

BAB III

METODE PENELITIAN

1.1. Metode Pengembangan Sistem Informasi Layanan *Home care*

Perancangan basis pengetahuan (*knowledge base*) berbasis semantik pada penelitian ini menggunakan metode yang dikembangkan oleh Noy dan McGuinness, atau yang lebih dikenal dengan sebutan Metode *Stanford 101*. Metode ini dipilih karena pendekatannya yang bersifat iteratif (berulang) dan sangat terstruktur, sehingga cocok untuk memodelkan domain pengetahuan dari sistem *home cleaning service* yang memiliki banyak entitas saling terkait.

Penerapan tahapan *Metode Stanford 101* pada penelitian ini dijabarkan ke dalam tujuh langkah berikut:

1. Menentukan Domain dan Ruang Lingkup (*Determine Domain and Scope*)

Langkah pertama adalah menetapkan batasan yang jelas agar yang dibangun tidak terlalu melebar dan tetap fokus pada tujuan sistem. Domain penelitian ini adalah layanan kebersihan rumah (*home cleaning service*). Ruang lingkup pengetahuan yang dimodelkan dibatasi pada entitas Pelanggan, Pekerja (penyedia jasa), Jenis Layanan (seperti *Deep Cleaning*, *Hydro Cleaning*), Alat Kebersihan, Bahan Pembersih, dan Area/Lokasi pengerjaan.

2. Mempertimbangkan Penggunaan Kembali yang Ada (*Consider Reusing Existing*)

Untuk menjaga standar baku *Semantic Web*, penelitian ini sebisa mungkin menggunakan kembali (*reuse*) beberapa kosakata (vocabulary) yang telah diakui secara global. Misalnya, penggunaan *FOAF (Friend of a Friend)* untuk merepresentasikan data dasar manusia seperti nama pekerja dan pelanggan.

3. Mendaftar Istilah-Istilah Penting (*Enumerate Important Terms*)

Tahap ini melibatkan inventarisasi kata kunci atau jargon operasional yang sering digunakan dalam proses bisnis *home cleaning*. Istilah yang berhasil dihimpun antara lain: Pelanggan, Pekerja, Layanan, Sapu, *Vacuum Cleaner*, Cairan Disinfektan, Ruang Tamu, Kamar Mandi, Tarif Layanan, dan Wilayah Operasional.

4. Mendefinisikan Kelas dan Hierarki Kelas (*Define Classes and Class Hierarchy*)

Istilah-istilah penting tersebut kemudian dikonstruksi menjadi struktur hierarki kelas (taksonomi) menggunakan pendekatan *top-down*. Kelas utama (Superclass) yang dibentuk

meliputi Layanan, Alat, Bahan, Area, dan Aktor (yang kemudian memiliki *Subclass* Pekerja dan Pelanggan).

5. Mendefinisikan Properti Kelas (*Define Properties of Classes - Slots*)

Setelah kelas terbentuk, ditentukan relasi (properti) yang menghubungkan antar kelas. Properti dalam OWL dibagi menjadi dua:

Object Property: Relasi yang menghubungkan dua individu dari kelas yang berbeda. Contoh: Kelas Pekerja dihubungkan ke kelas Layanan melalui relasi menguasaiKeahlian.

Data Property: Relasi yang menghubungkan individu dengan nilai tipe data primitif. Contoh: Kelas Layanan dihubungkan dengan nilai harga melalui relasi memilikiTarif (bertipe *integer*).

6. Mendefinisikan Batasan Properti (*Define Facets of Slots*)

Tahap ini menetapkan aturan batasan (restriksi) pada setiap properti, seperti nilai kardinalitas. Contohnya, properti memilikiTarif dibatasi hanya boleh memiliki tepat satu nilai (kardinalitas tunggal) bertipe angka.

7. Membuat Individu (*Create Instances*)

Langkah terakhir adalah melakukan instansiasi, yaitu memasukkan data konkret ke dalam kelas-kelas yang telah dibuat. Sebagai contoh, membuat individu bernama Layanan_CuciSofa sebagai bagian dari kelas Layanan, dan Budi_Santoso sebagai bagian dari kelas Pekerja.

Penelitian dilaksanakan melalui serangkaian tahapan yang disusun sistematis untuk memastikan proses perancangan hingga implementasi layanan *home care* pelanggan berjalan sesuai tujuan. Tahapan penelitian meliputi: analisis kebutuhan untuk menggali permasalahan aktual dan menentukan ruang lingkup domain layanan (jenis layanan, cakupan pekerjaan, area/ruang, peralatan/bahan, estimasi durasi, harga, serta pemesanan); perancangan yang menerjemahkan kebutuhan menjadi model konseptual berupa kelas, relasi, dan atribut (misalnya konsep Layanan, Area/Ruang, Paket Layanan, Alat/Bahan, dan Pemesanan) agar informasi layanan dapat direpresentasikan secara konsisten.

penyusunan *competency questions* sebagai daftar pertanyaan kebutuhan yang harus dapat dijawab oleh; implementasi menggunakan OWL pada model data *RDF* dengan bantuan *tool* pengembangan (misalnya *Protégé*); serta penyusunan kueri *SPARQL* untuk menguji kemampuan dalam menjawab kebutuhan informasi layanan pelanggan. Tahapan berikutnya adalah pengujian dan evaluasi, meliputi pemeriksaan konsistensi (*reasoning*), ketercakupan *competency questions* melalui hasil kueri, dan perbaikan iteratif jika ditemukan kekurangan konsep atau relasi. Penelitian diakhiri dengan penyusunan laporan/luaran berupa dokumentasi, *artefak* pemodelan menggunakan *UML*, serta hasil implementasi prototipe sistem informasi layanan *home care*

1.1.1 Analisis Kebutuhan & Ruang Lingkup Domain

Pada tahap ini, penelitian berfokus pada pengumpulan data mengenai berbagai jenis layanan *home care* yang tersedia, seperti *general cleaning*, *deep cleaning*, dan cuci karpet. Informasi terkait estimasi durasi, harga, serta detail peralatan yang dibutuhkan juga dikumpulkan untuk mendukung pengembangan yang sesuai dengan kebutuhan pelanggan.

1.1.2 Competency Questions (CQ)

Dalam rekayasa, *Competency Questions (CQ)* adalah indikator utama untuk menguji apakah model pengetahuan yang dibangun sudah *valid* atau belum. *CQ* merupakan sekumpulan pertanyaan dalam bahasa alami yang merepresentasikan kebutuhan pengguna sistem. dianggap berhasil apabila seluruh pertanyaan dalam *CQ* ini dapat dijawab oleh sistem menggunakan kueri *SPARQL*.

Penyusunan *Competency Questions (CQ)* dilakukan pada tahap awal rekayasa untuk membatasi ruang lingkup agar tidak meluas ke domain pengetahuan yang tidak relevan. Karakteristik utama dari *CQ* yang baik adalah kemampuannya dalam mengekstraksi informasi yang bersifat implisit menjadi eksplisit melalui relasi antar-kelas. Melalui daftar pertanyaan ini, pengembang dapat menentukan *concept*, *object property*, *data property*, serta *constraints* apa saja yang wajib dideklarasikan di dalam perangkat lunak *Protégé*.

Daftar pertanyaan kompetensi yang menjadi landasan perancangan kueri pada aplikasi *home cleaning* ini disajikan pada Tabel 3.1

Tabel 3. 1 competency question

No	Kode CQ	Pertanyaan Kompetensi (Bahasa Alami)	Keterlibatan Kelas	Luaran yang Diharapkan (Output)
1	CQ-01	Apa saja jenis layanan kebersihan yang ditawarkan di dalam sistem?	Layanan	Daftar nama seluruh layanan kebersihan.
2	CQ-02	Alat kebersihan apa saja yang harus disiapkan untuk layanan <i>Deep Cleaning</i> ?	Layanan, Alat	Daftar nama alat (<i>Vacuum Cleaner</i> , lap, dll).
3	CQ-03	Siapa saja pekerja yang berstatus aktif dan memiliki keahlian untuk <i>Hydro Cleaning</i> ?	Pekerja, Layanan	Daftar nama dan profil pekerja terkait.
4	CQ-04	Pekerja mana saja yang bersedia melayani pelanggan di area Surabaya Timur?	Pekerja, Area	Daftar nama pekerja di area tersebut.
5	CQ-05	Berapa tarif yang harus dibayarkan untuk layanan pembersihan Kamar Mandi?	Layanan, Area	Informasi harga atau tarif layanan.

1.1.3 Konseptualisasi Pengembangan Sistem Informasi Layanan *Home Care* Pelanggan

Tahap konseptualisasi melibatkan pembuatan struktur dasar dari, yang mencakup kelas-kelas seperti Layanan, Area Rumah, Alat/Bahan, dan Pemesanan. Relasi antar kelas ini akan dijelaskan dalam bentuk diagram dan dijadikan dasar dalam perancangan lebih lanjut.

1.1.4 Formalisasi & implementasi (OWL/RDF) menggunakan *Protégé*

Pada tahap ini, model yang telah dikonsepskan sebelumnya diterjemahkan menjadi format formal menggunakan OWL di atas RDF dengan bantuan *tool Protégé*. Proses ini termasuk pembuatan kelas, properti, dan contoh individu yang diperlukan.

1.1.5 Perumusan kueri *SPARQL* untuk menjawab CQs

Setelah diterapkan, tahap ini melibatkan pembuatan kueri *SPARQL* untuk mengeksekusi pertanyaan-pertanyaan dari *Competency Questions* yang telah disusun sebelumnya. *Kueri* ini akan diuji untuk memastikan dapat memberikan jawaban yang sesuai.

1.1.6 Evaluasi

Evaluasi dilakukan dengan cara memastikan konsistensi struktur, serta pengujian untuk memastikan bahwa semua *Competency Questions* dapat dijawab dengan benar menggunakan kueri *SPARQL*. Jika ditemukan adanya kesalahan atau ketidaksesuaian dalam jawaban, revisi dilakukan pada struktur.

1.2. Desain dan Artefak Teknis

Perancangan pada bab ini menerjemahkan kebutuhan *domain* ke dalam arsitektur sistem dan artefak teknis yang memanfaatkan sebagai *knowledge base* layanan *home care*. Desain meliputi pemodelan aktor dan fungsi utama (*use case*), alur kerja pencarian layanan berbasis (*flowchart*), arsitektur komponen sistem, model dalam bentuk *class diagram*, spesifikasi kebutuhan dalam bentuk *competency questions (CQs)* dan aturan sederhana, serta perumusan metrik *non-fungsional* dan jadwal penelitian sebagai dasar implementasi *prototipe*.

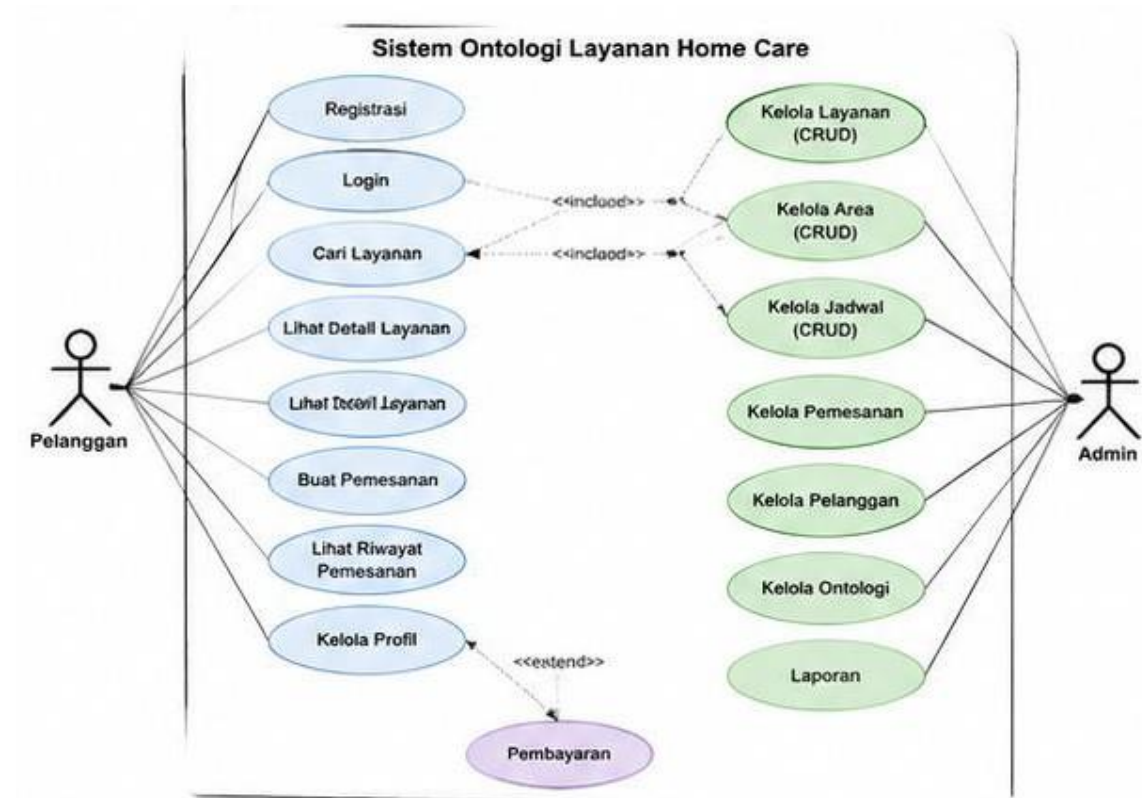
1.2.1 Use Case Diagram

Use Case Diagram digunakan untuk menggambarkan aktor yang terlibat dan fungsi utama prototipe sistem informasi yang memanfaatkan sistem layanan *home care*. Diagram *use case* pada penelitian ini dibuat menggunakan *software StarUML* sehingga notasi *use case* (termasuk relasi *<<include>>* dan *<<extend>>*) dapat ditampilkan sesuai standar *UML*.

Aktor yang terlibat meliputi Pelanggan dan *Admin/Operator*.

- Pelanggan: melihat katalog layanan/paket, melakukan pencarian layanan berdasarkan kebutuhan (area/jenis pekerjaan/paket), melihat detail layanan (cakupan, alat/bahan, durasi, harga), dan membuat pemesanan.

- *Admin/Operator*: mengelola data pengetahuan (layanan, area, alat/bahan, paket dan aturan), menjalankan uji *Competency Questions* melalui *query SPARQL*, serta memantau *log* penggunaan sistem.

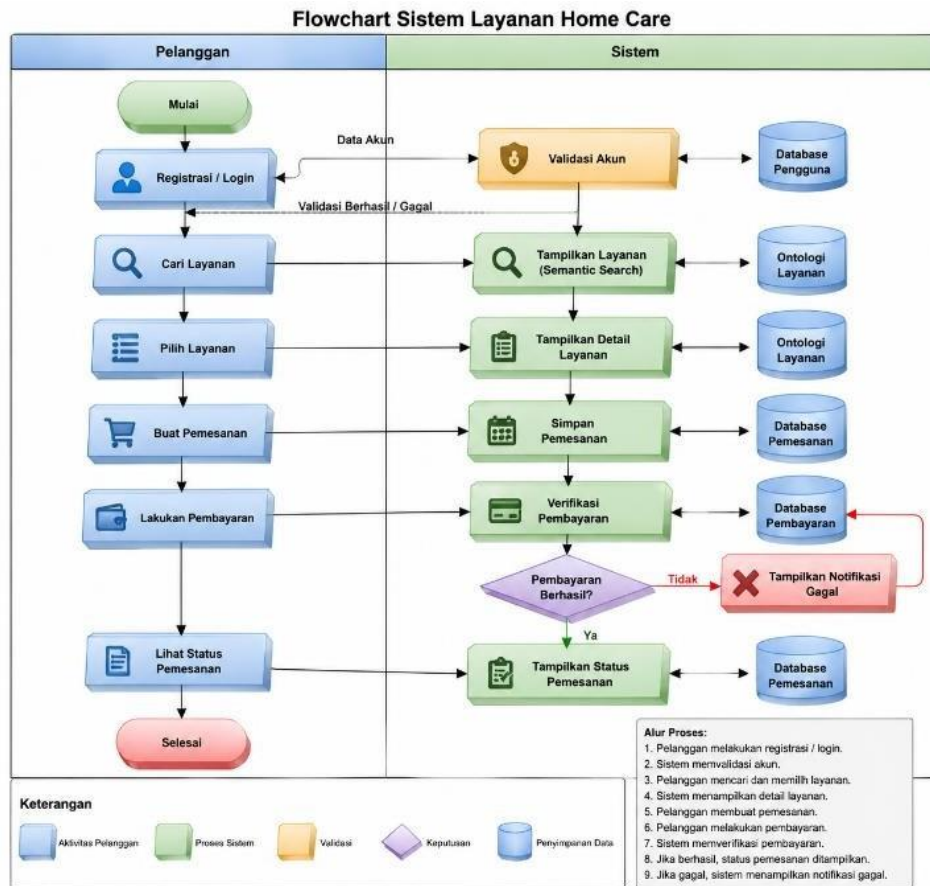


Gambar 3. 1 Menunjukkan interaksi mereka dengan sistem.

1.2.2 Flowchart Sistem

Flowchart digunakan untuk menjelaskan alur kerja sistem saat pelanggan melakukan pencarian layanan atau pemesanan. Alur dimulai dari pelanggan memasukkan kebutuhan (misalnya “*deep cleaning* untuk dapur”), kemudian sistem memetakan input tersebut ke konsep (kelas dan relasi yang relevan),

menjalankan kueri untuk mendapatkan daftar layanan/paket yang sesuai, lalu menampilkan hasil beserta detail cakupan pekerjaan, estimasi durasi, dan kebutuhan alat/bahan. Jika pelanggan memilih layanan/paket, sistem mencatat permintaan/pemesanan dan menyimpan ringkasan kebutuhan yang dipilih. *Flowchart* membantu memperjelas proses kerja *prototipe* dan hubungan antara input pengguna, *knowledge base*, dan keluaran informasi layanan.



Gambar 3. 2 ini menjelaskan alur proses dari awal hingga akhir

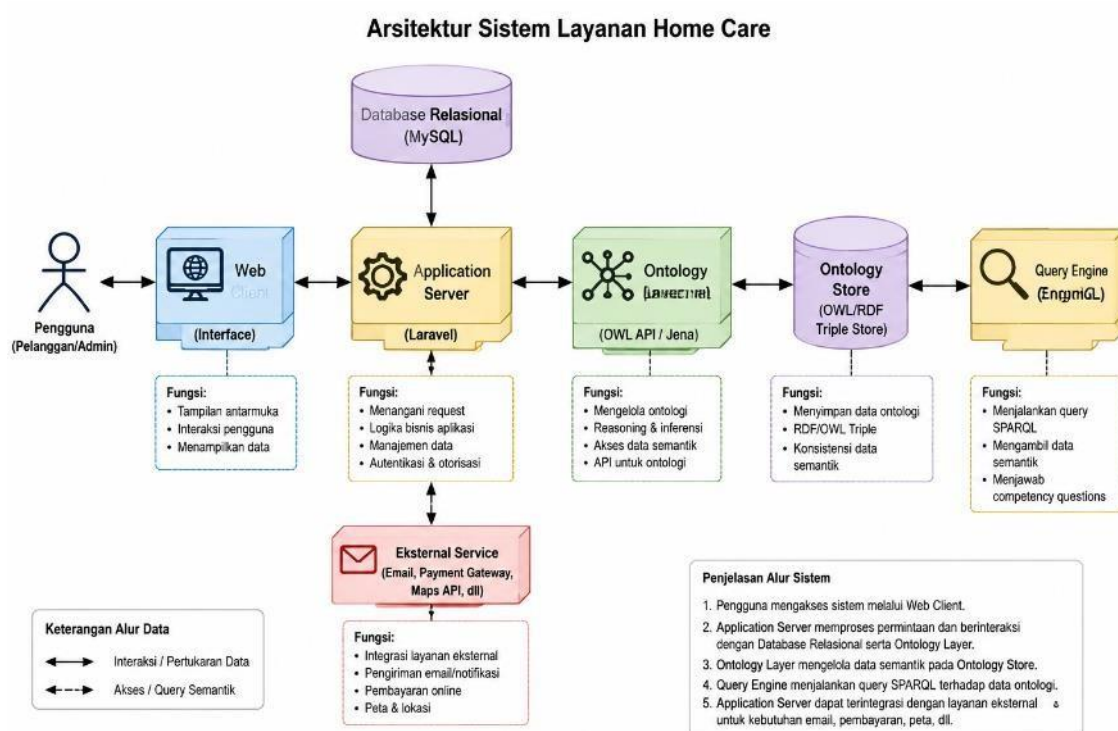
Flowchart menunjukkan alur ketika pelanggan mencari layanan. Dimulai dari input kebutuhan (contoh: “*deep cleaning* dapur”), sistem melakukan *mapping input* ke konsep (misalnya Area = Dapur, Layanan = *Deep Cleaning*), lalu menjalankan *query SPARQL* untuk mengambil layanan/paket yang sesuai. Sistem menampilkan hasil beserta detail (harga, durasi, alat/bahan). Jika pelanggan memilih salah satu hasil, sistem menyimpan pemesanan dan *log*; jika belum cocok, sistem meminta klarifikasi/ubah input.

1.2.3 Arsitektur Sistem

Arsitektur sistem menjelaskan komponen utama yang mendukung pemanfaatan sebagai *knowledge base* layanan *home care*. Komponen sistem terdiri dari:

1. Antarmuka pengguna (*UI*) untuk pelanggan/*admin* (bisa berupa *web* sederhana)
2. *Service/Backend* yang menerima *input* pengguna, melakukan validasi, dan menjalankan proses pencarian

3. *Knowledge Base* yang berisi konsep, relasi, dan aturan layanan (diimplementasikan dalam *OWL/RDF*)
4. *Query Engine* untuk menjalankan kueri *SPARQL* dan mengambil informasi dari *knowledge base*
5. *Database/Log* untuk menyimpan data pemesanan dan catatan aktivitas sistem. Arsitektur ini menunjukkan bagaimana tidak berdiri sendiri, tetapi dimanfaatkan oleh sistem informasi untuk menampilkan informasi layanan secara konsisten dan terstruktur.



Gambar 3.3 Arsitektur Sistem

Gambar 3. 3 Arsitektur Sistem

Arsitektur Integrasi Sistem (*Laravel*) Untuk mengimplementasikan pencarian berbasis semantik, aplikasi ini dirancang dengan menggunakan arsitektur *hybrid database*. Arsitektur ini memisahkan media penyimpanan antara data transaksional dengan data pengetahuan (semantik).

1. Basis Data Relasional (*MySQL*):

MySQL bertindak sebagai penyimpan data operasional aplikasi yang bersifat dinamis dan membutuhkan proses *Create, Read, Update, Delete (CRUD)* yang cepat. Data yang disimpan

di *MySQL* meliputi data autentikasi pengguna (*login/register*), keranjang pemesanan, riwayat transaksi, dan sesi (*sessions*). *Laravel* berinteraksi dengan *MySQL* menggunakan fitur Eloquent *ORM*.

2. Basis Data Semantik / *Triple Store*:

Berkas (berformat *.owl* atau *.ttl*) bertindak sebagai *knowledge base*. Basis data ini khusus menangani pencarian relasional kompleks yang mengandalkan penalaran (*reasoning*), seperti mencari kecocokan antara alat, layanan, dan keahlian pekerja.

3. Mekanisme Integrasi:

Karena *framework Laravel* berbasis *PHP* tidak dirancang untuk membaca *file* secara *native*, sistem menggunakan mekanisme integrasi khusus (seperti *library EasyRdf* atau komunikasi *API/HTTP Client* ke *endpoint server* seperti *Apache Jena*). Saat pengguna memasukkan kata kunci pencarian di *website*, *Controller* pada *Laravel* akan menerjemahkan permintaan tersebut menjadi bahasa kueri *SPARQL*, mengirimkannya ke *Triple Store*, lalu memetakan hasilnya kembali ke dalam antarmuka *web (HTML/Blade)* agar mudah dipahami oleh pelanggan.

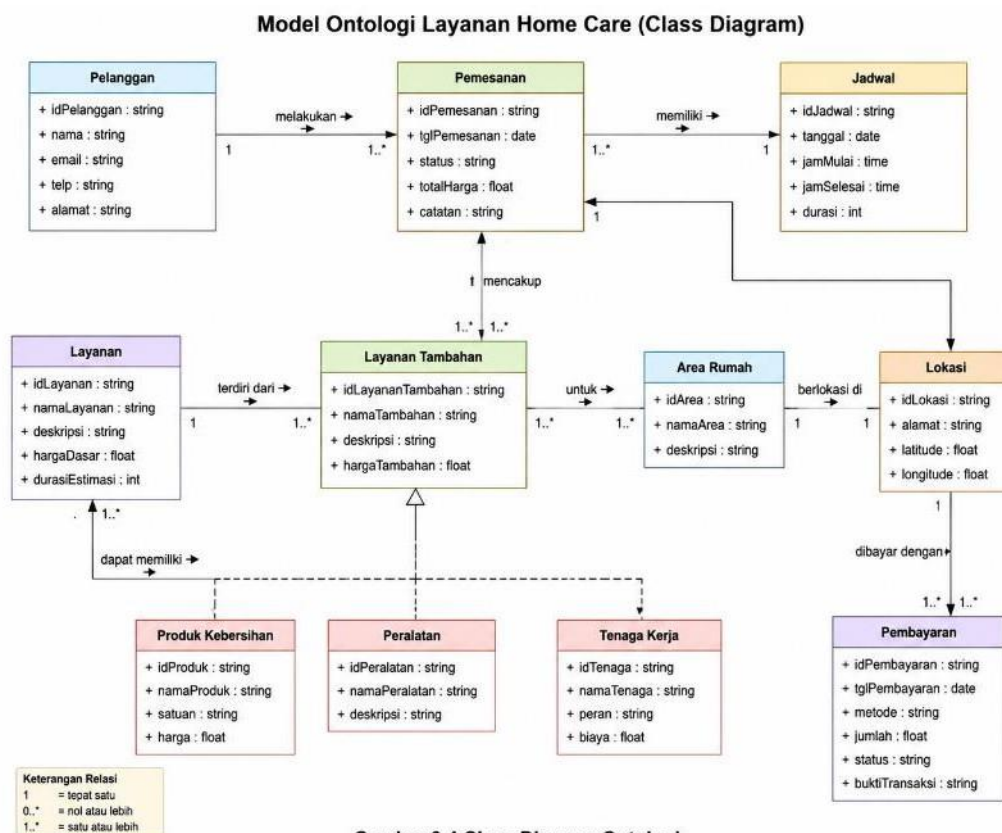
1.2.4 Perancangan Data (*Model dengan Class Diagram*)

Karena penelitian kamu fokusnya bagian “Perancangan Data” sebaiknya bukan *ERD* transaksi seperti sebelumnya, melainkan Model (bisa kamu gambarkan dalam bentuk *Class Diagram*). Model ini mendefinisikan konsep inti, misalnya:

1. Pelanggan
2. Pemesanan
3. Layanan
4. PaketLayanan
5. AreaRumah/Ruang
6. Pekerjaan/*Task*

7. Alat
8. Bahan
9. AturanHarga/Durasi

Gambar ini menunjukkan struktur dengan *Class Diagram*, yang menghubungkan konsep-konsep seperti Layanan, PaketLayanan, dan Pemesanan.



Gambar 3.4 Class Diagram Ontologi

Gambar 3. 4 Class Diagram

1. Class Pelanggan

Class Pelanggan merepresentasikan individu yang menggunakan layanan *home care*. Setiap pelanggan memiliki atribut seperti nama, alamat, nomor telepon, dan riwayat pemesanan. *Class* ini berhubungan dengan *class* Pemesanan melalui relasi melakukanPemesanan.

2. *Class Layanan*

Class Layanan merepresentasikan seluruh jenis layanan yang tersedia pada sistem, seperti *General Cleaning*, *Deep Cleaning*, Cuci Sofa, dan Cuci Karpet. Setiap layanan memiliki atribut nama layanan, estimasi durasi, harga, serta area yang dapat ditangani

3. *Class Pemesanan*

Class Pemesanan merepresentasikan transaksi layanan yang dilakukan oleh pelanggan. Setiap pemesanan menyimpan informasi seperti tanggal layanan, waktu layanan, alamat tujuan, status pemesanan, dan status pembayaran. *Class* ini menjadi penghubung antara pelanggan dan layanan yang dipilih. Melalui *class* ini, sistem dapat melacak seluruh aktivitas pemesanan yang terjadi dalam aplikasi.

4. *Class PaketLayanan*

Class PaketLayanan digunakan untuk mengelompokkan beberapa layanan menjadi satu paket yang dapat dipesan sekaligus oleh pelanggan.

Sebagai contoh, Paket *Deep Cleaning* dapat terdiri dari layanan pembersihan ruang tamu, kamar tidur, dapur, dan kamar mandi. Dengan adanya *class* ini, sistem dapat memberikan pilihan layanan yang lebih fleksibel serta meningkatkan efisiensi dalam proses pemesanan.

5. *Class AreaRumah*

Class AreaRumah merepresentasikan lokasi atau area rumah yang menjadi objek layanan. Contohnya meliputi ruang tamu, kamar tidur, dapur, kamar mandi, halaman, dan garasi. Setiap layanan dapat memiliki hubungan dengan satu atau lebih area rumah. Informasi ini digunakan untuk membantu sistem dalam mengidentifikasi cakupan pekerjaan yang termasuk dalam suatu layanan tertentu.

6. *Class Pekerjaan (Task)*

Class Pekerjaan merepresentasikan aktivitas atau tugas yang dilakukan dalam suatu layanan. Contohnya adalah menyapu lantai, mengepel, membersihkan kaca, mencuci sofa, membersihkan karpet, dan mengelap perabotan. Setiap layanan dapat terdiri dari beberapa pekerjaan yang saling berhubungan. Dengan memodelkan pekerjaan secara terpisah, sistem dapat memberikan informasi yang lebih rinci mengenai cakupan layanan yang akan diterima pelanggan.

7. Relasi Antar *Class*

Hubungan antar class dalam digunakan untuk menggambarkan keterkaitan konsep-konsep yang terdapat dalam *domain* layanan *home care*. Pelanggan memiliki relasi melakukan Pemesanan terhadap class Pemesanan. Pemesanan memiliki relasi memilih Layanan terhadap class Layanan. Selanjutnya, Layanan memiliki relasi mencakup Area terhadap Area Rumah, memiliki Pekerjaan terhadap Pekerjaan, membutuhkan Alat terhadap Alat, serta menggunakan Bahan terhadap Bahan. Selain itu, Layanan juga memiliki relasi menggunakan Aturan terhadap Aturan Harga Durasi untuk menentukan harga dan estimasi waktu pengerjaan.

Relasi-relasi tersebut membentuk struktur pengetahuan yang terorganisasi sehingga sistem dapat memahami hubungan antar konsep secara semantik. Dengan pendekatan ini, pencarian layanan menjadi lebih akurat dan informasi yang diberikan kepada pengguna menjadi lebih lengkap dibandingkan sistem berbasis pencarian kata kunci biasa.

Spesifikasi Struktur

Untuk memberikan gambaran detail mengenai basis pengetahuan yang dibangun, berikut adalah spesifikasi kelas dan properti yang didefinisikan dalam *Home Care*. Struktur ini dirancang untuk menangkap relasi semantik antar entitas layanan agar dapat diproses oleh mesin penalaran (*reasoning*).

Proses penalaran (*reasoning*) pada struktur ini melibatkan pemeriksaan konsistensi logis untuk memastikan tidak ada hierarki kelas yang tumpang tindih (*disjoint classes*). Penentuan spesifikasi ini juga membagi properti menjadi dua kategori utama, yaitu *Object Property* yang menghubungkan relasi antar-individu dari kelas berbeda,

serta *Data Property* yang menghubungkan suatu individu atau entitas dengan nilai tipe data primitif (literal), seperti teks, angka (*integer*), maupun format tanggal. Pembagian struktur properti secara eksplisit ini bertujuan agar mesin penalaran dapat membedakan secara logis antara keterkaitan relasional antar konsep dengan atribut deskriptif yang bersifat kuantitatif, sehingga akurasi pencarian informasi layanan menjadi lebih optimal."

Tabel 3. 2 Spesifikasi Kelas

Nama Kelas	Deskripsi
Layanan	Entitas utama yang merepresentasikan jasa kebersihan (contoh: <i>General Cleaning</i>).
PaketLayanan	Kumpulan dari beberapa entitas Layanan yang digabungkan dalam satu penawaran.
AreaRumah	Lokasi fisik pengerjaan layanan (contoh: Dapur, Kamar Mandi).
Peralatan	Perangkat keras yang dibutuhkan untuk melakukan tindakan layanan.
BahanPembersih	Material kimia atau pembersih yang dikonsumsi selama layanan.
TenagaKerja	<i>Subclass</i> dari Aktor yang memiliki keahlian spesifik (<i>Object Property</i>).

Tabel 3.2 di atas merinci hierarki kelas yang dibangun untuk membentuk struktur dasar basis pengetahuan sistem *Home Care*.

Spesifikasi kelas ini disusun untuk memberikan batasan domain yang jelas antara entitas utama, sehingga proses inferensi data dapat berjalan tanpa ambiguitas.

Sebagai contoh, pemisahan antara kelas Layanan dan PaketLayanan memungkinkan sistem untuk membedakan antara jasa tunggal yang bersifat spesifik dengan kumpulan layanan yang memiliki aturan harga serta durasi tersendiri.

Dengan struktur ini, sistem informasi dapat mengenali dan mengelola setiap konsep sebagai entitas unik yang memiliki karakteristik dan perilaku yang berbeda dalam domain layanan kebersihan rumah.

Tabel 3. 3 Spesifikasi Properti

Nama Properti	Tipe Properti	Relasi (<i>Domain -> Range</i>)
membutuhkanAlat	<i>Object Property</i>	Layanan > Peralatan
menggunakanBahan	<i>Object Property</i>	Layanan > BahanPembersih
mencakupArea	<i>Object Property</i>	Layanan > AreaRumah
memilikiTarif	<i>Data Property</i>	Layanan -> <i>xsd:integer</i>
estimasiDurasi	<i>Data Property</i>	Layanan -> <i>xsd:string</i>

tabel 3.3 mendefinisikan relasi semantik (*object property*) dan atribut data (*data property*) yang menghubungkan kelas-kelas dalam

Relasi seperti mencakup Area dan membutuhkan Alat berfungsi untuk mengintegrasikan berbagai konsep yang sebelumnya terpisah menjadi satu *graf* pengetahuan yang saling terhubung.

Sebagai contoh, relasi membutuhkan Alat didefinisikan untuk memastikan bahwa sistem mampu memberikan rekomendasi peralatan kebersihan secara otomatis ketika seorang pelanggan memilih layanan tertentu.

Dengan adanya properti ini, sistem dapat melakukan penalaran otomatis untuk menampilkan daftar alat yang harus disiapkan oleh petugas berdasarkan jenis layanan yang dipesan, sehingga meminimalisir kesalahan operasional di lapangan.

Sementara itu, data *property* seperti memiliki Tarif dan estimasi Durasi digunakan untuk memetakan nilai kuantitatif yang diperlukan dalam proses kalkulasi harga dan penjadwalan layanan secara *real-time*."

1.2.5 Spesifikasi Kebutuhan (Competency Questions & Aturan)

penjelasan tentang *Competency Questions (CQs)* sudah cukup jelas. *CQs* berfungsi sebagai alat untuk menguji apakah dapat memberikan jawaban yang sesuai dengan kebutuhan domain. Anda sudah menyebutkan contoh *CQs* yang relevan dengan layanan *home care*, seperti:

1. Layanan apa saja yang tersedia di dapur?
2. Paket layanan apa yang mencakup *deep cleaning*?
3. Alat apa yang dibutuhkan untuk layanan pembersihan sofa?

1.2.6 Metrik & Target Non-Fungsional

Metrik non-fungsional yang relevan untuk prototipe sistem informasi adalah:

1. Konsistensi = tidak menghasilkan kontradiksi saat *reasoning* (tidak ada *class/property* yang bertabrakan).

2. *Coverage Competency Questions* Minimal X% CQs berhasil dijawab oleh SPARQL query (misalnya target 90–100%).

3. Kualitas Informasi (*Clarity & Completeness*)

Definisi layanan jelas (cakupan pekerjaan, area, alat/bahan), dan konsisten antar layanan.

Metrik	Target	Cara Ukur
Konsistensi ontologi	Tidak ada konflik saat reasoning	Reasoner di Protégé / validasi OWL
Coverage CQs	$\geq 90\%$ CQs terjawab	SPARQL query untuk tiap CQ
Kejelasan & kelengkapan info	Definisi layanan lengkap & konsisten	Checklist (cakupan, area, alat, durasi, harga)
Waktu respon query (opsional)	$< 3-5$ detik (prototipe lokal)	Catat waktu eksekusi SPARQL
Usability prototipe (opsional)	SUS > 70	Kuesioner SUS pengguna terbatas

Gambar 3. 5 Target Metrik

1. Konsistensi tidak ada konflik saat *reasoning*
2. *Coverage CQs*: minimal 90% CQs dapat dijawab query SPARQL
3. Kejelasan & kelengkapan informasi layanan: definisi layanan lengkap (cakupan, area, alat, durasi, harga)