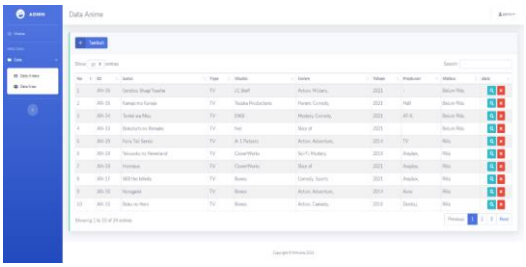
Pada Halaman Utama Admin merupakan halaman *dashboard*, dimana halaman *dashboard* ini merupakan halaman utama pada *backend*, pada halaman *dashboard* terdapat *banner* yang merupakan kata sambutan dan kalimat motivasi *website* MAnime, kemudian terdapat 2 *card* yang menampilkan jumlah *anime* dan jumlah *user* / pengguna. Pada halaman *dashboard* terdapat beberapa *menu* data yang terdiri dari *submenu* data *anime* dan data *user*, untuk lebih jelas ditampilkan seperti gambar 4.13.



Gambar 4.13 Halaman Utama Admin

10) Halaman Data *Anime*

Pada Halaman Data *Anime* dimana data *anime* akan dikelola, jika ingin menambahkan data baru admin dapat mengklik tombol tambah pada bagian sisi kiri, setelah tombol tambah diklik maka sistem akan menampilkan sebuah jendela kecil untuk menambahkan data *anime*, untuk lebih jelas ditampilkan seperti gambar 4.14



Gambar 4.14 Halaman Data *Anime*

11) Tambah Data *Anime*

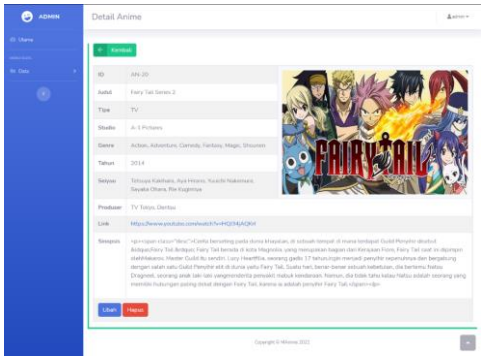
Pada Halaman Data *Anime* dimana data *anime* akan dikelola, jika ingin menambahkan data baru admin dapat mengklik tombol tambah pada bagian sisi kiri, setelah tombol tambah diklik maka sistem akan menampilkan sebuah jendela kecil untuk menambahkan data *anime*, untuk lebih jelas ditampilkan seperti gambar 4.15

Gambar 4.15 Tambah Data *Anime*

12) Halaman Detail *Anime*

Pada Halaman Detail *Anime* akan ditampilkan detail dari data *anime* yang diklik oleh admin sebelumnya, apabila admin ingin mengubah data dari *anime* tersebut maka admin harus terlebih dahulu mengklik tombol ubah pada

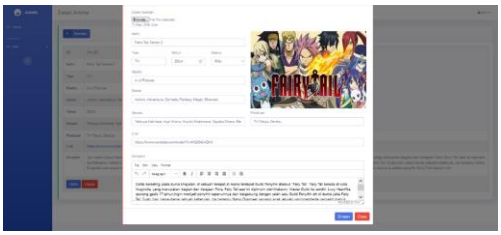
bagian bawah detail, untuk lebih jelas ditampilkan seperti gambar 4.16.



Gambar 4.16 Halaman Detail *Anime*

13) Ubah Data *Anime*

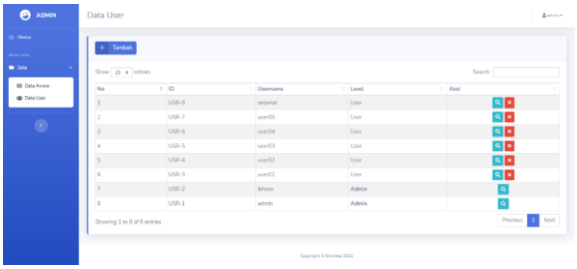
Pada Halaman Detail *Anime* akan ditampilkan detail dari data *anime* yang diklik oleh admin sebelumnya, apabila admin ingin mengubah data dari *anime* tersebut maka admin harus terlebih dahulu mengklik tombol ubah pada bagian bawah detail maka akan muncul jendela kecil untuk mengubah data *anime*, untuk lebih jelas ditampilkan seperti gambar 4.17.



Gambar 4.17 Ubah Data *Anime*

14) Halaman Data *User*

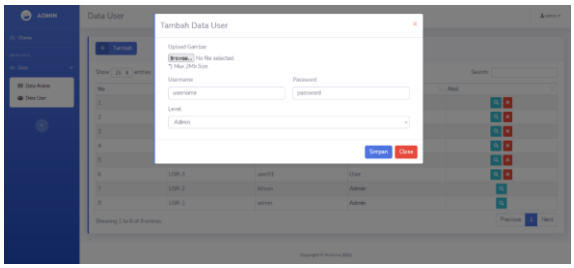
Pada Halaman Data *User* dimana data *user* akan dikelola, jika ingin menambahkan data baru admin dapat mengklik tombol tambah pada bagian sisi kiri, setelah tombol tambah diklik maka sistem akan menampilkan sebuah jendela kecil untuk menambahkan data *user*, kemudian terdapat *show entries* yang berfungsi untuk menampilkan berapa banyak data yang ingin ditampilkan, untuk lebih jelas ditampilkan seperti gambar 4.18



Gambar 4.18 Halaman Data *User*

15) Tambah Data *User*

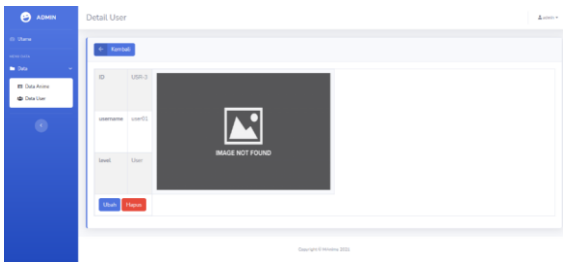
Pada Halaman Data *User* dimana data *user* akan dikelola, jika ingin menambahkan data baru admin dapat mengklik tombol tambah pada bagian sisi kiri, setelah tombol tambah diklik maka sistem akan menampilkan sebuah jendela kecil untuk menambahkan data *user*, untuk lebih jelas ditampilkan seperti gambar 4.19



Gambar 4.19 Tambah Data *User*

16) Ubah Data *User*

Pada Halaman Detail *User* akan ditampilkan detail dari data *user* yang diklik oleh admin sebelumnya, apabila admin ingin mengubah data dari *user* tersebut maka admin harus terlebih dahulu mengklik tombol ubah pada bagian bawah detail (hanya level *user* yang dapat dirubah), untuk lebih jelas ditampilkan seperti gambar 4.20



Gambar 4.20 Ubah Data *User*

6. KESIMPULAN

Dengan adanya hasil penelitian yang dilaksanakan dan berdasarkan uraian yang telah dibahas pada bab-bab sebelumnya, maka dapat ditarik kesimpulan bahwa:

1. Penerapan Metode *Content-Based Filtering* Pada Sistem Rekomendasi *Anime* Berbasis Semantik ini dibuat dengan menggunakan Bahasa pemrograman *PHP* dan *MySql* sebagai *database* nya dengan menggunakan alat bantu pengembangan sistem *Hierarchy Input Process Output System* (HIPO), *Flowchart* dan *Mapping Table*.
2. Telah dihasilkan perancangan dan pembuatan Penerapan Metode *Content-Based Filtering* Pada Sistem Rekomendasi *Anime* Berbasis Semantik.
3. Sistem Rekomendasi *Anime* ini dapat digunakan selalu oleh pengguna untuk mendapatkan info

ter*update* seputar *anime*, serta mendapatkan rekomendasi yang selalu *update*.

7. SARAN

Adapun saran saran yang dapat dikemukakan berdasarkan kesimpulan diatas yaitu sebagai berikut:

1. Diharapkan adanya pengembangan sistem menjadi bentuk aplikasi *mobile* sehingga lebih memudahkan pengguna dalam mengakses aplikasi.
2. Diharapkan pada aplikasi ini ada fitur pencarian *anime* agar mempermudah pengguna dalam mencari data *anime*.
3. Diharapkan aplikasi ini selalu memperluas *database anime* tidak hanya berdasarkan 1 sumber saja.
4. Diharapkan aplikasi ini juga dapat dikembangkan dengan menggabungkan metode yang lain agar dapat hasil yang mungkin lebih tepat dari sistem ini.
5. Diharapkan agar aplikasi ini dijadikan sebagai tambahan referensi bagi mahasiswa bidang Teknik Informatika.

8. DAFTAR PUSTAKA

Abdulloh, R. (2015). Web Programming is Easy. Jakarta: PT. Elex Media Komputindo, 7.

Ardhana, Y, M, Kusuma. (2012). Pemrograman PHP CodeIgniter Blackbox, Jakarta: Jasakom.

Asmara, R., Mubtada'i, N. R., & Bimantara, V. (2021). OPTIMASI MESIN PENCARI BUKU FIKSI BERDASARKAN PADA SEMANTIK IMPRESI. METHOMIKA: Jurnal Manajemen Informatika & Komputerisasi Akuntansi, 5(1), 1-8.

Badriyah, T., Fernando, R., & Syarif, I. (2018). Sistem Rekomendasi Content Based Filtering Menggunakan Algoritma Apriori. Konferensi Nasional Sistem Informasi (KNSI).

Cholifah, W. N., Yulianingsih, & Sagita, S. M. (2018). Pengujian Black Box Testing Pada Aplikasi Action & Strategy Berbasis Android Dengan Teknologi Phonegap. Jurnal String, 206-210.

Danor Pratama, Rifki Randa. (2021). Penerapan Metode Content-Based Filtering Untuk Rekomendasi Penawaran Paket Add On. Program Studi Teknik Informatika, STMIK Widya Cipta Dharma Samarinda.

Dini, H. (2017). Analisis dan Perancangan Sistem Informasi Pembahasan Secara Praktis dengan Contoh Kasus. Yogyakarta: Deepublish.

Fadilla, R., Andarsyah, R., Awangga, R. M., & Habibi, R. (2020). Data analytics: peningkatan performa algoritma rekomendasi collaborative filtering menggunakan K-means clustering. Bandung: Kreatif.

Fathurrahman, M., Nurjanah, D., & Rismala, R. (2017). Sistem Rekomendasi Pada Buku Dengan Menggunakan Metode Trust-aware Recommendation. eProceedings of Engineering, 4(3).

Fitri, R. (2020). Pemrograman Basis Data Menggunakan MySQL. Yogyakarta: Deepublish.

Hidayat, S. N., & Muhibbin, A. (2019). Representasi Persahabatan dalam Anime Movie "Koe No Katachi" (Doctoral dissertation, Universitas Muhammadiyah Surakarta).

INFEKSIEMERGING. (2020). QnA : Pertanyaan dan Jawaban Terkait COVID-19 (https://infeksiemerging.kemkes.go.id/uncategorized/qna-pertanyaan-dan-jawaban-terkait-covid-19, diakses 24 Februari 2021)

KEMENKOPMK. (2020). Pembatasan Sosial Berskala Besar (https://www.kemenkopmk.go.id/pembatasan-sosial-berskala-besar, diakses 30 Juni 2021)

Koesheryatin, T. S. (2014). Aplikasi Internet Menggunakan HTML, CSS, dan JavaScript. Elex Media Komputindo.

LiniMuda. (2020). 4 Cara Agar Nonton Anime Kamu Jadi Lebih Menyenangkan (https://linimuda.id/hobi/film/menonton-anime-lebih-menyenangkan-dengan-4-cara-ini/, diakses 30 Juni 2021)

LiniMuda. (2020). 4 Alasan Mengapa Anime Menjadi Hiburan yang Sangat Populer (https://linimuda.id/hobi/film/alasan-mengapa-anime-menjadi-hiburan-populer/, diakses 30 Juni 2021)

Luhn, Hans Peter (1957). "A Statistical Approach to Mechanized Encoding and Searching of Literary Information"

Poitras, Gilles. (2008). Contemporary Anime in Japanese Pop Culture. Japanese Visual Culture: Explorations in the World of Manga and Anime. New York: M.E. Sharpe.

Pratama, Gregorius Nofear Hadi. (2019). SISTEM REKOMENDASI FILM MENGGUNAKAN METODE CONTENT BASED FILTERING. Program Studi Teknik Informatika, Universitas Atma Jaya Yogyakarta.

Rahayudi, B. (2011). Dasar-Dasar Pemrograman: Disertai Berbagai Contoh Kasus dan Diimplementasikan dalam Bahasa Pascal. Malang: Universitas Brawijaya Press.

Ricci, F., Rokach, L., & Shapira, B. (2011). Introduction to recommender systems handbook. Boston, MA: Springer.

Sa'ad, Muhammad Ibu. (2020). Otodidak Web Programming Membuat Website Edutaiment. Jakarta: PT. Elex Media Komputindo.

Sianipar, R. H. (2015). Pemrograman Javascript: Teori Dan Implementasi (Vol. 1). Bandung: INFORMATIKA.

Sidik, Betha. (2012). Pemrograman web PHP, Edisi revisi. Bandung: Informatika.

Spärck Jones, K. (1972). A Statistical Interpretation of Term Specificity and Its Application in Retrieval. Journal of Documentation.

Station, T. J. (2015). Japanese Station Book. Jakarta: Bukune.

Suhartanto, M. (2017). pembuatan website sekolah menengah pertama negeri 3 delanggu dengan menggunakan php dan mysql. Speed-Sentra Penelitian Engineering dan Edukasi, 4(1).

Sukmana, H. T., Atinah, S., & Wardhani, L. K. (2019). IMPLEMENTASI CONTENT-BASED FILTERING PADA APLIKASI RADAR ZAKAT DALAM MEREKOMENDASIKAN PREFERENSI MUSTAHIK. Program Studi Teknik Informatika, Universitas Islam Negeri Syarif Hidayatullah Jakarta.

Supono, V. P. (2016). Pemrograman Web dengan menggunakan PHP dan FRAMEWORK CODEIGNITER. Yogyakarta: Deepublish.

T. Berners-Lee, J. Hendler, O. Lassila. (2001). The Semantic. Scientific American.

Tjandra, S., & Pickerling, C. (2015). Aplikasi Metode-Metode Software Testing Pada Configuration, Compatibility dan Usability Perangkat Lunak. In Seminar Nasional" Inovasi dalam Desain dan Teknologi (pp. 374-367).

Universitas Raharja, (2020). Metode Waterfall (https://raharja.ac.id/2020/04/04/metode-waterfall, diakses 3 Maret 2021)

Utami, F. H. (2015). Rekayasa Perangkat Lunak. Yogyakarta: Deepublish.

Wardana. (2016). Aplikasi Website Profesional dengan PHP dan jQuery. Jakarta: Elex Media Komputindo.

Winarno, E., Zaki, A., & Comunity, S. (2013). Buku Sakti Pemrograman PHP. Jakarta: PT Elex Media Komputindo.

Wikipedia contributors. "Tf–idf." (https://en.wikipedia.org/wiki/Tf–idf, diakses 10 Juli 2021).

Yuhefizar. (2013). Cara Mudah & Murah Membangun & Mengelola Website. Yogyakarta: Graha Ilmu.

Zanitri, V. (2018). Pengaruh Menonton Anime Jepang di Internet Terhadap Perilaku Imitasi di Kalangan Komunitas Japan Club East Borneo Kota Samarinda. Jurnal Ilmu Komunikasi, Fakultas Ilmu Sosial dan Ilmu Politik, Universitas Mulawarman.

Zebua, J., & Mustikasari, M. (2013). Aplikasi pencarian buku berbasis web semantik untuk perpustakaan Smk Yadika 7.