**Latar Belakang**

Alkitab merupakan kitab suci umat Kristiani di seluruh dunia yang berisi firman-firman Allah yang ditulis oleh tokoh-tokoh dari berbagai latar belakang seperti raja, nelayan, imam, pejabat pemerintah, petani, gembala, dan dokter dalam periode sekitar 1.500 tahun. (Yunianto & Rohayani, 2021). Alkitab juga menjadi otoritas final terhadap kepercayaan dan perbuatan Kristen. Hal-hal yang berkaitan dengan kehidupan umat Kristiani baik secara horizontal maupun vertikal diatur secara jelas melalui Alkitab.

Menurut Lembaga Alkitab Indonesia, Alkitab terjemahan Indonesia terdiri dari dua kontrak yaitu Perjanjian Lama dan Perjanjian Baru yang dibagi menjadi 66 bab/kitab (39 kitab dalam Perjanjian Lama dan 27 kitab dalam Perjanjian Baru), 1.189 sub-bab/pasal, dan 31.101 Ayat yang ditulis dari tahun ± 1440 S.M – 95 M (Budilaksono, et al., 2009). Hampir semua kitab pada Alkitab ditulis oleh penulis yang berbeda-beda, terlebih lagi, tidak semua kitab ditampilkan secara eksplisit siapa penulis kitab tersebut sehingga dengan hal tersebut cukup sulit untuk mengetahui sejarah kitab tersebut beserta penulisnya sehingga dibutuhkan sebuah automatisasi sistem pencarian pada Alkitab untuk mempermudah pencarian isi Alkitab sekaligus penulisnya. Automatisasi sistem pencarian Alkitab ini membutuhkan sebuah metode yang dapat mendefinisikan hierarki pengetahuan dari Alkitab itu sendiri dalam bentuk *knowledge management*. Untuk mengatasi itu, metode *Methontology* dapat digunakan sebagai solusi permasalahan tersebut.

*Methontology* merupakan salah satu metode konstruksi model ontologi yang terstruktur dengan baik, yang digunakan untuk mengonstruksi ontologi dari awal. Ontologi sendiri adalah suatu cara untuk merepresentasikan pengetahuan dari sekumpulan konsep dalam sebuah domain informasi dan hubungan antar konsep tersebut, oleh karena itu, ontologi dapat digunakan untuk menyajikan informasi secara semantik, dan mengatur serta memetakan kumpulan sumber informasi secara sistematis dan terstruktur. Dalam hal interoperabilitas data, hal ini sangat berguna karena dapat dilakukan dengan cara yang lebih efisien (Davies, et al., 2006). Keuntungan dari metode *Methontology* ini adalah menjelaskan setiap aktivitas yang harus dilakukan secara detail. Dengan menggunakan metode *Methontology*, ontologi yang dibangun dapat digunakan kembali untuk pengembangan sistem lebih lanjut (Fernández, et al., 1997)

Dengan memanfaatkan perkembangan teknologi yang sudah sangat pesat, proses pencarian isi Alkitab dapat dilakukan secara otomatis dan terstruktur. Oleh karena itu, penulis mengembangkan sebuah sistem otomatisasi pencarian isi Alkitab berdasarkan penulisnya dengan pendekatan ontologi. Ontologi ini nantinya akan berperan untuk merepresentasikan pengetahuan semantik yang ada pada Alkitab dalam bentuk makna objek (kontrak, pasal, ayat, dan penulis), properti dari suatu objek, serta relasi antar objek (Chandrasekaran & B., 1999), dimana semantik mengacu pada hubungan hierarki antara kontrak, kitab, pasal, dan ayat. Hubungan hierarki ini nantinya yang akan menjadi nilai penting dalam mengimplementasikan pendekatan ontologi untuk melakukan pencarian secara semantik karena makna objek yang saling berkaitan.

**Rumusan Masalah**

Sistem yang akan dikembangkan akan mengangkat rumusan-rumusan yang menjadi pokok permasalahan dalam sistem pencarian isi Alkitab yang mana menggunakan metode *Methontology* dan metode *prototyping* untuk pengembangan sistem tersebut.

Berikut merupakan rumusan-rumusan masalah yang diangkat pada penelitian ini :

1. Bagaimana menerapkan metode *Methontology* untuk mengembangkan ontologi dengan domain Alkitab?
2. Bagaimana menerapkan teknologi web semantik dalam mengembangkan sistem pencarian Alkitab berdasarkan penulis?
3. Bagaimana metode *prototyping* digunakan dalam mengembangkan sistem pencarian Alkitab berdasarkan penulis?

**Batasan Masalah**

Berikut ini merupakan batasan masalah yang diangkat pada penelitian ini :

1. Data yang digunakan merupakan dataset Alkitab digital yang diambil dari *website* *kaggle.com* dan jurnal-jurnal terkait Alkitab.
2. Sistem yang dikembangkan berbasis website dengan domain Alkitab dan bahasa pemrograman yang digunakan untuk implementasi sistem adalah PHP, Javascript, HTML, OWL, dan SPARQL.
3. Menggunakan metode *Methontology* sebagai metode pengembangan sistem ontologi dan metode *protoyping* sebagai metode pengembangan sistem.
4. Metode yang digunakan untuk menguji sistem adalah *Black Box* (*Regression testing)*.

**Tujuan Penelitian**

Berikut ini merupakan tujuan dari penelitian ini :

1. Tujuan dari penelitian ini untuk menyelesaikan penugasan pada mata kuliah yang diberikan oleh dosen pengampu
2. Tujuan dari penelitian ini untuk mengembangkan sebuah aplikasi berbasis website dengan pendekatan ontologi sebagai sistem pencarian isi Alkitab berdasarkan penulis

**Manfaat Penelitian**

Berikut ini merupakan tujuan dari penelitian ini :

1. Penelitian ini diharapkan dapat memudahkan pembaca dalam menjelajahi maupun pencarian mengenai isi Alkitab berdasarkan penulisnya dari setiap kitab.
2. Penelitian ini diharapkan bisa membantu dan menjadi referensi untuk penelitian selanjutnya yang memerlukan informasi mengenai penulis kitab-kitab pada Alkitab.

**Tinjauan Pustaka**

**Tinjauan Empiris**

Pada penelitian ini, terdapat beberapa penelitian sebelumnya yang menjadi acuan atau mendukung pengembangan “Otomatisasi Sistem Pencarian Ayat Alkitab Digital Berdasarkan Penulis dengan Metode Methontology” yaitu :

* ***Al-Quran Ontology Based On Knowledge Themes***(Ta'a, et al., 2017)

Tujuan utama dari penelitian ini adalah untuk mendemonstrasikan pengembangan ontologi dari Al-Quran dan metode yang digunakan untuk mencari pengetahuan Al-Quran menggunakan pendekatan pencarian semantik. Pendekatan pencarian didasarkan pada pola kata kunci dengan isi database. Dengan memberikan kata kunci, ayat-ayat yang relevan yang berkaitan dengan tema dan subtema diambil. Hasil yang dihasilkan dari pencarian dievaluasi berdasarkan pengukuran presisi dengan perhitungan 95% dan recall 90,4% untuk mengukur kinerja pencarian informasi.

* **Building Formal Ontologies for Theology and Systems of Belief** (Graf, 2018)

Penelitian ini mengembangkan dan membahas representasi teologi dan sistem kepercayaan dengan ontologi formal dan arsitektur informasi semantik dan mengevaluasi kegunaan arsitektur informasi tersebut. studi ini mengklarifikasi sejarah ontologi dari akar kuno, melalui penemuan modern awal hingga penggunaannya saat ini dalam ilmu informasi. Setelah pengenalan ontologi, tesis berkembang dan mewakili sistem teologis dan sistem kepercayaan dalam ontologi formal yang dirancang dengan perangkat lunak ontologi, Protégé.

Tesis ini menunjukkan manfaat ontologi ini dengan mengembangkan ontologi untuk teologi dan untuk sistem kepercayaan, Calvinisme. Ontologi, *TheOn* dan *CalvOn*, mengkonseptualisasikan domain teologi umum dan Calvinisme. TheOn mengkategorikan entitas teologis umum seperti "Teolog", "Subjek" dan "Sumber" dan dapat digunakan dalam basis pengetahuan teologis karena kapasitasnya untuk secara koheren menyajikan sejumlah besar data teologis yang saling terkait termasuk sudut pandang yang bertentangan dalam teologi. CalvOn mengkonseptualisasikan Calvinisme dalam *The Logic of Calvinism* karya A.N. Prior. Dalam arsitektur informasi ini, topik (doktrin) didefinisikan sebagai saling berkaitan dengan hubungan transitif, sehingga topik fundamental mewarisi isi subtopiknya. Topik-topiknya juga terkait dengan teks-teks alkitabiah yang digunakan sebagai dasar untuk kategorisasi doktrinal asli dan isi sistem kepercayaan. Untuk kedua ontologi, kueri yang berwawasan luas ditunjukkan.

* **Ontology-Based Approach for Knowledge Retrieval in Al- Qur’an Holy Book** (Ta'a, et al., 2016)

Penelitian ini membahas tentang pendemonstrasian metode pencarian ayat-ayat Al-Qur’an yang relevan dan akurat dengan menggunakan pendekatan ontologi.selain itu, penelitian ini juga mempelajari penelitian sebelumnya yang dilakukan dengan pencarian samantik Al-Qur’an dan pencarian berbasis ontologi. Penelitian ini mengusulkan kerangka penelitian semantik dan pendekatan temu kembali ilmu pengetahuan berbasis ontologi dalam Al-Qur’an. Pendekatan yang diusulkan dalam penelitian ini terdiri dari dua tahapan yaitu tahap pengembangan ontologi Al-Qur’an dan tahap pengembangan metode pencarian.

* **Semantic-based Ontology for Malay Qur’an Reader** (Ahmad, et al., 2016)

Penelitian ini merangkum teknik pencarian yang digunakan dalam penelitian yang ada tentang Al-Qur'an. Selain itu, tulisan ini juga mempelajari penelitian sebelumnya yang dilakukan pada Pencarian Semantik Alquran dan Pencarian Berbasis Ontologi Alquran yang berfokus pada Alquran Melayu. Ulasan ini membantu penelitian dalam mengatasi masalah umum dan keterbatasan dalam Al-Qur’an Melayu yang mempengaruhinya

aksesibilitas. Penelitian ini mengusulkan kerangka penelitian untuk ontologi berbasis semantik baru untuk Alquran Melayu. Hasil akhir akan menjadi alat yang dapat diakses yang dapat membantu pembaca Melayu untuk memahami Al-Qur'an dengan cara yang lebih baik.

* **Sistem Pencarian Program Studi Pada Perguruan Tinggi Di Bali Berbasis Semantik** (Novianti & Diaz, 2017)

Penelitian ini mengembangkan sebuah sistem pencarian program studi bertujuan untuk membantu calon mahasiswa menemukan informasi terkait dengan program studi yang diinginkan. Sistem pencarian berbasis semantik ini dibangun dan menggunakan ontologi sebagai representasi informasi berbasis pengetahuan. Hal ini dilakukan dengan harapan agar sistem dan pengguna memiliki pemahaman makna yang sama serta informasi program studi memiliki nilai inferensi semantik yang baik. Ontologi sebagai dasar representasi informasi memiliki 4 *class*, 9 *datatype properties* dan 4 *object properties*. Pengujian sistem menggunakan 5 buah kata kunci uji untuk mengetahui tingkat relevansi informasi yang dihasilkan oleh sistem dan hasil yang diperoleh adalah nilai *recall* sebesar 0.95 dan nilai *precision* sebesar 0.93, sehingga dapat dikatakan bahwa sistem memiliki tingkat relevansi yang tinggi ketika mengembalikan informasi yang diinginkan pengguna. Deskripsi informasi yang diberikan oleh sistem dapat digunakan sebagai pertimbangan bagi calon mahasiswa untuk memilih program studi pada perguruan tinggi tertentu.

**Tinjauan Teoritis**

* **Alkitab**

Secara harfiahnya, kata “Alkitab” dalam bahasa Inggris berasal dari *bíblia* dalam bahasa Latin dan *bíblos* dalam bahasa Yunani. Istilah tersebut memiliki arti buku dan kemungkinan berasal dari pelabuhan Mesir kuno Byblos (sekarang Lebanon) serta pembuatan buku dan gulungan menggunakan papiros untuk diekspor ke Yunani. Istilah alkitabiah lainnya adalah Kitab Suci, atau Tulisan Suci, yang berarti Buku Suci. Alkitab adalah surat yang ditulis oleh lebih dari 50 penulis selama sekitar 1.500 tahun dan terdiri dari dua kontrak yaitu Perjanjian Lama dan Perjanjian Baru yang dibagi menjadi 66 bab/kitab (39 kitab dalam Perjanjian Lama dan 27 kitab dalam Perjanjian Baru), 1.189 sub-bab/pasal, dan 31.101 Ayat yang ditulis dari tahun ± 1440 S.M – 95 M (Budilaksono, et al., 2009). Teks asli dikomunikasikan hanya dalam tiga bahasa. Perjanjian Lama ditulis terutama dalam bahasa Ibrani, dengan persentase kecil dalam bahasa Aram. Perjanjian Baru ditulis dalam bahasa Yunani Koine. Selain dua bagian utama, Perjanjian Lama dan Baru, Alkitab memiliki beberapa bagian lain: Pentateukh, Kitab Sejarah, Puisi dan Buku Kebijaksanaan, Nubuat, Injil dan Surat-Surat.

Bagi umat Kristiani, Alkitab adalah Firman Tuhan, sumber dari segala sumber, buku dari segala buku, dan dasar kehidupan manusia. Alkitab menjadi mujizat terbesar dalam sejarah umat manusia. Melalui Alkitab, umat Kristiani mendapat manfaat yang tak terhingga ketika mempelajarinya. Alkitab menjadi otoritas tertinggi dan sumber kebenaran yang sejati dalam kehidupan Kristen, selain itu Alkitab juga menjadi pedoman hidup umat Kristiani yang mengatur baik secara vertikal maupun horizontal.

* **Web Semantik**

WWW (*World Wide Web)* menganggap web semantik sebagai sebuah evolusi semenjak dicetuskannya web semantik pada 2002. Web semantik adalah sekumpulan teknologi, dimana mesin tidak hanya membaca data tetapi juga dapat memahami informasi dari data itu sendiri atau biasa disebut sebagai meta data. IEEE *transactions on learning technologies* menjelaskan web semantik sebagai berikut (Agustina, et al., 2012) :

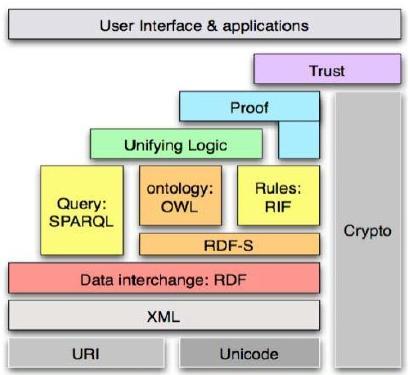
*The semantic web is conceptualized as an extension of the current web in which information is given well defined meaning betters enabling computers and people to work in cooperation* (web semantik dikonseptualisasikan sebagai perkembangan dari web saat ini di mana informasi yang diberikan didefinisikan dengan baik sehingga memungkinkan komputer dan orang-orang bekerja sama)

Dikutip dari *W3C Semantic FAQ (Frequently Asked Questions)*, belum memiliki definisi formal (Herman, 2011). Sedangkan Bapak Internet, Berners-Lee, mendefinisikan web semantik sebagai berikut :

*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* (web semantik adalah bukan bagian yang terpisah dari web, tetapi salah satu perkembangan darinya yang memiliki informasi yang didefinisikan agar bisa dimengerti dengan baik, sehingga memungkinkan komputer dan manusia dapat saling bekerja sama).

Inti dari semua aplikasi Web Semantik adalah penggunaan ontologi. Sebuah definisi ontologi yang disepakati secara umum adalah: 'Ontologi adalah eksplisit' dan spesifikasi formal dari konseptualisasi domain yang diminati' (Gruber, 1993). Definisi ini menekankan dua poin kunci: bahwa konseptualisasi bersifat formal dan karenanya memungkinkan penalaran dengan komputer; dan bahwa ontologi praktis dirancang untuk beberapa domain tertentu dari minat.

Menurut *World Wide Web Consortium (W3C),* teknologi web semantik terbagi dalam beberapa *layer* (lapisan) yang terdapat pada arsitekturnya. Model arsitektur web semantik disebut sebagai *Semantic Web Stack*. Model arsitektur pada teknologi web semantik dapat dilihat pada gambar di bawah :



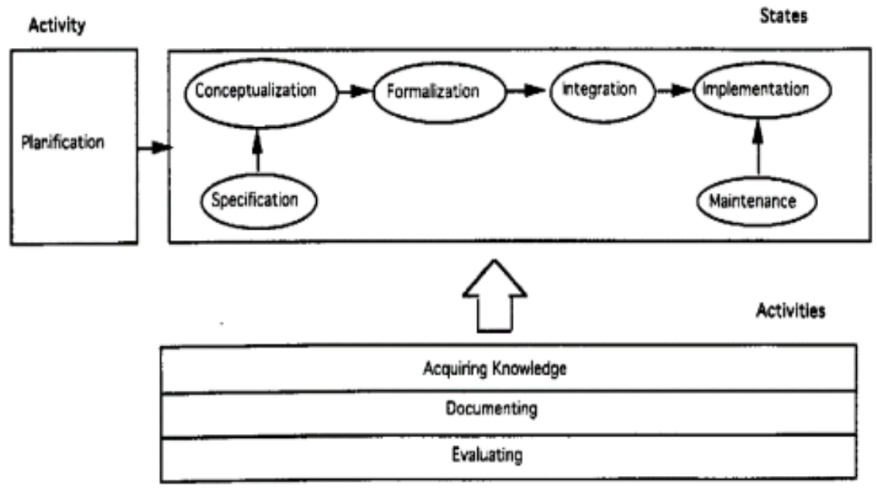
Gambar semantic web stack

* + - 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 di dalam *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. Lapisan ini bertujuan untuk mengintegrasikan definisi web semantik dengan dokumen XML lain yang scsuai standar. XML merupakan format standar untuk dokumen terstruktur dan sebagai cara paling fleksibel untuk menciptakan standar bagi format informasi dan kemudian menyediakan format tersebut beserta datanya di web,
      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*). RDF dan RDF Schema memungkinkan pengguna untuk membuat pernyataan tentang objek dan URI serta mendefinisikan kosakata yang dapat diacu dengan URI tersebut. Lapisan inilah yang menentukan tipe dari *resource* dan *link*. RDF Schema mendeklarasikan keberadaan kelas dan properti, termasuk subkelas, sub properti, *domain* dan *range*.
      4. Lapisan berikutnya yaitu OWL (*Ontology Web Language*) yang merupakan bahasa ontologi yang direkomendasikan oleh W3C. 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.
      5. Lapisan berikutnya yaitu *logic*. Lapisan *logic* merupakan aturan/*rule* dan sistem untuk melakukan penalaran pada ontologi. Lapisan *logic* menyediakan *framework* untuk menulis aksioma dari aturan dasar sistem. Dimana lapisan ini digunakan untuk meningkatkan bahasa ontologi dan memungkinkan penulisan aplikasi pengetahuan deklaratif khusus.
      6. Lapisan *Proof* mengeksekusi aturan dan mengevaluasi bersama-sama dengan mekanisme lapisan *Trust*. Lapisan *proof* juga mengeksekusi aturan dari lapisan *logic*, dimana lapisan ini melibatkan proses deduktif serta representasi *proof* dalam bahasa web dan validasi *proof*.
      7. 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.
      8. Lalu ada lapisan *Digital Signature*. Lapisan *digital signature* mendefinisikan blok dari data yang terenkripsi yang akan dimanfaatkan oleh komputer dan agen untuk memastikan apakah suatu informasi yang disediakan oleh sumber yang terpercaya serta mendeteksi adanya perubahan pada dokumen.
      9. Dan lapisan teratas yaitu user *interface* & *application* agar memungkinkan manusia dalam menggunakan aplikasi web semantik.
* **Ontologi**

Ontologi, bahasa Inggrisnya ‘*ontology’*, berasal dari bahasa Yunani, *ontos* dan *logos.* Dalam konteks ilmu komputer dan informasi, ontologi mendefinisikan seperangkat representasi primitif yang dapat digunakan untuk memodelkan domain pengetahuan atau wacana. Primitif representasional biasanya kelas (atau *set*), atribut (atau *property*), dan hubungan (atau *relationship* antara anggota *class*). Definisi primitif representasional mencakup informasi tentang makna dan batasannya pada aplikasi yang konsisten secara logis. Dalam konteks sistem basis data, ontologi dapat dilihat sebagai tingkat abstraksi model data, analog dengan model hierarkis dan relasional, tetapi dimaksudkan untuk memodelkan pengetahuan tentang individu, atributnya, dan hubungannya dengan individu lain.

Ontologi biasanya ditentukan dalam bahasa yang memungkinkan abstraksi jauh dari struktur data dan strategi implementasi; dalam praktiknya, bahasa ontologi lebih dekat dalam kekuatan ekspresif dengan logika orde pertama daripada bahasa yang digunakan untuk memodelkan *database*. Untuk alasan ini, ontologi dikatakan berada pada level “semantik”, sedangkan skema *database* adalah model data pada level “logis” atau “fisik”. Karena kemandiriannya dari model data tingkat yang lebih rendah, ontologi digunakan untuk mengintegrasikan *database* yang heterogen, memungkinkan interoperabilitas di antara sistem yang berbeda, dan menentukan antarmuka ke layanan berbasis pengetahuan yang independen. Dalam tumpukan teknologi standar Web Semantik (Berners-Lee, et al., May 2001), ontologi disebut sebagai lapisan eksplisit.

* ***Methontology***

*Methontology* adalah sebuah metode yang terstruktur dengan baik untuk membangun ontologi dari awal. *Methontology* merupakan salah satu metodologi untuk pengembangan ontologi dalam dunia *semantic web*. *Methontology* menawarkan pelaksanaan aktivitas konseptualisasi yang detail setiap tahapannya dan juga memiliki kemampuan untuk merekayasa ulang ontologi (Novianti, 2016). Methontologi merupakan salah satu metode pengembangan model ontologi yang memiliki keunggulan terkait dengan deskripsi setiap aktivitas. Selain itu, methontologi juga memiliki kemampuan yaitu ontologi yang dibangun dapat digunakan kembali untuk pengembangan sistem lebih lanjut (Fernandez & Juristo, 1997).

Gambar 2 Methontologi yang diadopsi dari Fernandez et al. (1997)

* ***Web Ontology Language (*OWL)**

*Web Ontology Language* (OWL) adalah keluarga bahasa representasi pengetahuan untuk membuat ontologi. Ontologi adalah cara formal untuk menggambarkan taksonomi dan jaringan klasifikasi, pada dasarnya mendefinisikan struktur pengetahuan untuk berbagai domain: kata benda yang mewakili kelas objek dan kata kerja yang mewakili hubungan antara objek.

Ontologi menyerupai hierarki kelas dalam pemrograman berorientasi objek tetapi ada beberapa perbedaan penting. Hirarki kelas dimaksudkan untuk mewakili struktur yang digunakan dalam kode sumber yang berkembang cukup lambat (mungkin dengan revisi bulanan) sedangkan ontologi dimaksudkan untuk mewakili informasi di Internet dan diharapkan berkembang hampir secara konstan. Demikian pula, ontologi biasanya jauh lebih fleksibel karena dimaksudkan untuk mewakili informasi di Internet yang berasal dari segala macam sumber data yang heterogen. Hirarki kelas di sisi lain cenderung cukup statis dan bergantung pada sumber data yang jauh lebih beragam dan lebih terstruktur seperti database perusahaan (Knublauch, et al., 2206).

Bahasa OWL dicirikan oleh semantik formal. Mereka dibangun di atas standar XML *World Wide Web Consortium* (W3C) untuk objek yang disebut *Resource Description Framework* (RDF) (W3C, 2012). OWL dan RDF telah menarik minat akademis, medis, dan komersial yang signifikan.

Pada bulan Oktober 2007, kelompok kerja W3C baru mulai memperluas OWL dengan beberapa fitur baru seperti yang diusulkan dalam pengajuan anggota OWL 1.1. W3C mengumumkan versi baru OWL pada 27 Oktober 2009. Versi baru ini, yang disebut OWL 2, segera menemukan jalannya ke editor semantik seperti Protégé dan penalaran semantik seperti Pellet, RacerPro, FaCT++ dan HermiT.

Keluarga OWL berisi banyak spesies, serialisasi, sintaksis, dan spesifikasi dengan nama yang mirip. OWL dan OWL2 digunakan untuk merujuk pada spesifikasi 2004 dan 2009, masing-masing. Nama spesies lengkap akan digunakan, termasuk versi spesifikasi (misalnya, OWL2 EL). Saat merujuk secara lebih umum, OWL *Family* akan digunakan (W3C, 2012) .

* **RDF (Resource Description Framework)**

*Resource Description Framewor*k (RDF) adalah standar *World Wide Web Consortium* (W3C) yang awalnya dirancang sebagai model data untuk metadata. Ini telah digunakan sebagai metode umum untuk deskripsi dan pertukaran data grafik. RDF menyediakan berbagai notasi sintaks dan format serialisasi data dengan *Turtle* (*Terse RDF Triple Language*) saat ini menjadi notasi yang paling banyak digunakan.

RDF adalah graf berarah yang terdiri dari tiga pernyataan. Pernyataan grafik RDF diwakili oleh: 1) simpul untuk subjek, 2) busur yang bergerak dari subjek ke objek untuk predikat dan 3) simpul untuk objek. Masing-masing dari tiga bagian pernyataan dapat diidentifikasi oleh URI. Sebuah objek juga bisa menjadi nilai literal. Model data yang sederhana dan fleksibel ini memiliki banyak kekuatan ekspresif untuk mewakili situasi yang kompleks, hubungan, dan hal-hal menarik lainnya, sementara juga abstrak dengan tepat.

RDF diadopsi sebagai rekomendasi W3C pada tahun 1999. Spesifikasi RDF 1.0 diterbitkan pada tahun 2004, spesifikasi RDF 1.1 pada tahun 2014. SPARQL adalah bahasa kueri standar untuk grafik RDF. RDFS, OWL dan SHACL adalah bahasa ontologi yang digunakan untuk mendeskripsikan data RDF.

Model data RDF mirip dengan pendekatan pemodelan konseptual klasik (seperti hubungan entitas atau diagram kelas). Ini didasarkan pada gagasan untuk membuat pernyataan tentang sumber daya (khususnya sumber daya web) dalam ekspresi bentuk subjek-predikat-objek, yang dikenal sebagai *triple*. Subjek menunjukkan sumber daya, dan predikat menunjukkan sifat atau aspek sumber daya, dan mengungkapkan hubungan antara subjek dan objek.

Misalnya, salah satu cara untuk merepresentasikan gagasan "Langit memiliki warna biru" di RDF adalah sebagai *triple*: subjek yang menunjukkan "langit", predikat yang menunjukkan "memiliki warna", dan objek yang menunjukkan "biru". Oleh karena itu, RDF menggunakan subjek alih-alih objek (atau entitas) berbeda dengan pendekatan khas model entitas-atribut-nilai dalam desain berorientasi objek: entitas (langit), atribut (warna), dan nilai (biru).

RDF adalah model abstrak dengan beberapa format serialisasi (pada dasarnya format *file* khusus), selain itu pengkodean khusus untuk sumber daya atau tiga kali lipat dapat bervariasi dari format ke format.

* ***Resource Description Framework Schema (RDFS)***

Skema RDF (Skema Kerangka Deskripsi Sumber Daya, disingkat RDFS, RDF(S), RDF-S, atau RDF/S) adalah sekumpulan kelas dengan properti tertentu menggunakan model data representasi pengetahuan yang dapat diperluas RDF, menyediakan elemen dasar untuk deskripsi dari ontologi. Ini menggunakan berbagai bentuk kosakata RDF, yang dimaksudkan untuk menyusun sumber daya RDF. RDF dan RDFS dapat disimpan di *triplestore*, kemudian seseorang dapat mengekstrak beberapa pengetahuan dari mereka menggunakan bahasa kueri, seperti SPARQL.

Versi pertama diterbitkan oleh *World-Wide Web Consortium* (W3C) pada April 1998, dan rekomendasi W3C terakhir dirilis pada Februari 2014. Banyak komponen RDFS disertakan dalam *Ontology Web Language* (OWL) yang lebih ekspresif (Brickley, et al., 1998).

Konstruksi RDFS adalah *class* RDFS, *properties* terkait dan *utility properties* yang dibangun di atas kosakata RDF (Lagoze & Carl, 2008).

**Classes**

**rdfs : Resource**

Mewakili *class* dari segalanya. Semua hal yang dijelaskan oleh RDF adalah sumber daya.

**Rdfs : Class**

Sebuah rdfs:Class mendeklarasikan sumber daya sebagai kelas untuk sumber daya lainnya.

Contoh khas dari rdfs:Class adalah foaf:Person dalam *Friend of a Friend* (FOAF) kosakata. *Instance* foaf:Person adalah sumber daya yang ditautkan ke kelas foaf:Person yang menggunakan properti rdf:type, seperti dalam ekspresi formal berikut dari kalimat bahasa alami: *'John is a Person'*.

**Ex:John rdf:type foaf:Person**

Definisi rdfs:Class adalah rekursif: rdfs:Class adalah *class* dari *classes*, dan karenanya merupakan *instance* dari dirinya sendiri.

**rdfs:Class rdf:type rdfs:Class**

*Class* lain yang dijelaskan oleh spesifikasi RDF dan RDFS adalah:

**rdfs:Literal**

nilai literal seperti *string* dan *integer*. Nilai *property* seperti *string* tekstual adalah contoh RDF *Literal*. *Literal* bisa polos atau diketik.

**rdfs:Datatype**

*class* dari *datatypes*. rdfs:Datatype adalah turunan dan *subclass* dari rdfs:Class. Setiap *instance* dari rdfs:Datatype adalah *subclass* dari rdfs:Literal.

**rdf:XMLLiteral**

*class* dari nilai *literal* XML. rdf:XMLLiteral adalah turunan dari rdfs:Datatype (dan dengan demikian merupakan *subclass* dari rdfs:Literal).

**rdf:Property**

*class* dari *property*

***Properties***

*Properties* adalah *instance* dari *class* rdf:Property dan menggambarkan hubungan antara sumber daya subjek dan sumber daya objek. Ketika digunakan sebagai *property* seperti itu adalah predikat

**rdfs:domain**

rdfs:domain dari rdf:Property mendeklarasikan *class* subjek dalam *triple* yang predikatnya adalah properti itu.

**rdfs:range**

rdfs:range dari rdf:Property mendeklarasikan kelas atau tipe data objek dalam *triple* yang predikatnya adalah properti itu.

Misalnya, deklarasi berikut digunakan untuk menyatakan bahwa properti

ex:employer menghubungkan subjek, yang bertipe foaf:Person, ke objek, yang bertipe foaf:Organization:

**ex:employer rdfs:domain foaf:Person**

**ex:employer rdfs:range foaf:Organization**

Misalnya dari dua deklarasi sebelumnya dari *triple :*

**ex:John ex:employer ex:CompanyX**

Dapat disimpulkan bahwa ex:John adalah foaf:Person, dan ex:CompanyX adalah foaf:Organization.

**rdf:type**

*Property* yang digunakan untuk menyatakan bahwa sumber daya adalah *instanve* dari *class*. QName yang diterima secara umum untuk *property* ini adalah "a" (DuCharme & Bob, 2011).

**rdfs:subClassOf**

Memungkinkan deklarasi hierarki kelas (Schreiber, et al., 2014).

Misalnya, berikut ini menyatakan bahwa ‘*Every Person is an Agent’*:

**foaf:Person rdfs:subClassOf foaf:Agent**

Hirarki kelas mendukung pewarisan *domain* dan *range* properti dari *class* ke *subclass*.

**rdfs:subPropertyOf**

turunan dari rdf:Properti yang digunakan untuk menyatakan bahwa semua sumber daya yang terkait oleh satu properti juga terkait dengan properti lainnya.

**rdfs: label**

turunan dari rdf:Properti yang dapat digunakan untuk menyediakan versi nama sumber daya yang dapat dibaca manusia.

**rdfs:komentar**

sebuah *instance* dari rdf:Properti yang dapat digunakan untuk memberikan deskripsi sumber daya yang dapat dibaca manusia.

***Utility Properties***

**rdfs:lihat Juga**

Contoh rdf:Properti yang digunakan untuk menunjukkan sumber daya yang mungkin memberikan informasi tambahan tentang sumber daya subjek.

rdfs:didefinisikanoleh

Sebuah *instance* dari rdf:Properti yang digunakan untuk menunjukkan sumber daya yang mendefinisikan sumber daya subjek. Properti ini dapat digunakan untuk menunjukkan kosakata RDF di mana sumber daya dijelaskan.

* ***Apache Jena Fuseki***

*Apache Jena* adalah kerangka kerja Web Semantik *open source* untuk Java. Ini menyediakan API untuk mengekstrak data dari dan menulis ke grafik RDF. Grafik direpresentasikan sebagai "*model*" abstrak. Sebuah model dapat bersumber dari data dari *file*, *database*, URL atau kombinasi dari semuanya. Sebuah model juga dapat ditanyakan melalui SPARQL 1.1.

*Jena* mirip dengan RDF4J (sebelumnya *OpenRDF Sesame*); meskipun, tidak seperti RDF4J, *Jena* menyediakan dukungan untuk OWL (*Web Ontology Language*). Kerangka kerja memiliki berbagai alasan internal dan alasan Pellet (penalaran Java OWL-DL open source) dapat diatur untuk bekerja di Jena.

Jena mendukung serialisasi grafik RDF untuk:

* + *Relational database*
  + *RDF/XML*
  + *Turtle*
  + *TriG*
  + *Notation 3*
  + *JSON-LD*

*Apache Jena Fuseki* bertindak sebagai server untuk mengeksekusi SPARQL dalam mengolah data RDF. Pada dasarnya SPARQL sama seperti SQL, yakni bahasa query data. Perbedaannya adalah SQL merupakan *command* untuk PHP, sedangkan SPARQL untuk RDF. Fuseki juga memungkinkan web mengakses *file* ontologi untuk proses *upload, update,* dan *query* di dalam *browser*, juga melihat hasil untuk proses yang terjadi tanpa mengubah file asal. Berbeda dengan Protégé, Fuseki merupakan *tools* untuk membina ontologi serta tidak menjalankan query di *browser*, melainkan di *local*. Oleh karena itu, *Fuseki* perlu di unduh untuk menjalankan proses secara protokol HTTP.

* **SPARQL**

SPARQL (singkatan dari *SPARQL Protocol and RDF Query Language* (Beckett, 2011)) adalah bahasa kueri RDF—yaitu, bahasa kueri semantik untuk *database*—mampu mengambil dan memanipulasi data yang disimpan dalam format *Resource Description Framework* (RDF) (Rapoza, 2006). SPARQL dibuat standar oleh *RDF* *Data Access Working Group* (DAWG) dari *World Wide Web Consortium*, dan diakui sebagai salah satu teknologi kunci dari web semantik. Pada 15 Januari 2008, SPARQL 1.0 diakui oleh W3C sebagai rekomendasi resmi, dan SPARQL 1.1 pada Maret 2013 .

SPARQL memungkinkan kueri terdiri dari pola *triple*, konjungsi, disjungsi, dan pola opsional. Implementasi untuk beberapa bahasa pemrograman ada. Ada alat yang memungkinkan seseorang untuk terhubung dan semi-otomatis membuat kueri SPARQL untuk *endpoint* SPARQL, misalnya ViziQuer. Selain itu, ada alat untuk menerjemahkan *query* SPARQL ke bahasa ku *query* eri lain, misalnya ke SQL dan ke XQuery.

Terdapat tiga klausa yang digunakan dalam query SPARQL, yaitu sebagai berikut: .

1. PREFIX

Pernyataan 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*.

2. SELECT

Pernyataan SELECT merupakan klausa yang mendefinisikan daftar dari variabel-variabel yang akan dikembalikan sebagai hasil dari eksekusi query.

3. WHERE

Pernyataan WHERE mendefinisikan *triple pattern* yang harus dimiliki oleh setiap hasil *query* yang valid. Seluruh pola yang merepresentasikan sebuah kalimat RDF harus sesuai dengan RDF *triple* yang terdiri dari subjek, predikat, dan objek.

* **Protégé**

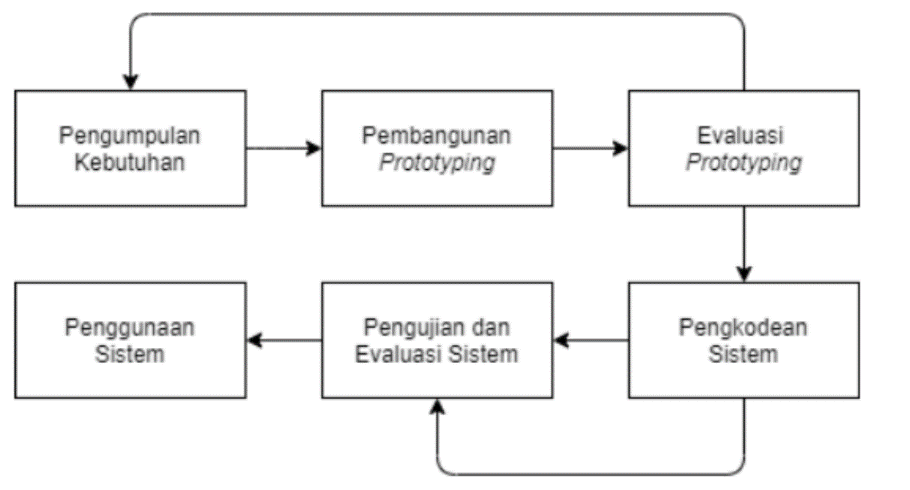
Protégé adalah editor ontologi *open source* gratis dan sistem manajemen pengetahuan. Alat meta Protégé pertama kali dibuat oleh Mark Musen pada tahun 1987 dan sejak itu dikembangkan oleh tim di Universitas Stanford (Gennari, et al., 2003). Perangkat lunak ini adalah editor ontologi paling populer dan banyak digunakan di dunia (Musen, 2015).

Protégé menyediakan antarmuka pengguna grafis untuk mendefinisikan ontologi. Ini juga termasuk pengklasifikasi deduktif untuk memvalidasi bahwa model konsisten dan untuk menyimpulkan informasi baru berdasarkan analisis ontologi. Seperti Eclipse, Protégé adalah kerangka kerja di mana berbagai proyek lain menyarankan *plugin* (Gašević, et al., 2009).

Protégé merupakan perangkat lunak yang bertujuan sebagai alat untuk membantu untuk mengembangkan 22 ontologi yang didasarkan pada *Knowledge Base System. Protégé* dapat digunakan untuk membuat, mengedit, dan menyimpan ontologi (Fahrurozzi & SN, 2017). Protégé memiliki format penyimpanan seperti OWL,RDF, XML.

* ***Prototyping***

*Prototyping* adalah metode pengembangan perangkat lunak yang akan menghasilkan *prototype* sistem sebagai perantara pengembang dan pengguna agar dapat berinteraksi dalam proses kegiatan pengembangan sistem. Agar proses pembuatan *prototype* berhasil maka pada tahap awal perlu didefinisikan aturan- aturan, yaitu pengembang dan penguna harus satu pemahaman bahwa *prototype* dibangun untuk mendefinisikan kebutuhan awal. *Prototype* akan dihilangkan atau ditambahkan pada bagiannya sehingga sesuai dengan perencanaan dan analisis yang dilakukan oleh pengembang sampai dengan uji coba dilakukan secara simultan seiiring dengan proses pengembangan. Metode *Prototyping* bertujuan untuk mengumpulkan informasi dari pengguna sehingga pengguna dapat berinteraksi dengan model *prototype* yang dikembangkan, sebab *prototype* menggambarkan versi awal dari sistem untuk kelanjutan sistem sesungguhnya yang lebih besar.

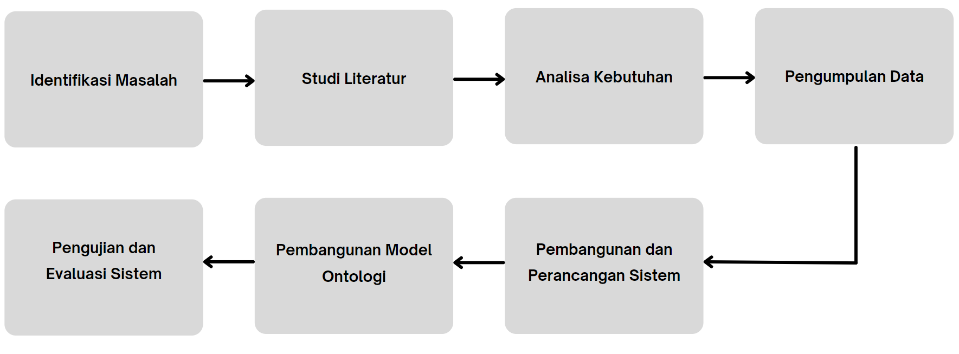


* ***Regression Testing***

**METODE PENELITIAN**

Pada bab ini akan dijelaskan mengenai langkah-langkah dan tahap-tahap perencanaan, perancangan, dan pengembangan sistem pencarian ayat Alkitab digital berdasarkan penulis dengan pendekatan ontologi.

* **Alur Penelitian**

Penelitian ini membahas mengenai sistem pencarian ayat Alkitab berdasarkan penulis dengan menggunakan metode *methontology.* Pembangunan sistem ini akan melalui beberapa tahapan. Tahapan-tahapan tersebut dapat dilihat pada diagram di bawah ini :

Gambar alur penelitian

Langkah pertama yang akan dilakukan dalam penelitian ini adalah mengidentifikasi permasalahan untuk mengetahui topik apa yang akan diangkat dalam penelitian ini dan pada penelitian ini topik yang diangkat adalah sistem pencarian pada Alkitab. Kemudian terdapat studi literatur, pada langkah ini, penulis melakukan studi dari berbagai sumber dan referensi yang berkaitan dengan topik yang diangkat. Kemudian terdapat langkah untuk menganalisa kebutuhan sistem. Tujuan langkah analisa kebutuhan yaitu untuk menganalisa kebutuhan apa saja yang diperlukan dalam proses pembangunan sistem pencarian pada Alkitab. Kemudian terdapat langkah pengumpulan data. Langkah ini merupakan langkah penting dalam penelitian ini karena data yang dikumpulkan nantinya akan digunakan sebagai dasar sistem. Kemudian terdapat langkah pembangunan model ontologi. Sesuai alur penelitian, pada langkah ini, data yang sudah dikumpulkan sebelumnya akan diolah menjadi model ontologi. Kemudian langkah selanjutnya adalah mulai membangun sistem berbasis web semantik dengan memanfaatkan model ontologi yang sudah dibuat sebelumnya. Pembangunan sistem akan menggunakan metode *prototyping.* Terakhir terdapat langkah pengujian dan evaluasi sistem. Pada langkah ini, sistem yang sudah dibangun akan melalui tahap pengujian sekaligus dievaluasi apakah susah sesuai dengan *acceptance criteria*. Metode pengujian akan menggunakan metode *black box* yaitu *regression testing.*

1. **Identifikasi Masalah**

Langkah pertama yang harus dilakukan dalam proses penelitian ini adalah mengidentifikasi permasalahan yang diangkat. Langkah ini merupakan yang paling penting dalam proses penelitian ini karena proses penelitian berjalan berdasarkan topik permasalahan yang diangkat. Permasalahan dalam penelitian ini adalah minimnya informasi pada Alkitab terkait penulis dari setiap kitab yang terdapat dalam Alkitab. Oleh karena hal tersebut, pembaca menjadi sulit mengetahui siapa tokoh/penulis yang berada di balik kisah-kisah dalam Alkitab

1. **Studi Literatur**

Langkah kedua yang harus dilakukan dalam proses penelitian ini adalah melakukan studi literatur. Langkah ini dilakukan dengan cara melakukan peninjauan terhadap berbagai sumber literatur baik itu artikel-artikel ilmiah dan penelitian-penelitian sebelumnya yang berkaitan dengan ontologi, web semantik, dan Alkitab.

1. **Analisa Kebutuhan**

Langkah selanjutnya yang akan dilakukan dalam penelitian ini adalah menganalisa kebutuhan sistem untuk mengetahui hal-hal apa saja yang dibutuhkan untuk membangun sistem ini. Kebutuhan-kebutuhan tersebut meliputi kebutuhan fungsional dan non-fungsional.

* + - **Analisa Kebutuhan Fungsional**

Analisa kebutuhan fungsional berkaitan dengan *user requirement* dari sistem itu sendiri, yaitu sebagai berikut :

* + - * Sistem yang dibangun memungkinkan pengguna untuk melakukan pencarian isi Alkitab.
      * Sistem yang dibangun memungkinkan pengguna untuk melakukan pencarian secara spesifik mengenai penuli dari setiap kitab dalam Alkitab.
    - **Analisa Kebutuhan Non-Fungsional**

Analisa kebutuhan non-fungsional berkaitan dengan *tools-tools* pendukung yang akan digunakan selama proses penelitian, yaitu sebagai berikut :

* *Hardware Requirement :*

*Hardware* yang dibutuhkan dalam penelitian ini yaitu PC baik itu laptop ataupun komputer yang memiliki standar mampu membangun program berbasis web dan sebagai wadah implementasi program

* *Software Requirement :*

*Software* yang dibutuhkan dalam penelitian ini yaitu Protégé untuk pembangunan ontologi, SPARQL sebagai *query* dari model ontologi, Vscode sebagai *text editor* dalam pembangunan program berbasis web, dan Apache Jena Fuseki sebagai penghubung antara ontologi semantik dengan web.

1. **Pengumpulan Data**

Proses penelitian ini akan menggunakan data sekunder, artinya data yang diperoleh atau dikumpulkan berasal dari orang lain yang sudah melakukan penelitian dan juga dari sumber-sumber yang sudah ada berkaitan dengan penelitian ini. Data-data ini nantinya akan digunakan untuk menjadi pendukung informasi yang telah diperoleh dari penelitian terdahulu, buku-buku, literatur-literatur, pustaka-pustaka, dan lainnya yang berkaitan dengan penelitian ini. Pada langkah ini, data dikumpulkan dari berbagai sumber baik internet meliputi artikel-artikel maupun jurnal, serta data Alkitab yang akan digunakan untuk membangun sistem ini diambil dari *website (kaggle.com).*

Setelah semua data terkumpul, data tersebut akan diolah dengan aplikasi Protégé untuk membuat model ontologi-nya dengan metode *methontology*. Setelah data diubah menjadi sebuah modek ontologi, model tersebut akan diuji dengan menggunakan SPARQL Query pada aplikasi Protégé melalui pemanggilan data dengan cara *querying* untuk mengetahui ketepatan dari model ontologi yang dibangun. Tepat atau tidaknya model ontologi akan bergantung pada *output* yang dihasilkan dari *query* tersebut. Jika model ontologi sudah dianggap benar dan tepat, proses dilanjutkan dengan membangun sistem.

<https://alkitab.sabda.org/resource.php?topic=553&res=jpz>

<https://www.gotquestions.org/Indonesia/apa-itu-Alkitab.html>

<https://www.researchgate.net/publication/352405930_Alkitab_Sebagai_Buku_Pegangan_Orang_Kristen_Ketidakbersalahan_Alkitab>

<https://merdinitbani.wordpress.com/2013/04/03/struktur-dan-pembagian-alkitab/>

<https://archive.org/details/bibleveryshortin0000rich>

<https://onesearch.id/Record/IOS4680.JATIM000000000041437>

<https://www.academia.edu/35857414/TERBENTUKNYA_ALKITAB_Makalah_Disusun_Oleh>

<https://arcusgpib.com/mau-tahu-siapa-siapa-penulis-alkitab-dan-kapan-ditulis-simak-ini/>

<https://labs.sabda.org/Statistik_Alkitab>

<http://p2k.itbu.ac.id/ind/1-3070-2950/Alkitab_27406_itbu_ensiklopedia-dunia-q-itbu.html#cite_ref-23>

IMPLEMENTASI TEKNOLOGI WEB SEMANTIK DALAM APLIKASI PENCARIAN KATALOG ONLINE PERPUSTAKAAN UNIVERSITAS BINA DARMA Andri Dosen Universitas Bina Darma Jalan Jenderal Ahmad Yani No.12 Palembang

https://www.researchgate.net/publication/320144427\_STUDI\_TENTANG\_PEMODELAN\_ONTOLOGI\_WEB\_SEMANTIK\_DAN\_PROSPEK\_PENERAPAN\_PADA\_BIBLIOGRAFI\_ARTIKEL\_JURNAL\_ILMIAH

<https://www.researchgate.net/publication/344853287_Thesis_by_Building_Formal_Ontologies_for_Theology_and_Systems_of_belief>

<https://old.datahub.io/sr_Latn/dataset/semantic_bible>

<https://www.researchgate.net/publication/329523074_Ontology-Based_Approach_for_Knowledge_Retrieval_in_Al-Quran_Holy_Book>

<https://archive.org/details/bibleveryshortin0000rich>

<https://archive.org/details/biblehistorymaki0000mill>

["Chapters and Verses: Who Needs Them?,"](http://www.biblestudymagazine.com/interactive/chaptersverses/) Christopher R. Smith, Bible Study Magazine (July-Aug 2009): 46-47.

<http://e-journals.unmul.ac.id/index.php/SAKTI/article/view/254>

<https://www.researchgate.net/publication/200026173_Semantic_Web_Technologies_Trends_and_Research_in_Ontology-based_Systems>

[1] Berners-Lee, T., Hendler, J. and Lassila, O.  [**The Semantic Web**](http://www.w3.org/2001/sw/), Scientific American, May 2001.  Also <http://www.w3.org/2001/sw/>

<https://repositori.usu.ac.id/handle/123456789/5289>

<https://www.w3.org/RDF/#:~:text=Resource%20Description%20Framework%20(RDF)&text=RDF%20is%20a%20standard%20model,data%20consumers%20to%20be%20changed>.

Knublauch, Holger; Oberle, Daniel; Tetlow, Phil; Wallace, Evan (March 9, 2006). ["A Semantic Web Primer for Object-Oriented Software Developers"](https://www.w3.org/2001/sw/BestPractices/SE/ODSD/). [*W3C*](https://en.wikipedia.org/wiki/W3C). Retrieved November 19, 2017.

[*"OWL 2 Web Ontology Language Document Overview (Second Edition)"*](https://www.w3.org/TR/owl2-overview/). [*W3C*](https://en.wikipedia.org/wiki/W3C). 11 December 2012.

 Sirin, E.; Parsia, B.; Grau, B. C.; Kalyanpur, A.; Katz, Y. (2007). [*"Pellet: A practical OWL-DL reasoner"*](https://web.archive.org/web/20070627202215/http:/pellet.owldl.com/papers/sirin05pellet.pdf) *(PDF)*. Web Semantics: Science, Services and Agents on the World Wide Web. **5** (2): 51–53. [*doi*](https://en.wikipedia.org/wiki/Doi_(identifier)):[*10.1016/j.websem.2007.03.004*](https://doi.org/10.1016%2Fj.websem.2007.03.004). Archived from [*the original*](http://pellet.owldl.com/papers/sirin05pellet.pdf) *(PDF)* on 2007-06-27.

 Beckett, Dave (6 October 2011). [*"What does SPARQL stand for?"*](http://lists.w3.org/Archives/Public/semantic-web/2011Oct/0041.html). [semantic-web@w3.org](mailto:semantic-web@w3.org).

Jim Rapoza (2 May 2006). ["SPARQL Will Make the Web Shine"](http://www.eweek.com/development/sparql-will-make-the-web-shine). *[eWeek](https://en.wikipedia.org/wiki/EWeek" \o "EWeek)*. Retrieved 17 January 2007.

Segaran, Toby; Evans, Colin; Taylor, Jamie (2009). Programming the Semantic Web. O’Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472. p. 84. [*ISBN*](https://en.wikipedia.org/wiki/ISBN_(identifier)) [*978-0-596-15381-6*](https://en.wikipedia.org/wiki/Special:BookSources/978-0-596-15381-6).

Gennari, John H.; Musen, Mark A.; Fergerson, Ray W.; Grosso, William E.; Crubézy, Monica; Eriksson, Henrik; Noy, Natalya F.; Tu, Samson W. (2003-01-01). ["The evolution of Protégé: an environment for knowledge-based systems development"](https://doi.org/10.1016/S1071-5819(02)00127-1). *International Journal of Human-Computer Studies*. **58** (1): 89–123. [doi](https://en.wikipedia.org/wiki/Doi_(identifier)):[10.1016/S1071-5819(02)00127-1](https://doi.org/10.1016%2FS1071-5819%2802%2900127-1). [ISSN](https://en.wikipedia.org/wiki/ISSN_(identifier)) [1071-5819](https://www.worldcat.org/issn/1071-5819).

Musen, Mark A. (2015-06-16). ["The protégé project"](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4883684). *AI Matters*. **1** (4): 4–12. [doi](https://en.wikipedia.org/wiki/Doi_(identifier)):[10.1145/2757001.2757003](https://doi.org/10.1145%2F2757001.2757003). [ISSN](https://en.wikipedia.org/wiki/ISSN_(identifier)) [2372-3483](https://www.worldcat.org/issn/2372-3483). [PMC](https://en.wikipedia.org/wiki/PMC_(identifier)) [4883684](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4883684). [PMID](https://en.wikipedia.org/wiki/PMID_(identifier)) [27239556](https://pubmed.ncbi.nlm.nih.gov/27239556).

 Dragan Gašević; Dragan Djurić; Vladan Devedžić (2009). [*Model Driven Engineering and Ontology Development*](https://books.google.com/books?id=s-9yu7ubSykC&pg=PA194) (2nd ed.). Springer. p. 194. [*ISBN*](https://en.wikipedia.org/wiki/ISBN_(identifier)) [*978-3-642-00282-3*](https://en.wikipedia.org/wiki/Special:BookSources/978-3-642-00282-3).

Smith MF *Software Prototyping: Adoption, Practice and Management*. McGraw-Hill, London (1991).