**SISTEM REKOMENDASI PEMILIHAN SEPEDA MOTOR DENGAN**

**PENDEKATAN ONTOLOGI**

**SKRIPSI**



I MADE CANTIAWAN GIRI KUSUMA

NIM. 1708561005

**INFORMATIKA**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

**UNIVERSITAS UDAYANA**

**JIMBARAN**

**2021**

LEMBAR PENGESAHAN TUGAS AKHIR

|  |  |  |
| --- | --- | --- |
| Judul | : | Sistem Rekomendasi Pemilihan Sepeda Motor dengan Pendekatan Ontologi |
| Kompetensi | : | Penemuan dan Manajemen Pengetahuan |
| Nama | : | I Made Cantiawan Giri Kusuma |
| NIM | : | 1708561005 |
| Tanggal Seminar | : | 18 Mei 2020 |

Disetujui oleh:

Pembimbing I Penguji I

Nama Pembimbing Nama Penguji

NIP. NIP.

Pembimbing II Penguji II

Nama Pembimbing Nama Penguji

NIP. NIP.

Penguji III

Nama Penguji

NIP.

Mengetahui,

Ketua Program Studi Informatika

Nama Ketua Prodi

NIP.

|  |  |  |
| --- | --- | --- |
| Judul | : | Sistem Rekomendasi Pemilihan Sepeda Motor dengan Pendekatan Ontologi |
| Nama | : | I Made Cantiawan Giri Kusuma (NIM. 1708561005) |
| Pembimbing | : | 1. Pak Cok  2. Pak Santiyasa |

ABSTRAK

Kata Kunci:

|  |  |  |
| --- | --- | --- |
| Title | : | Motorcycle Selection Recommendation System with Ontology Approach |
| Name | : | I Made Cantiawan Giri Kusuma (NIM. 1708561005) |
| Supervisor | : | 1. Pak Cok  2. Pak Santi |

ABSTRACT

Keywords:

**KATA PENGANTAR**

Proposal penelitian dengan judul “Sistem Rekomendasi Pemilihan Sepeda Motor dengan Pendekatan Ontologi” ini disusun dalam rangkaian pelaksanaan Tugas Akhir di Program Studi Informatika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Udayana. Proposal ini disusun dengan harapan dapat menjadi pedoman dan arahan dalam melaksanakan penelitian diatas.

Sehubung dengan telah diselesaikannya proposal ini, maka diucapkan terimakasih dan penghargaan kepada berbagai pihak yang telah membantu penulis dalam menyusun proposal ini, antara lain:

1. Bapak Dr. I Ketut Gede Suhartana, S.Kom., M.Kom. selaku koordinator Program Studi Informatika Fakultas MIPA Universitas Udayana;
2. Bapak Cokorda Rai Adi Pramartha, ST.MM.PhD sebagai calon Pembimbing I yang telah banyak mengkritisi, memeriksa, dan membantu menyempurnakan proposal ini;
3. Bapak-bapak dan ibu-ibu dosen di Program Studi Informatika Fakultas MIPA Universitas Udayana, yang telah meluangkan waktu dan memberikan saran dan masukan dalam penyempurnaan proposal ini;
4. Kawan-kawan di Program Studi Informatika yang telah memberikan dukungan moral dalam penyelesaian proposal ini.

Disadari pula bahwa proposal ini masih mengandung banyak kelemahan dan kekurangan. Memperhatikan hal ini, maka penulis sangat mengharapkan masukan dan saran-saran dalam penyempurnaan proposal ini.

Jimbaran, Maret 2021

Penulis

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

DAFTAR LAMPIRAN

**BAB I**

**PENDAHULUAN**

1. **Latar Belakang**

Ketersediaan transportasi saat ini menjadi kebutuhan yang penting untuk menunjang kehidupan sehari-hari dan sebagai penunjang kegiatan perekonomian seperti kelancaran usaha. Saat ini banyak alat transportasi yang disediakan, alat transportasi dapat dibedakan menjadi dua jenis yaitu alat transportasi umum dan alat transportasi pribadi. Alat transportasi umum merupakan layanan angkutan penumpang oleh sistem perjalanan kelompok yang tersedia untuk digunakan oleh masyarakat umum. Sedangkan alat transportasi pribadi adalah angkutan yang menggunakan kendaraan pribadi seperti mobil, sepeda motor, dan sebagainya.

Dari alat transportasi pribadi tersebut, sepeda motor merupakan alat transportasi yang banyak digunakan di Indonesia karena penggunaan transportasi sepeda motor sangat praktis terutama di daerah padat lalu lintas selain itu sepeda motor merupakan alat transportasi yang efisien dalam penggunaan bahan bakar dan biaya perawatan.

Melihat tingkat penggunaan sepeda motor di negara Indonesia saat ini industri sepeda motor berkembang dari tahun ke tahun. Dari data yang diambil dari website Asosiasi Industri Sepeda Motor Indonesia[[1]](#footnote-1) (AISI) bisa dilihat data *domestic distribution* dan *export* sepeda motor yang sering mengalami kenaikan dari tahun ketahunnya ditunjukan bahwa *domestic distribution* dan *export* sepeda motor pada tahun 2020 masing-masing adalah 3,660,616 dan 700,392.

Penggunaan kendaraan bermotor di Provinsi Bali juga mengalami peningkatan yang cukup pesat setiap tahunnya, salah satunya adalah penggunaan sepeda motor yang dapat dilihat penggunaan kendaraan pada tahun 2019 sebanyak 4,352,596. Data yang diambil dari website Badan Pusat Statistik Provinsi Bali[[2]](#footnote-2) (BPS Bali). Hal ini dikarenakan kurangnya minat masyarakat dalam menggunakan kendaraan umum karena beberapa alasan. Disamping itu, hal yang menyebabkan masyarakat lebih memilih untuk beralih dalam menggunakan sepeda motor adalah karena tingkat kemacetan lalu lintas di Bali sudah semakin meningkat karena banyaknya kendaraan dari luar Bali yang datang.

Melihat perkembangan industri sepeda motor saat ini, terdapat banyak sekali produsen sepeda motor di Indonesia. Setidaknya ada sekitar 4 merek besar produsen sepeda motor di Indonesia yaitu, Honda, Yamaha, Suzuki, dan Kawasaki. Banyaknya tipe sepeda motor yang diciptakan oleh perusahaan-perusahaan sepeda motor membuat calon pembeli perlu menentukan tipe sepeda motor yang tepat sesuai kriteria yang diinginkan oleh calon pembeli. Banyaknya kriteria yang diinginkan oleh calon pembeli sepeda motor menyebabkan sulitnya pihak dealer motor untuk memberikan rekomendasi tipe sepeda motor yang tepat untuk calon pembeli. Sehingga pemilihan sepeda motor menjadi suatu masalah bagi calon pembeli. Maka dari itu perlu dibuat suatu sistem rekomendasi pemilihan sepeda motor yang bisa diakses secara umum untuk mengatasi masalah pemilihan sepeda motor oleh calon pembeli.

Penelitian sebelumnya yang pernah dilakukan untuk mengatasi masalah yang sama ataupun hampir sama memberikan hasil yang cukup baik, namun pada penelitian tersebut hanya dapat melakukan pencarian sepeda motor. Sementara itu sudah banyak penelitian yang dilakukan dengan pendekatan ontologi dalam melakukan pencarian, pada penelitian itu sebagian besar memberikan hasil yang baik. Sehingga pada penelitian ini penulis akan menawarkan sistem dengan menggunakan pendekatan ontologi sebagai tulang punggung dari sistem dimana user dapat melakukan penjelajahan dan pencarian informasi tentang sepeda motor serta melihat rekomendasi sepeda motor. Sistem yang akan dibangun adalah Sistem Rekomendasi Pemilihan Sepeda Motor dengan Pendekatan Ontologi.

Penggunaan ontologi dalam merepresentasikan informasi sepeda motor dapat menjadi solusi dalam permasalahan ini. Ontologi merupakan sebuah cara untuk representasi pengetahuan dari sekumpulan konsep dalam sebuah domain dengan hubungan-hubungan (*relationships*) antara konsep-konsep tersebut. Salah satu metode yang akan digunakan dalam pembangunan model ontologi adalah metode Methontology. Metode Methontology merupakan salah satu metode yang dapat digunakan dalam mengembangkan ontologi, menggunakan kembali ontologi dan merekayasa ulang (C. & V, 2017). Metode ini memiliki kemampuan *life cycle* ontologi berdasarkan pada pengembangan *prototype* yang mengijinkan untuk melakukan penambahan, perubahan, dan penghapusan *term* (Gómez-Pérez et al., 2003 dalam Novianti & Diaz, 2017).

Untuk membantu dalam melakukan pengambilan keputusan pemilihan sepeda motor yang sesuai dengan keinginan calon pembeli berdasarkan kriteria yang diinginkan maka dibutuhkan suatu sistem pendukung keputusan. Maka dari itu perlu dibangun sistem pendukung keputusan untuk pemilihan sepeda motor. Metode yang akan digunakan pada fitur rekomendasi pemilihan sepeda motor adalah metode *Simple Additive Weighting* (SAW) yang memiliki kemampuan untuk memecahkan masalah multikriteria (Irawan & Abadan, 2019).

Berdasarkan hal tersebut, maka penulis mencoba untuk meneliti mengenai bagaimana merancang sebuah sistem rekomendasi untuk pemilihan sepeda motor berbasis teknologi web. Pada penelitian ini, penulis mengembangkan ontologi yang kemudian akan diterapkan pada sistem rekomendasi pemilihan sepeda motor. Pembangunan ontologi menggunakan metode yang sering digunakan yaitu, metode Methontology, sedangkan pembangunan sistem menggunakan metode Prototyping. Dalam melakukan pengambilan keputusan dalam pemilihan sepeda motor akan akan menggunakan metode *Simple Additive Weighting* karena mampu menyeleksi alternatif terbaik.

1. **Rumusan Masalah**

Pada penelitian ini mengangkat 4 (empat) rumusan masalah yang menjadi pokok permasalahan pendekatan metode Methontology dalam pembangunan model ontologi, pendekatan metode Prototyping dalam pembangunan sistem rekomendasi pemilihan sepeda motor, modul penjelajahan dan pencarian dalam sistem dan tingkat kemudahan dan kegunaan sistem.

1. Bagaimana metode Methontology digunakan untuk pembangunan model ontologi sepeda motor?
2. Bagaimana metode Prototyping digunakan untuk pembangunan sistem rekomendasi pemilihan sepeda motor?
3. Bagaimana modul penjelajahan semantik dan pencarian semantik dibangun dalam sistem rekomendasi pemilihan sepeda motor?
4. Bagaimana persepsi kemudahan dan persepsi kegunaan dalam sistem rekomendasi pemilihan sepeda motor?
5. **Batasan Masalah**

Supaya pengembangan sistem rekomendasi pemilihan sepeda motor ini lebih terfokus dan dilakukan secara tepat, maka batasan-batasan masalah antara lain:

1. Mengimplementasikan metode Methontology sebagai metode pengembangan ontologi dan metode *Prototyping* sebagai metode pengembangan sistem.
2. Mengimplementasikan ontologi sepeda motor pada program aplikasi berbasis website.
3. Kriteria-kriteria yang digunakan dalam pengambilan keputusan rekomendasi adalah harga, konsumsi bahan bakar, kapasitas bahan bakar, dan kecepatan.
4. Data sepeda motor yang digunakan adalah data sepeda motor yang baru yang masih diproduksi dan diambil dari produsen sepeda motor di Indonesia saat ini adalah Honda, Yamaha, Suzuki, dan Kawasaki.
5. Sistem rekomendasi yang dibuat tidak mendukung untuk rekomendasi pembelian sepeda motor bekas.
6. Sistem dibangun berbasis web menggunakan bahasa pemrograman PHP dan framework laravel.
7. **Tujuan Penelitian**

Adapun tujuan dari penelitian ini adalah sebagai berikut:

1. Membangun model ontologi untuk memudahkan pengklasifikasian pengetahuan mengenai sepeda motor yang kemudian diimplementasikan kedalah sistem berbasis website.
2. Membangun sistem informasi sepeda motor berbasis website, dimana sistem ini bertujuan untuk melakukan penjelajahan dan pencarian informasi sepeda motor serta sebagai sistem pengambilan keputusan sepeda motor.
3. Menguji seberapa mudahkan sistem yang dibangun untuk digunakan dan seberapa berguna sistem yang dibangun terhadap pengguna.
4. **Manfaat Penelitian**

Penelitian ini memiliki manfaat untuk beberapa komponen, antara lain sebagai berikut.

1. Masyarakat : penelitian ini dapat memudahkan masyarakat dalam melakukan pencarian sepeda motor berdasarkan informasi kriteria sepeda motor.
2. Keilmuan : penelitian ini dapat menjadi referensi pada penelitian lain yang memiliki karakteristik ontologi.

**BAB II**

**TINJAUAN PUSTAKA**

1. **Tinjauan Studi**

Terdapat beberapa penelitian serupa baik pendekatan, metode, maupun lingkup penelitian yang mirip dengan penelitian ini. Berikut ini beberapa penelitian tersebut.

1. Penelitian (habibulloh, 2017) membahas mengenai penerapan Tanimoto Similarity untuk rekomendasi pemilihan tipe sepeda motor. Penelitian ini dilatarbelakangi dengan hasil rekomendasi dealer yang masih dilakukan secara manual sehingga kurang efektif dan efisien dalam melakukan pemilihan tipe sepeda motor. Macam-macam sepeda motor Honda yang dijual terdapat 3 macam yaitu motor Sport, Motor Bebek, dan Motor Matic dengan berbagai tipe. Metode penelitian yang digunakan dalam penelitian ini adalah menggunakan Metode Tanimoto Similarity. Metode ini memberikan kemudahan kepada calon pembeli sepeda motor untuk memilih kriteria yang tepat untuk calon pembeli tersebut. Hasil dari penelitian ini adalah dihasilkan program aplikasi berbasis web memberikan kemudahan dan kepuasan pada *Customer* untuk melakukan pencarian sepeda motor yang sesuai dengan kriteria *Customer* dan memberikan Informasi yang lengkap untuk segala tipe sepeda motor.
2. Penelitian (Irawan & Abadan, 2019) membahas mengenai pengambilan keputusan untuk pembelian *Handphone* dengan menggunakan metode SAW. Penelitian ini dilakukan untuk merancang model sistem pendukung keputusan sesuai dengan data yang diperoleh. Penelitian ini menghasilkan infromasi berupa ranking dari sebuah *Handphone*. Hasil yang diperoleh setelah melakukan simulasi didapatkan urutan prioritas dengan perolehan nilai prioritas yang didapatkan dari hasil penilaian.
3. Penelitian (Novianti & Diaz, 2017) membahas mengenai Ontologi untuk pencarian program studi pada perguruan tinggi. Pencarian informasi dapat dilakukan dengan membangun sebuah sistem pencarian berbasis semantik. Ontologi menjadi dasar penerapan sistem semantik karena dapat merepresentasikan konten menjadi basis pengetahuan yang dapat dipahami oleh mesin. Sistem ini bertujuan untuk membantu calon mahasiswa untuk mendapatkan deskripsi informasi program studi yang diinginkan. Tahapan pengujian sistem, diperoleh hasil nilai *recall* sebesar 0.95 dari nilai *precision* sebesar 0.93, menunjukan bahwa sistem memiliki tingkat relevansi dan ketepatan pengambilan informasi yang tinggi.
4. Penelitian (Mutawalli, 2018) membahas mengenai penerapan ontologi dalam pemodelan sistem pendeteksian penyakit. Ontologi dalam konsep dalam membangun sistem dengan metode perhitungan menggunakan metode *jaccard similarity*. Hasil akurasi sistem penyakit demam tifoid sebanyak 88%, penyakit demam berdarah sebanyak 96%, dan penyakit malaria sebanyak 77% sehingga sistem dapat digunakan oleh *paramedic* sebagai referensi pengetahuan pada saat melakukan anammeses awal.
5. Penelitian (Jiang, Hagelien, & Natvig, 2019) membahas mengenai pencarian data terbuka memakan waktu yang banyak dan sulit menemukan dataset yang relevan. Sebagian besar katalog data terbuka hanya menyediakan pencarian berbasis kata kunci tanpa memahami maksud pengguna dan makna konseptual dari dataset. Penggunaan semantic ontologi dalam melakukan pencarian telah dieksplorasi dengan baik sebagai upaya untuk meningkatkan kualitas pencarian data yang relevan. Pada penelitian ini menyajikan pendekatan dalam pencarian kumpulan data terbuka, dimana semantik bergantung pada metadata yang dapat menjelaskan data tersebut. Natural Language Processing atau Pemrosesan Bahasa Alami (NLP) diterapkan untuk menghubungkan data dan mengindeks data dan kemungkinan independensi bahasa alami yang digunakan untuk menggambarkan dataset. Dengan menghubungkan manual dataset dengan konsep ontologi untuk domain dan penerbit data. Hasil menunjukan bahwa pencarian dengan semantik pendekatan ini meningkatkan kualitas dan efisiensi pencarian data terbuka.
6. Penelitian (Pramartha, Davis, & Kuan, 2017) bertujuan untuk merancang dan membangun portal digital yang akan mendokumentasikan dan melestarikan aspek warisan budaya Bali untuk kepentingan masyarakat luas khususnya generasi muda. Peneliti menyajikan rincian dengan menghubungkan satu aspek dari Budaya Bali dan sistem komunikasi tradisional Bali (Kulkul). Warisan budaya ini sebagian besar hanya diketahui secara tacit knowledge di masyarakat bali dan cenderung terdokumentasikan. Dibuatlah sebuah ontologi dasar kulkul untuk mendukung pencarian dan penelusuran semantik dari portal online dan sumber daya terkait.
7. Penelitian (Yunita, 2017) bertujuan untuk membangun sebuah ontologi dalam aplikasi yang berbasis web semantik untuk pencarian lowongan pekerjaan berdasarkan profil para pencari kerja yaitu IPK, keahlian, dan pendidikan. Banyaknya website yang menyediakan informasi lowongan pekerjaan yang sangat banyak menyebabkan kesulitan dalam melakukan pencarian pekerjaan yang cocok oleh pencari kerja, sehingga perlu adanya sistem pencari yang terintegrasi dengan website-website penyedia lowongan kerja dan dapat mengakomodasi pencari kerja mencari pekerjaan yang cocok. Penggunaan ontologi dapat menjadi solusi dari permasalahan tersebut, dimana ontologi menjadi *backbone* dari teknologi web semantik
8. Penelitian (Putra & Epriyano, 2017) membahas mengenai kesulitan konsumen dalam melakukan pemilihan sepeda motor *sport*. Salah satu cara untuk mengatasi masalah tersebut adalah pemanfaatan teknologi informasi dengan membuat aplikasi yang menggunakan metode *Analytical Hierarchy Process* (AHP). AHP adalah metode yang dapat digunakan untuk menunjang keputusan dengan mengubah nilai-nilai kualitatif menjadi nilai-nilai kuantitatif sehingga keputusan yang diambil menjadi lebih objektif. Pembuatan sistem pendukung keputusan berbasis web berfokus pada sepeda motor *sport* 150cc dengan menentukan prioritas utama dari beberapa kriteria serta alternative yang ada untuk mengambil sebuah keputusan.
9. **Ontologi**
10. Definisi Ontologi

Definisi mengenai ontologi yang sering digunakan, dimana disebutkan “An ontology is an explicit specification of a conceptualization”. Ontologi adalah suatu cara untuk merepresentasikan suatu domain pengetahuan secara eksplisit mengenai suatu konsep tertentu dengan memberikan makna, property, dan relasi pada konsep tersebut (Badron, Agus, & Hatta, 2017).

Sebuah ontologi mendefinisikan kosakata umum bagi para peneliti yang digunakan untuk berbagi informasi dalam domain. Hal ini mencakup definisi dari konsep dasar dalam domain dan hubungan diantara mereka. Terdapat beberapa tujuan untuk mengembangkan ontologi yaitu sebagai berikut (Novianti, 2016).

1. Berbagi pemahaman umum dari struktur informasi antar pengguna atau *software agent*.
2. Memungkinkan kembali penggunaan domain pengetahuan.
3. Membuat asumsi domain yang eksplisit.
4. Memisahkan domain pengetahuan dari operasional pengetahuan.
5. Menganalisis domain pengetahuan.
6. Manfaat Ontologi

Terdapat beberapa manfaat dalam menggunakan ontologi, yaitu : (Badron et al., 2017)

1. Menjelaskan suatu domain pengetahuan secara eksplisit; memberikan struktur hirarki dari konsep untuk menjelaskan sebuah domain dan bagaimana mereka berhubungan.
2. Berbagi pemahaman dari informasi yang terstruktur; sebagai contoh beberapa web yang berbeda memiliki informasi medis. Jika web tersebut dipakai bersama dan dipublikasikan dengan dasar ontologi yang sama maka perangkat lunak dapat mengekstrak dan mengumpulkan informasi dari situs yang berbeda.
3. Penggunaan ulang domain pengetahuan; apabila ingin membangun ontologi yang luas dapat mengembangkan ontologi yang telah ada sebelumnya dan mengintegrasikan dengan beberapa ontologi lainnya yang relevan dengan ontologi yang ingin dibangun.
4. Komponen Ontologi

Komponen pada ontologi web semantik terdiri akan *instances, property, class,* dan *axiom*. Berikut ini akan dijelaskan secara singkat elemen dasar pembentukan ontologi semantik web (Badron et al., 2017).

1. *Instances*

*Instance* atau yang disebut juga dengan *individual* adalah anggota dari suatu *classes*. *Instance* juga dapat dipandang sebagai objek yang ada pada domain.

1. *Property*

*Property* merupakan *binary relation*. Ada dua jenis *property* pada ontologi web semantik, yaitu *object property* dan *datatype property*. *Object property* digunakan untuk menghubungkan objek dengan objek lainnya. Sedangkan *datatype property* digunakan untuk menghubungkan objek dengan *datatype value*.

1. *Class*

*Class* menjelaskan sebuah konsep dalam suatu domain yang terdiri dari beberapa *instance*. *Class* juga dikenal sebagai *concept, object* dan *categories*. Sebuah *class* memiliki *subclass* yang ditujukan untuk menyatakan *concept* lebih spesifik dari *superclass*.

1. *Axiom*

*Axiom* merupakan aturan eksplisit yang digunakan untuk membatasi nilai dari *class* maupun *instance*. *Property* dari relasi adalah jenis *axiom*.

1. Pembangunan Ontologi

Pembangunan ontologi sering dikenal dengan *ontology* *engineering* atau *ontology* *development*. Pada umumnya dalam pembangunan ontologi melibatkan 3 (tiga) hal, yaitu metode, bahasa, dan perangkat lunak. Dalam pembangunan ontologi biasanya menggunakan metode Ontology Development 101, metode Methontology dan On-To Knowledge (OTK) (Badron et al., 2017). Bahasa Ontologi yang digunakan dalam pembuatan web adalah bahasa OWL (*Ontology Web Language*). Perangkat lunak dalam pembangunan ontologi ada bermacam-macam, salah satunya yang digunakan dalam penelitian ini adalah Protégé.

1. ***Ontology Web Language* (OWL)**

OWL (*Ontology Web Language*) adalah bahasa ontologi yang digunakan dalam pembuatan web. OWL merupakan bahasa yang digunakan untuk mendeskripsikan *class*, *property*, dan relasi antara objek dalam dalam suatu cara yang dapat diinterpretasikan oleh mesin. OWL menyediakan 3 (tiga) sub bahasa yang berbeda untuk kebutuhan tertentu dari pengguna, yaitu sebagai berikut (Novianti & Diaz, 2017).

1. OWL Lite

OWL Lite adalah bahasa OWL yang menyediakan pendefinisian hirarki *class* dan *property* dengan *constraint* yang sederhana. Penggunaan dari OWL Lite jika pengguna hanya membutuhkan hirarki *class* yang sederhana.

1. OWL DL

OWL DL adalah bahasa OWL yang mendukung pengguna yang menginginkan ekspresi maksimum tanpa kehilangan perhitungan. OWL DL meliputi semua konstruksi dalam OWL dengan batasan tertentu. OWL DL dapat menghasilkan hirarki klasifikasi secara otomatis dan mampu mengecek konsisten dalam suatu ontologi karena OWL mendukung *reasoning*.

1. OWL Full

OWL Full adalah bahasa OWL yang ditujukan kepada pengguna yang menginginkan sub bahasa yang sangat ekspresif dengan kebebasan sintaksis dari RDF tanpa ada jaminan komputasional. OWL Full memungkinkan ontologi untuk merubah arti dari suatu kosakata yang sudah didefinisikan.

Penggunaan OWL sangat diperlukan dalam pembangunan model ontologi agar dapat diinterpretasikan oleh perangkat lunak atau mesin. OWL dibuat berdasarkan RDF dan RDF *Schema*. Setiap *class* dalam OWL berhubungan dengan individu yang disebut perluasan *class*. Individu dalam perluasan *class* disebut dengan *instance*. OWL membedakan *property* menjadi 2 (dua) kategori, yaitu sebagai berikut.

1. *Object Property*, merupakan *property* yang menghubungkan individu yang satu dengan yang lain.
2. *Datatype* *Property*, merupakan *property* yang menghubungkan individu ke nilai data.
3. **RDF**

RDF (*Resource Development Framework*) adalah sebuah layer yang digunakan untuk merepresentasikan semantic dari isi halaman web. RDF merupakan suatu model sederhana yang digunakan untuk menghubungkan antara *properties* dan nilai. Model dari RDF merupakan suatu *triple* yang dinamakan *statement*, yang dimana *statement* tersebut terdiri dari *subject* yang dihubungkan dengan *object* melalui *predicate* (Novianti, 2016).

1. **Methontology**

Methontology merupakan salah satu metodologi untuk pengembangan ontologi. Methontology menawarkan pelaksanaan aktivitas konseptualisasi yang detail setiap tahapannya dan juga memiliki kemampuan untuk merekayasa ulang ontologi (Novianti & Wibawa, 2017). Methontology merupakan salah satu metode pengembangan model ontologi  yang memiliki keunggulan terkait dengan deskripsi setiap aktivitas. Methontology juga memiliki kemampuan yaitu ontologi yang dibangun dapat digunakan kembali untuk pengembangan sistem lebih lanjut (Fernández, Gómez-Pérez, & Juristo, 1997).



Gambar 2.1 Methontology yang diadopsi dari Fernández et al. (1997)

Adapun tahapan-tahapan dalam metode Methontology yaitu sebagai berikut:

1. Tahap spesifikasi

Tahap spesifikasi ini memiliki tujuan untuk menghasilkan dokumen spesifikasi ontologi informal, semi-forrmal atau formal yang ditulis dengan menggunakan bahasa alami. Tahap ini menggunakan sekumpulan representasi perantara atau menggunakan beberapa kompetensi pertanyaan

1. Tahap Akuisisi Pengetahuan

Tahap akuisisi pengetahuan merupakan aktivitas independen dalam pembangunan ontologi, sebagian besar tahap ini dilakaukan bersamaan dengan tahap spesifikasi. Para ahli, buku bahkan ontologi lain dapat mendadi dumber pengetahuan yang dapat dijelaskan dengan wawancara, curah pendapat, dan analisis teks.

1. Tahap Konseptualisasi

Tahap ini bertujuan untuk menyusun sebuah domain pengetahuan dalam model konseptual yang menjelaskan masalah dan solusi dalam kosakata domain yang diidentifikasi dalam aktivitas spesifikasi ontologi. Yang dilakukan pertama kali adalah membuat *Glosary of Terms*. *Terms* mencakup konsep, individu, kata kerja and properti. *Glosary of Terms* mengidentifikasi dan mengumpulkan semua domain pengetahuan yang dapat digunakan. Setelah hampir selesai menyelesaikan *Glosary of Terms* lalu kelompokanlah *terms* sebagai konsep dan kata kerja.

1. Tahap Integrasi

Pada tahap integrasi, pertimbangkan lagi penggunaan kembali definisi yang sudah dibangun ke dalam ontologi dengan tujuan mempercepat pembangunan ontologi.

1. Tahap Implementasi

Dalam tahapan ini adalah mengimplementasikan proses dari desain ontologi yang akan dibangun.

1. Tahap Evaluasi

Pada tahap evaluasi bertujuan untuk melakukan penilaian teknis atas ontologi, lingkungan perangkat lunaknya, dan dokumentasi yang berkaitan dengan kerangka acuan setiap tahap dalam siklus hidupnya. Istilah pada evaluasi ini adalah verifikasi dan validasi. Verifikasi adalah menjamin kebenaran ontologi,perangkat lunak dan dokumen yang berkaitan dengan kerangka acuan setiap fasenya. Sedngkan validasi adalah menjamin ontologi, perngakat lunak dan dokumen sesuai dengan sistem seharusnya.

1. Tahap Dokumentasi

Dalam dokumentasi ontologi, tidak ada pedoman yang disepakati. Dokumentasi yang tersedia hanyalah dalam bentuk kode ontologi, bahasa alami yang dilampirkan dan makalah yang diterbitkan dalam prosiding konferensi dan jurnal yang menetapkan pertanyaan-pertanyaan penting dari ontologi yang sudah dibangun.

1. **Web Semantik**
2. Definisi Web Semantik

Web semantik merupakan suatu pendekatan yang dikembangkan khusus pada teknologi World Wide Web (WWW) atau yang biasa disebut dengan istilah web. Pengertian web semantik yang sering dirujuk berasal dari Tim Berners-Lee yang menyatakan "*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.*" (Badron et al., 2017).

Dilihat dari penjelasan di atas, tujuan dari web semantik bukanlah untuk menggantikan web yang sudah ada saat ini, namun bertujuan untuk 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.

Web semantik memungkinkan suatu web menjadi lebih cerdas dikarenakan memiliki basis pengetahuan (*knowledge base*) didalamnya dalam bentuk ontologi. Dalam teknologi web semantik, ontologi berperan sebagai inti (*core technology*) sehingga dapat disebut sebagai ontologi web semantik (*semantic web ontology*).

Ontologi web semantik memungkinkan komputer agar sumber daya yang ada pada web (*web resource*) tidak hanya dapat dipahami oleh manusia (*human-readable*) namun juga dapat dipahami oleh mesin/komputer (*machine-readable*) sehingga memungkinkan untuk melakukan pengelolaan kumpulan sumber daya web (*web resource*) secara sistematis dan terstruktur.

Dengan demikian, ontologi web semantik dapat mengatasi permasalahan pada sumber daya informasi yang akan terus berkembang namun belum terstruktur.

1. Arsitektur Web Semantik

Teknologi web semantik terbagi dalam beberapa *layer* (lapisan) yang terdapat pada arsitekturnya. Model arsitektur web semantik disebut sebagai *Semantic Web Stack* (Badron et al., 2017).

1. Lapisan/*layer* pertama yaitu URI (*Uniform Resource Identifier*) dan Unicode merupakan fitur penting dari sebuah web. URI merupakan standard untuk lokasi dan identitas suatu sumber daya web (*web resource*). Sedangkan *Unicode* merupakan standar pengkodean set karakter internasional yang memungkinkan semua bahasa manusia dapat digunakan didalam web menggunakan satu bentuk standar dari URI.
2. Lapisan kedua yaitu XML (*Extensible* Markup *Language*). XML merupakan sintaks yang umum digunakan dalam web terutama web semantik. XML merupakan bahasa *markup* untuk dokumen yang berisi informasi yang terstruktur.
3. Lapisan selanjutnya yaitu RDF (*Resource Description Framework*) yang merupakan format representasi data untuk web semantik. RDF merupakan *framework* yang berbentuk *graph* untuk merepresentasikan dan mendeskripsikan informasi pada sumber daya web (*web resource*).
4. Lapisan berikutnya yaitu OWL (*Ontology Web Language*) yang merupakan bahasa ontologi yang direkomendasikan oleh W3C.
5. OWL merupakan bahasa yang lebih kaya dan kompleks untuk mendeskripsikan *resource*. Untuk melakukan *query* data RDF dan OWL maka hadirlah SPARQL (*SPARQL Protocol and RDF Query Language*). *Query* diperlukan untuk mengambil informasi untuk web semantik.
6. Lapisan-lapisan berikutnya yaitu *Logic*, *Proof* dan *Trust*. Lapisan *logic* merupakan aturan/*rule* dan sistem untuk melakukan penalaran pada ontologi, lapisan *proof* mengeksekusi aturan dan mengevaluasi bersama-sama dengan mekanisme lapisan *trust* untuk mempercayai bukti yang diberikan pada aplikasi atau tidak. Untuk masukan yang terpercaya, kriptografi perlu digunakan, seperti tanda tangan digital (*digital signature*) untuk verifikasi asal-usul sumber data. Dan lapisan teratas yaitu *user interface & application* agar memungkinkan manusia dalam menggunakan aplikasi web semantik.



Gambar 2.2 Arsitektur Web Semantik (Badron et al., 2017)

1. ***Prototyping***

Metode *prototyping* merupakan metode yang digunakan dalam pembangunan sistem rekomendasi pemilihan sepeda motor pada penelitian ini. Penggunaan metode *prototyping* memiliki kelebihan dalam mendapatkan umpan balik yang cepat dari user. Tahapan awal dalam model ini adalah perencanaan yang dilakukan secara cepat yang kemudian dilanjutkan dengan perancangan. Setelah tahapan perancangan dilakukan, maka dilanjutkan dengan membuat *prototype* aplikasi. Setelah itu *prototype* akan diberikan kepada user untuk mendapatkan umpan balik (Subhiyakto & Utomo, 2017).



Gambar 2.3 Metode Prototyping

1. **SPARQL**

SPARQL merupakan standar yang dikeluarkan W3C untuk melakukan *query* dari untuk memperoleh data dari sumber daya web (*web resource*) yang terdapat pada dokumen RDF dan OWL. SPARQL terdiri dari *triple pattern* yang sama dengan RDF *triple* yang terdiri dari *subject*, *predicate*, dan *object*, dimana masing-masing dari *subject*, *predicate*, dan *object* dapat menjadi variabel pada SPARQL. Klausa yang digunakan dalam *query* SPARQL, yaitu sebagai berikut (Badron et al., 2017).

1. PREFIX

*Statement* PREFIX merupakan sebuah metode yang digunakan sebagai penunjuk yang membawa suatu *resource* yang dalam hal ini diwakili URI (*Uniform Resource Identifier*). PREFIX biasanya digunakan untuk menyingkat *resource*.

1. SELECT

*Statement* SELECT merupakan klausa yang mendefinisikan daftar dari variabel-variabel yang akan dikembalikan sebagai hasill dari eksekusi *query*.

1. WHERE

*Statement* WHERE mendefinisikan *triple* *pattern* yang harus dimiliki oleh setiap hasill *query* yang valid. Seluruh pola yang merepresentasikan sebuah kalimat RDF harus sesuai dengan RDF *triple* yang terdiri dari *subject*, *predicate*, dan *object*.

1. **Apache Jena Fuseki**

Apache Jena merupakan sebuah *framework java* yang dapat digunakan secara *open source* untuk membangun sistem maupun *web semantic*. Framework ini terdiri dari berbagai macam Application Programming Interface (API) yang berintegrasi untuk memproses data RDF. Apache Jena memiliki metode untuk membaca dan menulis RDF sebagai *XML* (Mutawalli, Suhriani, & Supardianto, 2018).

1. **Protégé**

Protégé merupakan perangkat lunak yang dikembangkan oleh *Stanford* *Center for Biomedical Informatics Research at the Stanford University School of Medicine*. Perangkat lunak protégé merupakan perangkat lunak yang bersifat *Open Source* dibawah lisensi Mozilla Public License (MPL). Protégé merupakan perangkat lunak alat bantu untuk membantu untuk mengemabngkan ontologi yang didasarkan pada basis pengetahuan *Knowledge Base System*. Protégé dapat digunakan untuk membuat, mengedit, dan menyimpan ontologi (Fahrurrozi & SN, 2017).

1. ***Simple Additive Weighting* (SAW)**

*Simple Additive Weighting* (SAW) merupakan salah satu metode yang digunakan dalam sistem pendukung keputusan. Metode SAW dapat juga disebut dengan metode penjumlahan terbobot. Metode SAW mencari penjumlahan dari rating kinerja pada suatu alternatif pada setiap atribut (Irawan & Abadan, 2019). Berikut adalah formulasi dari metode SAW.

Dimana rij merupakan rating kinerja ternormalisasi dari alternatif Ai pada atribut Cj; i = 1, 2, 3, …, m dan j = 1, 2, 3, …, n. Nilai preferensi untuk setiap alternatif (Vi) dihitung sebagai berikut.

Dimana nilai Vi yang lebih besar akan mengindikasikan bahwa alternatif Ai lebih baik untuk dipilih.

1. ***Technology* *Accpetance* *Model* (TAM)**

*Technology Accpetance Model* (TAM) merupakan suatu model yang mengidentifikasi suatu kemungkinan dalam penerimaan penggunaan dan memberikan langkah yang tepat. *Technology Accpetance Model* (TAM) terdiri dari 2 (dua) faktor, yaitu sebagai berikut (Kurniawan, Saputra, & Prasetyawan, 2018).

1. Persepsi kegunaan (*perceived usefulness*) adalah sejauh mana seseorang percaya bahwa menggunakan teknologi akan meningkatkan kinerja pekerjaannya.
2. Persepsi kemudahan (*perceived easy of use*) adalah sejauh mana seseorang percaya bahwa menggunakan teknologi akan memudahkan pekerjaan.

Menurut Robey (1979) “A system that does not help people perform their jobs is not likely to be received favorably in spite of careful implementation efforts" yang berarti suatu sistem yang tidak mampu untuk membantu sesorang dalam melakukan pekerjaannya kemungkinan besar akan tidak akan diterima dengan baik. Persepsi kemudahan (*perceived easy of use*) dan persepsi kegunaan (*perceived usefulness*) penggunaan teknologi diindikasikan sebagai konstruksi dasar yang berpengaruh dalam keputusan untuk menggunakan teknologi informasi. Definisi konseptual persepsi kegunaan dan persepsi kemudahan dapat dieksplorasi dengan menggunakan item kuesioner di setiap dimensi (Davis, 1989).



Gambar 2.4 Diagram alur dari TAM, diadopsi dari Davis (1989)

**BAB III**

**METODOLOGI PENELITIAN**

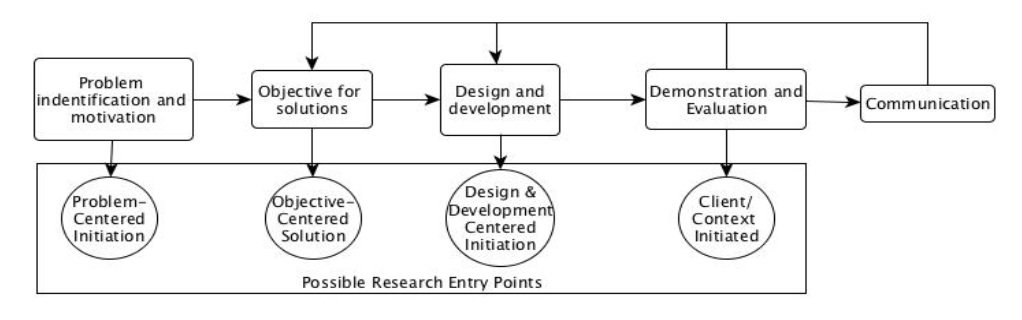
Pada bagian ini menjelaskan mengenai metodologi penelitian, dimana akan menjelaskan gambaran langkah-langkah yang akan dilakukan dalam melaksanakan penelitian ini. Langkah-langkah tersebut meliputi pengumpulan data dan alur penelitian yang menggunakan metode *Design Science Research Methodology* (DSRM).

1. **Pengumpulan Data**

Pada pengumpulan data, data yang akan digunakan dalam penelitian ini dibagi menjadi 2 (dua), yaitu data untuk membangun model ontologi dan data hasil pengujian sistem. Dalam sistem rekomendasi pemilihan sepeda motor ini, sistem membutuhkan data yang akan digunakan dalam pembangunan model ontologi yang kemudian akan diimplementasikan kedalam sistem untuk ditampilkan kepada pengguna dalam bentuk informasi. Data yang digunakan dalam membangun model ontologi adalah data mengenai kumpulan informasi sepeda motor yang dibutuhkan untuk sistem. Data tersebut didapatkan dengan pengambilan langsung dengan datang ke dealer-dealer terkait dan dari literatur terkait yang termasuk dalam *domain* sepeda motor. Data yang diambil adalah nama unit sepeda motor, merek sepeda motor, jenis transmisi sepeda motor, tipe sepeda motor, tahun produksi sepeda motor, volume silinder sepeda motor, harga sepeda motor, kecepatan maksimal sepeda motor, kapasitas bahan bakar sepeda motor, dimensi sepeda motor, dan konsumsi bahan bakar sepeda motor.

1. ***Design Science Research Methodology***

Alur penelitian ini menggunakan metode *Design Science Research Methodology* (DSRM). Metode DSRM menawarkan pendekatan yang berguna untuk melakukan penelitian yang bertujuan untuk menciptakan dan mengevaluasi desain dalam mengatasi masalah. Seperti pada Gambar 3.1, DSRM memiliki beberapa tahapan, yaitu (1) *Problem identification and motivation*; (2) *Objective for solutions*; (3) *Design and development*; (4) *Demonstration and Evaluation*; dan (5) *Communication* (Pramartha et al., 2017).



Gambar 3.1 Metode DSRM (Pramartha et al., 2017)

1. ***Problem indentification and motivation***

Tahapan ini merupakan tahapan untuk indentifikasi masalah yang diangkat pada penelitian ini. Permasalahan dalam penelitian ini adalah perkembangan industri sepeda motor saat ini sangat meningkat, disamping itu penggunaan sepeda motor juga semakin tahun semakin meningkat. Pemilihan sepeda motor yang tepat menjadi sebuah permasalahan untuk calon pembeli sepeda motor.

1. ***Objective for solution***

Tahapan ini merupakan tahapan untuk menentukan solusi yang digunakan untuk mengatasi permasalahan tersebut, yaitu dengan membuat sebuah aplikasi sistem yang dapat membantu dalam melakukan penjelajahan dan pencarian informasi mengenai sepeda motor. Sistem yang akan dibangun menggunakan model ontologi, karena model ontologi dapat digunakan dalam penyajian informasi secara semantik serta melakukan pengorganisasian dan pemetaan kumpulan sumber informasi secara sistematis dan terstruktur. Sehingga ketika pengetahuan dari sepeda motor dikumpulkan secara ontologi, maka akan memberikan kemudahan dalam pengorganisasian dan manajemen data.

1. ***Design and development***

Tahapan ini merupakan tahapan untuk melakukan desain dan pengembangan sistem dari penelitian yang dilakukan yaitu meliputi pembangunan model, analisis kebutuhan dan pembangunan sistem.

1. **Pembangunan Model**

Metode yang digunakan dalam pembangunan model ontologi pada penelitian ini menggunakan metode Methontology. Metode Methontology merupakan salah satu metodologi pembangunan model ontologi yang memiliki keunggulan terkait dengan deskripsi setiap aktivitas yang harus dilakukan secara mendetail. Selain itu, metode Methontology memiliki kemampuan yaitu ontologi yang dibangun dapat digunakan kembali untuk pengembangan sistem lebih lanjut (Fernández et al., 1997).

1. **Tahap Spesifikasi**

Tujuan dari tahap spesifikasi adalah untuk menghasilkan dokumen spesifikasi ontologi informal, semi formal atau formal yang ditulis dalam bahasa alami, menggunakan representasi menengah atau menggunakan pertanyaan kompetensi. Berikut deskripsi dari ontologi sepeda motor.

1. Domain : Sepeda Motor
2. Tujuan : Untuk membangun model ontologi untuk pendukung pengambilan keputusan pemilihan sepeda motor.
3. Tingkat Formalitas : Semi formal
4. Lingkup : Sepeda motor
5. Sumber Pengetahuan : Observasi langsung, leteratur terkait
6. **Tahap Akuisisi Pengetahuan**

Dalam proses pengembangan ontologi ini, sebagaian besar akuisisi pengetahuan dilakukan dalam hubungannya dengan tahap spesifikasi kebutuhan dengan proses pengembangan ontologi. Dalam tahap akuisisi pengetahuan ontologi Sepeda Motor menggunakan teknik sebagai berikut.

1. Berdiskusi bersama ontology engineer untuk menyusun draft awal dokumen spesifikasi kebutuhan.
2. Analisis teks informal, untuk mempelajari konsep-konsep utama.
3. Analisis teks formal. Mengidentifikasi struktur yang akan dideteksi (definisi, penegasan, dll.) dan jenis pengetahuan yang dikontribusikan oleh masing-masing (konsep, atribut, nilai, dan hubungan).

Dalam penelitian ini mengunakan data sepeda motor merk Yamaha, Honda, Suzuki, dan Kawasaki. Data tersebut diperoleh dari hasil observasi langsung, literatur maupun sumber internet yang dapat dipercaya.

1. **Tahap Konseptualisasi**

Konseptulisasi memiliki tujuan menyusun sebuah domain pengetahuan dalam bentuk konseptual serta mengatur dan mengelola pengetahuan yang didapat pada proses akuisisi pengetahuan. Sehingga menghasilkan konseptual dari ontologi sepeda motor.

1. **Tahap Integrasi**

Dalam tahap integrasi ini bertujuan untuk pertimbangkan lagi penggunaan kembali definisi yang sudah dibangun ke dalam ontologi dengan tujuan mempercepat pembangunan ontologi.

1. **Tahap Implementasi**

Dalam tahap ini, dilakukan proses pendefinisian kembali dan proses implementasi dari rancangan ontologi sepeda motor dengan menggunakan perangkat lunak Protégé.

1. **Tahap Evaluasi**

Pada tahap evaluasi, penulis melakukan proses evaluasi dengan melakukan SPARQL query pada aplikasi Protégé.

1. **Tahap Dokumentasi**

Pada tahap ini, dilakukan proses dokumentasi ontologi sepeda motor baik dalam kode ontologi, teks bahasa alami yang dilampirkan pada definisi formal, maupun makalah yang diterbitkan dalam proses konferensi dan jurnal yang mengatur pertanyaan-pertanyaan penting dari ontologi yang sudah dibangun.

1. **Merancang Sistem Pendukung Keputusan**

Dalam penelitian ini, metode sistem pendukung keputusan yang digunakan adalah sistem pendukung keputusan *Simple Additive Weighting* (SAW). *Simple Additive Weighting* (SAW) merupakan salah satu metode yang digunakan dalam sistem pendukung keputusan. Metode SAW dapat juga disebut dengan metode penjumlahan terbobot.

Dimana rij merupakan rating kinerja ternormalisasi dari alternatif Ai pada atribut Cj. Nilai preferensi untuk setiap alternatif (Vi) dihitung sebagai berikut. Dimana nilai Vi yang lebih besar akan mengindikasikan bahwa alternatif Ai lebih baik untuk dipilih.

Kriteria yang digunakan dalam metode SAW ini untuk sistem rekomendasi pemilihan sepeda motor ini antara lain, Harga (C1), Kapasistas Bahan Bakar (C2), Kecepatan (C3), dan Konsumsi Bahan Bakar (C4).

1. **Pembangunan sistem**

Metode pengembangan sistem yang digunakan untuk membangun sistem rekomendasi pemilihan sepeda motor adalah metode *Prototyping*. Metode *Prototyping* memiliki beberapa tahapan yaitu pengumpulan kebutuhan, membangun *prototyping*, evaluasi *prototyping*, mengkodekan sistem, menguji sistem, evaluasi sistem dan penggunaan sistem.



Gambar 3.2 Tahapan Pembangunan Sistem dengan Metode Prototyping

1. **Pengumpulan Kebutuhan**

Tahap ini menjelaskan mengenai analisis kebutuhan sistem meliputi analisis kebutuhan fungsional dan analisis kebutuhan non-fungsional.

1. Analisis Kebutuhan Fungsional

Analsisi kebutuhan fungsional meliputi kegunaan dari sistem, maka dapat dijabarkan kebutuhan fungsional sistem pada Tabel 3.1.

Tabel 3.1 Kebutuhan Fungsional Sistem

|  |  |  |
| --- | --- | --- |
| **Kode** | **Deskripsi** | **Target Pengguna** |
| F1 | Sistem yang dirancang memungkinkan pengguna dapat menjelajahi informasi sepeda motor (*borwsing*, sistem yang dirancang dapat memungkinkan pengguna melakukan pencarian spesifik mengenai sepeda motor berdasarkan batasan-batasan terntentu (*searching* , dan sistem yang dirancang dapat memungkinkan pengguna untuk dapat mengetahui hasil rekomendasi sepeda motor berdasarkan batasan-batasan kriteria tertentu. | *Guest User* |

1. Analisis Kebutuhan Non-fungsional

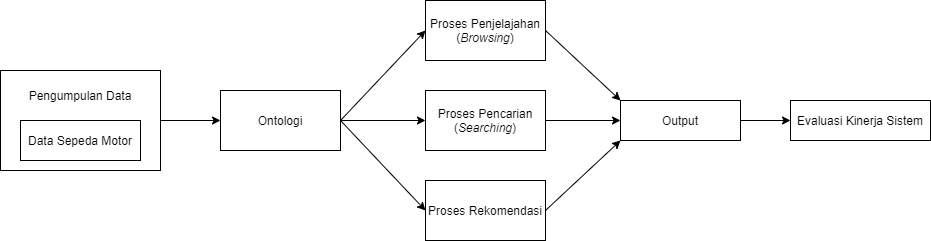
Analisis kebutuhan non-fungsional dari sistem ini dimana sistem ini dapat menampilkan antarmuka pengguna yang mudah dipahami, yang dapat memberikan kenyamanan dan kemudahan bagi pengguna saat menggunakan sistem, baik dalam jangka waktu yang lama maupun ketika penggunaan pertama.

1. **Membangun *Prototyping***

Pada tahap ini, dilakukan pembangunan *prototyping* dari sistem dengan membuat perancangan sementara yang berfokus pada penyajian, yaitu dengan membuat format input dan format output. Bentuk perancangan sistem yang akan dibangun antara lain dengan membuat desain umum dari sistem, *use case diagram*, *activity diagram*, dan perancangan *user interface* pengguna.

1. Desain Umum Sistem

Desain sistem yang akan dibuat berupa tahapan-tahapan yang akan dilalui. Dimulai dari tahapan pengumpulan dan penyimpanan data, proses penjelajahan, proses pencarian, dan proses rekomendasi hingga tahap evaluasi kinerja sistem yang dapat dilihat pada Gambar 3.3 yang menunjukan rancangan desain umum sistem rekomendasi pemilihan sepeda motor.



Gambar 3.3 Desain Umum Sistem

Pada tahap pengumpulan data, peneliti mengumpulkan data mengenai informasi yang berkaitan dengan sepeda motor. Data akan diinputkan oleh peneliti kedalam model ontologi yang telah dibangun sebelumnya yang kemudian akan diimplementasikan kedalam sistem.

Selanjutnya setelah melakukan tahap penjelajahan dan pencarian , maka sistem akan mengeluarkan hasil atau *output* sistem berupa informasi sepeda motor yang relevan terhadap pencarian *user* dan hasil pencarian akan saling berkaitan secara semantik. Pada saat melakukan tahap rekomendasi dimana sistem akan mengeluarkan hasil atau *output* sistem yang sebelumnya telah dilakukan perhitungan untuk menentukan nilai dengan menggunakan metode SAW.

Terkahir adalah tahap evaluasi kinerja sistem yang bertujuan untuk mengetahui kinerja dari sistem yang telah dibangun. Pada tahap ini akan dilakukan pengujian sistem dari segi logika, fungsi-fungsi pada sistem dan akurasi yang dari hasil penjelajahan, pencarian dan rekomendasi.

1. *Use Case Diagram*

Dalam sistem rekomendasi pemilihan sepeda motor ini, *use case diagram* digambarkan dalam Aktor atau *user*. Aktor merupakan seseorang atau segala sesuatu diluar sistem yang harus berinteraksi dengan sistem. Aktor dalam sistem ini adlaah *guest user*. Pada Tabel 3.2 menunjukan definisi aktor pada *use case diagram* pada sistem.

Tabel 3.2 Deskripsi Aktor pada Use Case Diagram

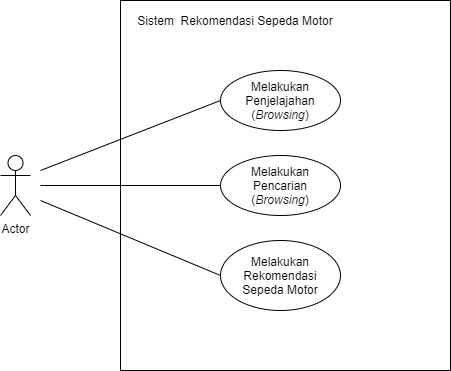
|  |  |  |
| --- | --- | --- |
| **No.** | **Aktor** | **Deskripsi** |
| 1. | *Guest User* | Merupakan pengguna yang memiliki hak akses untuk melakukan penjelajahan pengetahuan, pencarian pengetahuan dan rekomendasi sepeda motor pada sistem. |

Selanjutnya pada Tabel 3.3 menunjukan deskripsi *use case diagram* dari sistem yang dibangun.

Tabel 3.3 Deskripsi Use Case Diagram pada Sistem

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Proses** | **Deskripsi** | **Kode** |
| 1. | Melakukan penjelajahan (*browsing*) pengetahuan sepeda motor | Proses penjelajahan pengetahuan sepeda motor dapat dilakukan oleh *guest* *user* dengan mengklik salah satu *hyperlink* yang diinginkan. | UC1 |
| 2. | Melakukan pencarian (*searching*) pengetahuan sepeda motor | Proses pencarian pengetahuan sepeda motor dapat dilakukan oleh *guest* *user* dengan menentukan *input* pada *form* pencarian. | UC2 |
| 3. | Melakukan rekomendasi sepeda motor | Proses rekomendasi sepeda motor dapat dilakukan oleh *guest* *user* dengan menentukan variabel-variabel yang diinginkan pada *form* *input* rekomendasi. | UC3 |

*Use case diagram* dari Sistem Rekomendasi Pemilihan Sepeda Motor dapat dilihat pada Gambar 3.4.

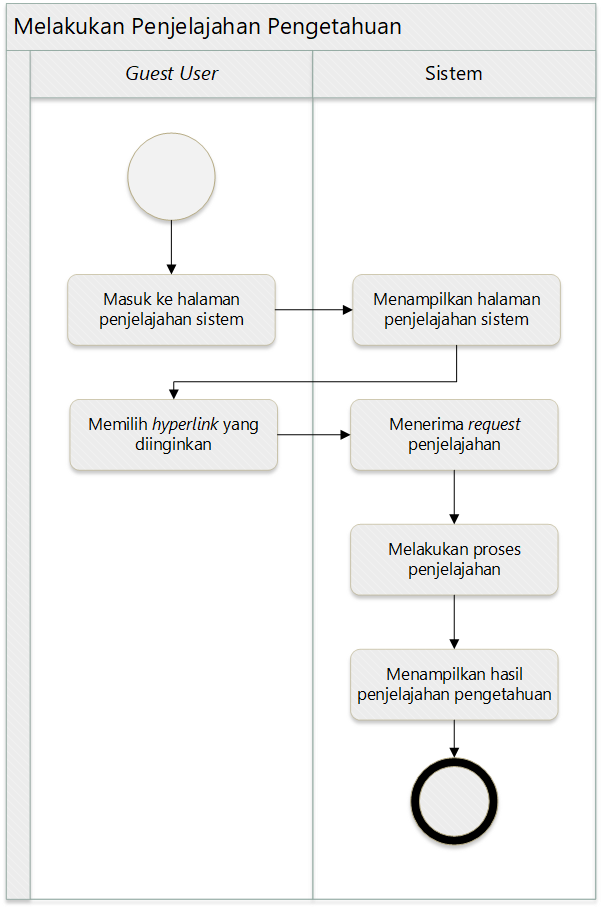


Gambar 3.4 Use Case Diagram Sistem

1. *Activity Diagram*

Tahap *activity diagram* menggambarkan alur aktivitas sekuensial dari tiap *use case diagram*, proses, dan logika dari sistem. Serta menggambarkan sebuah pekerjan dalam *workflow*. Berikut ini merupakan *activity diagram* dari masing-masing proses yang ada pada sistem.

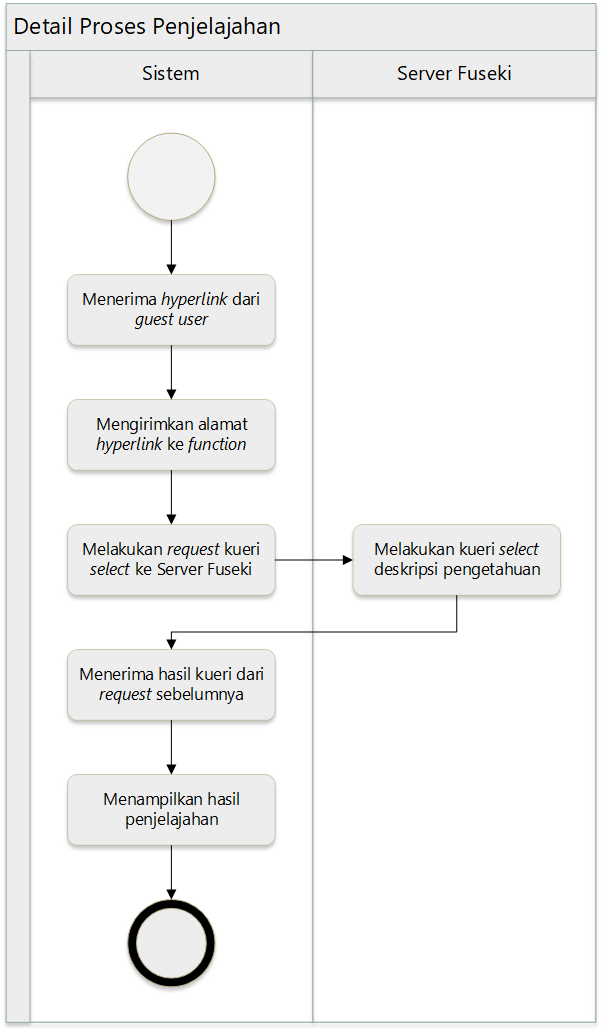
1. Melakukan Penjelajahan Pengetahuan
2. *Guest user* masuk ke halaman penjelajahan sistem.
3. Pada *dashboard* sistem terdapat beberapa *hyperlink* untuk melakukan penjelajahan sistem.
4. *Guest* *user* memilih salah satu *hyperlink* yang diinginkan.
5. Sistem menerima *request* penjelajahan dan melakukan proses penjelajahan.
6. Sistem menampilkan hasil penjelajahan pada halaman hasil penjelajahan.



Gambar 3.5 Activity diagram Melakukan Penjelajahan Pengetahuan

Pada Gambar 3.5 menunjukan *activity diagram* dari melakukan penjelajahan pengetahuan pada sistem.

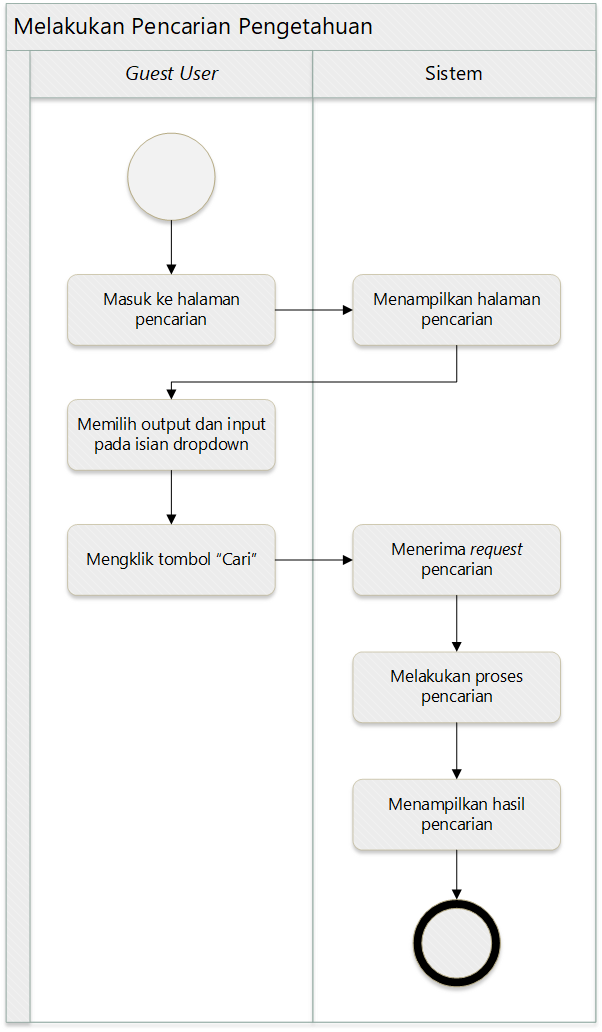
1. Detail Proses Penjelajahan Pengetahuan
2. Sistem menerima *hyperlink* penjelahan pengetahuan yang dipilih oleh *guest* *user*.
3. Alamat *hyperlink* penjelahan yang diterima sistem akan dikirimkan pada *function* yang bersesuaian.
4. Sistem akan melakukan *request* untuk melakukan kueri *select* deskripsi pengetahuan pada ontologi melalui server Fuseki.
5. Server Fuseki melakukan kueri *select* deskripsi pengetahuan pada ontologi sesuai dengan *request* dari sistem.
6. Sistem menerima hasil kueri dari *request* yang sebelumnya diminta.
7. Sistem menampilkan hasil penjelajahan pada halaman hasil penjelajahan.



Gambar 3.6 Activity diagram Detail Proses Penjelajahan Pengetahuan

Pada Gambar 3.6 menunjukan proses *activity diagram* dari detail proses penjelajahan pengetahuan.

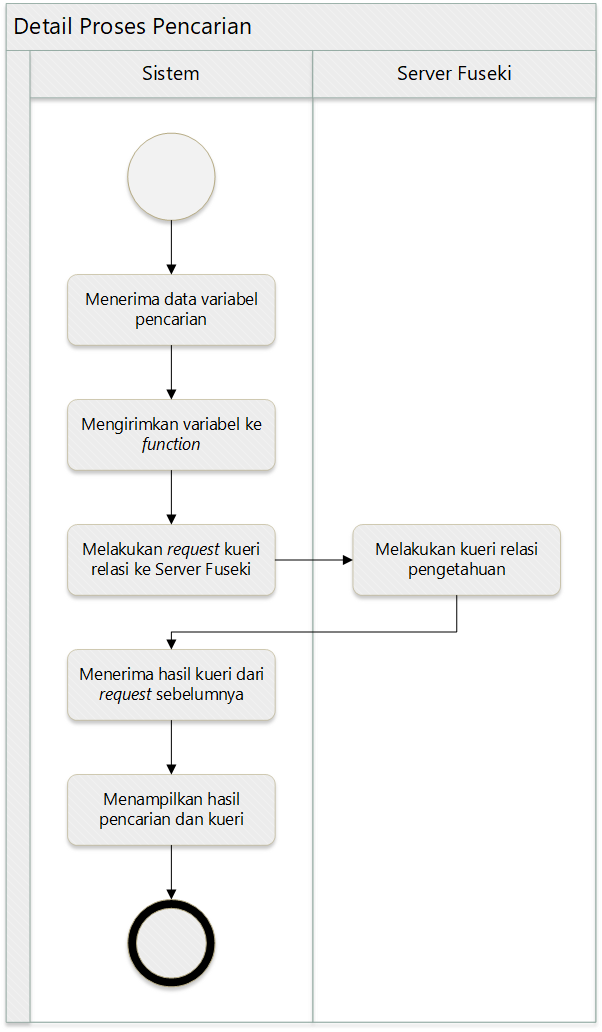
1. Melakukan Pencarian Pengetahuan
2. *Guest* *user* masuk ke halaman pencarian sistem.
3. *Guest* *user* akan memilih *output* dan *input* pada isian *dropdown*, lalu mengeksekusi pencarian pengetahuan dengan melakukan klik pada tombol “Cari”.
4. Sistem akan menerima *request* pencarian pengetahuan dan melakukan proses pencarian.
5. Sistem menampilkan hasil pencarian dan hasil kueri pada halaman hasil pencarian.



Gambar 3.7 Activity diagram Melakukan Pencarian Pengetahuan

Pada Gambar 3.7 menunjukan *activity diagram* dari melakukan pencarian pengetahuan pada sistem.

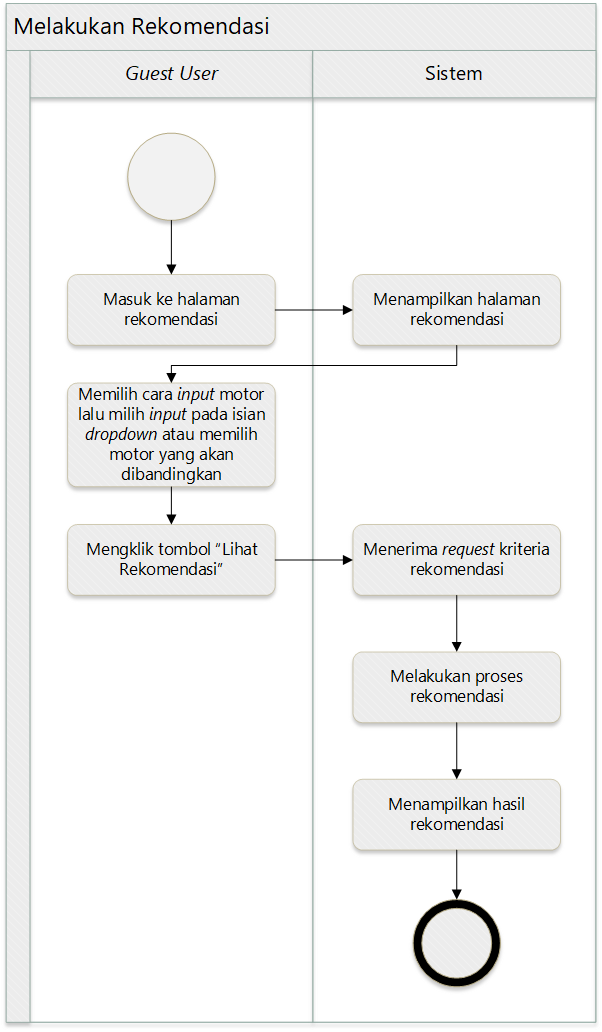
1. Detail Proses Pencarian Pengetahuan
2. Sistem menerima data pada variabel pencarian pengetahuan yang dipilih oleh *guest* *user*.
3. Variabel pencarian yang diterima sistem akan dikirimkan ke *function* yang bersesuaian.
4. Sistem melakukan *request* untuk melakukan kueri *select* relasi pengetahuan pada ontologi dengan variabel yang didapatkan kepada server Fuseki.
5. Server Fuseki melakukan kueri *select* relasi pengetahuan pada ontologi sesuai dengan *request* dari sistem.
6. Sistem menerima hasil kueri *select* dari *request* yang sebelumnya diminta.
7. Sistem akan menampilkan hasil pencarian berserta kueri yang digunakan pada server Fuseki pada halaman pencarian pengetahuan.



Gambar 3.8 Activity diagram Detail Proses Pencarian

Pada Gambar 3.8 menunjukan proses *activity diagram* dari detail proses pencarian pengetahuan.

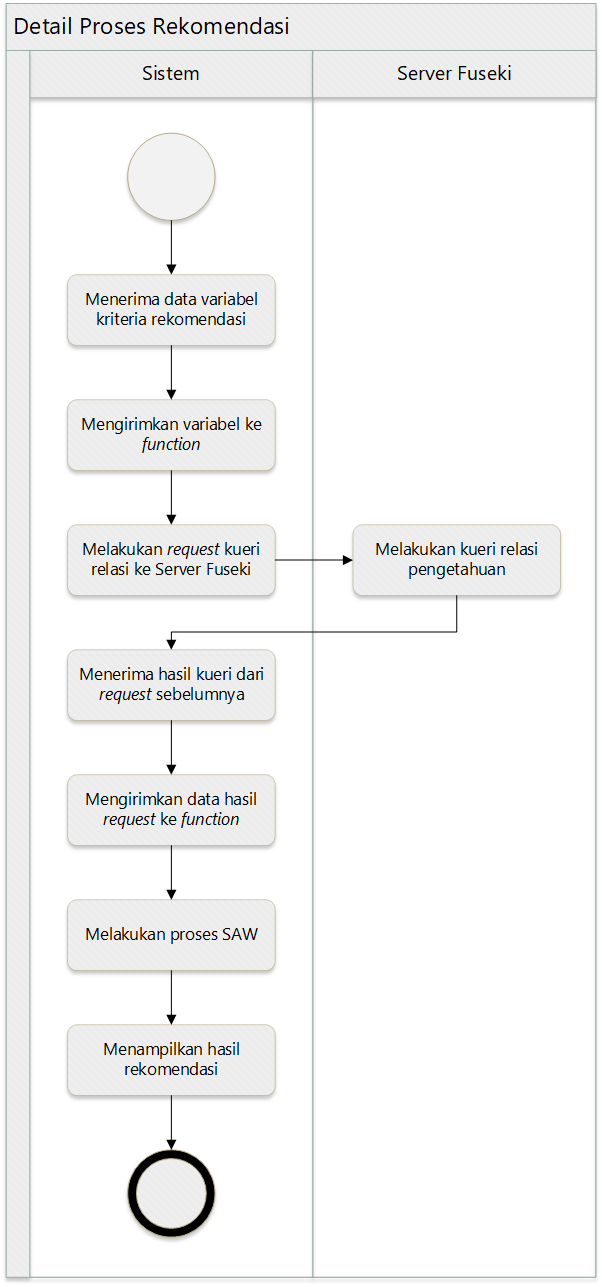
1. Melakukan Rekomendasi Sepeda Motor
2. *Guest* *user* masuk ke halaman rekomendasi sistem.
3. *Guest* *user* akan memilih cara *input* motor dan memilih *input* *filter* kriteria dibandingkan pada isian *dropdown* atau memilih motor, lalu mengeksekusi rekomendasi sepeda motor dengan melakukan klik pada tombol “Lihat Rekomendasi”.
4. Sistem akan menerima *request* rekomendasi sepeda motor dan melakukan proses rekomendasi.
5. Sistem menampilkan hasil rekomendasi sepeda motor.



Gambar 3.9 Activity diagram Melakukan Rekomendasi Sepeda Motor

Pada Gambar 3.9 menunjukan *activity* *diagram* dari melakukan rekomendasi sepeda motor pada sistem.

1. Detail Proses Rekomendasi Sepeda Motor
2. Sistem menerima data pada variabel rekoemndasi sepeda motor yang dipilih oleh *guest* *user*.
3. Variabel rekomendasi yang diterima sistem akan dikirimkan ke *function* yang bersesuaian.
4. Sistem melakukan *request* untuk melakukan kueri *select* relasi pengetahuan pada ontologi dengan variabel yang didapatkan kepada server Fuseki.
5. Server Fuseki melakukan kueri *select* relasi pengetahuan pada ontologi sesuai dengan *request* dari sistem.
6. Sistem menerima hasil kueri *select* dari *request* yang sebelumnya diminta.
7. Data sepeda motor yang didapatkan kemudian dikirimkan ke function untuk melakukan perhitungan dari metode *Simple Additive Weighting*.
8. Sistem akan menampilkan hasil rekomendasi pada halaman hasil rekomendasi sepeda motor

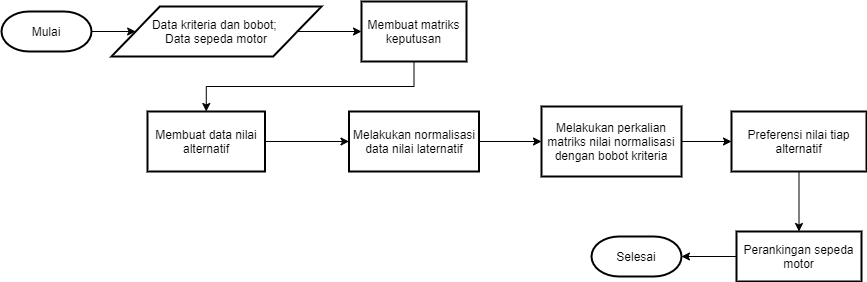


Gambar 3.10 Activity diagram Detail Proses Rekomendasi

Pada Gambar 3.10 menunjukan proses *activity* *diagram* dari detail proses rekomendasi sepeda motor pada sistem.

1. Detail Proses *Simple Additive Weighting* (SAW)

Pada Gambar 3.11 menunjukan alur peoses dari metode Simple Additive Weighting yang ada pada sistem rekomendasi sepeda motor ini.

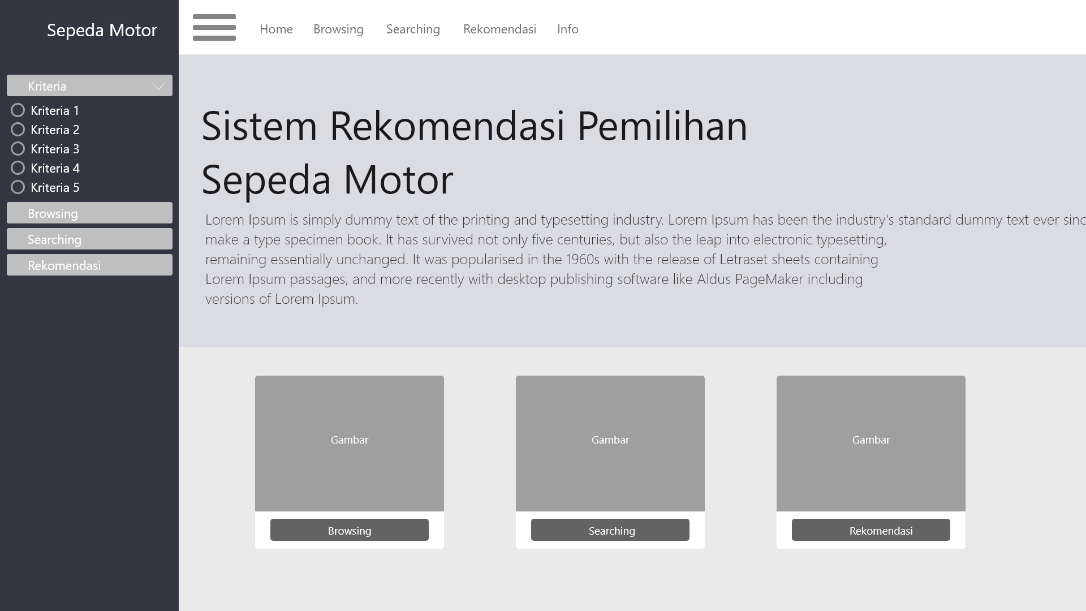


Gambar 3.11 Detail Proses Simple Additive Weighting

1. Perancangan *User Interface*

Pada sistem yang akan dibangun ini, rancang *user interface* sistem hanya akan ditujukan untuk *guest user*. Berikut penjabaran dari rancang antarmuka pada sistem.

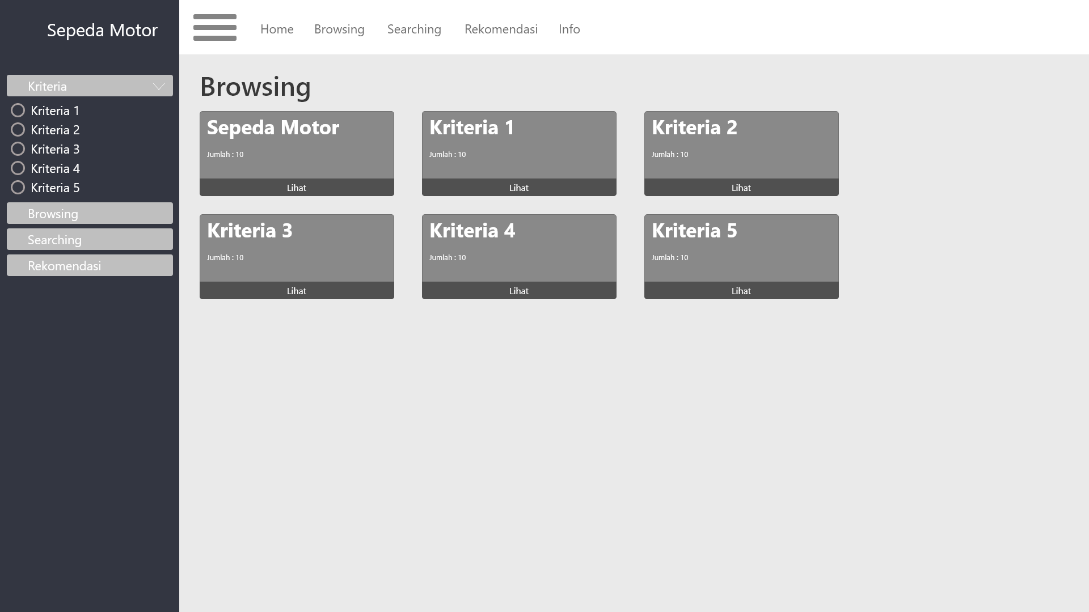
1. Rancangan *User Interface* Halaman Utama



Gambar 3.12 Rancangan user interface Halaman Utama

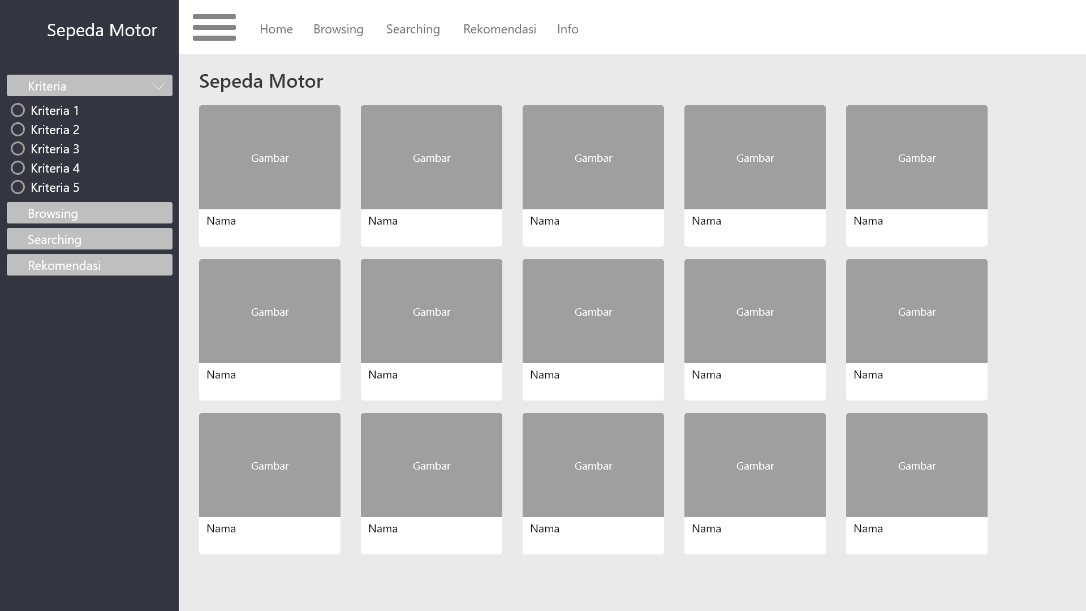
Gambar 3.12merupakan rancangan *interface* dari halaman utama. Pada halaman utama akan menampilkan deskripsi singkat dari sistem dan *hyperlink* ke halaman penjelajahan (*browsing*), halaman pencarian (*searching*), rekomendasi dan kuesioner pengujian dan evaluasi sistem.

1. Rancangan *User Interface* Halaman Penjelajahan



Gambar 3.13 Rancangan user interface Halaman Penjelajahan

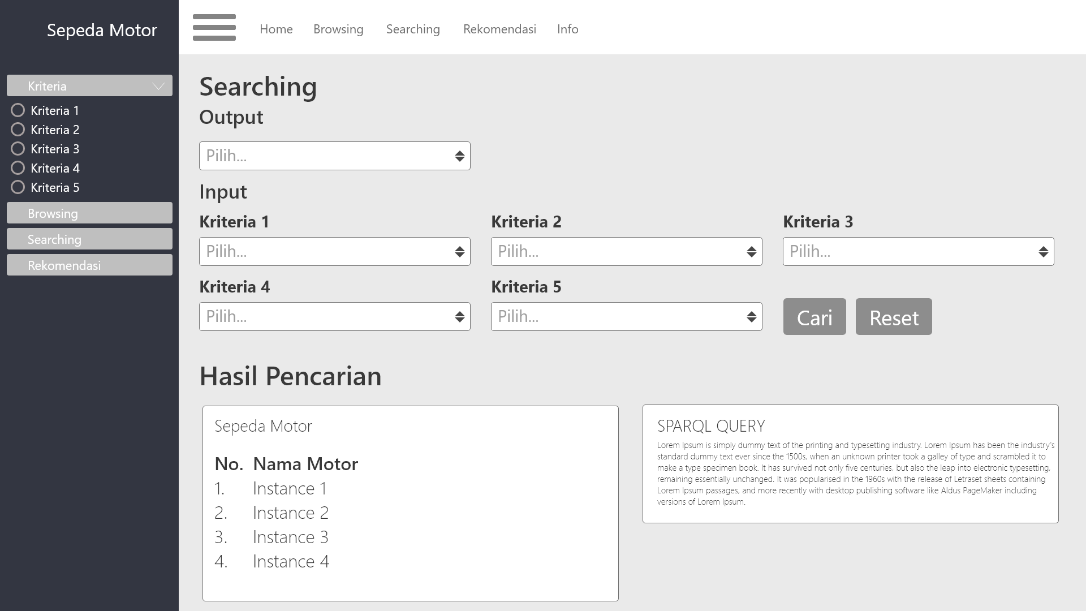
Gambar 3.13merupakan rancangan *interface* dari halaman penjelajahan. Pada halaman ini akan menampilkan pilihan kriteria *browsing* yang memiliki *hyperlink* yang akan menghubungkan ke halaman tertentu.



Gambar 3.14 Rancangan user interface Halaman Penjelajahan Sepeda Motor

Gambar 3.14merupakan rancangan *interface* dari halaman penjelajahan. Pada halaman ini akan menampilkan *list* dari kriteria yang telah dipilih oleh *guest* *user*. Dimana *list* tersebut memiliki *hyperlink* untuk menuju halaman tertentu.

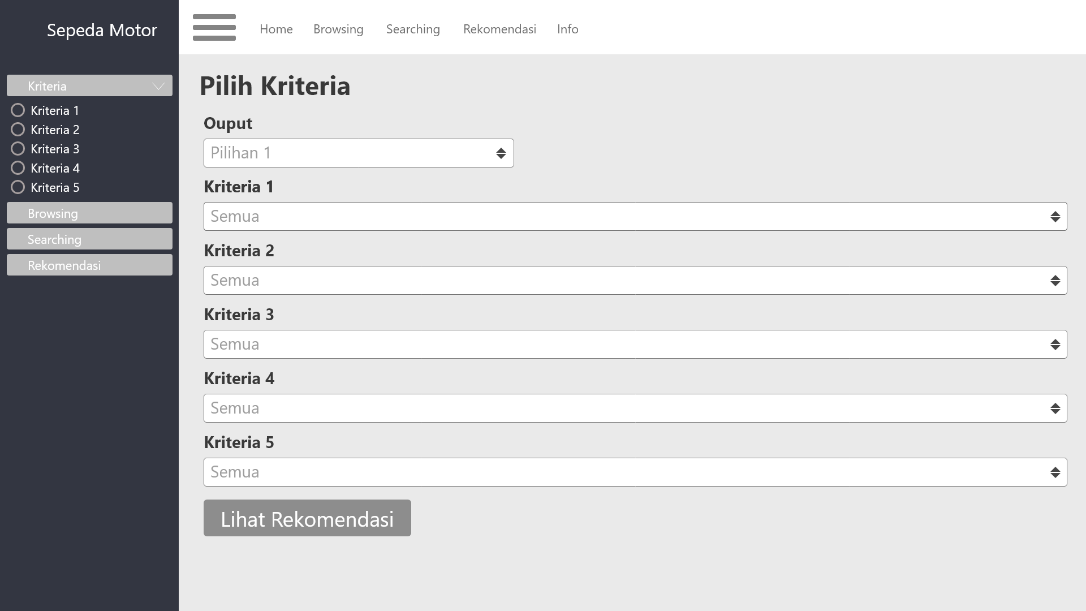
1. Rancangan *User Interface* Halaman Pencarian



Gambar 3.15 Rancangan user interface Halaman Pencarian

Gambar 3.15merupakan rancangan *interface* dari halaman pencarian. Pada halaman pencarian tersebut ditampilkan form *input* pencarian berupa *dropdown* yang menjadi dasar dari pencarian. Pada halaman tersebut juga akan ditampilkan hasil pencarian dan kueri yang digunakan untuk melakukan pencarian.

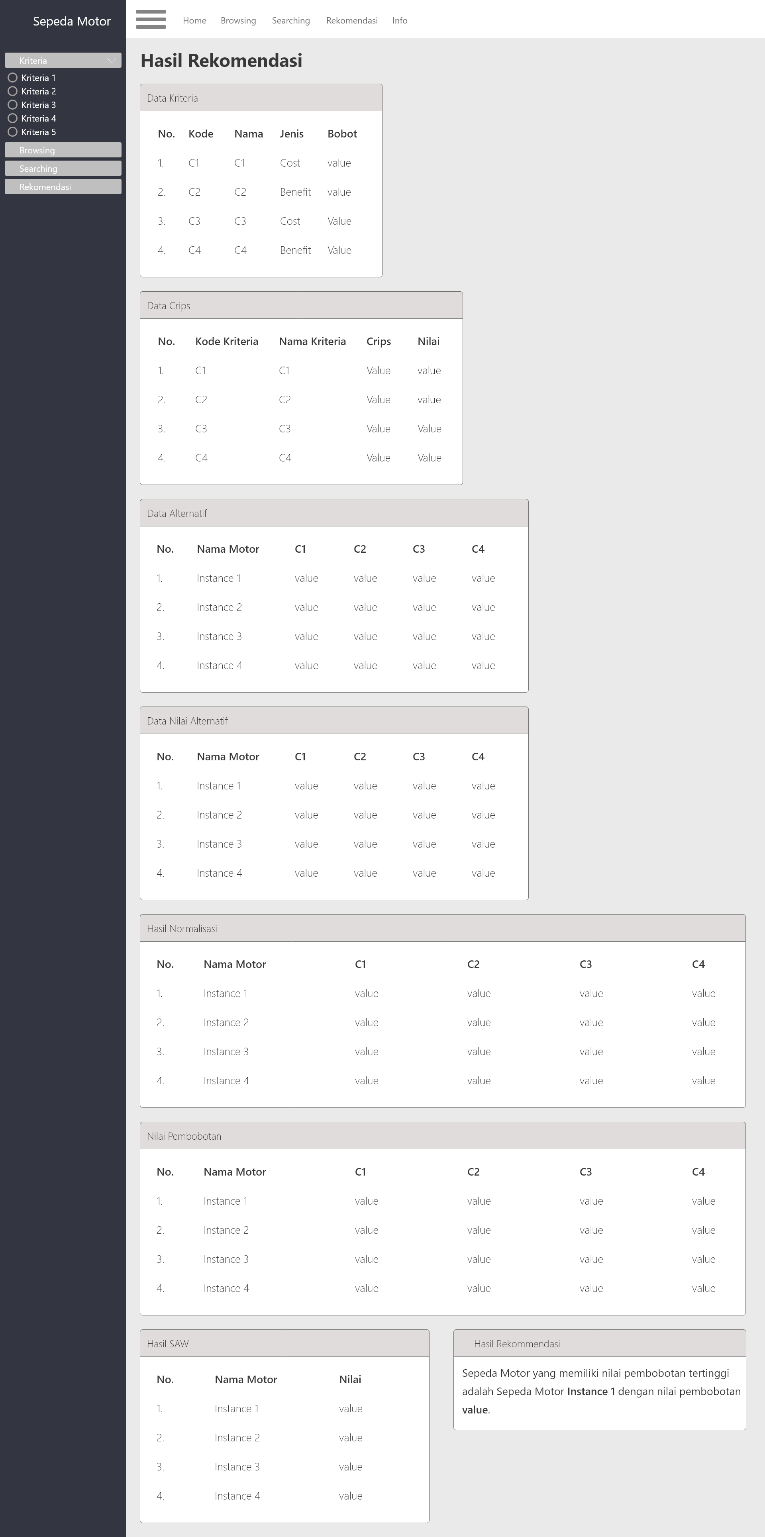
1. Rancangan *User Interface* Halaman Input Rekomendasi



Gambar 3.16 Rancangan user interface Input Rekomendasi

Gambar 3.16 Rancangan user interface Input Rekomendasimerupakan rancangan *interface* dari halaman input rekomendasi. Pada halaman tersebut ditampilkan form *input* kriteria sepeda motor berupa *dropdown* yang menjadi dasar dari pilihan sepeda motor.

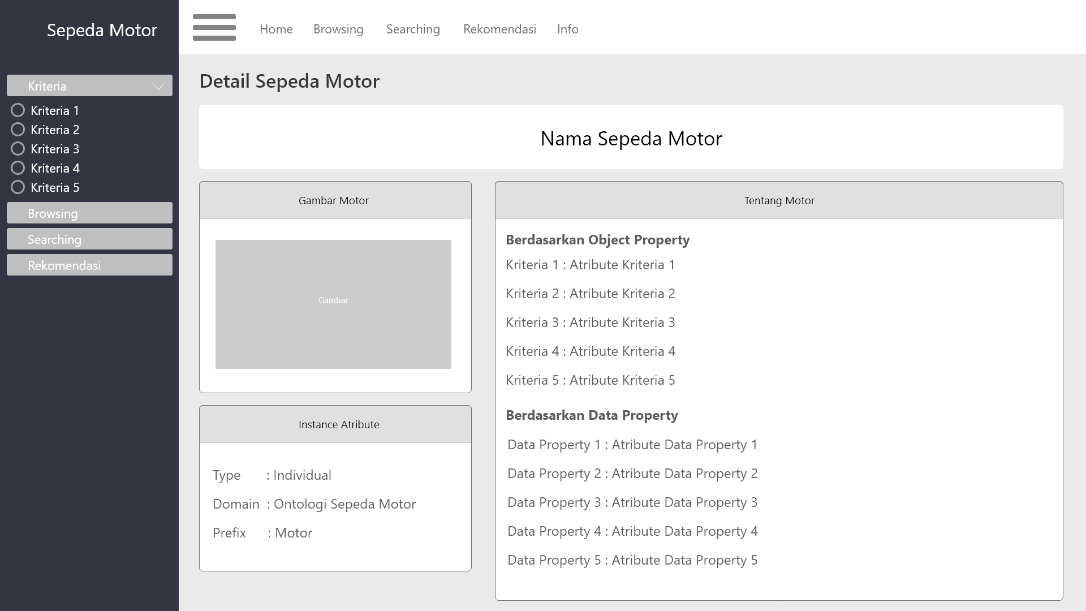
1. Rancangan *User Interface* Halaman Hasil Rekomendasi



Gambar 3.17 Rancangan user interface Halaman Hasil Rekomendasi

Gambar 3.17 Rancangan user interface Halaman Hasil Rekomendasimerupakan rancangan *interface* dari halaman hasil rekomendasi sepeda motor. Pada halaman tersebut ditampilkan informasi detail mengenai hasil perhitungan kriteria dengan menggunakan SAW hingga menampilkan motor yang memiliki nilai pembobotan tertinggi.

1. Rancangan *User Interface* Halaman Detail *Instance*



Gambar 3.18 Rancangan user interface Detail Instances

Gambar 3.18 Rancangan user interface Detail Instancesmerupakan rancangan *interface* dari halaman detail *instance*s. Pada halaman tersebut ditampilkan informasi detail mengenai *instances* yang dipilih berdasarkan hasil penjelajahan maupun hasil pencarian.

1. Fitur-fitur pada sistem

Pada Tabel 3.4 meenjabarkan fitur-fitur yang tersedia pada Sistem Rekomendasi Pemilihan Sepeda Motor. Fitur yang ddibuat sebagai tambahan dari kebutuhan sistem berdasarkan ide-ide dari peneliti.

Tabel 3.4 Fitur-fitur Sistem Rekomendasi Pemilihan Sepeda Motor

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Fungsionalitas** | **Tujuan** | **Deskripsi** |
| 1. | Penjelajahan | Memudahkan *guest* *user* dalam melakukan penjelajahan setiap bagian pengetahuan yang terdapat pada sistem. | *Guest* *user* dapat melakukan penjelahjahan setiap bagian pengetahuan yang ada dengan mengklik *hyperlink* yang diinginkan secara berkesinambungan. |
| 2. | Pencarian | Memudahkan *guest* *user* dalam melakukan pencarian pengetahuan yang diinginkan berdasarkan variabel-variabel tertentu. | *Guest* *user* dapat melakukan pencarian sepeda motor yang ingin dicari dengan memasukan variabel-variabel terkait pencarian. Hasil pencarian yang relevan dengan variabel-variabel yang sebelumnya dimasukkan akan ditampilkan. |
| 3. | Rekomendasi | Memudahkan *guest* *user* dalam mengetahui hasil rekomendasi sepeda motor dengan menggunakan metode SAW (*Simple* *Additive* *Weighting*). | *Guest* *user* dapat melihat hasil rekoemndasi sepeda motor yang dicari dengan memasukan variabel-variabel. Hasil rekoemndasi dengan variabel-variabel yang sebelumnya dimasukkan akan ditampilkan. |

1. **Evaluasi *Prototyping***

Pada tahap evaluasi *prototyping* ini bertujuan untuk mengetahui apakah *prototype* yang dibangun telah sesuai dengan keinginan. Selain dilakukan evaluasi, terdapat kegiatan revisi atau perbaikan prancangan hingga sistem dinyatakan benar dan layak untuk dibuat.

1. **Pembangunan Sistem**

Pada tahap pembangunan ini akan mulai membangun sistem yang sesuai dengan perancangan atau *prototyping* yang sebelumnya telah dibangun. Pembangunan sistem akan dilakukan dengan beberapa tahapan antara lain sebagai berikut.

1. Penyiapan basis data sistem, dalam hal ini ontologi sepeda motor yang sebelumnya dibangun. Dalam tahap ini, ontologi yang sebelumnya dibuat menggunakan perangkat Protégé 4.3 dilakukan proses *reasoning* terlebih dahulu. Ontologi yang telah diinferensi dengan benar kemudian diekspor sebagai ontologi baru sehingga menghasilkan file OWL (*Ontology Web Language*) yang nantinya akan menjadi basis data dari sistem.
2. Menyiapkan *environment* sebagai tempat melakukan *deployment* sistem. Dalam tahap ini, *environment* yang dimaksud adalah komputer atau laptop yang digunakan sebagai server lokal (*localhost*). Selain Apache, sistem juga memerlukan Apache Jena Fuseki yang bertindak sebagai server antar ontologi dengan sistem. Oleh karena itu, perlu dilakukan instalasi Apache Jena Fuseki pada komputer atau laptop yang digunakan sebagai server lokal (*localhost*).
3. Pengkodean sistem. Pada tahap ini dilakukan proses mengintegrasikan *library* EasyRDF ke dalam bahaha pemrograman PHP dan bahasa *query* SPARQL. *Library* EasyRDF ini diperlukan sebagar *parser* dari file OWL yang menjadi basis data dari sistem.
4. Menyiapkan *environment* sebagai tempat *running* sistem secara *online* untuk memudahkan dalam tahap pengujian sistem, dimana mengikutsertakan responden dari berbagai tempat maupun *platform* yang berbeda. Peneliti menggunakan *Virtual Private Server Linux* (VPS) sebagai *server* *hosting* dan Plesk sebagai *control* *panel* dari *virtual* *server*. Dalam tahap ini, dilakukan konfigurasi *virtual* *server*, instalasi Apache Jena Fuseki pada *virtual* *server* dengan menggunakan *Tomcat*. Setelah *environmoent* siap digunakan, *source* *code* sistem di unggah melalui Git pada Plesk. Terakhir, dilakukan proses pengunggahan ontologi menuju server Fuseki serta melakukan konfigurasi untuk menghubungkan sistem secara *online* dan dapat melakukan *request* terhadap *server* Fuseki dengan baik.
5. ***Demonstration***

Dalam tahap ini akan dilakukan pengujian sistem. Setelah sistem dibangun dan dapat berjalan secara *online*, kemudian akan dilakukan pengujian terhadap sistem. Dalam subbab ini akan dijelaskan mengenai skenario pengujian sistem akan menggunakan 2 (dua) langkah pengujian sistem yaitu, Pengujian *Black-Box* atau pengujian terhadap fungsional sistem dan pengujian akurasi. Pengujian sistem bertujuan untuk memastikan bahwa sistem yang dibangun telah berjalam sesuai dengan metode yang digunakan (*usefullness*) dan juga memastikan bahwa sistem yang dibangun merupakan sistem yang berguna dan mudah digunakan (*easy of use*). Berikut merupakan langkah-langkah pengujian sisem.

1. Pengujian Fungsionalitas

Pengujian fungsionalitas pada sistem akan menggunakan pengujian *Black-Box Testing*, yang merupakan pengujian yang dilakukan pada *user interface* untuk menguji apakah hasil eksekusi telah sesuai dengan masukan yang diberikan dan memeriksa fungsional dari perangkat lunak. Dalam hal ini akan menguji kemampuan sistem dalam melakukan proses-proses yang didefinisikan pada analisis kebutuhan. *Black-Box Testing* dinyatakan berhasil apabila fungsi yang ada pada sistem sesuai dengan yang diharapkan pengguna.

Tabel 3.5, Tabel 3.6, dan Tabel 3.7 memaparkan skenario pengujian *Black-Box Testing* yang diguakan sebagai panduan oleh peneliti dalam melakukan pengujian *Black-Box Testing* sistem rekomendasi sepeda motor ini.

Tabel 3.5 Skenario Black-Box Testing Penjelajahan Pengetahuan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Kode Kebutuhan : F1 | | | Kode Pengujian : U20 | |
| Kasus:  Penjelajahan *Guest User* | | | | |
| **No.** | **Kode** | **Nama Skenario** | **Hasil Pengujian** | **Kesimpulan** |
| 1. | UC4-1-1 | Menampilkan halaman penjelajahan |  |  |
| 2. | UC4-1-2 | Penjelajahan berhasil dilakukan |  |  |
| 3. | UC4-1-3 | Hasil penjelajahan berhasil ditampilkan |  |  |

Tabel 3.6 Skenario Black-Box Testing Pencarian Pengetahuan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Kode Kebutuhan : F1 | | | Kode Pengujian : U21 | |
| Kasus:  Pencarian *Guest User* | | | | |
| **No.** | **Kode** | **Nama Skenario** | **Hasil Pengujian** | **Kesimpulan** |
| 1. | UC4-1-1 | Menampilkan halaman pencarian |  |  |
| 2. | UC4-1-2 | Pencarian berhasil dilakukan |  |  |
| 3. | UC4-1-3 | Hasil pencarian berhasil ditampilkan |  |  |

Tabel 3.7 Skenario Black-Box Testing Rekomendasi

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Kode Kebutuhan : F1 | | | Kode Pengujian : U22 | |
| Kasus:  Rekomendasi *Guest User* | | | | |
| **No.** | **Kode** | **Nama Skenario** | **Hasil Pengujian** | **Kesimpulan** |
| 1. | UC4-1-1 | Menampilkan halaman rekomendasi |  |  |
| 2. | UC4-1-2 | Pencarian dan perhitungan nilai berhasil dilakukan |  |  |
| 3. | UC4-1-3 | Hasil rekomendasi berhasil ditampilkan |  |  |

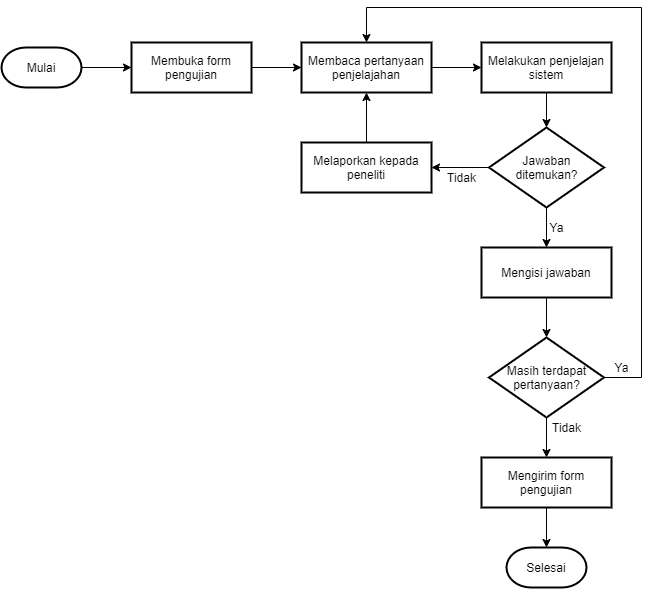
1. Pengujian Akurasi Sistem

Pengujian akurasi sistem bertujuan untuk mengetahui keakuratan sistem rekomendasi sepeda motor dalam menampilkan data hasil penjelajahan dan pencarian sepeda motor oleh pengguna serta keakuratan hasil perhitungan dari fitur rekomendasi. Pengujian yang dilakukan adalah pengujian semantic browsing, semantic searching dan perbandingan nilai perhitungan SAW dengan perhitungan manual.

Pada tahap pengujian ini, peneliti mencari sejumlah peserta yang bersedia untuk melakukan pengujian sistem. Peserta dikumpulkan untuk melakukan sesi pelatihan singkat yang bertujuan untuk menguraikan langkah-langkah yang perlu dilakukan oleh peserta dalam pengujian sistem. Setelah melakukan sesi pelatihan, peserta akan diminta untuk menyelesaikan berbagai tugas penjelajahan (*borwsing*) dan pencarian (*searching*) menggunakan fitur dan fasilitas yang tersedia disistem manajemen pengetahuan. Tugas-tugas yang dilakukan oleh peserta dijabarkan sebagai berikut.

Pertama, peserta diminta untuk menyelesaikan berbagai tugas penjelajahan (eskprlorasi sistem dengan megikuti salah satu tautan menarik kee yang lain) pada modul penjelajahan. Di setiap tugas penjelajahan, peserta diminta untuk menjawab pertanyaan dengan membuat beberapa elemen *query* menggunakan modul penjelajahan sistem manajemen pengetahuan. Berikut ini contoh pernyataan penjelajahan.

1. Sebutkan 5 sepeda motor yang menggunakan sistem transmisi auto!

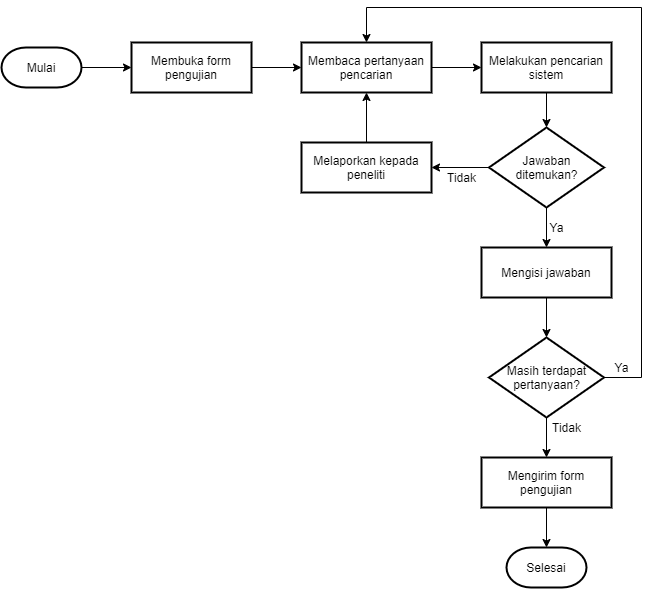


Gambar 3.19 Diagram Alir Proses Pengujian Penjelajahan

Gambar 3.19 menunjukan diagram alir proses pengujian penjelajahan yang dilakukan oleh pesert pengujian.

Kedua, peserta diminta untuk menyelesaikan berbagai tugas pencarian (meminta sebagian informasi dari database) menggunakan modul pencarian sistem. Agar dapat menjawab pertanyaa menggunakan fasilitas pencarian, peserta akan membangun elemen dari *query* sebagai *filter* *input* yang kemudian diikuti dengan mengklik tombol pencarian. Berikut ini contoh dari pertanyaan pencarian.

1. Sebutkan sepeda motor yang termasuk dalam merek Honda yang menggunakan transmisi Manual, merupakan type motor Sport dan diproduksi tahun 2020!



Gambar 3.20 Diagram Alir Proses Pengujian Pencarian

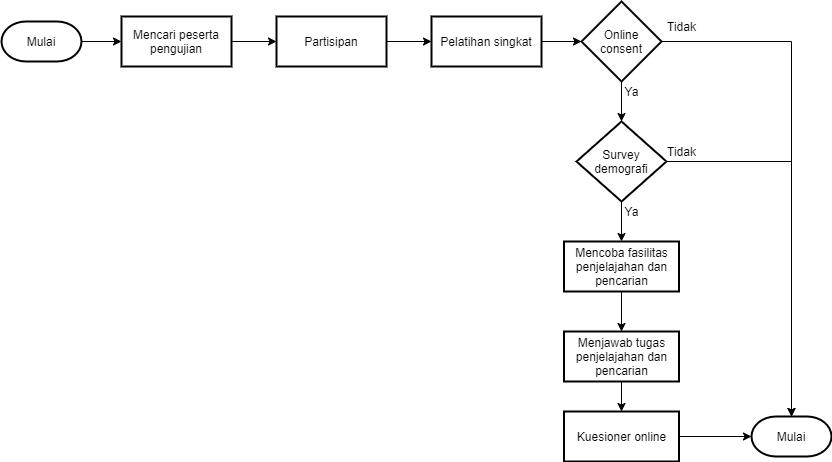
Gambar 3.20 menunjukan diagram alir proses pengujian pencarian yang dilakukan oleh peserta pengujian.

Ketiga, pengujian terhadap perhitungan manual yang bertujuan untuk melakukan pengujian pada modul rekomendasi. Pengujian ini dilakukan dengan dara menccokan hasil perhitungan sistem dengan perhitungan secara manual untuk menguji kesesuaian sistem dengan rancangan yang dibuat.

1. ***Evaluation***

Tahap akhir yang dilakukan setelah melakukan pengujian sistem adalah melakukan evaluasi sistem. Evaluasi sistem ini dilakukan untuk mengetahui sistem yang dibangun seberapa mudah digunakan dan seberapa berguna sistem ini dalam pandangan para pengguna sistem. Dalam tahap ini sistem akan dievaluasi dari segi kemudahan dan kebermanfaatan penggunaan sistem untuk mengetahui apakan pengguna menemukan sistem rekomendasi mudak digunakan dan bermanfaat dari segi perspektif belajar tentang sepeda motor dan praktik terkait.

Evaluasi sistem dirancang untuk mengetahui persepsi pengguna tentang kegunaan sisitem dan kemudahan penggunaan sistem. Kegunaan yang dirasakan dapat didefinisikan sebagai “sejauh mana seseorang percaya bahwa menggunakan sistem tertentu akan meningkatkan kinerja pekerjaaannya” (Davis, 1989). Sedangkan dari persepsi kemudahan penggunaan sistem didefinifikan sebagai “sejauh mana seseorang percaya bahwa sistem tertentu akan bebas dari upaya” (Davis, 1989).



Gambar 3.21 Diagram Alir Proses Evaluasi Sistem

Gambar 3.21 menunjukan diagram alir proses evaluasi sistem rekomendasi sepeda motor yang telah dibangun.

Proses evaluasi dan analisis dijabarkan sebagai berikut. Setelah melakukan tugas penjelajahan dan pencarian, dilanjutkan peserta *testing* dengan menjawab serangkaian pertanyaan mengenai kemudahan penggunaan dan kegunaan sistem rekomendasi ini. Peneliti mengadopsi kuesioner yang dibangun oleh Davis (1989), dimana peneliti berfokus pada dua dimensi yaitu persepsi kegunaan (*perceived usefullness*, PU) dan persepsi kemudahan penggunaan (*perceived* *easy* *of* *use*, PE). Berikut merupakan persepsi dari kegunaan (PU) yang terdiri dari 6 (enam) *item*.

1. Menggunakan Sistem Rekomendasi Sepeda Motor memungkinkan saya menyelesaikan tugas lebih cepat.
2. Menggunakan Sistem Rekomendasi Sepeda Motor memungkinkan saya meningkatkan kinerja tugas saya.
3. Menggunakan Sistem Rekomendasi Sepeda Motor memungkinkan saya meningkatkan produktivitas dalam pekerjaan saya.
4. Menggunakan Sistem Rekomendasi Sepeda Motor memungkinkan saya meningkatkan efektivitas dalam pekerjaan saya.
5. Menggunakan Sistem Rekomendasi Sepeda Motor memungkinkan saya memudahkan untuk melakukan pekerjaan saya.
6. Saya menemukan bahwa Sistem Rekomendasi Sepeda Motor berguna dalam pekerjaan saya.

Persepsi kemudahan penggunaan (PE) terdiri dari 6 (enam) *item*, yaitu sebagai berikut.

1. Saya menemukan bahwa Sistem Rekomendasi Sepeda Motor mudah untuk saya pelajari cara menggunakannya.
2. Saya menemukan bahwa Sistem Rekomendasi Sepeda Motor mudah digunakan untuk melakukan apa yang saya inginkan.
3. Saya menemukan bahwa Sistem Rekomendasi Sepeda Motor jelas dan dapat dimengerti untuk berinteraksi dengan sistem.
4. Saya menemukan bahwa Sistem Rekomendasi Sepeda Motor adalah sistem yang jelas dan mudah dimengerti.
5. Saya menemukan bahwa Sistem Rekomendasi Sepeda Motor memudahkan saya untuk terampil dalam menggunakan sistem ini.
6. Saya menemukan bahwa Sistem Rekomendasi Sepeda Motor mudah untuk digunakan.

*Item* diukur menggunakan skala Likert 7 poin (sangat setuju = 7, setuju = 6, agak setuju = 5, tidak setuju maupun setuju (netral) = 4, agak tidak setuju = 3, tidak setuju = 2, dan sangat tidak setuju = 1).

1. ***Communication***

Pada tahap komunikasi ini merupakan tahap akhir dari penelitian yang dilakukan. Dimana penelitian ini akan dituliskan pada buku tugas akhir, dan akan diterbitkan dalam jurnal ilmiah.

1. **Pengolahan dan Analisis Data**

Selanjutnya setelah melakuka tahap pembangunan sistem, yang dimana telah dilakukan pengujian dan evaluasi sistem, maka data yang diperoleh dari pengujian dan evaluasi sistem rekomendasi sepeda motor ini akan diolah dan dianalisis melalui beberaa proses. Terdapat 4 (empat) pengolahan data, yaitu sebagai berikut.

1. Pengolahan Data Pengujian Tugas Penjelajahan (*Browsing*)

Setelah peserta selesai melakukan tugas penjelajahan, peneliti menandai masing-masing kiriman dan mengklasifikasikan skim penandaan menjadi 3 (tiga) kategori.

1. Salah, dengan memberi nilai skor 0 (nol). Nilai skor ini diberikan ketika peserta tidak memberikan jawaban yang benar untuk tugas yang diberikan.
2. Sebagian benar, dengan memberi nilai skor 1 (satu). Nilai skor ini diberikan ketika jawaban yang diberikan oleh peserta sebagian benar.
3. Sepenuhnya benar, dengan memberi nilai skor 2 (dua). Nilai skor ini diberikan ketika jawahab yang diberikan olhe peserta sepenuhnya benar.

Berdasarkan ketiga kategori di atas, akan didapatkan hasil klasifikasi skim penandaan terhadap seluruh kiriman peserta yang kemudian akan dimasukkan pada tabel seperti Tabel 3.8 berikut.

Tabel 3.8 Data Pengujian Penjelajahan

|  |  |  |
| --- | --- | --- |
| **Kategori** | **Jumlah** | |
| **Kiriman** | **Persen** |
| Salah |  |  |
| Sebagian benar |  |  |
| Sepenuhnya benar |  |  |
| **Total** |  | 100% |

Selanjutnya, dari hasil skim penandaan seluruh kiriman peserta, peneliti menerapkan analisis statistik sebagai berikut.

1. Rerata (*mean*). Dengan analisis ini, didapatkan rerata ketepatan jawaban peserta pada masing-masing pertanyaan. Rerata ini akan menggambarkan keakuratan sistem dalam menampilkan data hasil penjelajahan.
2. Nilai tengah (*median*). Dengan analisis ini, didapatkan nilai tengah dari seluruh nilai skor peserta pada masing-masing pertanyaan.
3. Nilai terendah (*minimum*). Dengan analisis ini, didapatkan nilai terendah dari seluruh nilai skor peserta pada masing-masing pertanyaan.
4. Nilai tertinggi (*maximum*). Dengan analisis ini, didapatkan nilai tertinggi dari seluruh nilai skor peserta pada masing-masing pertanyaan.

Berdasarkan analisis statistik tersebut, akan dapat disimpulkan seberapa akurasi sistem dalam menampilkan data penjelajahan.

1. Pengolahan Data Pengujian Tugas Pencarian (*Searching*)

Setelah peserta selesai melakukan tugas pencarian, peneliti menandai masing-masing kiriman dan mengklasifikasikan skim penandaan menjadi 3 (tiga) kategori.

1. Salah, dengan memberi nilai skor 0 (nol). Nilai skor ini diberikan ketika peserta tidak memberikan jawaban yang benar untuk tugas yang diberikan.
2. Sebagian benar, dengan memberi nilai skor 1 (satu). Nilai skor ini diberikan ketika jawaban yang diberikan oleh peserta sebagian benar.
3. Sepenuhnya benar, dengan memberi nilai skor 2 (dua). Nilai skor ini diberikan ketika jawahab yang diberikan olhe peserta sepenuhnya benar.

Berdasarkan ketiga kategori di atas, akan didapatkan hasil klasifikasi skim penandaan terhadap seluruh kiriman peserta yang kemudian akan dimasukkan pada tabel seperti Tabel 3.9berikut.

Tabel 3.9 Data Pengujian Pencarian

|  |  |  |
| --- | --- | --- |
| **Kategori** | **Jumlah** | |
| **Kiriman** | **Persen** |
| Salah |  |  |
| Sebagian benar |  |  |
| Sepenuhnya benar |  |  |
| **Total** |  | 100% |

Selanjutnya, dari hasil skim penandaan seluruh kiriman peserta, peneliti menerapkan analisis statistik sebagai berikut.

1. Rerata (*mean*). Dengan analisis ini, didapatkan rerata ketepatan jawaban peserta pada masing-masing pertanyaan. Rerata ini akan menggambarkan keakuratan sistem dalam menampilkan data hasil pencarian.
2. Nilai tengah (*median*). Dengan analisis ini, didapatkan nilai tengah dari seluruh nilai skor peserta pada masing-masing pertanyaan.
3. Nilai terendah (*minimum*). Dengan analisis ini, didapatkan nilai terendah dari seluruh nilai skor peserta pada masing-masing pertanyaan.
4. Nilai tertinggi (*maximum*). Dengan analisis ini, didapatkan nilai tertinggi dari seluruh nilai skor peserta pada masing-masing pertanyaan.

Berdasarkan analisis statistik tersebut, akan dapat disimpulkan seberapa akurasi sistem dalam menampilkan data pencarian.

1. Pengolahan Data Evaluasi Persepsi Kegunaan Yang Dirasakan

Setelah seluruh peserta menjawab kuesioner yang berisi serangkaian pertanyaan terkait dengan kegunaan sistem yang dirasakan, selanjutnya peneliti melakukan pengolahan data hasil kuesioner. Karena hasil kuesioner telah memiliki penandaan skor secara otomatis, peneliti langsung melanjutkan dengan melakukan analisis hasil kuesioner. Dalam melakukan analisis hasil kuesioner, analisis statistik berikut dilakukan dengan menggunakan perangkat lunak SPSS.

1. Rerata (*mean*). Dengan analisis ini, didapatkan rerata nilai skor yang diberikan peserta pada masing-masing pertanyaan. Rerata ini akan menggambarkan seberapa berguna dan mudah digunakan sistem dalam persepsi peserta.
2. Nilai tengah (*median*). Dengan analisis ini, didapatkan nilai tengah dari seluruh nilai skor peserta pada masing-masing pertanyaan.
3. Nilai terendah (*minimum*). Dengan analisis ini, didapatkan nilai terendah dari seluruh nilai skor peserta pada masing-masing pertanyaan.
4. Nilai tertinggi (*maximum*). Dengan analisis ini, didapatkan nilai tertinggi dari seluruh nilai skor peserta pada masing-masing pertanyaan.

Berdasarkan analisis statistik tersebut, akan dapat disimpulkan seberapa berguna dan mudah digunakan dalam persepsi peserta.

1. Pengolahan Data Evaluasi Persepsi Kemudahan Penggunaan

Setelah seluruh peserta menjawab kuesioner yang berisi serangkaian pertanyaan terkait dengan kemudahan penggunaan sistem, selanjutnya peneliti melakukan pengolahan data hasil kuesioner. Karena hasil kuesioner telah memiliki penandaan skor secara otomatis, peneliti langsung melanjutkan dengan melakukan analisis hasil kuesioner. Dalam melakukan analisis hasil kuesioner, analisis statistik berikut dilakukan dengan menggunakan perangkat lunak SPSS.

1. Rerata (*mean*). Dengan analisis ini, didapatkan rerata nilai skor yang diberikan peserta pada masing-masing pertanyaan. Rerata ini akan menggambarkan seberapa berguna dan mudah digunakan sistem dalam persepsi peserta.
2. Nilai tengah (*median*). Dengan analisis ini, didapatkan nilai tengah dari seluruh nilai skor peserta pada masing-masing pertanyaan.
3. Nilai terendah (*minimum*). Dengan analisis ini, didapatkan nilai terendah dari seluruh nilai skor peserta pada masing-masing pertanyaan.
4. Nilai tertinggi (*maximum*). Dengan analisis ini, didapatkan nilai tertinggi dari seluruh nilai skor peserta pada masing-masing pertanyaan.

Berdasarkan analisis statistik tersebut, akan dapat disimpulkan seberapa berguna dan mudah digunakan dalam persepsi peserta.

**BAB IV**

**HASIL DAN PEMBAHASAN**

1. **Implementasi Ontologi**

Pada bagian implementasi ontologi akan dijabarkan mengenai implementasi dan pembangunan ontologi sesuai dengan tahapan yang telah ditentukan. Implementasi ontologi ini tidak menggunaka keseluruh tahapan yang ada karena pada tahapan-tahapan tertentu komponen tersebut tidak dapat didefinisikan. Berikut merupakan implementasi dari tahapan metode pembangunan ontologi yaitu dengan menggunakan metode Methontology.

1. **Tahap Spesifikasi**

Dalam tahap sesifikasi, dihasilkan deskripsi dari ontologi sepeda motor sebagai berikut.

1. Domain : Sepeda Motor
2. Tujuan : Untuk membangun model ontologi untuk pendukung pengambilan keputusan pemilihan sepeda motor.
3. Tingkat Formalitas : Semi formal
4. Lingkup : Sepeda motor
5. Sumber Pengetahuan : Observasi langsung, leteratur terkait
6. **Tahap Akuisisi Pengetahuan**

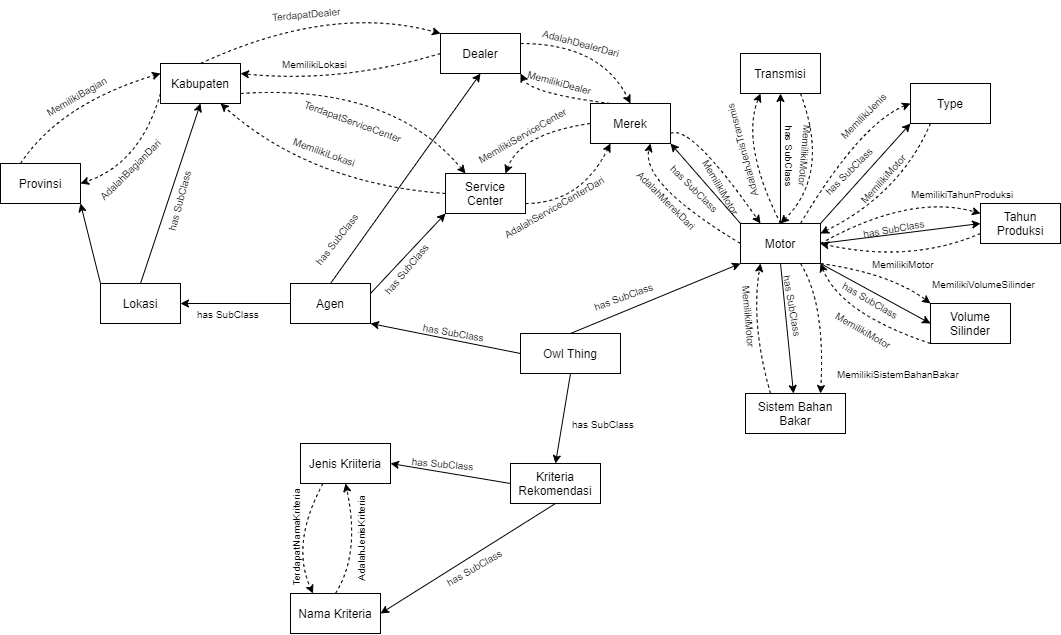
Dalam tahap akuisisi pengetahuan ontologi Sepeda Motor menggunakan teknik sebagai berikut.

1. Berdiskusi bersama *ontology engineer* untuk menyusun draft awal dokumen spesifikasi kebutuhan.
2. Analisis teks informal, untuk mempelajari konsep-konsep utama.
3. Analisis teks formal. Mengidentifikasi struktur yang akan dideteksi (definisi, penegasan, dll.) dan jenis pengetahuan yang dikontribusikan oleh masing-masing (konsep, atribut, nilai, dan hubungan).

Dalam penelitian ini mengunakan data sepeda motor merk Yamaha, Honda, Suzuki, dan Kawasaki. Data tersebut diperoleh dari pengumpulan data langsung di salah satu dealer di daerah Kintamani selama 1 hari, literatur maupun sumber internet yang dapat dipercaya yang merupakan website resmi dealer motor.

1. **Tahap Konseptualisasi**

Tahap konseptualisasi ontologi (Gomez-Perez, 2003) bertujuan untuk mengatur dan mengelola pengetahuan yang diperoleh selama proses akuisisi pengetahuan. Saat model konseptual telah dibangun, metodologi mengusulkan untuk mengubah model konseptual menjadi model formal, yang kemudian diimplementasikan dengan bahasa implementasi ontologi.



Gambar 4.1 Konseptual Ontologi

Gambar 4.2 merupakan cuplikan dari triplet subjek, predikat, dan objek yang ada pada ontologi.



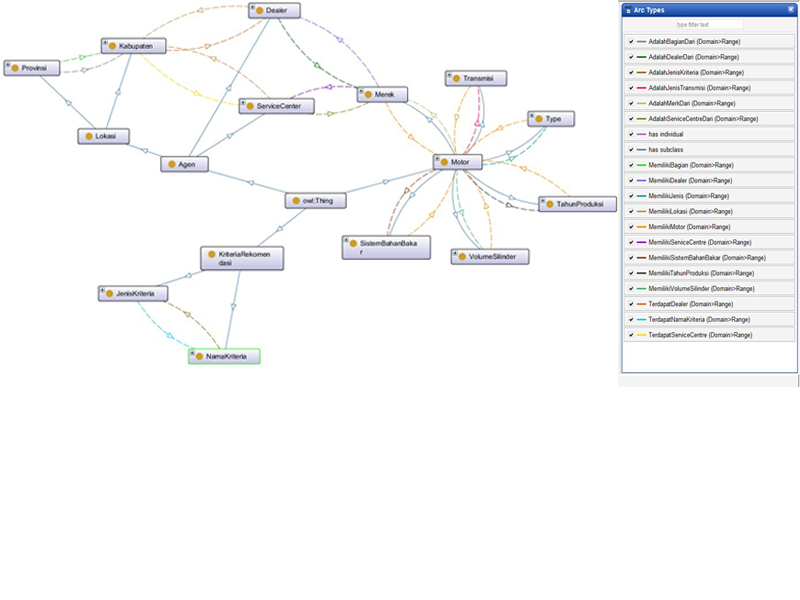
Gambar 4.2 Cuplikan Triplet pada Ontologi Sepeda Motor

1. **Tahap Integrasi**

Dalam tahap ini, peneliti mengintegrasikan model ontologi yang dibuat kedalam kerangka yang telah dikonsepkan bersama *ontology engineer* sebelumnya.

1. **Tahap Implementasi**

Implementasi perancangan konseptual ontologi yang telah dilakukan menggunakan Methontology kemudian diformalisasikan menggunakan perangkat lunak Protégé 4.3. Pada perangkat lunak Protégé 4.3, setial bagian ontologi didefinisikan sesuai dengan hasil dari tiap tahapan tugas pada Methontology, dimana *concept* didefinisikan sebagai *class*, *ad-hoc binary relation* didefinisikan sebagai *object properties*, dan *instances* didefinisikan sebagai *individual*. Pada Gambar 4.3 menunjukan hasil perancangan ontologi sepeda motor yang merupakan ontologi yang dihasilkan dari perancangan sebelumnya.



Gambar 4.3 Diagram Ontograf Sistem Rekomendasi Sepeda Motor

*Object properties* merupakan suatu relasi yang menghubungkan menghubungkan dua *instance* atau individual yang berbeda. Jika sebuah *object* *property* menghubungkan antara individual *a* dan individual *b*, maka *inverse* *property* akan menghubungkan individual *b* dengan individual *a*. Gambar 4.4 menunjukan *object properties* yang ada pada ontologi sepeda motor.

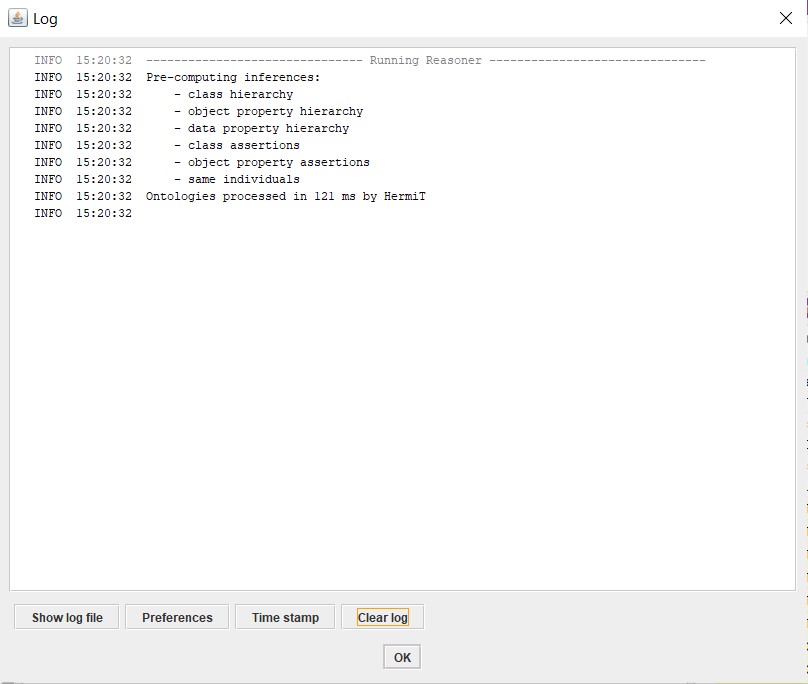


Gambar 4.4 Object Properties pada Ontologi Sepeda Motor

*Individual* dari setiap atribut yang dimiliki masing-masing *class* didaftarkan pada ontologi sepeda motor yang diimplementasikan menggunakan Protégé 4.3.0.

1. **Tahap Evaluasi**

Setelah model ontologi dibangun, maka dilakukan inferensi menggunakan Reasoner untuk mengecek konsistenti ontologi. Saat ontologi sudah dianggap konsiten, maka ontolog dapat diimplementasikan pada suatu sistemyang ingin dibangun. Dari proses *resoning* dilakukan, ontologi sepeda motor telah konsisten, yang dibuktikan dengan tidak munculnya “Resoner Error”. Pada Gambar 4.5 menunjukan hasil *resoning* pada ontologi sepeda motor menggunakan HermiT 1.4.3.456.

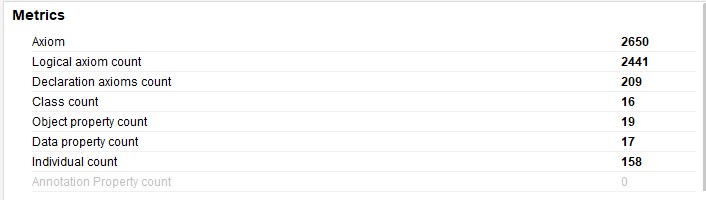


Gambar 4.5 Resoner menggunakan HermiT 1.4.3.456

1. **Tahap Dokumentasi**

Pada tahap ini, akan dilakukan roses dokumentasi ontologi sepeda motor baik dalam kode ontologi, teks bahasa alami yang dilampirkan pada desfinisi formal, maupun makalah yang diterbitkan dalam proses konfrensi dan jurnal yang mengatur pertanyaan-pertanyaan dari ontologi yang sudah dibangun.

Dari ontologi sepeda motor yang telah dibanyun, tersusun *metric* ontologi yang memberikan gambaran secara matematis komponen yang ditunjukan pada Gambar 4.6.



Gambar 4.6 Metric Ontologi

1. **Implementasi Sistem**

Pada bagian implementasi sistem ini akan dijabarkan mengenai implementasi sistem sesuai dengan tahapan yang telah ditentukan. Dalam sistem akan 1 (satu) jenis user saja yaitu *Guest User* yang akan dapat melakukan aktivitas penjelajahan, pencarian dan rekomendasi sepeda motor.

Penjelajahan, merupakan aktivitas yang dapat dilakukan oleh *Guest User* dengan melakukan penelusuran dalam sistem dengan memilih salah satu tautan sehingga akan menuju halaman berikutnya sesuai dengan tautan yang telah dipilih.

Pencarian, merupakan aktivitas yang dapat dilakukan oleh *Guest User* dengan melakukan pencarian dengan memilih *output* dan *input* terlebih dahulu sehingga akan mendapatkan informasi sesuai dengan *output* dan *input* yang telah dipilih.

Rekomendasi, merupakan proses dimana *Guest User* dapat melihat hasil rekomendasi sepeda motor dengan memasukan *input* terlebih dahulu, baik dengan melakukan *input* kriteria maupun dengan memilih sepeda motor yang akan dibandingkan. Berikut ini merupakan implementasi sistem.

1. **Lingkungan Implementasi**

Dalam tahap implementasi sistem, terdapat aplikasi atau perangkat lunak penunjang maupun *library* yang digunakan dalam pembuatan sistem rekomendasi pemilihan sepeda motor, yaitu sebagai berikut.

1. Windows 10 Home 64-bit (*Client*) dan *Linux (Vitual Private Server)*
2. Protégé 4.3
3. XAMPP Control Panel v3.2.2
4. PHP 7.2.3
5. Visual Studio Code 1.53.2
6. Bootstrap 4.0.0
7. Laravel 6.18.43
8. Apache Jena Fuseki 3.17.0
9. *Library* EasyRDF
10. Google Chrome 88.0.4324.182
11. Plesk
12. Microsoft Office Visio 2016
13. Mecrosoft Office Excel 2016
14. Adobe XD
15. **Implementasi Ontologi ke Dalam Sistem**

Tahap implementasi ontologi ke dalam sistem mencakup beberapa proses yaitu proses unggah ontologi ke server fuseki agar dapat digunakan oleh sistem. Kemudian dilakukan proses kondeksi ontologi dengan sistem menggunakan *library* EasyRDF. Pada Tabel 4.1 adalah *source code* untuk mengkoneksikan ontologi dengan sistem.

Tabel 4.1 Source Code Proses Koneksi Ontologi dengan Sistem

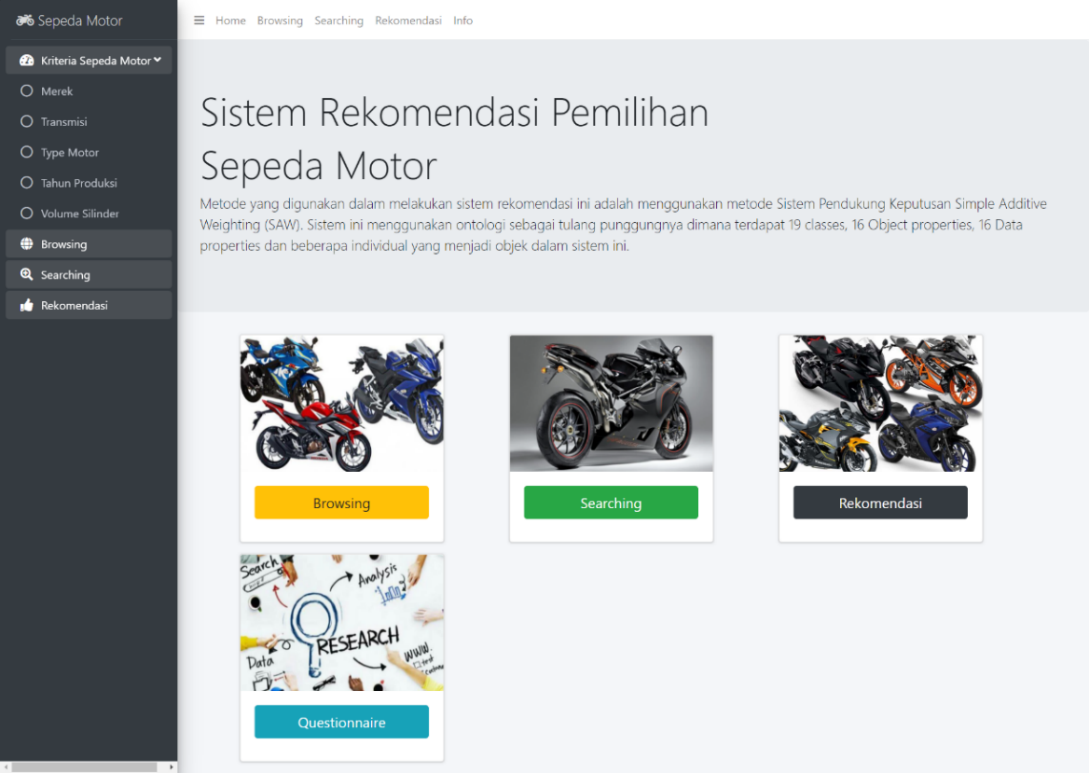
|  |
| --- |
| use EasyRdf\RdfNamespace;  use EasyRdf\Sparql\Client;  RdfNamespace::set('rdf', 'http://www.w3.org/1999/02/22-rdf-syntax-ns#');  RdfNamespace::set('rdfs', 'http://www.w3.org/2000/01/rdf-schema#');  RdfNamespace::set('owl', 'http://www.w3.org/2002/07/owl#');  RdfNamespace::set('motor', 'http://www.semanticweb.org/girikusuma/OntologiSepedaMotor#');  $this->sparql = new Client('https://jena.balidigitalheritage.com/fuseki/motor/query'); |

1. **Implementasi *User Interface* (Antarmuka) Sistem**

*User interface* atau antarmuka sistem yang sebelumnya dirancang pada Bab 3 selanjutnya akan diimplementasikan dengan menggunakan bahasa pemrograman PHP, HTML dengan menggunakan *framework* Laravel 6.18.43 dan CSS dengan menggunakan *framework* Bootstrap 4.0.0. Berikut akan dipaparkan *capture* hasil implementasi rancangan antarmuka yang telah dibuat.

1. **User interface Halaman Utama**

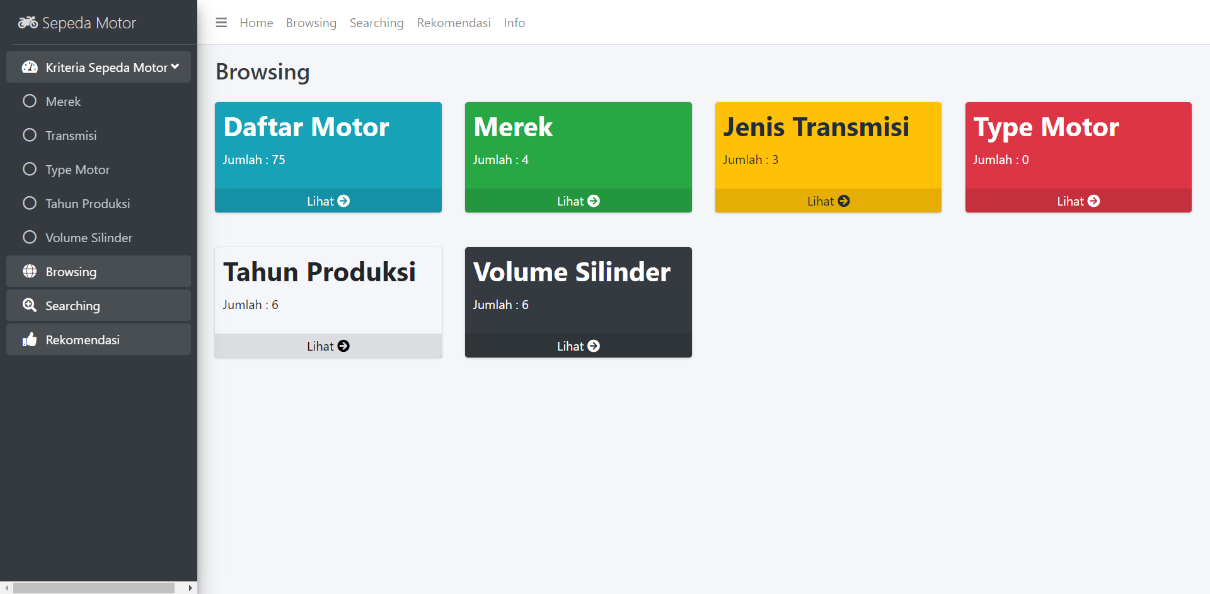
Pada Gambar 4.7adalah implementasi dari halaman utama sistem untuk *guest user*. Pada halaman ini akan terdapat deskripsi singkat mengenai sistem dan daftar tautan utama pada sistem, yaitu *browsing*, *searching*, dan rekomendasi. Pada halaman ini *guest user* dapat memilih tautan yang diinginkan.



Gambar 4.7 User interface Halaman Utama

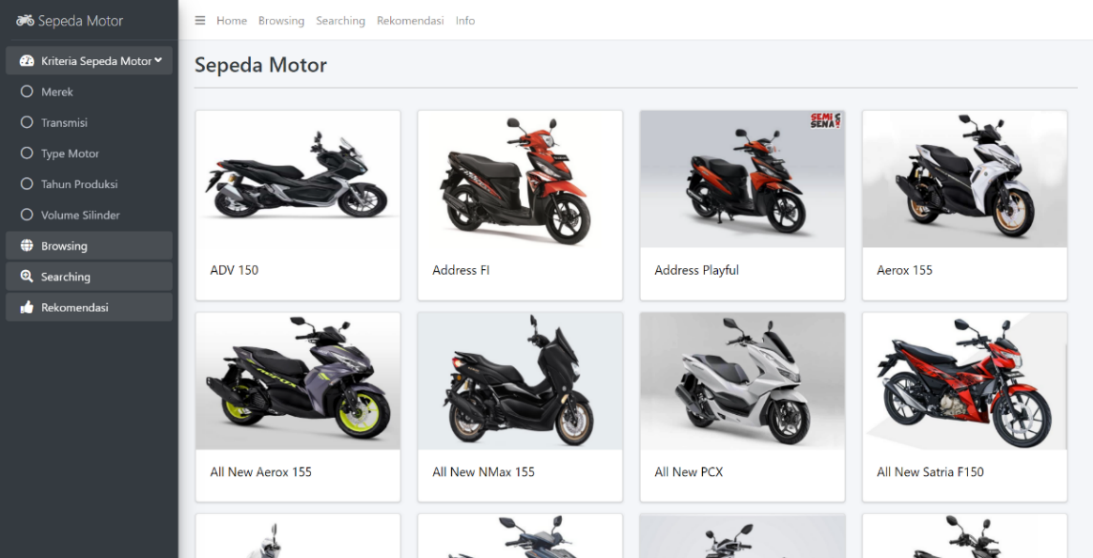
1. **User interface Halaman Penjelajahan**

Pada Gambar 4.8adalah implementasi dari halaman penjelajahan sistem untuk *guest user*. Pada halaman ini akan terdapat beberapa daftar tautan penjelahan pada sistem, yaitu kriteria-kriteria yang dapat dipilih oleh *guest user*.



Gambar 4.8 User interface Halaman Penjelajahan

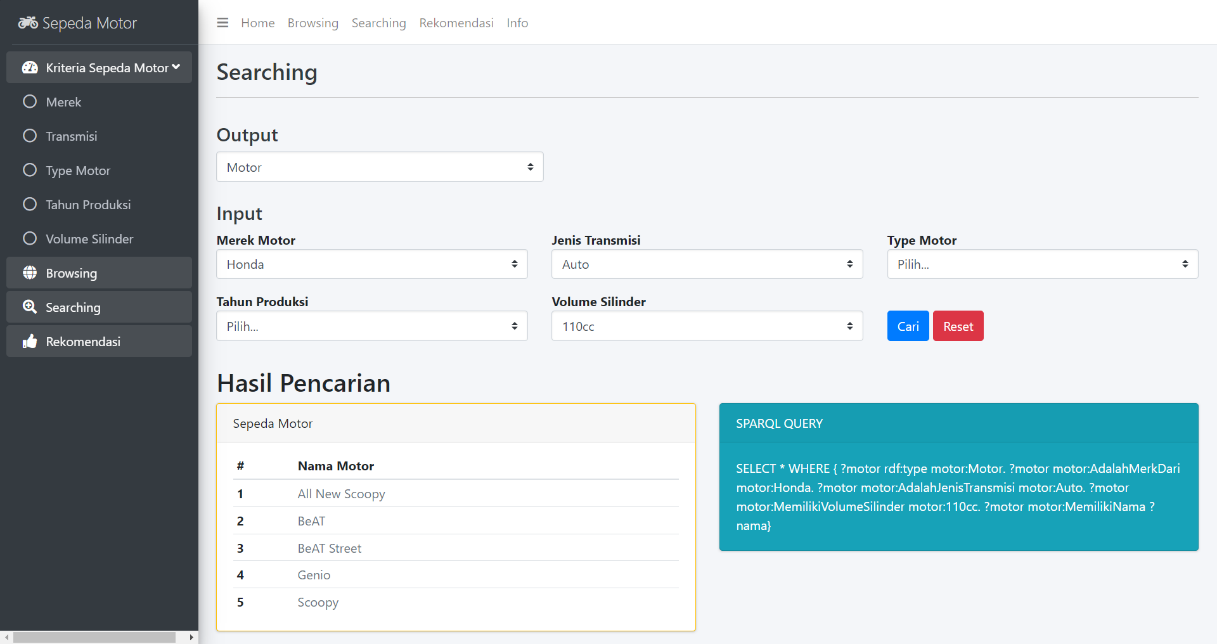
Pada Gambar 4.9adalah implementasi dari halaman penjelajahan sistem untuk *guest user* setelah memilih salah satu tautan pada halaman sebelumnya. Pada halaman ini akan terdapat beberapa list dari kriteria yang telah dipilih *guest user*.



Gambar 4.9 User interface Halaman Penjelajahan Sepeda Motor

1. **User interface Halaman Pencarian**

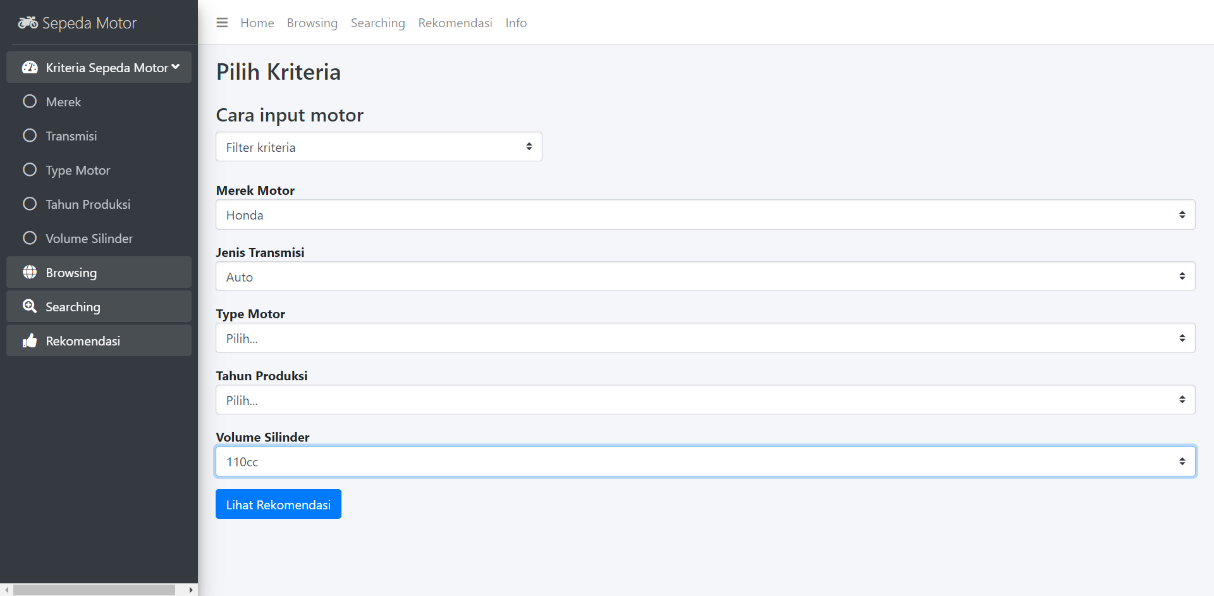
Pada Gambar 4.10adalah implementasi halaman pencarian untuk *guest user*. Pada halaman tersebut terdapat *form* untuk melakukan pencarian suatu *instances* informasi yang terkait dengan sepeda motor berdasarkan *input* yang diinginkan. Pada halaman tersebut, *guest* *user* dapat melakukan pencarian *instances* dengan cara mengisi *form output* dan minimal sebuah *form input* yang diinginkan, lalu mengklik tombol “Cari”. Hasil pencarian akan ditampilkan secara *realtime* beserta *query* SPARQL yang digunakan untuk melakukan pencarian. *Guest* *user* kemudian dapat mengakses tautan output yang diinginkan.



Gambar 4.10 User interface Halaman Pencarian

1. **User interface Halaman Input Rekomendasi**

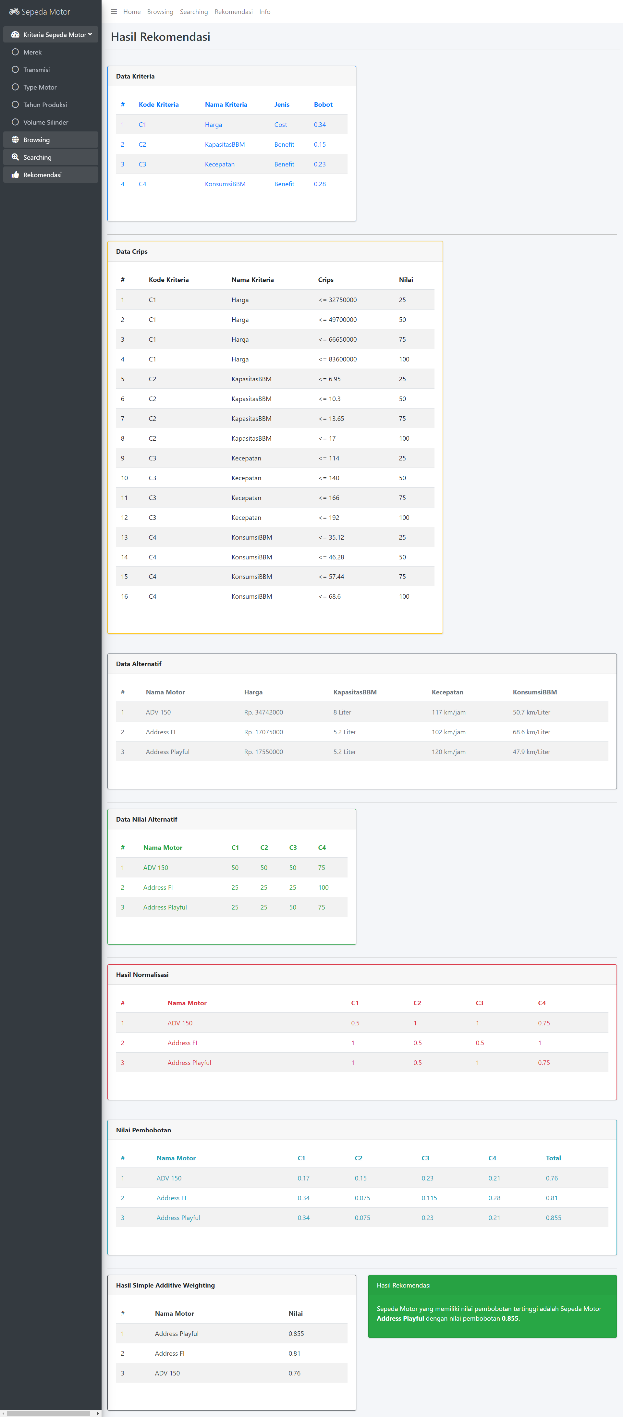
Pada Gambar 4.11adalah implementasi halaman input rekomendasi untuk *guest* *user*. Pada halaman tersebut terdapat *form* untuk melakukan *filter* terhadap sepeda motor yang akan dipilih. Pada halaman tersebut, *guest user* dapat melakukan *filter* kriteria instances dengan cara mengisi minimal sebuah *form* *input* yang diinginkan, lalu mengklik tombol “Lihat Rekomendasi”.



Gambar 4.11 User interface Halaman Input Rekomendasi

1. **User Interface Halaman Hasil Rekomendasi**

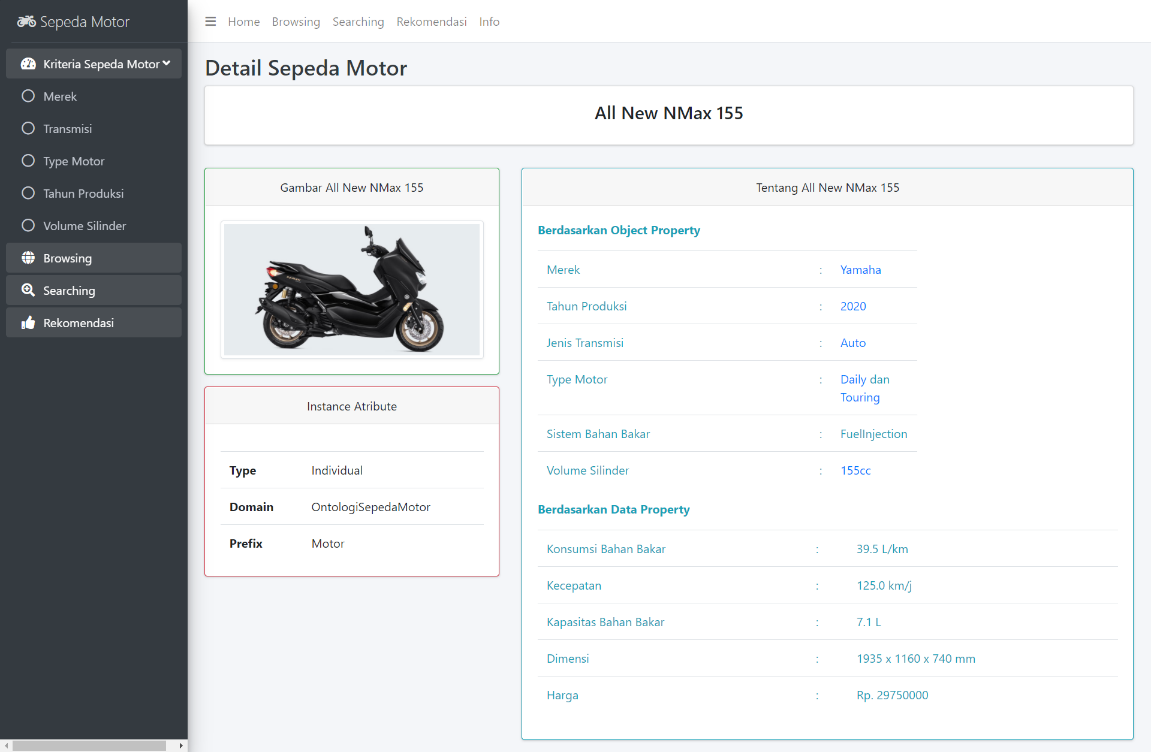
Pada Gambar 4.12adalah implementasi halaman hasil rekomendasi untuk *guest* *user*. Pada halaman ini menampilkan hasil perhitungan pembobotan dari sepeda motor menggunakan metode *Simple Additive Weighting* (SAW).



Gambar 4.12 User interface Halaman Hasil Rekomendasi

1. **User interface Halaman Detail Instance**

Pada Gambar 4.13adalah implementasi halaman detail *instances* untuk *guest* *user*. Pada halaman ini akan menampilkan informasi detail yang ada pada *instances* tertentu.



Gambar 4.13 User interface Halaman Detail Instances

1. **Implementasi Pengujian dan Evaluasi Sistem**

Pada bagian ini akan dijabarkan mengenai implementasi pengujian dan evaluasi sistem sesuai dengan tahapan yang telah ditentukan sebelumnya. Berikut ini merupakan implementasi pengujian dan evaluasi sistem rekomendasi pemilihan sepeda motor.

1. **Pengujian Fungsionalitas**

Pengujian fungsionalitas dilakukan terhadap sistem dengan menggunakan pengujian *Black-Box Testing*. Pengujian *Black-Box Testing* merupakan pengujian yang dilakukan pada *user interface* untuk menguji apakah hasil eksekusi telah sesuai dengan masukan yang diberikan dengan memeriksa fungsional dari perangkat lunak. Pada pengujian fungsionalitas ini akan diuji kemampuan sistem dengan melakukan proses-proses yang didefinisikan pada analisis kebutuhan. Pengujian ini dilakukan oleh peneliti dengan 10 kali iterasi atau perulangan untuk setiap modul penjelajahan, modul pencarian dan modul rekomendasi.

Tabel 4.2 Kode Kebutuhan Sistem

|  |  |  |
| --- | --- | --- |
| **Kode** | **Pengguna** | **Status** |
| F1 | *Guest User* | OK |

Tabel 4.2 menunjukan kode kebutuhan yang merujuk pada hasil analisis kebutuhan yang dipaparkan pada Bab 3. Pada Tabel 4.3, Tabel 4.4, dan Tabel 4.5 dijelaskan rincian dari pengujian *Black-Box Testing* pada hasil *checklist* kebutuhan dan kode skenario pada tabel pengujian merujuk pada *test case* yang dirancang sebelumnya yang digunakan sebagai panduan oleh peneliti dalam melakukan pengujian *Black-Box Testing* sistem rekomendasi sepeda motor ini.

Tabel 4.3 Black-Box Testing Penjelajahan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Kode Kebutuhan : F1 | | | Kode Pengujian : U20 | |
| Kasus:  Penjelajahan *Guest User* | | | | |
| **No.** | **Kode** | **Nama Skenario** | **Hasil Pengujian** | **Kesimpulan** |
| 1. | UC4-1-1 | Menampilkan halaman penjelajahan | Sistem menampilkan halaman penjelajahan | Sesuai |
| 2. | UC4-1-2 | Penjelajahan berhasil dilakukan | * Sistem menampilkan *list hyperlink* * Sistem berhasil melakukan penejelajahan | Sesuai |
| 3. | UC4-1-3 | Hasil penjelajahan berhasil ditampilkan | Sistem menampilkan penjelajahan pada halaman penjelajahan | Sesuai |

Tabel 4.4 Black-Box Testing Pencarian

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Kode Kebutuhan : F1 | | | Kode Pengujian : U21 | |
| Kasus:  Pencarian *Guest User* | | | | |
| **No.** | **Kode** | **Nama Skenario** | **Hasil Pengujian** | **Kesimpulan** |
| 1. | UC4-1-1 | Menampilkan halaman pencarian | Sistem menampilkan halaman pencarian | Sesuai |
| 2. | UC4-1-2 | Pencarian berhasil dilakukan | * Sistem menampilkan *form output* dan *form input* pencarian * Sistem berhasil melakukan *query* pencarian | Sesuai |
| 3. | UC4-1-3 | Hasil pencarian berhasil ditampilkan | Sistem menampilkan hasil pencarian dan *query* pencarian pada halaman yang sama | Sesuai |

Tabel 4.5 Black-Box Testing Rekomendasi

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Kode Kebutuhan : F1 | | | Kode Pengujian : U22 | |
| Kasus:  Rekomendasi *Guest User* | | | | |
| **No.** | **Kode** | **Nama Skenario** | **Hasil Pengujian** | **Kesimpulan** |
| 1. | UC4-1-1 | Menampilkan halaman rekomendasi | Sistem menampilkan halaman rekomendasi | Sesuai |
| 2. | UC4-1-2 | Pencarian dan perhitungan nilai berhasil dilakukan | * Sistem menampilkan *form filter* kriteria rekomendasi * Sistem berhasil melakukan *query filter* dan melakukan perhitungan dengan metode SAW | Sesuai |
| 3. | UC4-1-3 | Hasil rekomendasi berhasil ditampilkan | Sistem menampilkan hasil rekomendasi | Sesuai |

1. **Partisipan dan Pengumpulan Data**

Setelah melakukan pengujian fungsionalitas dengan menggunakan pengujian *Black-Box Testing*, selanjutnya dilakukan pencarian partisipan untuk selanjutnya dilakukan pengumpulan data pengujian akurasi dan evaluasi. Proses pengumpulan data dari partisipan pada penelitian ini dilakukan selama 1 (satu) minggu. Partisiasi dalam penelitian ini bersifat sukarela dan tidak ada satu pun perserta yang terlibat dalam survei akuisisi data peneliti. Peserta yang direkrut dari jaringan mahasiswa.

Setelah peserta setuju untuk mengambil bagian dalam penelitian ini, peneliti mengumpulkan peserta melalui sesi pelatihan singkat menggunakan perangkat *online video conference* Google Meets. Perangkat ini dipilih karena seluruh peserta pengujian telah mengintal aplikasi ini dan cukup sering menggunakan perangkat lunak tersebut. Pada sesi tersebut, peneliti menjelaskan mengenai cara kerja sistem, termasuk cara menggunakan penjelajahan semantik, pencarian semantik dan rekomendasi dan apa yang harus dilakukan oleh peserta selama melakukan pengujian dan evaluasi sistem ini. Peneliti menekankan kepada peserta bahwa pada pencarian semantik pada sistem sangat berbeda dengan layanan pencarian berbasis teks dan kata kunci yang umumnya tersedia dibanyak aplikasi berbasis web. Pada percarian berbasis teks, pengguna mengetik *string* teks apa pun dan sistem akan mencocokannya dengan data yang tersedia. Namun, dalam pencarian semantik pada sistem, pengguna hanya memilih kelas dan *instance* yang tersedia yang merupakan bagian dari ontologi dan penyimpanan data RDF.

Setelah menyimak sesi pelatihan dan mengerti penjelasan yang disampaikan, semua peserta diminta untuk memberikan persetujuan online mereka dan kemudian melakukan berbagai tugas penjelajahan dan pencarian menggunakan fitur dan fasilitas yang tersedia di sistem. Akhirnya, semua peserta diundang untuk menjawab serangkaian pertanyaan kecil mengenai kegunaan yang dirasakan dan kemudahan penggunaan sistem. Studi ini dilakukan selama seminggu untuk memungkinkan peneliti meningkatkan kinerja sistem ketika para peserta melaporkan bug pada sistem.

1. **Implementasi Pengujian Akurasi**

Pada bagian pengujian akurasi akan dijabarkan tugas penjelajahan (*browsing*) dan pencaian (*searching*) yang diberikan kepada peserta untuk selanjutnya diselesaikan dengan menggunakan fitur dan fasilitas yang tersedia dalam sistem serta melakukan pengujian terhadap perhitungan manual untuk mengetahui hasil yang diberikan pada fitur rekomendasi. Pengujiin akurasi yang dilakukan dijabarkan sebagai berikut.

1. Pengujian Akurasi Penjelajahan (*Browsing*)

Pada pengujian akurasi penjelajahan, peserta diminta untuk menjawab 5 (lima) pertanyaan tugas penjelajahan yang telah disediakan. Jawaban didapatkan dari hasil melakukan penjelajahan (eksplorasi sistem dengan mengikuti suatu tautan menarik ke yang lain) pada fitur penjelajahan sistem.

1. Sebutkan 5 sepeda motor yang menggunakan sistem transmisi auto!
2. Sebutkan konsumsi bahan bakar motor Address FI!
3. Sebutkan nama-nama merek!
4. Sebutkan 3 motor yang menggunakan volume silinder 175cc!
5. Sebutkan 5 sepeda motor yang menggunakan jenis transmisi Bebek!
6. Pengujian Akurasi Pencarian (*Searching*)

Pada pengujian akurasi pencarian, peserta diminta untuk menjawab 5 (lima) pertanyaan tugas pencarian yang telah disediakan. Jawaban didapatkan dari hasil melakukan pencarian (meminta sebagian informasi dari *database*) pada fitur penjelajahan sistem. Agar dapat menjawab pertanyaan menggunakan fasilitas pencarian, peserta diminta untuk membangun beberapa elemen dari *query* sebagai *filter* *input* dan membentuk satu *query* sebagai *filter* *output*, lalu diikuti dengan mengklik tombol pencarian yang tersedia.

1. Sebutkan 4 sepeda motor yang termasuk dalam merek Honda yang menggunakan transmisi Manual, merupakan type motor Sport dan diproduksi tahun 2020!
2. Sebutkan 5 sepeda motor yang termasuk dalam merek Suzuki yang menggunakan transmisi Auto dan memiliki volume silinder 110cc!
3. Sebutkan 5 dealer yang merupakan dealer Honda yang berlokasi di Denpasar!
4. Sebutkan 5 service center yang merupakan service center Honda!
5. Sebutkan 4 sepeda motor yang termasuk dalam merek Kawasaki, menggunakan transmisi Manual dan memiliki type Touring!

Dalam implementasi pengujian akurasi penjelajahan dan pencarian, data yang digunakan dalam penelitian ini dikumpulkan menggunakan kuesioner online Google Forms yang sebelumnya telah peneliti kembangkan. Data yang dikumpulkan kemudian diekspor pada *spreadsheets*. Setelah melakukan pengujian dan evaluasi sistem, data yang diperoleh dari pengujian dan evaluasi sistem ini selanjutnya akan diolah dan dianalisis sesuai dengan tahapan yang telah ditentukan.

1. Pengujian terhadap perhitungan manual

Contoh kasus untuk perhitungan oleh sistem sebagai berikut. *Guest user* sudah memilih kriteria *filter* sepeda motor yang akan dilihat hasil rekomendasinya, kriteria *filter* yang dipilih adalah jenis transmisi bebek dan type motor *daily* sehingga didapatkan motor diantara lain Jupiter Z1, New Smash FI, Revo X, Supra X 125 FI, dan Vega Force. Perhitungan manualnya adalah sebagai berikut:

1. Data Kriteria

Data kriteria berisi kode, nama, jenis, dan bobot. Bobot kriteria menentukan seberapa penting kriteria tersebut. Pada Tabel 4.6 menunjukan kriteria yang digunakan untuk perhitungan metode Simple Additive Weighting pada sistem.

Tabel 4.6 Data Kriteria

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Kode** | **Nama** | **Jenis** | **Bobot** |
| 1. | C1 | Harga | *Cost* | 0,34 |
| 2. | C2 | KapasitasBBM | *Benefit* | 0,15 |
| 3. | C3 | Kecepatan | *Benefit* | 0,23 |
| 4. | C4 | KonsumsiBBM | *Beenefit* | 0,28 |

Jenis kriteria terdiri dari *cost* dan *benefit*. *Cost* memiliki arti dimana jika semakin kecil nilai maka akan semakin bagus, sedangkan *benefit* memiliki arti jika semakin besar nilai maka semakin bagus. KonsumsiBBM memiliki jenis *benefit* karena satuan nilai yang dipakai adalah km/L dimana semakin besar nilainya semakin bagus.

1. Data *Crips*

Data *crips* berisi kode kriteria, nama kriteria, keterangan *crips*, dan nilai. *Crips* bersifat optional yaitu sebagai pembatas dari nilai setiap kriteria, setiap *crips* memiliki nilai masing-masing. Jika tidak menggunakan crips maka nilai kriteria yang akan diproses langsung pada metode *Simple Additive Weghting* (SAW) pada sistem. Pada Tabel 4.7 menunjukan data *crips* pada sistem.

Tabel 4.7 Data Crips

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Kode Kriteria** | **Nama Kriteria** | **Crips** | **Nilai** |
| 1. | C1 | Harga | <= 32750000 | 25 |
| 2. | C1 | Harga | <= 49700000 | 50 |
| 3. | C1 | Harga | <= 66650000 | 75 |
| 4. | C1 | Harga | <= 83600000 | 100 |
| 5. | C2 | KapasitasBBM | <= 6,95 | 25 |
| 6. | C2 | KapasitasBBM | <= 10,3 | 50 |
| 7. | C2 | KapasitasBBM | <= 13,65 | 75 |
| 8. | C2 | KapasitasBBM | <= 17 | 100 |
| 9. | C3 | Kecepatan | <= 114 | 25 |
| 10. | C3 | Kecepatan | <= 140 | 50 |
| 11. | C3 | Kecepatan | <= 166 | 75 |
| 12. | C3 | Kecepatan | <= 192 | 100 |
| 13. | C4 | KonsumsiBBM | <= 35,12 | 25 |
| 14. | C4 | KonsumsiBBM | <= 46,28 | 50 |
| 15. | C4 | KonsumsiBBM | <= 57,44 | 75 |
| 16. | C4 | KonsumsiBBM | <= 68,6 | 100 |

Dalam mencari data crips tersebut, peneliti menggunakan rumus sebagai beikut.

1. Mencari nilai selisih pada masing-masing kriteria dengan melakukan pengurangan nilai terbesar pada kriteria di seluruh sepeda motor dikurangi nilai terkecil pada kriteria di seluruh sepeda motor dan kemudian dibagi 4 (jumlah *crips* yang digunakan).
2. Untuk mencari nilai *crips* 1 dengan menjumlahkan nilai terkecil pada kriteria seluruh motor dengan selisih yang didapatkan. Kemudian untuk mencari nilai *crips* berikutnya dengan menjumlahkan nilai *crips* sebelumnya dengan selisih yang didapatkan. Ulangin sebanyak 4 (jumlah *crips* yang digunakan).
3. Ulangin proses (a) dan (b) sebanyak jumlah kriteria yang digunakan dalam melakukan proses rekomendasi.
4. Data Alternatif

Data alternatif berisi nama, dan nilai dari setiap alternatif berdasarkan data kriteria. Tabel 4.8 menjabarkan data alternatif yang terpilih sesuai dengan kriteria *filter* yang diinputkan oleh *guest user*.

Tabel 4.8 Data Alternatif

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Nama Motor** | **Harga** | **KapasitasBBM** | **Kecepatan** | **KonsumsiBBM** |
| 1. | Jupiter Z1 | Rp. 18110000 | 4.1 Liter | 110 km/jam | 53,2 km/Liter |
| 2. | New Smash FI | Rp. 16100000 | 3.7 Liter | 100,9 km/jam | 42,8 km/Liter |
| 3. | Revo X | Rp. 16605000 | 4 Liter | 110 km/jam | 59,8 km/Liter |
| 4. | Supra X 125 FI | Rp. 18171000 | 4 Liter | 118 km/jam | 57,2 km/Liter |
| 5. | Vega Force | Rp. 16665000 | 4 Liter | 120 km/jam | 55,6 km/Liter |

1. Tahap Analisa

Pada tahap ini dimana mengubah nilai pada alternatif sesuai bobot pada data *crips*, sehingga diperoleh data seperti tabel berikut.

Tabel 4.9 Data Penyesuaian Nilai alternatif dengan Data Crips

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Nama Motor** | **C1** | **C2** | **C3** | **C4** |
| 1. | Jupiter Z1 | 25 | 25 | 25 | 75 |
| 2. | New Smash FI | 25 | 25 | 25 | 50 |
| 3. | Revo X | 25 | 25 | 25 | 100 |
| 4. | Supra X 125 FI | 25 | 25 | 50 | 75 |
| 5. | Vega Force | 25 | 25 | 50 | 75 |

1. Tahap Normalisasi

Pada tahap ini untuk melakukan normalisasi pada tanel tahap analisa dengan menggunakan rumus sebagai berikut.

* Kriteria Harga (C1)

Harga merupakan *cost*, maka rumus yang digunakan adalah,

Sehingga,

* Kriteria KapasitasBBM (C2)

KapasitasBBM merupakan *benefit*, maka rumus yang digunakan adalah,

Sehingga,

* Kriteria Kecepatan (C3)

Kecepatan merupakan *benefit*, maka rumus yang digunakan adalah,

Sehingga,

* Kriteria KonsumsiBBM (C4)

KonsumsiBBM merupakan *benefit*, maka rumus yang digunakan adalah,

Sehingga,

Sehingga didapatkan tabel hasil normalisasi sebagai berikut.

Tabel 4.10 Hasil Normalisasi

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Nama Motor** | **C1** | **C2** | **C3** | **C4** |
| 1. | Jupiter Z1 | 1 | 1 | 0,5 | 0,75 |
| 2. | New Smash FI | 1 | 1 | 0,5 | 0,5 |
| 3. | Revo X | 1 | 1 | 0,5 | 1 |
| 4. | Supra X 125 FI | 1 | 1 | 1 | 0,75 |
| 5. | Vega Force | 1 | 1 | 1 | 0,75 |

1. Tahap Pembobotan

Pada tahap pembobotan, akan dilakukan dengan mengalikan bobot kriteria dengan setiap baris matriks nilai normalisasi.

0,815

0,745

0,885

0,93

0,93

Sehingga didapatkan tabel hasil pembobotan sebagai berikut.

Tabel 4.11 Hasil Pembobotan

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Nama Motor** | **C1** | **C2** | **C3** | **C4** | **Nilai** |
| 1. | Jupiter Z1 | 0,34 | 0,15 | 0,115 | 0,21 | 0,815 |
| 2. | New Smash FI | 0,34 | 0,15 | 0,115 | 0,14 | 0,745 |
| 3. | Revo X | 0,34 | 0,15 | 0,23 | 0,21 | 0,885 |
| 4. | Supra X 125 FI | 0,34 | 0,15 | 0,23 | 0,21 | 0,93 |
| 5. | Vega Force | 1 | 1 | 1 | 0,75 | 0,93 |

1. Tahap Perankingan

Pada tahap perankingan akan dilakukan pengurutan alternatif berdasarkan nilai pembobotan tertinggi, sehingga didapatkan hasil sebagai berikut.

Tabel 12 Hasil Perankingan

|  |  |  |
| --- | --- | --- |
| **No.** | **Nama Motor** | **Nilai** |
| 1. | Supra X 125 FI | 0,93 |
| 2. | Vega Force | 0,93 |
| 3. | Revo X | 0,885 |
| 4. | Jupiter Z1 | 0,845 |
| 5. | New Smash FI | 0,745 |

Hasil perthitungan sistem cocok dengan hasil perhitungan manual, sehingga sistem proses rekomendasi berjalan dengan baik sesuai dengan rancangannya.

1. **Hasil Pengolahan dan Analisis Data**

Dalam pengujian akurasi penjelajahan dan pencarian dalam sistem ini, terdapat sebanyak 30 peserta yang menanggapi undangan peneliti. Namun, peneliti hanya dapat menggunakan 25 tanggapan yang dapat digunakan karena beberapa dari peserta menyelesaikan kuesioner tanpa melakukan tugas ataupun tidak menanggapi semua pertanyaan dari tugas penjelajahan dan tugas pencarian. Karena data yang tidak lengkap tersebut maka penulis tidak dapat dimasukan kedalam analisis.

Diantara 25 peserta pengujian dalam penelitian ini, dapat dirangkum karakteristik responden seperti pada Tabel 4.13, dimana 32% berumur 20 tahun; 76% adalah laki-laki; 88% beragama Hindu; dan 48% tinggal di wilayah perkotaan.

Tabel 4.13 Karakteristik Peserta Pengujian

|  |  |  |  |
| --- | --- | --- | --- |
| Karakteristik | | Jumlah (orang) | Presentase |
|  | **Umur** |
| 1 | 20 | 8 | 32% |
| 2 | 21 | 12 | 48% |
| 3 | 22 | 4 | 16% |
| 4 | 23 | 1 | 4% |
|  | **Jenis kelamin** |  |  |
| 1 | Laki-Laki | 19 | 76% |
| 2 | Perempuan | 6 | 24% |
|  | **Agama** |  |  |
| 1 | Hindu | 22 | 88% |
| 2 | Islam | 2 | 8% |
| 3 | Budha | 0 | 0% |
| 4 | Kristen | 1 | 4% |
| 5 | Katolik | 0 | 0% |
| 6 | Konghucu | 0 | 0% |
|  | **Wilayah tempat tinggal** |  |  |
| 1 | Perkotaan | 12 | 48% |
| 2 | Pedesaan | 13 | 52% |

Selanjutnya akan dijabarkan hasil pengolahan dan analisis data pengujian akurasi dan evaluasi sistem rekomendasi pemilihan type sepeda motor sesuai dengan tahapan yang telah ditentukan. Berikut ini penjelasan dari proses pengolahan dan analisis data tersebut.

1. **Pengujian Akurasi**

Selanjutnya setelah melakuka tahap pembangunan sistem, yang dimana telah dilakukan pengujian dan evaluasi sistem, maka data yang diperoleh dari pengujian dan evaluasi sistem rekomendasi sepeda motor ini akan diolah dan dianalisis melalui beberaa proses. Terdapat 4 (empat) pengolahan data, yaitu sebagai berikut.

1. Pengolahan Data Pengujian Tugas Penjelajahan (*Browsing*)

Setelah peserta selesai melakukan tugas penjelajahan, peneliti menandai masing-masing kiriman dan mengklasifikasikan skim penandaan menjadi 3 (tiga) kategori.

1. Salah, dengan memberi nilai skor 0 (nol). Nilai skor ini diberikan ketika peserta tidak memberikan jawaban yang benar untuk tugas yang diberikan.
2. Sebagian benar, dengan memberi nilai skor 1 (satu). Nilai skor ini diberikan ketika jawaban yang diberikan oleh peserta sebagian benar.
3. Sepenuhnya benar, dengan memberi nilai skor 2 (dua). Nilai skor ini diberikan ketika jawahab yang diberikan olhe peserta sepenuhnya benar.

Berdasarkan ketiga kategori di atas, akan didapatkan hasil klasifikasi skim penandaan terhadap seluruh kiriman peserta yang kemudian akan dimasukkan pada tabel seperti Tabel 4.14 berikut.

Tabel 4.14 Hasil Klasifikasi Skim Penandaan Tugas Penjelajahan

|  |  |  |
| --- | --- | --- |
| **Kategori** | **Jumlah** | |
| **Kiriman** | **Persen** |
| Salah | 11 | 9% |
| Sebagian benar | 9 | 7% |
| Sepenuhnya benar | 105 | 84% |
| **Total** | 125 | 100% |

Berdasarkan hasil klasifikasi skim penandaan tugas penjelajahan tersebut, dapat disimpulkan bahwa dari seluruh pertanyaan, sebagian besar peserta memberikan jawaban yang sepenuhnya benar.

Tabel 4.15 menunjukan hasil dari analisis statistik data pengujuan akurasi penjelajahan sistem yang tlah dilakukan dengan menggunakan perangkat lunak Microsoft Excel.

Tabel 4.15 Hasil Analisis Statistik Penujian Akurasi Penjelajahan

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Statistik** | | | | | | **Keseluruhan** |
|  | **P1** | **P2** | **P3** | **P4** | **P5** |
| ***Mean*** | 1,84 | 1,44 | 1,92 | 1,84 | 1,72 | 1,752 |
| ***Median*** | 2 | 0 | 2 | 2 | 2 | 2 |
| ***Minimum*** | 1 | 0 | 1 | 0 | 0 | 0 |
| ***Maximum*** | 2 | 2 | 2 | 2 | 2 | 2 |

1. Rerata (*mean*). Dengan analisis ini, didapatkan rerata skor yang dimiliki peserta pengujian pada keseluruhan pertanyaan adalah 1,752 (yang dibulatkan sehingga menjadi 2) yang dikonversikan menjadi “Sepenuhnya benar”. Rerata ini menggambarkan skala jawaban yang diberikan peserta dalam menyelesaikan tugas penjelajahan adalah sepenuhnya benar.
2. Nilai tengah (*median*). Dengan analisis ini, didapatkan nilai tengah dari seluruh nilai skor peserta pengujian pada keseluruhan adalah 2, dimana jika dikonversikan akan menjadi “Sepenuhnya benar”. Nilai tengah ini menggambarkan skala nilai paling tengah dari jawaban yang diberikan peserta dalam menyelesaikan tugas penjelajahan adalah sepenuhnya benar.
3. Nilai terendah (*minimum*). Dengan analisis ini, didapatkan nilai terendah dari seluruh nilai skor peserta pada keseluruhan adalah 0, dimana jika dikonversikan akan menjadi “Salah”. Nilai terendah ini menggambarkan skalan dari nilai paling rendah yang diberikan peserta dalam menyelesaikan tugas penjelajahan adalah salah.
4. Nilai tertinggi (*maximum*). Dengan analisis ini, didapatkan nilai tertinggi dari seluruh nilai skor peserta pada keseluruhan adalah 2, dimana jika dikonversikan akan menjadi “Sepenuhnya benar”. Nilai tertinggi ini menggambarkan skala nilai yang paling tinggi yang diberikan peserta dalam menyelesaikan tugas penjelajahan adalah sepenuhnya benar.

Berdasarkan analisis statistik tersebut, dapat dilihat rerata yang dimiliki peserta pengujian dari masing-masing pertanyaan pada .



Berdasarakan **gambar,** didapatkan informasi sebagai berikut.

1. Rerata nilai tertinggi terdapat pada pertanyaan ketiga (P3) dengan nilai 1,92. Hal ini menandakan bahwa pertanyaan ketiga adalah pertanyaan yang dapat dijawab peserta dengan tingkat kebenaran tertinggi.
2. Rerata nilai terendah terdapat pada pertanyaan kedua (P2) dengan nilai 1,44. Hal ini menandakan bahwa pertanyaan kedua adalah pertanyaan yang dapat dijawab oleh peserta dengan tingkat kebenaran terendah.

Berdasarkan hasil analisa statistik yang telah dijabarkan sebelumnya, maka didapatkan sebuah kesimpulan bahwa rerata peserta dapat menyelesaikan tugas penjelajahan pada sistem dengan jawaban yang sepenuhnya benar.

1. Pengolahan Data Pengujian Tugas Pencarian (*Searching*)

Setelah peserta selesai melakukan tugas pencarian, peneliti menandai masing-masing kiriman dan mengklasifikasikan skim penandaan menjadi 3 (tiga) kategori.

1. Salah, dengan memberi nilai skor 0 (nol). Nilai skor ini diberikan ketika peserta tidak memberikan jawaban yang benar untuk tugas yang diberikan.
2. Sebagian benar, dengan memberi nilai skor 1 (satu). Nilai skor ini diberikan ketika jawaban yang diberikan oleh peserta sebagian benar.
3. Sepenuhnya benar, dengan memberi nilai skor 2 (dua). Nilai skor ini diberikan ketika jawahab yang diberikan olhe peserta sepenuhnya benar.

Berdasarkan ketiga kategori di atas, akan didapatkan hasil klasifikasi skim penandaan terhadap seluruh kiriman peserta yang kemudian akan dimasukkan pada tabel seperti berikut.

Tabel 4.16 Hasil Klasifikasi Skim Penandaan Tugas Pencarian

|  |  |  |
| --- | --- | --- |
| **Kategori** | **Jumlah** | |
| **Kiriman** | **Persen** |
| Salah | 8 | 6% |
| Sebagian benar | 17 | 14% |
| Sepenuhnya benar | 100 | 80% |
| **Total** | 125 | 100% |

Berdasarkan hasil klasifikasi skim penandaan tugas penjelajahan tersebut, dapat disimpulkan bahwa dari seluruh pertanyaan, sebagian besar peserta memberikan jawaban yang sepenuhnya benar.

Tabel 4.15 menunjukan hasil dari analisis statistik data pengujuan akurasi penjelajahan sistem yang tlah dilakukan dengan menggunakan perangkat lunak Microsoft Excel.

Tabel 4.17 Hasil Analisis Statistik Penujian Akurasi Penjelajahan

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Statistik** | | | | | | **Keseluruhan** |
|  | **P1** | **P2** | **P3** | **P4** | **P5** |
| ***Mean*** | 1,6 | 1,58 | 1,72 | 1,88 | 1,8 | 1,716 |
| ***Median*** | 2 | 0 | 1 | 2 | 2 | 1 |
| ***Minimum*** | 0 | 0 | 0 | 0 | 0 | 0 |
| ***Maximum*** | 2 | 2 | 2 | 2 | 2 | 2 |

1. Rerata (*mean*). Dengan analisis ini, didapatkan rerata skor yang dimiliki peserta pengujian pada keseluruhan pertanyaan adalah 1,716 (yang dibulatkan sehingga menjadi 2) yang dikonversikan menjadi “Sepenuhnya benar”. Rerata ini menggambarkan skala jawaban yang diberikan peserta dalam menyelesaikan tugas penjelajahan adalah sepenuhnya benar.
2. Nilai tengah (*median*). Dengan analisis ini, didapatkan nilai tengah dari seluruh nilai skor peserta pengujian pada keseluruhan adalah 1, dimana jika dikonversikan akan menjadi “Sebagian benar”. Nilai tengah ini menggambarkan skala nilai paling tengah dari jawaban yang diberikan peserta dalam menyelesaikan tugas penjelajahan adalah sebagian benar.
3. Nilai terendah (*minimum*). Dengan analisis ini, didapatkan nilai terendah dari seluruh nilai skor peserta pada keseluruhan adalah 0, dimana jika dikonversikan akan menjadi “Salah”. Nilai terendah ini menggambarkan skalan dari nilai paling rendah yang diberikan peserta dalam menyelesaikan tugas penjelajahan adalah salah.
4. Nilai tertinggi (*maximum*). Dengan analisis ini, didapatkan nilai tertinggi dari seluruh nilai skor peserta pada keseluruhan adalah 2, dimana jika dikonversikan akan menjadi “Sepenuhnya benar”. Nilai tertinggi ini menggambarkan skala nilai yang paling tinggi yang diberikan peserta dalam menyelesaikan tugas penjelajahan adalah sepenuhnya benar.

Berdasarkan analisis statistik tersebut, dapat dilihat rerata yang dimiliki peserta pengujian dari masing-masing pertanyaan pada .



Berdasarakan **gambar,** didapatkan informasi sebagai berikut.

1. Rerata nilai tertinggi terdapat pada pertanyaan keempat (P4) dengan nilai 1,88. Hal ini menandakan bahwa pertanyaan keempat adalah pertanyaan yang dapat dijawab peserta dengan tingkat kebenaran tertinggi.
2. Rerata nilai terendah terdapat pada pertanyaan kedua (P2) dengan nilai 1,58. Hal ini menandakan bahwa pertanyaan kedua adalah pertanyaan yang dapat dijawab oleh peserta dengan tingkat kebenaran terendah.

Berdasarkan hasil analisa statistik yang telah dijabarkan sebelumnya, maka didapatkan sebuah kesimpulan bahwa rerata peserta dapat menyelesaikan tugas penjelajahan pada sistem dengan jawaban yang sepenuhnya benar.

1. Pengolahan Data Evaluasi Persepsi Kegunaan Yang Dirasakan

Pada Tabel 4.18 menunjukan hasil dari analisi statistik data evaluasi persepsi kegunaan yang dirasakan yang telah dilakukan dengan menggunakan perangkat lunak Microsoft Excel.

Tabel 4.18 Hasil Analisis Statistik Evaluasi Persepsi Kegunaan Yang Dirasakan

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Statistik** | | | | | | | **Keseluruhan** |
|  | **P1** | **P2** | **P3** | **P4** | **P5** | **P6** |
| ***Mean*** | 6,76 | 6,56 | 6,48 | 6,6 | 6,64 | 6,44 | 6,58 |
| ***Median*** | 7 | 7 | 6 | 7 | 7 | 6 | 7 |
| ***Minimum*** | 6 | 6 | 5 | 5 | 6 | 5 | 5 |
| ***Maximum*** | 7 | 7 | 7 | 7 | 7 | 7 | 7 |

Berdasarkan Tabel 4.18, maka didapatkan informasi sebagai berikut.

1. Rerata (*mean*). Dengan analisis ini, didapatkan rerata nilai skor yang diberikan peserta pada keseluruhan pertanyaan adalah 6,58 (jika dibulatkan menjadi 7) yang jika dikonversikan kedalam skala Linkert menjadi “Sangat Setuju”. Rerata ini akan menggambarkan skala rerata dari evaluasi persepsi kegunaan yang dirasakan adalah sangat setuju.
2. Nilai tengah (*median*). Dengan analisis ini, didapatkan nilai tengah dari seluruh nilai skor peserta pada keseluruhan pertanyaan adalah 7, jika dikonversikan ke dalam skala Linkert menjadi “Sangat Setuju”. Nilai tengah ini menggambarkan skala nilai paling tengah yang diberikan oleh peserta pengujian dari evaluasi persepsi kegunaan yang dirasakan adalah sangat setuju.
3. Nilai terendah (*minimum*). Dengan analisis ini, didapatkan nilai terendah dari seluruh nilai skor peserta pada keseluruhan pertanyaan adalah 5, jika dikonversikan ke dalam skala Linkert menjadi “Agak Setuju”. Nilai terendah ini menggambarkan skala terendah yang diberikan oleh peserta pengujian dari evaluasi persepsi kegunaan yang dirasakan adalah agak setuju.
4. Nilai tertinggi (*maximum*). Dengan analisis ini, didapatkan nilai tertinggi dari seluruh nilai skor peserta pada keseluruhan pertanyaan adalah 7, jika dikonversikan ke dalam skala Linkert menjadi “Sangat Setuju”. Nilai tertinggi menggambarkan skala tertinggi yang diberikan oleh peserta pengujian dari evaluasi persepsi kegunaan yang dirasakan adalah sangat setuju.

Gambar x merupakan grafik batang yang menunjukan rerata nilai yang diberikan peserta pengujian dari masing-masing pertanyaan.



Berdasarkan gambar x, maka didapatkan informasi yaitu sebagai berikut.

1. Rerata skor tertinggi terdapat pada pertanyaan pertama (P1) dengan skor 6,76. Hal ini menandakan bahwa persepsi kesetujuan peserta pengujian bahwa “sistem yang dibangun memungkinkan untuk menyelesaikan tugas lebih cepat” adalah yang tertinggi.
2. Rerata skor terendah terdapat pada pertanyaan keenam (P6) dengan skor 6,44. Hal ini menandakan bahwa persepsi kesetujuan peserta pengujian bahwa “sistem yang dibangun berguna dalam pekerjaan” adalah yang terendah.

Berdasarkan analisis statistik yang telah dijabarkan, dapat disimpulkan bahwa rata-rata peserta pengujian sangat setuju bahwa sistem yang dibangun adalah sistem yang berguna atau memiliki kebergunaan.

1. Pengolahan Data Evaluasi Persepsi Kemudahan Penggunaan

Pada Tabel 4.19 menunjukan hasil dari analisi statistik data evaluasi persepsi kemudahan penggunaan yang telah dilakukan dengan menggunakan perangkat lunak Microsoft Excel.

Tabel 4.19 Hasil Analisis Statistik Evaluasi Persepsi Kemudahan Penggunaan

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Statistik** | | | | | | | **Keseluruhan** |
|  | **P1** | **P2** | **P3** | **P4** | **P5** | **P6** |
| ***Mean*** | 6,68 | 6,4 | 6,6 | 6,4 | 6,68 | 6,56 | 6,553 |
| ***Median*** | 7 | 6 | 7 | 6 | 7 | 6 | 6,5 |
| ***Minimum*** | 6 | 5 | 5 | 5 | 5 | 5 | 5 |
| ***Maximum*** | 7 | 7 | 7 | 7 | 7 | 7 | 7 |

Berdasarkan Tabel 4.19, maka didapatkan informasi sebagai berikut.

1. Rerata (*mean*). Dengan analisis ini, didapatkan rerata nilai skor yang diberikan peserta pada keseluruhan pertanyaan adalah 6,553 (jika dibulatkan menjadi 7) yang jika dikonversikan kedalam skala Linkert menjadi “Sangat Setuju”. Rerata ini akan menggambarkan skala rerata dari evaluasi persepsi kemudahan penggunaan adalah sangat setuju.
2. Nilai tengah (*median*). Dengan analisis ini, didapatkan nilai tengah dari seluruh nilai skor peserta pada keseluruhan pertanyaan adalah 6,5 (jika dibulatkan menjadi 7), jika dikonversikan ke dalam skala Linkert menjadi “Sangat Setuju”. Nilai tengah ini menggambarkan skala nilai paling tengah yang diberikan oleh peserta pengujian dari evaluasi persepsi kemudahan penggunaan adalah sangat setuju.
3. Nilai terendah (*minimum*). Dengan analisis ini, didapatkan nilai terendah dari seluruh nilai skor peserta pada keseluruhan pertanyaan adalah 5, jika dikonversikan ke dalam skala Linkert menjadi “Agak Setuju”. Nilai terendah ini menggambarkan skala terendah yang diberikan oleh peserta pengujian dari evaluasi persepsi kemudahan penggunaan adalah agak setuju.
4. Nilai tertinggi (*maximum*). Dengan analisis ini, didapatkan nilai tertinggi dari seluruh nilai skor peserta pada keseluruhan pertanyaan adalah 7, jika dikonversikan ke dalam skala Linkert menjadi “Sangat Setuju”. Nilai tertinggi menggambarkan skala tertinggi yang diberikan oleh peserta pengujian dari evaluasi persepsi kemudahan penggunaan adalah sangat setuju.

Gambar x merupakan grafik batang yang menunjukan rerata nilai yang diberikan peserta pengujian dari masing-masing pertanyaan.



Berdasarkan gambar x, maka didapatkan informasi yaitu sebagai berikut.

1. Rerata skor tertinggi terdapat pada pertanyaan pertama (P1) dan pertanyaan kelima (P5) dengan skor 6,68. Hal ini menandakan bahwa persepsi kesetujuan peserta pengujian bahwa “sistem yang dibangun mudah untuk dipelajari cara menggunakannya” dan “sistem yang dibangun memudahkan untuk berbuat terampul dalam menggunakan sistem” adalah yang tertinggi.
2. Rerata skor terendah terdapat pada pertanyaan kedua (P2) dengan skor 6,4. Hal ini menandakan bahwa persepsi kesetujuan peserta pengujian bahwa “sistem yang dibangun mudah digunakan untuk melakukan apa yang diinginkan” adalah yang terendah.

Berdasarkan analisis statistik yang telah dijabarkan, dapat disimpulkan bahwa rata-rata peserta pengujian sangat setuju bahwa sistem yang dibangun adalah sistem yang mudah digunakan.

**BAB V**

**KESIMPULAN**

1. **Kesimpulan**

Dari hasil perancangan dan pembangunan Sistem Rekomendasi Pemilihan Sepeda Motor ini dapat ditarik beberapa kesimpulan, yaitu sebagai berikut.

1. Dalam penelitian ini telah diimplementasikan metode Methontology dalam mengembangkan ontologi sepeda motor, yang terdiri dari beberapa tahapan, yaitu tahap spesifikasi, tahap akuisisi pengetahuan, tahap konseptualisasi, tahap integrasi, tahap implementasi, tahap evaluasi, dan tahap dokumentasi.
2. Dalam penelitian ini telah diimplementasikan metode Prototyping untuk merancang bangun sistem rekomendasi sepeda motor berbasis web beserta implementasi ontologi sepeda motor ke dalam sistem, yang terdiri dari beberapa tahapan, yaitu pengumpulan kebutuhan sistem, pembangunan *prototype*, evaluasi *prototype*, pembangunan sistem, pengujuan sistem, dan evaluasi sistem.
3. Dalam penelitian ini telah diimplementasikan metode *Simple Additive Weighting* (SAW) untuk memberikan pembobotan pada masing-masing sepeda motor dalam fitur rekomendasi. Dimana hasil perhitungan sistem cocok dengan hasil perhitungan manual, sehingga sistem mampu memberikan rekomendasi pemilihan sepeda motor sesuai dengan kriteria yang dipilih user.
4. Dari pengujian yang dilakukan terhadap sistem dengan melibatkan peserta pengujian didapatkan hasil pengujian berupa nilai rata-rata akurasi dari fitur penjelajahan dan pencarian yaitu masing-masing sebesar 1,752 dan 1,716. Berdasarkan hasil tersebut, maka dapat disimpulkan bahwa hasil dari penjelajahan dan pencarian sistem telah dapat diangap relatif akurat, dengan rata-rata peserta dapat menjawab tugas penjelajahan dan tugas pencarian sepenuhnya benar. Sedangkan berdasarkan hasil evaluasi yang diberikan peserta terhadap sistem mendapatkn hasil evaluasi berupa nilai rata-rata persepsi kegunaan yang dirasakan dan persepsi kemudahan penggunaan masing-masing sebesar 6,58 dan 6,53. Berdasarkan hasil tersebut, maka daat disimpulkan bahwa hasil rata-rata peserta sangat setuju bahwa sistem yang dibangun adalah sistem yang memiliki kegunaan dan mudah digunakan.
5. **Saran**

Untuk pengembangan sistem lebih lanjut dan menjadi lebih baik dari penelitian ini, adapun saran yang dapat digunakan adalah sebagai berikut.

1. Menambahkan lebih banyak kriteria lagi supaya lebih banyak pilihan bagi *user* untuk memilih sepeda motor sesuai dengan kriteria-kriteria yang diinginkan.
2. Menambahkan fitur *simple* searching, yaitu fitur yang memungkinkan *user* untuk melakukan pencarian berbasis teks yang kemudian menghasilkan keluaran berupa ontologi yang memiliki konteks yang sama dengan teks yang dimasukan oleh *user*.

**DAFTAR PUSTAKA**

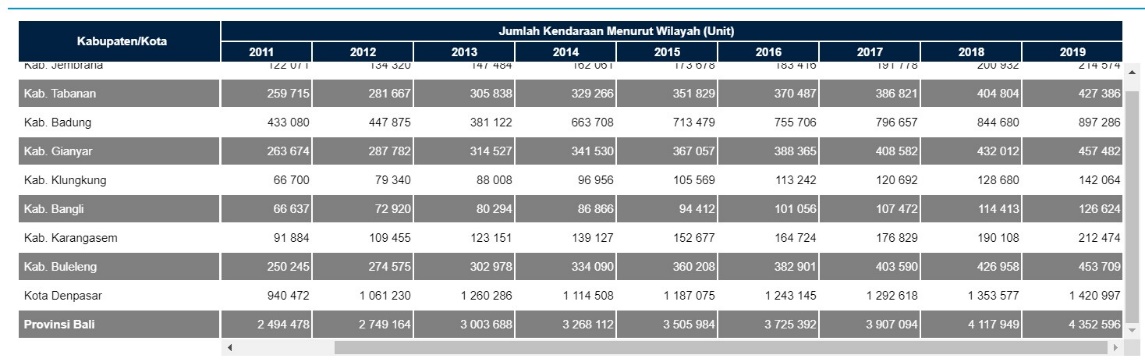
**LAMPIRAN A**

**DATA DISTRIBUSI SEPEDA MOTOR DAN PENGGUNAAN KENDARAAN PROVINSI BALI**

1. Data Distribusi dan *Export* Sepeda Motor dari tahun 2011-2020



1. Data Penggunaan Kendaraan di Provinsi Bali pada tahun 2019



**LAMPIRAN B**

**DAFTAR CLASS, OBJECT PROPERTIES, DATA PROPERTIES DAN INDIVIDUAL PADA ONTOLOGI**

1. Daftar *class* pada ontologi Sepeda Motor

|  |  |
| --- | --- |
| **Nama** | **Deskripsi** |
| Agen | Merupakan *class* untuk agen-agen seperti dealer dan service center. |
| Dealer | Merupakan *class* untuk menampung individual dealer. |
| Lokasi | Merupakan *class* untuk dearah yaitu provinsi dan kabupaten. |
| Kabupaten | Merupakan *class* untuk menampung individual kabupaten. |
| Provinsi | Merupakan *class* untuk menampung individual provinsi. |
| ServiceCenter | Merupakan *class* untuk menampung individual service center. |
| KriteriaRekomendasi | Merupakan *class* untuk kriteria pada rekomendasi, yaitu jenis kriteria dan nama kriteria. |
| JenisKriteria | Merupakan *class* untuk menampung individual jenis-jenis kriteria. |
| NamaKriteria | Merupakan *class* untuk menampung individual nama-nama kriteria pada rekomendasi. |
| Motor | Merupakan *class* untuk menampung individual motor. |
| Merek | Merupakan *class* untuk menampung individual merek. |
| SistemBahanBakar | Merupakan *class* untuk menampung individual sistem bahan bakar. |
| TahunProduksi | Merupakan *class* untuk menampung individual tahun produksi motor. |
| Transmisi | Merupakan *class* untuk menampung individual jenis transmisi motor. |
| Type | Merupakan *class* untuk menampung individual type motor. |
| VolumeSilinder | Merupakan *class* untuk menampung individual volume silinder motor. |

1. Daftar *object properties* pada ontologi Sepeda Motor

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Relasi** | **Class Sumber** | **Class Tujuan** | **Relasi Inverse** |
| AdalahBagianDari | Kabupaten | Provinsi | MemilikiBagian |
| AdalahDealerDari | Dealer | Merek | MemilikiDealer |
| AdalahJenisKriteria | NamaKriteria | JenisKriteria | TerdapatNamaKriteria |
| AdalahJenisTransmisi | Motor | Transmisi | MemilikiMotor |
| AdalahMerkDari | Motor | Merek | MemilikiMotor |
| AdalahServiceCentreDari | ServiceCenter | Merek | MemilikiMotor |
| MemilikiBagian | Provinsi | Kabupaten | MemilikiServiceCentre |
| MemilikiDealer | Merek | Dealer | AdalahBagianDari |
| MemilikiJenis | Motor | Type | AdalahDealerDari |
| MemilikiLokasi | Dealer, ServiceCentre | Kabupaten | TerdapatDealer, TerdapatServiceCentre |
| MemilikiMotor | TahunProduksi, VolumeSilinder, Transmisi, SistemBahanBakar, Merek, Type | Motor | MemilikiVolumeSilinder, MemilikiSistemBahanBakar, MemilikiTahunProduksi, AdalahMerkDari, AdalahJenisTransmisi |
| MemilikiServiceCentre | Merek | ServiceCenter | AdalahServiceCentreDari |
| MemilikiSistemBahanBakar | Motor | SistemBahanBakar | MemilikiMotor |
| MemilikiTahunProduksi | Motor | TahunPoduksi | MemilikiMotor |
| MemilikiVolumeSilinder | Motor | VolumeSilinder | MemilikiMotor |
| TerdapatDealer | Kabupaten | Dealer | MemilikiLokasi |
| TerdapatNamaKriteria | JenisKriteria | NamaKriteria | AdalahJenisKriteria |
| TerdapatServiceCentre | Kabupaten | ServiceCenter | MemilikiLokasi |

1. Daftar *data properties* pada ontologi Sepeda Motor

|  |  |
| --- | --- |
| **Nama Relasi** | **Deskripsi** |
| MemilikiAlamat | Merupakan relasi untuk memberikan nilai *string* alamat pada individual dealer dan service center. |
| MemilikiBobot | Merupakan relasi untuk memberikan nilai *float* bobot pada individu nama kriteria. |
| MemilikiDimensi | Merupakan *data property* yang memiliki sub *data property* lainnya. |
| MemilikiDimensiLebar | Merupakan relasi yang memberikan nilai *integer* dimensi lebar pada individual motor. |
| MemilikiDimensiPanjang | Merupakan relasi yang memberikan nilai *integer* dimensi panjang pada individual motor. |
| MemilikiDimensiTinggi | Merupakan relasi yang memberikan nilai *integer* dimensi tinggi pada individual motor. |
| MemilikiGambar | Merupakan relasi yang memberikan nilai *string* nama gambar pada individual motor. |
| MemilikiHarga | Merupakan relasi yang memberikan nilai *integer* harga pada individual motor. |
| MemilikiHariBuka | Merupakan relasi yang memberikan nilai *string* hari buka pada individual dealer dan service center. |
| MemilikiJamBuka | Merupakan relasi yang memberikan nilai *string* jam buka pada individual dealer dan service center. |
| MemilikiJamTutup | Merupakan relasi yang memberikan nilai *string* jam tutup pada individual dealer dan service center. |
| MemilikiJumlahTransmisi | Merupakan relasi yang memberikan nilai *integer* jumlah transmisi pada individual motor. |
| MemilikiKapasitasBahanBakar | Merupakan relasi yang memberikan nilai *float* kapasitas bahan bakar pada individual motor. |
| MemilikiKecepatan | Merupakan relasi yang memberikan nilai *float* kecepatan pada individual motor. |
| MemilikiNama | Merupakan relasi yang memberikan nilai *string* nama pada individual di seluruh *class* yang ada. |
| MemilikiNoTelp | Merupakan relasi yang memberikan nilai *string* nomor telepon pada individual dealer dan service center. |
| MemilikiTingkatKonsumsiBahanBakar | Merupakan relasi yang memberikan nilai *float* tingkat konsumsi bahan bakar pada individual motor. |

1. Daftar individual pada ontologi Sepeda Motor

**LAMPIRAN C**

**PARTISIPAN PENGUJIAN AKURASI DAN EVALUASI SISTEM**

|  |  |  |
| --- | --- | --- |
| **No.** | **Nama Responden** | **Keterangan** |
| 1. | I Wayan Widnyana | Mahasiswa |
| 2. | I Gusti Lanang Ary kresnawan | Mahasiswa |
| 3. | I Gusti Bagus Dharma Putra | Mahasiswa |
| 4. | I Putu Agus Wahyu Widiatmika | Mahasiswa |
| 5. | Putu Mas Anggita Putra | Mahasiswa |
| 6. | I Kadek Anom Sukawirasa Putra | Mahasiswa |
| 7. | Muhammad Firdaus Zulkarnain | Mahasiswa |
| 8. | Made Rusdinda Hartani | Mahasiswa |
| 9. | Ni Luh Putu Dianti Parasmitha Sari | Mahasiswa |
| 10. | I Putu Indie Surya Jayadi | Mahasiswa |
| 11. | Wayan Yogi Astira | Mahasiswa |
| 12. | I Made Yoga Mahendra | Mahasiswa |
| 13. | Ni Kadek Sukma Putri Rahayu | Mahasiswa |
| 14. | Safira | Mahasiswa |
| 15. | Gusto Ginting | Mahasiswa |
| 16. | I Made Wasanta Bhaskara | Mahasiswa |
| 17. | I Nengah Aryadi Suputra | Mahasiswa |
| 18. | Ni Made Elvina Aryadhika Putri | Mahasiswa |
| 19. | i gede acintia udayana | Mahasiswa |
| 20. | Anak Agung Gde Ramananda kartikeya pattraksha | Mahasiswa |
| 21. | Gde Deva Dimastawan Saputra | Mahasiswa |
| 22. | I Wayan Gede Indrayasa | Mahasiswa |
| 23. | Anggi kusumadewi | Mahasiswa |
| 24. | I Made Suastika | Mahasiswa |
| 25. | Jro Mangku Deny Saputra | Mahasiswa |

**LAMPIRAN D**

**LEMBAR KUESIONER DEMOGRAFI PARTISIPAN**

**Demugrafi Pertisipan**

Bagian ini bertujuan untuk mendata demografi partisipan koesioner ini. Jawaban pertanyaan di bawah ini sesuai dengan biodata dan keadaan Anda saat ini.

|  |  |  |
| --- | --- | --- |
| **No.** | **Pertanyaan** | **Jawaban** |
| 1. | Nama |  |
| 2. | Alamat |  |
| 3. | HP/Telepon |  |
| 4. | Umur |  |
| 5. | Jenis Kelamin | ○ Laki-laki  ○ Perempuan |
| 6. | Agama | ○ Hindu  ○ Islam  ○ Budha  ○ Kristen  ○ Katholik  ○ Konghucu |
| 7. | Status Perkawinan | ○ Belum Kawin  ○ Kawin  ○ Cerai  ○ Duda/Janda |
| 8. | Wilayah Tempat Tinggal | ○ Perkotaan  ○ Perdesaan |
| 9. | Pekerjaan | ○ PNS/ASN/Polri  ○ Wiraswasta  ○ Dosen/Guru  ○ Petani/Nelayan  ○ Mahasiswa  ○ Lainnya |
| 10. | Nama Instansi (Jika Pekerjaan Mahasiswa) |  |
| 11. | Berapa lama anda telah menggunakan Komputer/Internet? | ○ Tidak pernah menggunakan  ○ Kurang dari 6 bulan  ○ 6 sampai 12 bulan  ○ 1 tahun atau lebih |
| 12. | Pilih keanggotaan komunitas lokal Anda. | ○ Banjar  ○ Sekehe teruna/teruni  ○ Tidak ada  ○ Lainnya |

**LAMPIRAN E**

**LEMBAR KUESIONER PENGUJIAN AKURASI SISTEM DISERTAI HASILNYA**

**Pengujian Akurasi Fitur Penjelajahan**

Bagian ini bertujuan untuk melakukan pengujian terhadap fitur penjelajahan pada sistem.

Lakukan penjelajahan pada sistem untuk menemukan informasi sesuai dengan pertanyaan. Kemudian, jawablah pertanyaan di bawah ini sesuai dengan hasil yang didapatkan pada sistem.

|  |  |
| --- | --- |
| Tugas 1: Sebutkan 5 sepeda motor yang menggunakan sistem transmisi auto! | |
| Jawaban: | Sepenuhnya benar: 21  Sebagian benar: 4  Salah: 0 |
| Tugas 2: Sebutkan konsumsi bahan bakar motor Address FI! | |
| Jawaban: | Sepenuhnya benar: 18  Sebagian benar: 0  Salah: 7 |
| Tugas 3: Sebutkan nama-nama merek! | |
| Jawaban: | Sepenuhnya benar: 23  Sebagian benar: 2  Salah: 0 |
| Tugas 4: Sebutkan 3 motor yang menggunakan volume silinder 175cc! | |
| Jawaban: | Sepenuhnya benar: 23  Sebagian benar: 0  Salah: 2 |
| Tugas 5: Sebutkan 5 sepeda motor yang menggunakan jenis transmisi Bebek! | |
| Jawaban: | Sepenuhnya benar: 20  Sebagian benar: 3  Salah: 2 |

**Pengujian Akurasi Fitur Pencarian**

Bagian ini bertujuan untuk melakukan pengujian terhadap fitur pencarian pada sistem.

Lakukan pencarian pada sistem untuk menemukan informasi sesuai dengan pertanyaan. Kemudian, jawablah pertanyaan di bawah ini sesuai dengan hasil yang didapatkan pada sistem.

|  |  |
| --- | --- |
| Tugas 1: Sebutkan 4 sepeda motor yang termasuk dalam merek Honda yang menggunakan transmisi Manual, merupakan type motor Sport dan diproduksi tahun 2020! | |
| Jawaban: | Sepenuhnya benar: 17  Sebagian benar: 6  Salah: 2 |
| Tugas 2: Sebutkan 5 sepeda motor yang termasuk dalam merek Suzuki yang menggunakan transmisi Auto dan memiliki volume silinder 110cc! | |
| Jawaban: | Sepenuhnya benar: 19  Sebagian benar: 4  Salah: 2 |
| Tugas 3: Sebutkan 5 dealer yang merupakan dealer Honda yang berlokasi di Denpasar! | |
| Jawaban: | Sepenuhnya benar: 19  Sebagian benar: 5  Salah: 1 |
| Tugas 4: Sebutkan 5 service center yang merupakan service center Honda! | |
| Jawaban: | Sepenuhnya benar: 23  Sebagian benar: 1  Salah: 1 |
| Tugas 5: Sebutkan 4 sepeda motor yang termasuk dalam merek Kawasaki, menggunakan transmisi Manual dan memiliki type Touring! | |
| Jawaban: | Sepenuhnya benar: 22  Sebagian benar: 1  Salah: 2 |

**LAMPIRAN F**

**LEMBAR KUESIONER EVALUASI SISTEM DISERTAI HASILNYA**

**Evaluasi Kegunaan Yang Dirasakan dan Kemudahan Penggunaan Sistem**

Bagian ini bertujuan untuk mengevaluasi kegunaan dan kemudahan dalam penggunaan sistem.

Setelah melakukan berbagai tugas pada sistem, jawablah 6 pertanyaan persepsi kegunaan yang dirasakan dan 6 pertanyaan persepsi kemudahan penggunaan sistem sesuai dengan yang Anda rasakan.

***Keterangan:*** *Dimensi persepsi kegunaan yang dirasakan (perceived usefullness, PU) dan dimensi persepsi kemudahan penggunaan (perceived easy of use, PE) dalam kuesioner ini mengadopsi kuesioner Technology Acceptence Model yang dibangun oleh Davis (1989).*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Menggunakan Sistem Rekomendasi Sepeda Motor akan memungkinkan saya...** | | | | | | | | | |
|  | **Sangat tidak setuju** | **Tidak setuju** | **Agak tidak setuju** | **Netral** | **Agak setuju** | **Setuju** | **Sangat setuju** | **Jumlah Orang** |
| ...menyelesaikan tugas lebih cepat. | 0 | 0 | 0 | 0 | 0 | 6 | 19 | **25** |
| ...meningkatkan kinerja tugas saya. | 0 | 0 | 0 | 0 | 0 | 11 | 14 | **25** |
| ...meningkatkan produktivitas dalam pekerjaan saya. | 0 | 0 | 0 | 0 | 2 | 9 | 14 | **25** |
| ...meningkatkan efektivitas dalam pekerjaan saya. | 0 | 0 | 0 | 0 | 1 | 8 | 16 | **25** |
| ...lebih mudah untuk melakukan pekerjaan saya. | 0 | 0 | 0 | 0 | 0 | 9 | 16 | **25** |
| ...menemukan bahwa sistem ini berguna dalam pekerjaan saya. | 0 | 0 | 0 | 0 | 2 | 10 | 13 | **25** |
| **Total Masukan** | **0** | **0** | **0** | **0** | **5** | **53** | **92** | **125** |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Saya akan menemukan bahwa Sistem Rekomendasi Sepeda Motor...** | | | | | | | | | |
|  | **Sangat tidak setuju** | **Tidak setuju** | **Agak tidak setuju** | **Netral** | **Agak setuju** | **Setuju** | **Sangat setuju** | **Jumlah Orang** |
| ...mudah untuk saya pelajari cara menggunakannya. | 0 | 0 | 0 | 0 | 0 | 8 | 17 | **25** |
| ...mudah digunakan untuk melakukan apa yang saya inginkan. | 0 | 0 | 0 | 0 | 1 | 13 | 11 | **25** |
| ...jelas dan dapat dimengerti untuk berinteraksi dengan sistem. | 0 | 0 | 0 | 0 | 2 | 6 | 17 | **25** |
| ...adalah sistem yang jelas dan mudah dimengerti. | 0 | 0 | 0 | 0 | 3 | 9 | 13 | **25** |
| ...lebih mudah untuk melakukan pekerjaan saya. | 0 | 0 | 0 | 0 | 1 | 6 | 18 | **25** |
| ...mudah untuk digunakan. | 0 | 0 | 0 | 0 | 1 | 9 | 15 | **25** |
| **Total** | **0** | **0** | **0** | **0** | **0** | **51** | **91** | **125** |

**LAMPIRAN G**

***SOURCE CODE PROGRAM***

1. Controller
2. Controller.php

|  |
| --- |
| <?php  namespace App\Http\Controllers;  use Illuminate\Foundation\Auth\Access\AuthorizesRequests;  use Illuminate\Foundation\Bus\DispatchesJobs;  use Illuminate\Foundation\Validation\ValidatesRequests;  use Illuminate\Routing\Controller as BaseController;  use EasyRdf\RdfNamespace;  use EasyRdf\Sparql\Client;  class Controller extends BaseController  {  use AuthorizesRequests, DispatchesJobs, ValidatesRequests;  public $sparql;  function \_\_construct(){    RdfNamespace::set('rdf', 'http://www.w3.org/1999/02/22-rdf-syntax-ns#');  RdfNamespace::set('rdfs', 'http://www.w3.org/2000/01/rdf-schema#');  RdfNamespace::set('owl', 'http://www.w3.org/2002/07/owl#');  RdfNamespace::set('motor', 'http://www.semanticweb.org/girikusuma/OntologiSepedaMotor#');  $this->sparql = new Client('https://jena.balidigitalheritage.com/fuseki/sepeda-motor/query');  }  public function parseData($str){  return str\_replace('http://www.semanticweb.org/girikusuma/OntologiSepedaMotor#','', $str);  }  } |

1. DealerController.php

|  |
| --- |
| <?php  namespace App\Http\Controllers;  use Illuminate\Http\Request;  class DealerController extends Controller  {  public function index()  {  //query untuk mengambil data provinsi dan menyimpan pada variabel result  $provinsi = $this->sparql->query('SELECT \* WHERE {?s rdf:type motor:Provinsi}');  $result = [];  foreach($provinsi as $item){  array\_push($result, [  'provinsi' => $this->parseData($item->s->getUri())  ]);  }  $data = [  'hasilprovinsi' => $result  ];  return view('dealer/index', $data);  }  public function location($provinsi)  {  //query untuk mengambil data kabupaten berdasarkan provinsi tertentu dan menyimpan pada variabel result  $kabupaten = $this->sparql->query('SELECT \* WHERE {?s motor:AdalahBagianDari motor:'.$provinsi.'}');  $result = [];  foreach($kabupaten as $item){  $nama = $this->parseData($item->s->getUri());  $jumlah = 0;  $getNamaMotor = $this->sparql->query('SELECT \* WHERE {?s motor:MemilikiLokasi motor:'.$nama.'. ?s rdf:type motor:Dealer}');  foreach($getNamaMotor as $motor){  $jumlah = $jumlah + 1;  }  array\_push($result, [  'namaKabupaten' => $nama,  'jumlah' => $jumlah  ]);  }  $data = [  'kabupaten' => $result,  'provinsi' => $provinsi  ];  return view('dealer/daerah', $data);  }  public function show($provinsi, $kabupaten)  {  //query untuk mengambil data merek dan menyimpan pada variabel resultMerek  $getMerek = $this->sparql->query('SELECT \* WHERE {?merek rdf:type motor:Merek}');  $resultMerek = [];  $resultDealer = [];  foreach($getMerek as $item){  array\_push($resultMerek, [  'merek' => $this->parseData($item->merek->getUri())  ]);  }  //perulangan untuk mengambil data dealer berdasarkan merek tertentu dan daerah tertentu dan disimpan pada variabel resultDealer  foreach($resultMerek as $getDealer){  $Dealer = $this->sparql->query('SELECT \* WHERE {?s rdf:type motor:Dealer. ?s motor:MemilikiLokasi motor:'.$kabupaten.'. ?s motor:AdalahDealerDari motor:'.$getDealer['merek'].'. ?s motor:MemilikiNama ?nama}');  foreach($Dealer as $item){  array\_push($resultDealer, [  'id' => $this->parseData($item->s->getUri()),  'merekDealer' => $getDealer['merek'],  'nama' => $this->parseData($item->nama->getValue())  ]);  }  }  $data = [  'getMerek' => $resultMerek,  'getDealer' => $resultDealer,  'provinsi' => $provinsi,  'kabupaten' => $kabupaten  ];  return view('dealer/list', $data);  }  public function detail($id)  {  //query untuk mengambil data pada satu dealer tertentu  $getDetail = $this->sparql->query("SELECT \* WHERE {motor:".$id." motor:AdalahDealerDari ?merek. motor:".$id." motor:MemilikiAlamat ?alamat. motor:".$id." motor:MemilikiJamBuka ?jamBuka. motor:".$id." motor:MemilikiJamTutup ?jamTutup. motor:".$id." motor:MemilikiNoTelp ?noTelp. motor:".$id." motor:MemilikiHariBuka ?hari. motor:".$id." motor:MemilikiNama ?nama. motor:".$id." motor:MemilikiGambar ?gambar}");  $result = [];  foreach($getDetail as $item){  array\_push($result, [  'merek' => $this->parseData($item->merek->getUri()),  'nama' => $this->parseData($item->nama->getValue()),  'alamat' => $this->parseData($item->alamat->getValue()),  'jamBuka' => $this->parseData($item->jamBuka->getValue()),  'jamTutup' => $this->parseData($item->jamTutup->getValue()),  'noTelp' => $this->parseData($item->noTelp->getValue()),  'hariBuka' => $this->parseData($item->hari->getValue()),  'gambar' => $this->parseData($item->gambar->getValue())  ]);  }  $jumlah = count($result);  $data = [  'dealer' => $result,  'id' => $id,  'jumlah' => $jumlah  ];  return view('dealer/detail', $data);  }  } |

1. MerekController.php

|  |
| --- |
| <?php  namespace App\Http\Controllers;  use Illuminate\Http\Request;  class MerekController extends Controller  {  public function index()  {  //query untuk mengambil data merek motor dan disimpan pada variabel result  $merek = $this->sparql->query("SELECT \* WHERE {?s rdf:type motor:Merek}");  $result = [];  foreach($merek as $item){  array\_push($result, [  'namamerek' => $this->parseData($item->s->getUri())  ]);  }  $data = [  'merek' => $result  ];  return view('merekmotor/index', $data);  }    public function show($merek)  {  //query untuk mengambil data motor berdasarkan merek tertentu dan disimpan pada variabel result  $getnama = $this->sparql->query("SELECT \* WHERE {?s motor:AdalahMerkDari motor:".$merek.". ?s motor:MemilikiNama ?n. ?s motor:MemilikiGambar ?gambar}");  $result = [];  $jumlah = 0;  foreach($getnama as $item){  array\_push($result, [  'id' => $this->parseData($item->s->getUri()),  'nama' => $this->parseData($item->n->getValue()),  'gambar' => $this->parseData($item->gambar->getValue())  ]);  $jumlah = $jumlah + 1;  }  $data = [  'motor' => $result,  'jumlah' => $jumlah,  'merek' => $merek  ];  return view('merekmotor/list', $data);  }  } |

1. MotorController.php

|  |
| --- |
| <?php  namespace App\Http\Controllers;  use Illuminate\Http\Request;  use Illuminate\Pagination\Paginator;  use Illuminate\Support\Collection;  use Illuminate\Pagination\LengthAwarePaginator;  class MotorController extends Controller  {  public function menu()  {  //query untuk mengambil data motor, merek, transmisi, type, tahun, dan volume  $motor = $this->sparql->query('SELECT \* WHERE {?s rdf:type motor:Motor. ?s motor:MemilikiNama ?o}');  $merek = $this->sparql->query('SELECT \* WHERE {?s rdf:type motor:Merek}');  $transmisi = $this->sparql->query('SELECT \* WHERE {?s rdf:type motor:Transmisi}');  $type = $this->sparql->query('SELECT \* WHERE {?s rdf:type motor:Type}');  $tahun = $this->sparql->query('SELECT \* WHERE {?s rdf:type motor:TahunProduksi}');  $volume = $this->sparql->query('SELECT \* WHERE {?s rdf:type motor:VolumeSilinder}');  $jumlahMotor = $jumlahMerek = $jumlahTransmisi = $jumlahType = $jumlahTahun = $jumlahVolume = 0;  foreach($motor as $item){  $jumlahMotor = $jumlahMotor + 1;  }  foreach($merek as $item){  $jumlahMerek = $jumlahMerek + 1;  }  foreach($transmisi as $item){  $jumlahTransmisi = $jumlahTransmisi + 1;  }  foreach($type as $item){  $jumlahType = $jumlahType + 1;  }  foreach($tahun as $item){  $jumlahTahun = $jumlahTahun + 1;  }  foreach($volume as $item){  $jumlahVolume = $jumlahVolume + 1;  }  $data = array(  'jumlahMotor' => $jumlahMotor,  'jumlahMerek' => $jumlahMerek,  'jumlahTransmisi' => $jumlahTransmisi,  'jumlahType' => $jumlahType,  'jumlahTahun' => $jumlahTahun,  'jumlahVolume' => $jumlahVolume  );  return view ('sepedamotor/menu', ['data' => $data]);  }    public function index()  {  //query untuk mengambil data motor dan disimpan pada variabel result  $motor = $this->sparql->query("SELECT \* WHERE {?s rdf:type motor:Motor. ?s motor:MemilikiNama ?o. ?s motor:MemilikiGambar ?gambar}");  $result = [];  foreach($motor as $item){  array\_push($result, [  'hasilmotor' => $this->parseData($item->o->getValue()),  'idmotor' => $this->parseData($item->s->getUri()),  'gambar' => $this->parseData($item->gambar->getValue())  ]);  }  $data = [  'motor' => $result  ];  // parsing $result ke fungsi getPagination  // $data = $this->getPagination($result);  return view ('sepedamotor/index', $data);    // return view untuk pagination  // return view ('sepedamotor/index', compact('data'));  }  // fungsi getPagination untuk melakukan paging  public function getPagination($items, $perPage = 5, $page = null, $options = [])  {  $page = $page ?: (Paginator::resolveCurrentPage() ?: 1);  $items = $items instanceof Collection ? $items : Collection::make($items);  return new LengthAwarePaginator($items->forPage($page, $perPage), $items->count(), $perPage, $page, $options);  }  public function show($idmotor)  {  //query untuk mengambil data nama motor  $getnama = $this->sparql->query("SELECT \* WHERE {motor:".$idmotor." motor:MemilikiNama ?n}");  foreach($getnama as $i){  $nama = $this->parseData($i->n->getValue());  }  //query untuk mengambil semua data pada satu motor  $sql = $this->sparql->query("SELECT \* WHERE {motor:".$idmotor." motor:AdalahJenisTransmisi ?t. motor:".$idmotor." motor:AdalahMerkDari ?m. motor:".$idmotor." motor:MemilikiSistemBahanBakar ?sb. motor:".$idmotor." motor:MemilikiJenis ?j. motor:".$idmotor." motor:MemilikiVolumeSilinder ?v . motor:".$idmotor." motor:MemilikiTahunProduksi ?tp. motor:".$idmotor." motor:MemilikiTingkatKonsumsiBahanBakar ?konsumsi. motor:".$idmotor." motor:MemilikiKecepatan ?kecepatan. motor:".$idmotor." motor:MemilikiKapasitasBahanBakar ?kapasitas. motor:".$idmotor." motor:MemilikiDimensiLebar ?L. motor:".$idmotor." motor:MemilikiDimensiTinggi ?T. motor:".$idmotor." motor:MemilikiDimensiPanjang ?P. motor:".$idmotor." motor:MemilikiHarga ?harga. motor:".$idmotor." motor:MemilikiGambar ?gambar}");  $jumlah = 0;  foreach($sql as $item){  $jumlah = $jumlah + 1;  }  $result = [];  foreach($sql as $item){  array\_push($result, [  'transmisi' => $this->parseData($item->t->getUri()),  'type' => $this->parseData($item->j->getUri()),  'merek' => $this->parseData($item->m->getUri()),  'sistem' => $this->parseData($item->sb->getUri()),  'volume' => $this->parseData($item->v->getUri()),  'tahun' => $this->parseData($item->tp->getUri()),  'konsumsi' => $this->parseData($item->konsumsi->getValue()),  'kecepatan' => $this->parseData($item->kecepatan->getValue()),  'kapasitas' => $this->parseData($item->kapasitas->getValue()),  'lebar' => $this->parseData($item->L->getValue()),  'tinggi' => $this->parseData($item->T->getValue()),  'panjang' => $this->parseData($item->P->getValue()),  'harga' => $this->parseData($item->harga->getValue()),  'namagambar' => $this->parseData($item->gambar->getValue())  ]);  }  $data = [  'motor' => $result,  'nama' => $nama,  'jumlah' => $jumlah  ];  return view('sepedamotor/detail', $data);  }  } |

1. PagesController.php

|  |
| --- |
| <?php  namespace App\Http\Controllers;  use Illuminate\Http\Request;  class PagesController extends Controller  {  public function home(){  return view ('index');  }  public function browsing(){  return view ('browsing');  }  public function info(){  return view ('info');  }  } |

1. RekomendasiController.php

|  |
| --- |
| <?php  namespace App\Http\Controllers;  use Illuminate\Http\Request;  class RekomendasiController extends Controller  {  public function index()  {  //query untuk mengambil data dari fuseki untuk select option  $merek = $this->sparql->query('SELECT \* WHERE {?merek rdf:type motor:Merek}');  $transmisi = $this->sparql->query('SELECT \* WHERE {?transmisi rdf:type motor:Transmisi}');  $typemotor = $this->sparql->query('SELECT \* WHERE {?typemotor rdf:type motor:Type}');  $tahun = $this->sparql->query('SELECT \* WHERE {?tahun rdf:type motor:TahunProduksi}');  $volume = $this->sparql->query('SELECT \* WHERE {?volume rdf:type motor:VolumeSilinder}');    $resultMerek = [];  $resultTransmisi = [];  $resultType = [];  $resultTahun = [];  $resultVolume = [];    //mengambl data dari fuseki server  foreach($merek as $item){  array\_push($resultMerek, [  'hasilMerek' => $this->parseData($item->merek->getUri())  ]);  }  foreach($transmisi as $item){  array\_push($resultTransmisi, [  'hasilTransmisi' => $this->parseData($item->transmisi->getUri())  ]);  }  foreach($typemotor as $item){  array\_push($resultType, [  'hasilType' => $this->parseData($item->typemotor->getUri())  ]);  }  foreach($tahun as $item){  array\_push($resultTahun, [  'hasilTahun' => $this->parseData($item->tahun->getUri())  ]);  }  foreach($volume as $item){  array\_push($resultVolume, [  'hasilVolume' => $this->parseData($item->volume->getUri())  ]);  }  $dataMotor = $this->sparql->query("SELECT \* WHERE {?motor rdf:type motor:Motor. ?motor motor:MemilikiNama ?nama}");  $resultMotor = [];  foreach($dataMotor as $item){  array\_push($resultMotor, [  'id' => $this->parseData($item->motor->getUri()),  'nama' => $this->parseData($item->nama->getValue())  ]);  }  //menyimpan nilai pada satu variabel data  $data = [  'getMerek' => $resultMerek,  'getTransmisi' => $resultTransmisi,  'getType' => $resultType,  'getTahun' => $resultTahun,  'getVolume' => $resultVolume,  'getMotor' => $resultMotor  ];  return view ('rekomendasi/index', $data);  }  public function getSAW(Request $request)  {  if($request->has('cari\_filter')){  //query untuk mengambil data sepeda motor dari fuseki server berdasarkan kriteria yang dipilih  $sql = "SELECT \* WHERE {?motor rdf:type motor:Motor";  if($request->merek != 'semua'){  $sql = $sql." .?motor motor:AdalahMerkDari motor:".$request->merek;  }  if($request->transmisi != 'semua'){  $sql = $sql." .?motor motor:AdalahJenisTransmisi motor:".$request->transmisi;  }  if($request->typemotor != 'semua'){  $sql = $sql." .?motor motor:MemilikiJenis motor:".$request->typemotor;  }  if($request->tahun != 'semua'){  $sql = $sql." .?motor motor:MemilikiTahunProduksi motor:".$request->tahun;  }  if($request->volume != 'semua'){  $sql = $sql." .?motor motor:MemilikiVolumeSilinder motor:".$request->volume;  }    $sql = $sql.". ?motor motor:MemilikiHarga ?harga. ?motor motor:MemilikiTingkatKonsumsiBahanBakar ?tingkatbbm. ?motor motor:MemilikiKecepatan ?kecepatan. ?motor motor:MemilikiKapasitasBahanBakar ?kapasitas. ?motor motor:MemilikiNama ?nama}";  $resultMotor = [];    $query = $this->sparql->query($sql);  //menyimpan data sepeda motor pada variabel $resultMotor  foreach($query as $item){  array\_push($resultMotor, [  0 => $this->parseData($item->nama->getValue()),  1 => intval($this->parseData($item->harga->getValue())),  2 => floatval($this->parseData($item->kapasitas->getValue())),  3 => floatval($this->parseData($item->kecepatan->getValue())),  4 => floatval($this->parseData($item->tingkatbbm->getValue()))  ]);  }  } else {  $resultMotor = [];  $jumlahMotor = count($request->motor);  //mengambil data motor dari request dan menyimpan ke sebuah array  for($x = 0; $x < $jumlahMotor; $x++){  $query = $this->sparql->query('SELECT \* WHERE {motor:'.$request->motor[$x].' rdf:type motor:Motor. motor:'.$request->motor[$x].' motor:MemilikiHarga ?harga. motor:'.$request->motor[$x].' motor:MemilikiTingkatKonsumsiBahanBakar ?tingkatbbm. motor:'.$request->motor[$x].' motor:MemilikiKecepatan ?kecepatan. motor:'.$request->motor[$x].' motor:MemilikiKapasitasBahanBakar ?kapasitas. motor:'.$request->motor[$x].' motor:MemilikiNama ?nama}');  foreach($query as $item){  array\_push($resultMotor, [  0 => $this->parseData($item->nama->getValue()),  1 => intval($this->parseData($item->harga->getValue())),  2 => floatval($this->parseData($item->kapasitas->getValue())),  3 => floatval($this->parseData($item->kecepatan->getValue())),  4 => floatval($this->parseData($item->tingkatbbm->getValue()))  ]);  }  }  }    $jumlah = count($resultMotor);  //query untuk mengambil data kriteria dari fuseki server  $query = $this->sparql->query("SELECT \* WHERE {?kriteria rdf:type motor:NamaKriteria. ?kriteria motor:MemilikiBobot ?bobot. ?kriteria motor:AdalahJenisKriteria ?jenis}");    //menyimpan data kriteria pada variabel $getKriteria  $getKriteria = [];  $kode = 1;  foreach($query as $item){  array\_push($getKriteria, [  'kriteria' => $this->parseData($item->kriteria->getUri()),  'jenis' => $this->parseData($item->jenis->getUri()),  'bobot' => floatval($this->parseData($item->bobot->getValue())),  'kode' => "C".($kode)  ]);  $kode += 1;  }  //memanggil fungsi getCrips untuk memberi nilai pada table Data Crips  $cripsData = $this->getCrips($getKriteria);  //memanggil fungsi getNilaiLaternatif untuk memberi nilai pada tabel Data Nilai Alternatif  $nilaiAlternatif = $this->getNilaiAlternatif($getKriteria, $resultMotor, $cripsData);    //memanggil fungsi getNormalisasi untuk memberi nilai pada tabel Hasil Normalisasi  $normalisasi = $this->getNormalisasi($getKriteria, $nilaiAlternatif, $jumlah);  //memanggil fungsi getRanking untuk memberi nilai pada tabel Nilai Pembobotan  $rankingData = $this->getRanking($normalisasi);  //memanggil fungsi getResultSAW untuk memberi nilai pada tabel Hasil Simple Additive Weighting  $hasilSAW = $this->getResultSAW($rankingData);  //menyimpan semua variabel pada variabel $data  $data = [  'motor' => $resultMotor,  'normalisasi' => $normalisasi,  'ranking' => $rankingData,  'hasilSAW' => $hasilSAW,  'bobot' => $getKriteria,  'crips' => $cripsData,  'alternatif' => $nilaiAlternatif,  'jumlah' => $jumlah  ];    return view('rekomendasi/hasil', $data);  }  public function getCrips($kriteria)  {  //query untuk mengambil data seluruh sepeda motor  $query = $this->sparql->query("SELECT \* WHERE {?motor rdf:type motor:Motor. ?motor motor:MemilikiHarga ?harga. ?motor motor:MemilikiTingkatKonsumsiBahanBakar ?tingkatbbm. ?motor motor:MemilikiKecepatan ?kecepatan. ?motor motor:MemilikiKapasitasBahanBakar ?kapasitas}");  $motor = [];  //meyimpan data sepeda motor pada variabel $motor  foreach($query as $item){  array\_push($motor, [  'motor' => $this->parseData($item->motor->getUri()),  'harga' => intval($this->parseData($item->harga->getValue())),  'kapasitas' => floatval($this->parseData($item->kapasitas->getValue())),  'kecepatan' => floatval($this->parseData($item->kecepatan->getValue())),  'konsumsi' => floatval($this->parseData($item->tingkatbbm->getValue()))  ]);  }  //menghitung jumlah motor dan kriteria  $jumlahKriteria = count($kriteria);  $jumlahMotor = count($motor);  //inisialisasi variabel array $crips  $crips = [];  //cek maxmin harga harga dan mencari nilai crips harga => $crips[0]  $arrayHarga = [];  for($x = 0; $x < $jumlahMotor; $x++){  $arrayHarga[$x] = $motor[$x]['harga'];  }  $maxHarga = max($arrayHarga);  $minHarga = min($arrayHarga);  $rasio = ($maxHarga - $minHarga)/4;  $tempHarga = $minHarga;  for($x = 0; $x < 4; $x++){  $crips[0][$x] = $tempHarga + $rasio;  $tempHarga = $crips[0][$x];  }  //cek maxmin kapasitas dan mencari nilai crips kapasitas => $crips[1]  $arrayKapasitas = [];  for($x = 0; $x < $jumlahMotor; $x++){  $arrayKapasitas[$x] = $motor[$x]['kapasitas'];  }  $maxKapasitas = max($arrayKapasitas);  $minKapasitas = min($arrayKapasitas);  $rasio = ($maxKapasitas - $minKapasitas)/4;  $tempKapasitas = $minKapasitas;  for($x = 0; $x < 4; $x++){  $crips[1][$x] = $tempKapasitas + $rasio;  $tempKapasitas = $crips[1][$x];  }  //cek maxmin kecepatan dan mencari nilai crips kecepatan => $crips[2]  $arrayKecepatan = [];  for($x = 0; $x < $jumlahMotor; $x++){  $arrayKecepatan[$x] = $motor[$x]['kecepatan'];  }  $maxKecepatan = max($arrayKecepatan);  $minKecepatan = min($arrayKecepatan);  $rasio = ($maxKecepatan - $minKecepatan)/4;  $tempKecepatan = $minKecepatan;  for($x = 0; $x < 4; $x++){  $crips[2][$x] = $tempKecepatan + $rasio;  $tempKecepatan = $crips[2][$x];  }  //cek maxmin konsumsi dan mencari nilai crips konsumsi => $crips[3]  $arrayKonsumsi = [];  for($x = 0; $x < $jumlahMotor; $x++){  $arrayKonsumsi[$x] = $motor[$x]['konsumsi'];  }  $maxKonsumsi = max($arrayKonsumsi);  $minKonsumsi = min($arrayKonsumsi);  $rasio = ($maxKonsumsi - $minKonsumsi)/4;  $tempKonsumsi = $minKonsumsi;  for($x = 0; $x < 4; $x++){  $crips[3][$x] = $tempKonsumsi + $rasio;  $tempKonsumsi = $crips[3][$x];  }  //memberi nilai crips  $arrayValueCrips = array(  array(25, 50, 75, 100),  array(25, 50, 75, 100),  array(25, 50, 75, 100),  array(25, 50, 75, 100)  );  //menyimpan nilai crips total  $getCripsData = [];  $iterasi = 0;  for($i = 0; $i <$jumlahKriteria; $i++){  for($j = 0; $j < 4; $j++){  $iterasi = $iterasi + 1;  $getCripsData[$i][$j]['iterasi'] = $iterasi;  $getCripsData[$i][$j]['kode'] = $kriteria[$i]['kode'];  $getCripsData[$i][$j]['nama'] = $kriteria[$i]['kriteria'];  $getCripsData[$i][$j]['crips'] = $crips[$i][$j];  $getCripsData[$i][$j]['nilai'] = $arrayValueCrips[$i][$j];  }  }  return $getCripsData;  }  public function getNilaiAlternatif($kriteria, $motor, $crips)  {  //dd($crips);  //menghitung jumlah motor dan kriteria  $jumlahMotor = count($motor);  $jumlahKriteria = count($kriteria);  //memberi nilai nilai alternatif sepeda motor  $getNilaiLaternatif = [];  for($i = 0; $i < $jumlahMotor; $i++){  $getNilaiLaternatif[$i]['nama'] = $motor[$i][0];  for($j = 0; $j < $jumlahKriteria; $j++){  for($k = 0; $k < 4; $k++){  if($motor[$i][$j+1] <= $crips[$j][$k]['crips'])  {  $getNilaiLaternatif[$i][$j] = $crips[$j][$k]['nilai'];  break;  }  }  }  }    return $getNilaiLaternatif;  }  public function getNormalisasi ($kriteria, $data, $jumlahMotor)  {  //menghitung jumlah kriteria  $jumlahKriteria = count($kriteria);  //inisilalisasi nilai maxmin  $MaxMin = [];  foreach($kriteria as $item){  if($item['jenis'] == 'Cost'){  array\_push($MaxMin, 99999999);  } else {  array\_push($MaxMin, 0);  }  }  //memberi nilai minimal dan maksimal dari kriteria  for($x = 0; $x < $jumlahKriteria; $x++){  if($kriteria[$x]['jenis'] == 'Cost'){  for($i = 0; $i < $jumlahMotor; $i++){  for($j = 0; $j < $jumlahMotor; $j++){  if($data[$j][$x] < $MaxMin[$x]){  $MaxMin[$x] = $data[$i][$x];  }  }  }  }  else {  for($j = 0; $j < $jumlahMotor; $j++){  if($data[$j][$x] > $MaxMin[$x]){  $MaxMin[$x] = $data[$j][$x];  }  }  }  }    //perhitungan rating normalisasi  $ratingNormalisasi = array();  for($i = 0; $i < $jumlahKriteria; $i++){  for($j = 0; $j < $jumlahMotor; $j++){  if($kriteria[$i]['jenis'] == 'Cost'){  $ratingNormalisasi[$j][$i] = $MaxMin[$i] / $data[$j][$i];  }  else {  $ratingNormalisasi[$j][$i] = $data[$j][$i] / $MaxMin[$i];  }  }  }  //menambahkan nama motor pada array ratingNormalisasi  for($x = 0; $x < $jumlahMotor; $x++){  $ratingNormalisasi[$x][$jumlahKriteria] = $data[$x]['nama'];  }  return $ratingNormalisasi;  }  public function getRanking($data)  {  //menghitung jumlah motor  $jumlahMotor = count($data);  //query untuk mengambil data kriteria  $query = $this->sparql->query("SELECT \* WHERE {?kriteria rdf:type motor:NamaKriteria. ?kriteria motor:MemilikiBobot ?bobot}");    //menyimpan data kriteria pada variabel  $bobotKriteria = [];  foreach($query as $item){  array\_push($bobotKriteria, [  'kriteria' => $this->parseData($item->kriteria->getUri()),  'bobot' => floatval($this->parseData($item->bobot->getValue()))  ]);  }  //menghitung jumlah kriteria  $jumlahKriteria = count($bobotKriteria);  $hasilRekomendasi = [];  //mengalikan nilai pada data motor dengan bobot pada kriteria  for($i = 0; $i < $jumlahMotor; $i++){  for($j = 0; $j < $jumlahKriteria; $j++){  $hasilRekomendasi[$i][$j] = $data[$i][$j] \* $bobotKriteria[$j]['bobot'];  }  }  //menyimpan nama dan menghitung total pada masing-masing alternatif motor  for($x = 0; $x < $jumlahMotor; $x++){  $hasilRekomendasi[$x][$jumlahKriteria] = $data[$x][4];  }  $tempTotal = 0;  for($i = 0; $i < $jumlahMotor; $i++){  for($j = 0; $j < $jumlahKriteria; $j++){  $tempTotal = $tempTotal + $hasilRekomendasi[$i][$j];  }  $hasilRekomendasi[$i][5] = $tempTotal;  $tempTotal = 0;  }  return $hasilRekomendasi;  }  public function getResultSAW($data)  {  //menghitung jumlah motor  $jumlahMotor = count($data);    //query untuk mengambil nama kriteria  $query = $this->sparql->query("SELECT \* WHERE {?kriteria rdf:type motor:NamaKriteria}");  //menyimpan data kriteria dan menghitung jumlah kriteria  $kriteria = [];  foreach($query as $item){  array\_push($kriteria, $this->parseData($item->kriteria->getUri()));  }  $jumlahKriteria = count($kriteria);  //menyimpan data nama dan nilai pada variabel hasilSAW dari variabel data  $hasilSAW = [];  for($i = 0; $i <$jumlahMotor; $i++){  $hasilSAW[$i]['nama'] = $data[$i][4];  $hasilSAW[$i]['nilai'] = $data[$i][5];  }  //melakukan sorting dengan menggunakan bubblesort  for($j = 0; $j < $jumlahMotor; $j++){  for($i = 0; $i <$jumlahMotor; $i++){  if(($i+1) < ($jumlahMotor)){  if($hasilSAW[$i]['nilai'] < $hasilSAW[$i+1]['nilai']){  $tempNilai = $hasilSAW[$i]['nilai'];  $hasilSAW[$i]['nilai'] = $hasilSAW[$i+1]['nilai'];  $hasilSAW[$i+1]['nilai'] = $tempNilai;  $tempNama = $hasilSAW[$i]['nama'];  $hasilSAW[$i]['nama'] = $hasilSAW[$i+1]['nama'];  $hasilSAW[$i+1]['nama'] = $tempNama;  }  }  }  }    return $hasilSAW;  }  } |

1. SearchingController.php

|  |
| --- |
| <?php  namespace App\Http\Controllers;  use Illuminate\Http\Request;  class SearchingController extends Controller  {  public function index(Request $request){  //untuk mengetahui apakan terdapat request dari user  if($request->has('cari')){  $status = 1;  //untuk membangun query berdasarkan kriteria filter motor  $sql = 'SELECT \* WHERE { ?motor rdf:type motor:Motor';  if($request->cari\_merek != 'semua'){  $sql = $sql.'. ?motor motor:AdalahMerkDari motor:'.$request->cari\_merek;  }  if($request->cari\_transmisi != 'semua'){  $sql = $sql.'. ?motor motor:AdalahJenisTransmisi motor:'.$request->cari\_transmisi;  }  if($request->cari\_typemotor != 'semua'){  $sql = $sql.'. ?motor motor:MemilikiJenis motor:'.$request->cari\_typemotor;  }  if($request->cari\_tahun != 'semua'){  $sql = $sql.'. ?motor motor:MemilikiTahunProduksi motor:'.$request->cari\_tahun;  }  if($request->cari\_volume != 'semua'){  $sql = $sql.'. ?motor motor:MemilikiVolumeSilinder motor:'.$request->cari\_volume;  }  $sql = $sql.'. ?motor motor:MemilikiNama ?nama}';  $querydata = $this->sparql->query($sql);  $resultMotor = [];  $resultDealer = [];  $resultService = [];  //meyimpan data nama dan id motor  foreach($querydata as $item){  array\_push($resultMotor, [  'id' => $this->parseData($item->motor->getUri()),  'nama' => $this->parseData($item->nama->getValue())  ]);  }  $getMerek = $request->cari\_merek;  $getTransmisi = $request->cari\_transmisi;  $getJenis = $request->cari\_typemotor;  $getTahun = $request->cari\_tahun;  $getVolume = $request->cari\_volume;  $getLokasi = '';  $jumlahMotor = count($resultMotor);  $jumlahDealer = 0;  $jumlahService = 0;  } else if($request->has('cari\_dealer')){  $status = 2;  //untuk membangun query berdasarkan kriteria filter dealer  $sql = 'SELECT \* WHERE { ?dealer rdf:type motor:Dealer';  if($request->cari\_merek\_dealer != 'semua'){  $sql = $sql.'. ?dealer motor:AdalahDealerDari motor:'.$request->cari\_merek\_dealer;  }  if($request->cari\_lokasi\_dealer != 'semua'){  $sql = $sql.'. ?dealer motor:MemilikiLokasi motor:'.$request->cari\_lokasi\_dealer;  }  $sql = $sql.'. ?dealer motor:MemilikiNama ?nama}';  $querydata = $this->sparql->query($sql);  $resultMotor = [];  $resultDealer = [];  $resultService = [];  //meyimpan data nama dan id dealer  foreach($querydata as $item){  array\_push($resultDealer, [  'id' => $this->parseData($item->dealer->getUri()),  'nama' => $this->parseData($item->nama->getValue())  ]);  }  $getMerek = $request->cari\_merek\_dealer;  $getTransmisi = '';  $getJenis = '';  $getTahun = '';  $getVolume = '';  $getLokasi = $request->cari\_lokasi\_dealer;  $jumlahMotor = 0;  $jumlahDealer = count($resultDealer);  $jumlahService = 0;  } else if($request->has('cari\_service')){  $status = 3;  //untuk membangun query berdasarkan kriteria filter service center  $sql = 'SELECT \* WHERE { ?service rdf:type motor:ServiceCenter';  if($request->cari\_merek\_service != 'semua'){  $sql = $sql.'. ?service motor:AdalahServiceCentreDari motor:'.$request->cari\_merek\_service;  }  if($request->cari\_lokasi\_service != 'semua'){  $sql = $sql.'. ?service motor:MemilikiLokasi motor:'.$request->cari\_lokasi\_service;  }  $sql = $sql.'. ?service motor:MemilikiNama ?nama}';  $querydata = $this->sparql->query($sql);  $resultMotor = [];  $resultDealer = [];  $resultService = [];  //meyimpan data nama dan id service center  foreach($querydata as $item){  array\_push($resultService, [  'id' => $this->parseData($item->service->getUri()),  'nama' => $this->parseData($item->nama->getValue())  ]);  }  $getMerek = $request->cari\_merek\_service;  $getTransmisi = '';  $getJenis = '';  $getTahun = '';  $getVolume = '';  $getLokasi = $request->cari\_lokasi\_service;  $jumlahMotor = 0;  $jumlahDealer = 0;  $jumlahService = count($resultService);  } else {  $sql = '';  $getMerek = '';  $getTransmisi = '';  $getJenis = '';  $getTahun = '';  $getVolume = '';  $getLokasi = '';  $resultMotor = [];  $resultDealer = [];  $resultService = [];  $status = 0;  $jumlahMotor = 0;  $jumlahDealer = 0;  $jumlahService = 0;  $merek = "semua";  $transmisi = "semua";  $jenis = "semua";  $tahun = "semua";  $volume = "semua";  }  //melakukan query untuk mengambil data merek, transmisi, type, tahun, volume, dan lokasi  $merek = $this->sparql->query('SELECT \* WHERE {?merek rdf:type motor:Merek}');  $transmisi = $this->sparql->query('SELECT \* WHERE {?transmisi rdf:type motor:Transmisi}');  $typemotor = $this->sparql->query('SELECT \* WHERE {?typemotor rdf:type motor:Type}');  $tahun = $this->sparql->query('SELECT \* WHERE {?tahun rdf:type motor:TahunProduksi}');  $volume = $this->sparql->query('SELECT \* WHERE {?volume rdf:type motor:VolumeSilinder}');  $lokasi = $this->sparql->query('SELECT \* WHERE {?lokasi rdf:type motor:Kabupaten}');    $resultMerek = [];  $resultTransmisi = [];  $resultType = [];  $resultTahun = [];  $resultVolume = [];  $resultLokasi = [];    foreach($merek as $item){  array\_push($resultMerek, [  'hasilMerek' => $this->parseData($item->merek->getUri())  ]);  }  foreach($transmisi as $item){  array\_push($resultTransmisi, [  'hasilTransmisi' => $this->parseData($item->transmisi->getUri())  ]);  }  foreach($typemotor as $item){  array\_push($resultType, [  'hasilType' => $this->parseData($item->typemotor->getUri())  ]);  }  foreach($tahun as $item){  array\_push($resultTahun, [  'hasilTahun' => $this->parseData($item->tahun->getUri())  ]);  }  foreach($volume as $item){  array\_push($resultVolume, [  'hasilVolume' => $this->parseData($item->volume->getUri())  ]);  }  foreach($lokasi as $item){  array\_push($resultLokasi, [  'hasilLokasi' => $this->parseData($item->lokasi->getUri())  ]);  }  $data = [  'getMerek' => $resultMerek,  'getTransmisi' => $resultTransmisi,  'getType' => $resultType,  'getTahun' => $resultTahun,  'getVolume' => $resultVolume,  'getMotor' => $resultMotor,  'getDealer' => $resultDealer,  'getService' => $resultService,  'getLokasi' => $resultLokasi,  'status' => $status,  'query' => $sql,  'jumlahMotor' => $jumlahMotor,  'jumlahDealer' => $jumlahDealer,  'jumlahService' => $jumlahService,  'merek' => $getMerek,  'transmisi' => $getTransmisi,  'jenis' => $getJenis,  'tahun' => $getTahun,  'volume' => $getVolume,  'lokasi' => $getLokasi  ];  return view ('searching', $data);  }  } |

1. ServiceController.php

|  |
| --- |
| <?php  namespace App\Http\Controllers;  use Illuminate\Http\Request;  class ServiceController extends Controller  {  public function index()  {  //query untuk mengambil data provinsi dan menyimpan pada variabel result  $provinsi = $this->sparql->query('SELECT \* WHERE {?s rdf:type motor:Provinsi}');  $result = [];  foreach($provinsi as $item){  array\_push($result, [  'provinsi' => $this->parseData($item->s->getUri())  ]);  }  $data = [  'hasilprovinsi' => $result  ];  return view('servicecentre/index', $data);  }  public function location($provinsi)  {  //query untuk mengambil data kabupaten berdasarkan provinsi tertentu dan menyimpan pada variabel result  $kabupaten = $this->sparql->query('SELECT \* WHERE {?s motor:AdalahBagianDari motor:'.$provinsi.'}');  $result = [];  foreach($kabupaten as $item){  $nama = $this->parseData($item->s->getUri());  $jumlah = 0;  $getNamaMotor = $this->sparql->query('SELECT \* WHERE {?s motor:MemilikiLokasi motor:'.$nama.'. ?s rdf:type motor:ServiceCenter}');  foreach($getNamaMotor as $motor){  $jumlah = $jumlah + 1;  }  array\_push($result, [  'namaKabupaten' => $nama,  'jumlah' => $jumlah  ]);  }  $data = [  'kabupaten' => $result,  'provinsi' => $provinsi  ];  return view('servicecentre/daerah', $data);  }  public function show($provinsi, $kabupaten)  {  //query untuk mengambil data merek dan menyimpan pada variabel resultMerek  $getMerek = $this->sparql->query('SELECT \* WHERE {?merek rdf:type motor:Merek}');  $resultMerek = [];  $resultService = [];  foreach($getMerek as $item){  array\_push($resultMerek, [  'merek' => $this->parseData($item->merek->getUri())  ]);  }  //perulangan untuk mengambil data service center berdasarkan merek tertentu dan daerah tertentu dan disimpan pada variabel resultService  foreach($resultMerek as $getService){  $Service = $this->sparql->query('SELECT \* WHERE {?s rdf:type motor:ServiceCenter. ?s motor:MemilikiLokasi motor:'.$kabupaten.'. ?s motor:AdalahServiceCentreDari motor:'.$getService['merek'].'. ?s motor:MemilikiNama ?nama}');  foreach($Service as $item){  array\_push($resultService, [  'id' => $this->parseData($item->s->getUri()),  'merekService' => $getService['merek'],  'nama' => $this->parseData($item->nama->getValue())  ]);  }  }  $data = [  'getMerek' => $resultMerek,  'getService' => $resultService,  'provinsi' => $provinsi,  'kabupaten' => $kabupaten  ];  return view('servicecentre/list', $data);  }  public function detail($id)  {  //query untuk mengambil data pada satu service center tertentu  $getDetail = $this->sparql->query("SELECT \* WHERE {motor:".$id." motor:AdalahServiceCentreDari ?merek. motor:".$id." motor:MemilikiAlamat ?alamat. motor:".$id." motor:MemilikiJamBuka ?jamBuka. motor:".$id." motor:MemilikiJamTutup ?jamTutup. motor:".$id." motor:MemilikiNoTelp ?noTelp. motor:".$id." motor:MemilikiHariBuka ?hari. motor:".$id." motor:MemilikiNama ?nama. motor:".$id." motor:MemilikiGambar ?gambar}");  $result = [];  foreach($getDetail as $item){  array\_push($result, [  'merek' => $this->parseData($item->merek->getUri()),  'nama' => $this->parseData($item->nama->getValue()),  'alamat' => $this->parseData($item->alamat->getValue()),  'jamBuka' => $this->parseData($item->jamBuka->getValue()),  'jamTutup' => $this->parseData($item->jamTutup->getValue()),  'noTelp' => $this->parseData($item->noTelp->getValue()),  'hariBuka' => $this->parseData($item->hari->getValue()),  'gambar' => $this->parseData($item->gambar->getValue())  ]);  }  $jumlah = count($result);  $data = [  'service' => $result,  'id' => $id,  'jumlah' => $jumlah  ];  return view('servicecentre/detail', $data);  }  } |

1. TahunController.php

|  |
| --- |
| <?php  namespace App\Http\Controllers;  use Illuminate\Http\Request;  class TahunController extends Controller  {    public function index()  {  //query untuk mengambil data tahun motor dan disimpan pada variabel result  $tahun = $this->sparql->query("SELECT \* WHERE {?s rdf:type motor:TahunProduksi}");  $result = [];  foreach($tahun as $item){  array\_push($result, [  'tahun' => $this->parseData($item->s->getUri())  ]);  }  $data = [  'hasiltahun' => $result  ];  return view('tahunproduksi/index', $data);  }  public function show($tahun)  {  //query untuk mengambil data motor berdasarkan tahun tertentu dan disimpan pada variabel result  $getnama = $this->sparql->query("SELECT \* WHERE {?s motor:MemilikiTahunProduksi motor:".$tahun.". ?s motor:MemilikiNama ?n. ?s motor:MemilikiGambar ?gambar}");  $result = [];  $jumlah = 0;  foreach($getnama as $item){  array\_push($result, [  'id' => $this->parseData($item->s->getUri()),  'nama' => $this->parseData($item->n->getValue()),  'gambar' => $this->parseData($item->gambar->getValue())  ]);  $jumlah = $jumlah + 1;  }  $data = [  'motor' => $result,  'jumlah' => $jumlah,  'tahun' => $tahun  ];  return view('tahunproduksi/list', $data);  }  } |

1. TransmisiController.php

|  |
| --- |
| <?php  namespace App\Http\Controllers;  use Illuminate\Http\Request;  class TransmisiController extends Controller  {    public function index()  {  //query untuk mengambil data transmisi motor dan disimpan pada variabel result  $transmisi = $this->sparql->query("SELECT \* WHERE {?s rdf:type motor:Transmisi}");  $result = [];  foreach($transmisi as $item){  array\_push($result, [  'transmisi' => $this->parseData($item->s->getUri())  ]);  }  $data = [  'hasiltransmisi' => $result  ];  return view('transmisi/index', $data);  }  public function show($transmisi)  {  //query untuk mengambil data motor berdasarkan transmisi tertentu dan disimpan pada variabel resul  $getnama = $this->sparql->query("SELECT \* WHERE {?s motor:AdalahJenisTransmisi motor:".$transmisi.". ?s motor:MemilikiNama ?n. ?s motor:MemilikiGambar ?gambar}");  $result = [];  $jumlah = 0;  foreach($getnama as $item){  array\_push($result, [  'id' => $this->parseData($item->s->getUri()),  'nama' => $this->parseData($item->n->getValue()),  'gambar' => $this->parseData($item->gambar->getValue())  ]);  $jumlah = $jumlah + 1;  }  $data = [  'motor' => $result,  'jumlah' => $jumlah,  'transmisi' => $transmisi  ];  return view('transmisi/list', $data);  }  } |

1. TypeController.php

|  |
| --- |
| <?php  namespace App\Http\Controllers;  use Illuminate\Http\Request;  class TypeController extends Controller  {  public function index()  {  //query untuk mengambil data type motor dan disimpan pada variabel result  $type = $this->sparql->query("SELECT \* WHERE {?s rdf:type motor:Type}");  $result = [];  foreach($type as $item){  array\_push($result, [  'type' => $this->parseData($item->s->getUri())  ]);  }  $data = [  'hasiltype' => $result  ];  return view('typemotor/index', $data);  }  public function show($type)  {  //query untuk mengambil data motor berdasarkan type tertentu dan disimpan pada variabel result  $getnama = $this->sparql->query("SELECT \* WHERE {?s motor:MemilikiJenis motor:".$type.". ?s motor:MemilikiNama ?n. ?s motor:MemilikiGambar ?gambar}");  $result = [];  $jumlah = 0;  foreach($getnama as $item){  array\_push($result, [  'id' => $this->parseData($item->s->getUri()),  'nama' => $this->parseData($item->n->getValue()),  'gambar' => $this->parseData($item->gambar->getValue())  ]);  $jumlah = $jumlah + 1;  }  $data = [  'motor' => $result,  'jumlah' => $jumlah,  'type' => $type  ];  return view('typemotor/list', $data);  }  } |

1. VolumeController.php

|  |
| --- |
| <?php  namespace App\Http\Controllers;  use Illuminate\Http\Request;  class VolumeController extends Controller  {    public function index()  {  //query untuk mengambil data volume silinder motor dan disimpan pada variabel result  $volume = $this->sparql->query("SELECT \* WHERE {?s rdf:type motor:VolumeSilinder}");  $result = [];  foreach($volume as $item){  array\_push($result, [  'volume' => $this->parseData($item->s->getUri())  ]);  }  $data = [  'hasilvolume' => $result  ];  return view('volumesilinder/index', $data);  }  public function show($volume)  {  //query untuk mengambil data motor berdasarkan volume silinder tertentu dan disimpan pada variabel result  $getnama = $this->sparql->query("SELECT \* WHERE {?s motor:MemilikiVolumeSilinder motor:".$volume.". ?s motor:MemilikiNama ?n. ?s motor:MemilikiGambar ?gambar}");  $result = [];  $jumlah = 0;  foreach($getnama as $item){  array\_push($result, [  'id' => $this->parseData($item->s->getUri()),  'nama' => $this->parseData($item->n->getValue()),  'gambar' => $this->parseData($item->gambar->getValue())  ]);  $jumlah = $jumlah + 1;  }  $data = [  'motor' => $result,  'jumlah' => $jumlah,  'volume' => $volume  ];  return view('volumesilinder/list', $data);  }  } |

1. https://www.aisi.or.id/statistic/ [↑](#footnote-ref-1)
2. https://bali.bps.go.id/dynamictable/2018/02/02/217/banyaknya-kendaraan-bermotor-menurut-kabupaten-kota-di-bali-2010-2016.html [↑](#footnote-ref-2)