LAPORAN TUGAS BESAR
IF4070 REPRESENTASI PENGETAHUAN DAN PENALARAN

Ontologi dan Sistem Berbasis Pengetahuan

Disusun oleh:

Ibrahim Ihsan Rasyid	13522018
Farhan Nafis Rayhan	13522037
Muhammad Syarafi Akmal	13522076

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung

2025

DAFTAR ISI

BAB I
DESKRIPSI PERSOALAN

Secara fundamental, ontologi adalah sebuah spesifikasi formal dan eksplisit dari istilah-istilah dalam suatu domain beserta hubungan di antara mereka. Ontologi berfungsi untuk menangkap pengetahuan tentang domain yang diminati dengan mendefinisikan kosakata umum yang dapat digunakan untuk berbagi informasi di antara manusia atau agen perangkat lunak. Ini mencakup definisi konsep-konsep dasar, atribut, dan relasi yang dapat ditafsirkan oleh mesin , serta batasan, aturan, dan aksioma yang mengatur domain tersebut. Pada intinya, ontologi menyediakan representasi pengetahuan yang terstruktur, fleksibel, dan dapat digunakan kembali yang cocok untuk penalaran Kecerdasan Buatan (AI).

Sistem berbasis pengetahuan (*knowledge-based system*), yang sering disebut sebagai sistem pakar (*expert system*), adalah sebuah program dalam cabang kecerdasan buatan yang dirancang untuk meniru kemampuan pengambilan keputusan seorang ahli manusia dalam domain yang kompleks dan spesifik. Tidak seperti program tradisional yang kaku mengikuti alur prosedural, sistem berbasis pengetahuan dirancang untuk menyelesaikan masalah-masalah rumit dengan cara menarik kesimpulan dari basis pengetahuan yang terstruktur, sama seperti cara seorang ahli manusia bekerja. Tujuannya adalah untuk mengambil pengetahuan domain yang spesifik—seperti aturan hukum, diagnosis medis, atau perencanaan keuangan—dan membuatnya dapat diakses untuk memberikan saran atau solusi.

PEMILIHAN DOMAIN

Tugas ini menuntut penerjemahan fakta dan aturan dari domain tertentu ke dalam representasi formal untuk memungkinkan penalaran simbolik. Domain yang dipilih untuk proyek ini adalah Hukum Waris Islam, yang secara teknis dikenal sebagai *ʿilm al-farāʾiḍ* (ilmu tentang bagian-bagian yang telah ditentukan).

Pemilihan domain ini didasarkan pada kesesuaiannya yang tinggi dengan tujuan tugas. *Fara'id* pada dasarnya adalah sistem berbasis pengetahuan yang telah ada selama ribuan tahun. Penalaran dalam domain ini mengikuti aturan-aturan *if-then* yang jelas dan dapat dipahami manusia. Sebagai contoh:

- Aturan 1: JIKA seorang istri wafat dan memiliki anak, MAKA suami mendapat bagian 1/4.
- Aturan 2: JIKA seorang istri wafat dan TIDAK memiliki anak, MAKA suami mendapat bagian 1/2.
- Aturan 3: JIKA seorang pewaris adalah 'Anak Lelaki', MAKA ia menghalangi (melakukan *Hajb*) 'Saudara Lelaki' untuk mendapat warisan.

Kumpulan aturan yang terstruktur, presisi matematis, dan logika kondisional yang kompleks ini menjadikan *Fara'id* sebagai domain yang ideal untuk diterjemahkan ke dalam representasi ontologi formal dan sistem pakar berbasis aturan.

BATASAN DOMAIN

Fikih Islam (yurisprudensi) adalah domain yang sangat luas. Untuk memastikan proyek ini dapat dikelola dan tidak terlalu sempit, kami menetapkan batasan-batasan berikut:

- Batasan *Mazhab* (School of Law): Terdapat perbedaan signifikan antara aturan waris Sunni dan Syiah (Ja'fari), terutama dalam klasifikasi ahli waris. Proyek ini akan fokus secara eksklusif pada aturan waris Sunni berdasarkan konsensus empat mazhab utama (Hanafi, Maliki, Syafi'i, Hanbali). Perbedaan antar mazhab Sunni (misalnya, status kakek vs. saudara) juga tidak akan dimodelkan untuk menjaga konsistensi.

- Batasan Ahli Waris: Sistem akan memodelkan dua kelas ahli waris utama: *Ashab al-Furud* (Penerima Bagian Tetap) dan *'Asabah* (Residuari/Penerima Sisa). Kelas ketiga, *Dhawu al-Arham* (Kerabat Jauh), yang memiliki aturan kompleks, tidak akan diimplementasikan.

- Batasan Proses: Fokus sistem adalah pada kalkulasi pembagian warisan (persentase). Proses pra-distribusi, seperti administrasi pemakaman, pembayaran utang almarhum, dan eksekusi wasiat (yang dibatasi 1/3 harta) , diasumsikan telah selesai.

ASUMSI DOMAIN

Fikih Islam tidak terlepas dari terjadinya perbedaan pendapat. Salah satu perdebatan utama mengenai *ilm al-farāʾiḍ* adalah besaran dari waris yang diterima anggota keluarga pada beberapa kasus tertentu. Perdebatan ini muncul akibat perbedaan penafsiran dan pemahaman, yang dapat menciptakan ketidakpastian dalam ontologi. Terdapat dua buah kasus perbedaan pendapat yang perlu dibahas:

- Apakah bagian Ibu turun dari 1/3 menjadi 1/6 ketika almarhum memiliki dua saudara atau lebih, meskipun para saudara terhalang dari waris: Sebagian ulama berpendapat bahwa sekadar keberadaan mereka sudah cukup untuk menurunkan bagian Ibu. Ulama lain berpendapat bahwa jika mereka terhalang, maka mereka tidak seharusnya dihitung sebagai pengurang bagiannya.

- Apakah Kakek dari pihak Ayah dapat menghalangi Saudara Kandung dari almarhum untuk menerima warisan, sebagaimana Ayah menghalanginya: Aturan utama pada diagram menunjukkan bahwa ia memang menghalangi, tetapi ada catatan bahwa mazhab Malikiyyah tidak sependapat. Dalam pandangan mereka, Kakek diperlakukan seperti saudara, sehingga keduanya harus berbagi harta warisan, bukan saling menggugurkan.

Untuk menjaga integritas dari ontologi yang akan dibangun, kami perlu menentukan aturan mana saja yang dianggap berlaku walau aturan ini masih diperdebatkan. Oleh karena itu, kami menentukan asumsi yang akan diterapkan pada ontologi. Pada kedua kasus, kami mengasumsikan bahwa penghalangan tidak berlaku.

RUJUKAN DOMAIN

Untuk memberikan konteks kepada pembaca serta memvalidasi batasan dan aturan terhadap domain yang dipilih, kami mencantumkan beberapa rujukan. Berikut adalah rujukan utama kami:

1. Sumber primer: Aturan-aturan fundamental domain ini bersumber dari Al-Qur'an, khususnya surat An-Nisa' (ayat 4:11, 4:12, dan 4:176), serta Hadis Nabi (As-Sunnah). Aturan-aturan formal dan sistematis dari *`ilm al-farāʾiḍ* (ilmu waris) yang menjadi dasar basis pengetahuan (knowledge base) sistem ini diambil dari kodifikasi dan *istinbath* (derivasi hukum) yang telah mapan dalam yurisprudensi Islam (fikih). Rujukan utama untuk derivasi aturan-aturan ini adalah kitab-kitab tafsir *ahkam* (hukum) seperti Tafsir al-Qurtubi, serta kitab-kitab fikih klasik yang menjadi standar dalam domain ini, terutama Matan Ar-Rahbiyyah dan karya perbandingan mazhab seperti Bidayat al-Mujtahid.

2. Sumber Akademik (Pemodelan): Untuk implementasi teknis dan pemodelan ontologi, kami merujuk pada penelitian akademis di bidang *knowledge engineering* yang telah memodelkan domain ini. Rujukan kunci adalah karya Zouaoui dan Rezeg (2021) mengenai "AraFamOnto", sebuah sistem kalkulasi waris Islam yang berbasis ontologi.[1]

BAB II
PEMBAHASAN

ONTOLOGI

Ontologi hukum waris ini dirancang berdasarkan klasifikasi ahli waris serta relasi kekerabatan yang menentukan siapa yang berhak menerima warisan dan siapa yang terhalang oleh keberadaan ahli waris lainnya. Meskipun struktur ontologi merujuk pada konsep pohon keluarga, secara fungsi dan semantik ontologi ini tidak bertujuan untuk merepresentasikan silsilah keluarga secara eksplisit. Sebaliknya, ontologi memodelkan hubungan kekerabatan dengan berpusat pada seorang subjek yang telah meninggal (*deceased*) untuk kemudian mengidentifikasi kategori ahli waris beserta ketentuan porsinya. Dengan pendekatan ini, ontologi bersifat konseptual dan normatif, karena menekankan aturan dan prioritas dalam pewarisan, bukan sekadar menggambarkan struktur genealogis keluarga.

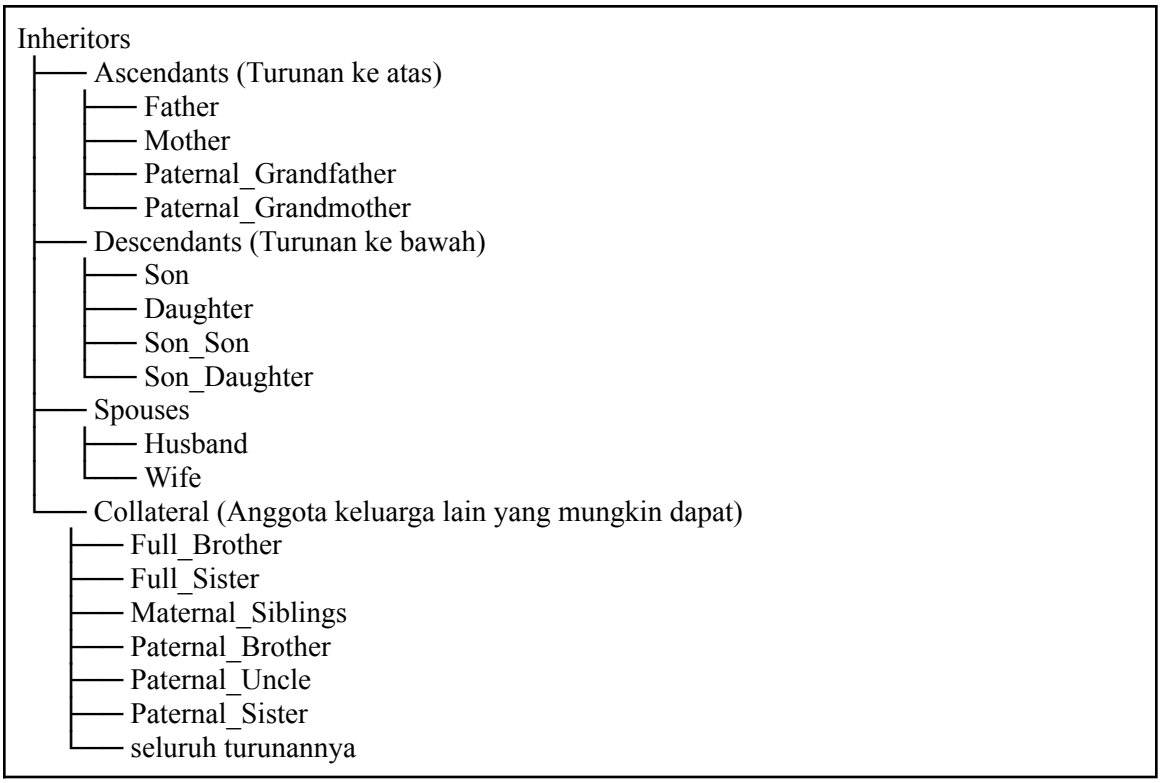
TBox dan ABox dalam Ontologi Hukum Waris

Representasi pengetahuan dalam ontologi dibangun melalui dua komponen utama, yaitu TBox (Terminological Box) dan ABox (Assertion Box). Keduanya memisahkan definisi konseptual dari fakta konkret sehingga ontologi dapat digunakan untuk mengeksekusi aturan pewarisan pada berbagai kasus.

TBox berisi definisi kelas, hirarki kelas, serta object properties yang mendeskripsikan konsep-konsep hukum waris. Di dalam TBox, ahli waris diklasifikasikan ke dalam kelompok *Ascendants*, *Descendants*, *Spouses*, dan *Footnotes*. Masing-masing kategori memiliki subclass yang menggambarkan tipe pewaris tertentu, seperti *Father*, *Mother*, *Son*, *Daughter*, *Full_Brother*, *Paternal_Uncle*, dan sebagainya. TBox juga memuat definisi object property seperti Father_Blocks, Son_Blocks, atau Full_Brother_Blocks, yang merepresentasikan aturan *hajb* (penghalangan waris). Setiap property memiliki domain dan range yang menunjukkan pihak yang menghalangi dan pihak yang terhalang sesuai ketentuan fikih. Dengan demikian, TBox menyediakan kerangka semantik yang stabil bagi proses inferensi dan pembagian warisan.

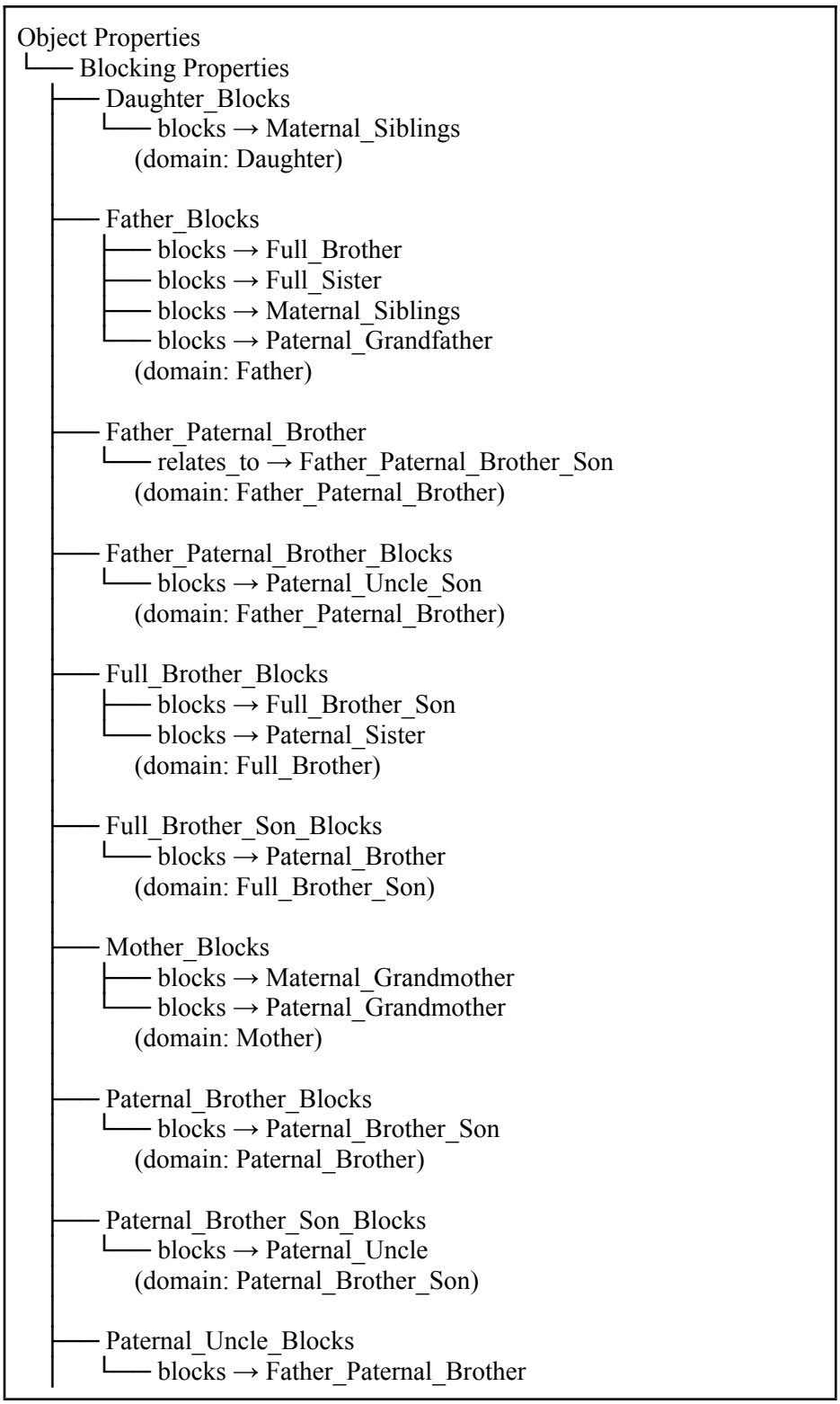
ABox, di sisi lain, memuat fakta-fakta konkret mengenai individu dalam suatu kasus. ABox berisi instansiasi kelas—misalnya *Mas_Amba* sebagai *Son*, *Megawati_Akmalputri* sebagai *Daughter*, atau *Mulyono_Widodo* sebagai *Son*—beserta atribut dan relasinya (berdasarkan kasus uji). Nilai tertentu seperti *Portion* dapat diberikan pada individu sebagai informasi deskriptif mengenai hak waris. ABox bersifat dinamis dan berubah setiap kali pengguna memasukkan struktur keluarga baru, karena ia menggambarkan kondisi nyata dari kasus pewarisan yang sedang dianalisis. Dengan input ABox dan aturan TBox, sistem dapat menentukan siapa saja ahli waris yang sah, siapa yang terhalang, serta bagian waris yang harus diberikan sesuai aturan syariah.

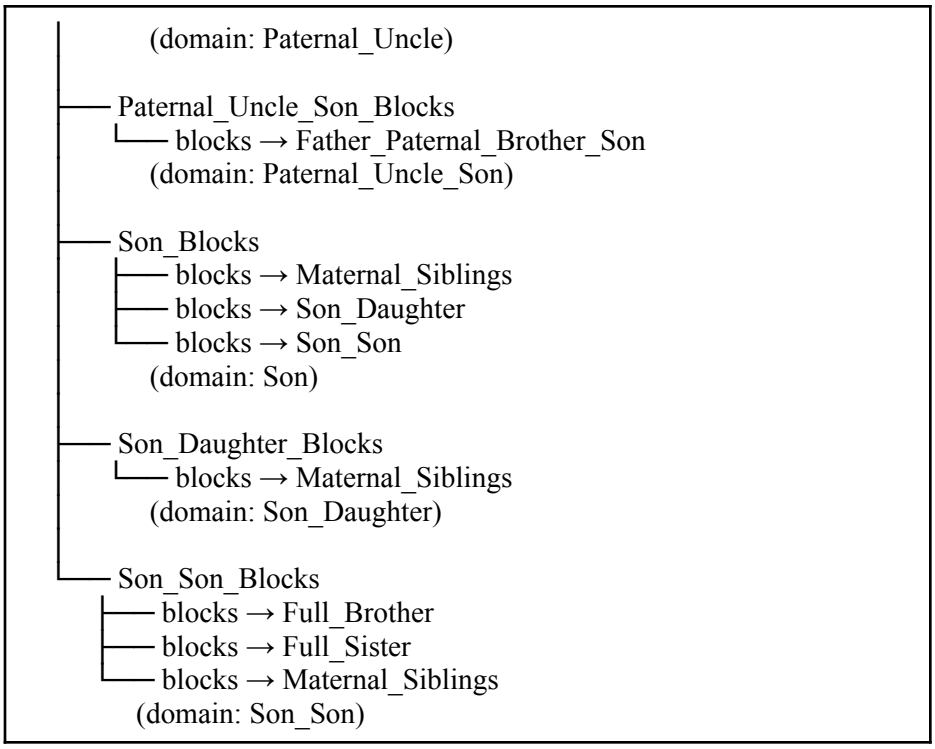
Berikut adalah struktur ontologi:


Inheritors
Ascendants (Turunan ke atas)
Father
Mother
Paternal_Grandfather
Paternal_Grandmother
Descendants (Turunan ke bawah)
Son
Daughter
Son_Son
Son_Daughter
Spouses
Husband
Wife
Collateral (Anggota keluarga lain yang mungkin dapat)
Full_Brother
Full_Sister
Maternal_Siblings
Paternal_Brother
Paternal_Uncle
Paternal_Sister
seluruh turunannya

Terdapat beberapa properti di ontologi ini:

1. X_Blocks (properti objek), X adalah nama kelas domain yang akan menghalangi warisan kepada kelas range yang didefinisikan. Berikut adalah properti objek yang telah dirancang di Protege:





2. Portion adalah properti data yang memberikan range nilai porsi pendapatan pada domain kelas inheritor berdasarkan hubungan ontologi yang sudah didefinisikan. Perlu diketahui bahwa properti ini bersifat deskriptif berdasarkan ketetapan hubungan antar keluarga, maka nilai bukan hasil kalkulasi apapun.

IMPLEMENTASI SISTEM

Sistem Berbasis Pengetahuan diimplementasikan menggunakan bahasa pemrograman Prolog (SWI Prolog) berdasarkan ontologi yang telah dibangun. Implementasi dibedakan menjadi tiga: fact, rule, dan alur utama program

a. Fact

Fakta-fakta dalam sistem berbentuk pohon keluarga yang menyimpan tiga jenis fakta, yaitu person/4, parent_of/2, dan married/2. Predikat person menyimpan ID, nama, jenis kelamin, dan status kehidupan (true berarti hidup, dan false berarti mati). Predikat parent_of(A, B) menyatakan bahwa B adalah orang tua dari A. Predikat married(A, B) menyatakan bahwa A dan B pernah menikah

Fakta yang disimpan bersifat informasi dasar. Informasi turunan lainnya seperti ayah, ibu, kakak, anak, dan lain-lain diturunkan berdasarkan fakta tersebut dengan rule yang dibuat

```
:- dynamic(person/4).
:- dynamic(parent_of/2).
:- dynamic(married/2).
:- discontiguous(person/4).
:- discontiguous(parent_of/2).
:- discontiguous(married/2).

% --- Persons ---
person(p1, betta, male, false).
person(p2, rina, female, true).
person(p3, aldi, male, false).
person(p4, wati, female, false).
person(p5, devi, female, true).
person(p6, lia, female, true).

% --- Marriages ---
married(p1, p2).
married(p3, p4).

% --- Parent-Child Relationships ---
parent_of(p3, p1).
parent_of(p5, p3).
parent_of(p6, p3).
parent_of(p3, p2).
parent_of(p5, p4).
parent_of(p6, p4).
```

b. Rule

Terdapat berbagai rule yang dibuat dalam sistem ini, yaitu rule yang berkaitan dengan hubungan kerabat, aturan bagian warisan seorang kerabat, dan aturan jumlah warisan yang didapatkan yang disebabkan oleh kehadiran kerabat lain.

Aturan yang berkaitan dengan hubungan kerabat disimpan dalam file kinship.pl. Terdapat berbagai rule di dalamnya yang menjelaskan hubungan dua orang, seperti father, mother, son, full_sibling_of, full_uncle, dan lain-lain. Di dalam file tersebut terdapat predikat living_relative_type/3 yang mengidentifikasi tipe apa seorang kerabat terhadap pewaris.

```
/* Relasi dasar */
father(Child, Father) :-
    parent_of(Child, Father),
    person(Father, _, male, _).

mother(Child, Mother) :-
    parent_of(Child, Mother),
    person(Mother, _, female, _).

son(Parent, Son) :-
    parent_of(Son, Parent),
    person(Son, _, male, _).

daughter(Parent, Daughter) :-
    parent_of(Daughter, Parent),
```

```
    person(Daughter, _, female, _).

husband(Wife, Husband) :-
    married(Husband, Wife).

wife(Husband, Wife) :-
    married(Husband, Wife).

/* Periksa apakah kerabat masih hidup */
is_living_relative(Person) :-
    person(Person, _, _, true).

full_sibling_of(P1, P2) :-
    father(P1, F), father(P2, F),
    mother(P1, M), mother(P2, M),
    P1 \== P2.

paternal_sibling_of(P1, P2) :-
    father(P1, F), father(P2, F),
    mother(P1, M1), mother(P2, M2),
    M1 \== M2,
    P1 \== P2.

maternal_sibling_of(P1, P2) :-
    mother(P1, M), mother(P2, M),
    father(P1, F1), father(P2, F2),
    F1 \== F2,
    P1 \== P2.

full_uncle_paternal(Uncle, NieceNephew) :-
    (full_sibling_of(Uncle, Parent),
    person(Parent, _, male, _),
    parent_of(Parent, NieceNephew)
    ;
    husband(Aunt, Uncle),
    full_sibling_of(Parent, Aunt),
    parent_of(Parent, NieceNephew)
    ).

ancestor_of(Anc, Desc) :-
    parent_of(Desc, Anc).

ancestor_of(Anc, Desc) :-
    parent_of(Desc, Parent),
    ancestor_of(Anc, Parent).

descendant_of(Desc, Anc) :-
    ancestor_of(Anc, Desc).

son_descendant(Desc, Anc) :-
    son(Anc, Desc).
son_descendant(Desc, Anc) :-
    son(Anc, S),
    son_descendant(Desc, S).

/* living_relative_type/3 - identifikasi tipe relasi kerabat yang masih
```

```
hidup */
living_relative_type(DeceasedID, HeirID, wife) :-
    person(HeirID, _, female, true),
    (married(DeceasedID, HeirID) ; married(HeirID, DeceasedID)).

living_relative_type(DeceasedID, HeirID, husband) :-
    person(HeirID, _, male, true),
    (married(DeceasedID, HeirID) ; married(HeirID, DeceasedID)).

living_relative_type(DeceasedID, HeirID, father) :-
    person(HeirID, _, male, true),
    parent_of(DeceasedID, HeirID).

living_relative_type(DeceasedID, HeirID, mother) :-
    person(HeirID, _, female, true),
    parent_of(DeceasedID, HeirID).

living_relative_type(DeceasedID, HeirID, son) :-
    person(HeirID, _, male, true),
    parent_of(HeirID, DeceasedID).

living_relative_type(DeceasedID, HeirID, daughter) :-
    person(HeirID, _, female, true),
    parent_of(HeirID, DeceasedID).

living_relative_type(DeceasedID, HeirID, grandfather_paternal) :-
    person(HeirID, _, male, true),
    parent_of(DeceasedID, ParentID),
    person(ParentID, _, male, false),
    parent_of(ParentID, HeirID).

living_relative_type(DeceasedID, HeirID, grandmother_maternal) :-
    person(HeirID, _, female, true),
    parent_of(DeceasedID, ParentID),
    person(ParentID, _, female, false),
    parent_of(ParentID, HeirID).

living_relative_type(DeceasedID, HeirID, grandmother_paternal) :-
    person(HeirID, _, female, true),
    parent_of(DeceasedID, ParentID),
    person(ParentID, _, male, false),
    parent_of(ParentID, HeirID).

living_relative_type(DeceasedID, HeirID, grandson_son) :-
    person(HeirID, _, male, true),
    parent_of(HeirID, ParentID),
    person(ParentID, _, male, false),
    parent_of(ParentID, DeceasedID),
    person(ParentID, _, male, false).

living_relative_type(DeceasedID, HeirID, granddaughter_son) :-
    person(HeirID, _, female, true),
    parent_of(HeirID, ParentID),
    person(ParentID, _, male, false),
    parent_of(ParentID, DeceasedID),
    person(ParentID, _, male, false).
```

```
living_relative_type(DeceasedID, HeirID, full_brother) :-
    person(HeirID, _, male, true),
    full_sibling(DeceasedID, HeirID).

living_relative_type(DeceasedID, HeirID, full_sister) :-
    person(HeirID, _, female, true),
    full_sibling(DeceasedID, HeirID).

living_relative_type(DeceasedID, HeirID, maternal_sister) :-
    person(HeirID, _, female, true),
    maternal_sibling(DeceasedID, HeirID).

living_relative_type(DeceasedID, HeirID, maternal_brother) :-
    person(HeirID, _, male, true),
    maternal_sibling(DeceasedID, HeirID).

living_relative_type(DeceasedID, HeirID, full_brothers_son) :-
    person(HeirID, _, male, true),
    full_sibling(DeceasedID, SiblingID),
    parent_of(HeirID, SiblingID).

living_relative_type(DeceasedID, HeirID, paternal_sister) :-
    person(HeirID, _, female, true),
    paternal_sibling(DeceasedID, HeirID).

living_relative_type(DeceasedID, HeirID, paternal_brother) :-
    person(HeirID, _, male, true),
    paternal_sibling(DeceasedID, HeirID).

living_relative_type(DeceasedID, HeirID, paternal_brothers_son) :-
    person(HeirID, _, male, true),
    paternal_sibling(DeceasedID, SiblingID),
    parent_of(HeirID, SiblingID).

living_relative_type(DeceasedID, HeirID, full_uncle_paternal) :-
    person(HeirID, _, male, true),
    full_uncle_paternal(HeirID, DeceasedID).

living_relative_type(DeceasedID, HeirID, fathers_paternal_brother) :-
    person(HeirID, _, male, true),
    parent_of(DeceasedID, FatherID),
    paternal_sibling(FatherID, HeirID).

living_relative_type(DeceasedID, HeirID, full_paternal_uncles_son) :-
    person(HeirID, _, male, true),
    full_uncle_paternal(UncleID, DeceasedID),
    parent_of(HeirID, UncleID).

living_relative_type(DeceasedID, HeirID, fathers_paternal_brothers_son)
:-
    person(HeirID, _, male, true),
    parent_of(DeceasedID, FatherID),
    paternal_sibling(FatherID, UncleID),
    parent_of(HeirID, UncleID).
```

```
full_sibling(Person1, Person2) :-
    Person1 \= Person2,
    parent_of(Person1, Father),
    parent_of(Person2, Father),
    person(Father, _, male, _),
    parent_of(Person1, Mother),
    parent_of(Person2, Mother),
    person(Mother, _, female, _).

paternal_sibling(Person1, Person2) :-
    Person1 \= Person2,
    parent_of(Person1, Father),
    parent_of(Person2, Father),
    person(Father, _, male, _),
    \+ full_sibling(Person1, Person2).

maternal_sibling(Person1, Person2) :-
    Person1 \= Person2,
    parent_of(Person1, Mother),
    parent_of(Person2, Mother),
    person(Mother, _, female, _),
    \+ full_sibling(Person1, Person2).

/* living_relative_type/2 - mencari seluruh kerabat yang masih hidup dan
tipe mereka */
/* Digunakan dengan findall */
living_relative_type(DeceasedID, Type) :-
    person(HeirID, _, _, true),
    living_relative_type(DeceasedID, HeirID, Type).
```

Aturan bagian warisan seorang kerabat disimpan dalam file provisional-share.pl. Predikat get_provisional_share/3 menyatakan porsi warisan yang didapatkan oleh kerabat tersebut, baik berupa pecahan pasti atau asabah (residual). Porsi warisan yang didapat pun beragam tergantung jumlah anggota atau kehadiran kerabat lain

```
/* get_provisional_share(HeirType, HeirsList, Share) */
/* Menentukan bagian warisan sementara untuk tipe ahli waris tertentu
berdasarkan daftar ahli waris */

/* --- Spouses --- */
get_provisional_share(wife, HeirsList, 1/8) :-
    has_descendant(HeirsList), !.
get_provisional_share(wife, _, 1/4) :- !.

get_provisional_share(husband, HeirsList, 1/4) :-
    has_descendant(HeirsList), !.
get_provisional_share(husband, _, 1/2) :- !.

/* --- Parents --- */
get_provisional_share(mother, HeirsList, 1/6) :-
    ( has_descendant(HeirsList)
    ; count_heir_type(HeirsList, full_brother, FB), FB >= 2
    ; count_heir_type(HeirsList, full_sister, FS), FS >= 2
    ; count_heir_type(HeirsList, paternal_brother, PB), PB >= 2
```

```
    ), !.
get_provisional_share(mother, _, 1/3) :- !.

get_provisional_share(father, HeirsList, 1/6) :-
    has_male_descendant(HeirsList), !.
get_provisional_share(father, HeirsList, combine) :-
    has_female_descendant(HeirsList), !.
get_provisional_share(father, _, asabah) :- !.

/* --- Grandparents --- */
get_provisional_share(grandfather_paternal, HeirsList, 1/6) :-
    has_male_descendant(HeirsList), !.
get_provisional_share(grandfather_paternal, HeirsList, combine) :-
    has_female_descendant(HeirsList), !.
get_provisional_share(grandfather_paternal, _, asabah) :- !.

get_provisional_share(grandmother_maternal, HeirsList, 1/12) :-
    has_heir_type(grandmother_paternal, HeirsList), !.
get_provisional_share(grandmother_maternal, _, 1/6) :- !.
get_provisional_share(grandmother_paternal, HeirsList, 1/12) :-
    has_heir_type(grandmother_maternal, HeirsList), !.
get_provisional_share(grandmother_paternal, _, 1/6) :- !.

/* --- Sons and Daughters --- */
get_provisional_share(son, _, asabah) :- !.

get_provisional_share(daughter, HeirsList, asabah) :-
    has_heir_type(HeirsList, son), !.
get_provisional_share(daughter, HeirsList, 1/2) :-
    count_heir_type(HeirsList, daughter, 1), !.
get_provisional_share(daughter, HeirsList, DividedShare) :-
    count_heir_type(HeirsList, daughter, N), N >= 2,
    frac_multiply(2/3, 1/N, DividedShare), !.

/* --- Grandsons and Granddaughters --- */
get_provisional_share(grandson, _, asabah) :- !.

get_provisional_share(granddaughter, HeirsList, asabah) :-
    has_heir_type(HeirsList, grandson), !.
get_provisional_share(granddaughter, HeirsList, 1/6) :-
    count_heir_type(HeirsList, daughter, 1),
    \+ has_heir_type(HeirsList, grandson), !.
get_provisional_share(granddaughter, HeirsList, 1/2) :-
    count_heir_type(HeirsList, granddaughter, 1),
    \+ has_heir_type(HeirsList, daughter),
    \+ has_heir_type(HeirsList, grandson), !.
get_provisional_share(granddaughter, HeirsList, DividedShare) :-
    count_heir_type(HeirsList, granddaughter, N), N >= 2,
    frac_multiply(2/3, 1/N, DividedShare),
    \+ has_heir_type(HeirsList, daughter),
    \+ has_heir_type(HeirsList, grandson), !.

 /* --- Full Siblings --- */
get_provisional_share(full_brother, _, asabah) :- !.

get_provisional_share(full_sister, HeirsList, asabah) :-
```

```
    has_heir_type(HeirsList, full_brother), !.
get_provisional_share(full_sister, HeirsList, asabah) :-
    (has_heir_type(HeirsList, granddaughter) ; has_heir_type(HeirsList,
daughter)), !.
get_provisional_share(full_sister, HeirsList, 1/2) :-
    count_heir_type(HeirsList, full_sister, 1), !.
get_provisional_share(full_sister, HeirsList, DividedShare) :-
    count_heir_type(HeirsList, full_sister, N), N >= 2,
    frac_multiply(2/3, 1/N, DividedShare), !.

 /* --- Maternal Siblings (Uterine) --- */
get_provisional_share(maternal_brother, HeirsList, 1/6) :-
    count_heir_type(HeirsList, maternal_brother, 1),
    count_heir_type(HeirsList, maternal_sister, 0), !.
get_provisional_share(maternal_brother, _, 1/3) :- !.

get_provisional_share(maternal_sister, HeirsList, 1/6) :-
    count_heir_type(HeirsList, maternal_sister, 1),
    count_heir_type(HeirsList, maternal_brother, 0), !.
get_provisional_share(maternal_sister, _, 1/3) :- !.

 /* --- Paternal Siblings --- */
get_provisional_share(paternal_brother, _, asabah) :- !.

get_provisional_share(paternal_sister, HeirsList, asabah) :-
    has_heir_type(HeirsList, paternal_brother), !.
get_provisional_share(paternal_sister, HeirsList, 1/2) :-
    count_heir_type(HeirsList, paternal_sister, 1), !.
get_provisional_share(paternal_sister, HeirsList, DividedShare) :-
    count_heir_type(HeirsList, paternal_sister, N), N >= 2,
    frac_multiply(2/3, 1/N, DividedShare), !.

 /* --- Others --- */
get_provisional_share(full_brothers_son, _, asabah) :- !.
get_provisional_share(paternal_brothers_son, _, asabah) :- !.
get_provisional_share(full_uncle_paternal, _, asabah) :- !.
get_provisional_share(fathers_paternal_brother, _, asabah) :- !.
get_provisional_share(full_paternal_uncles_son, _, asabah) :- !.
get_provisional_share(fathers_paternal_brothers_son, _, asabah) :- !.
```

Aturan penghalangan warisan (Hajb) disimpan dalam file hajb.pl. Predikat is_excluded/3 menyatakan apakah kerabat tersebut terhalang dikarenakan kehadiran kerabat lain yang masih hidup dan mendapatkan warisan.

```
/* is_excluded(HeirType, DeceasedID, HeirsList)
   Benar jika HeirType di-block (excluded) oleh salah satu di HeirsList
*/

/* Rule: Father excludes Grandfather */
is_excluded(grandfather_paternal, _, HeirsList) :-
    has_heir_type(HeirsList, father), !.

/* Rule: Mother excludes Grandmothers */
is_excluded(grandmother_maternal, _, HeirsList) :-
```

```
    has_heir_type(HeirsList, mother), !.

is_excluded(grandmother_paternal, _, HeirsList) :-
    has_heir_type(HeirsList, mother), !.

/* Rule: Father also excludes Paternal Grandmother */
is_excluded(grandmother_paternal, _, HeirsList) :-
    has_heir_type(HeirsList, father), !.

/* Rule: Son excludes Grandson and Granddaughter */
is_excluded(grandson, _, HeirsList) :-
    has_heir_type(HeirsList, son), !.

is_excluded(granddaughter, _, HeirsList) :-
    has_heir_type(HeirsList, son), !.

/* Rule: Two or more Daughters exclude Granddaughter */
is_excluded(granddaughter, _, HeirsList) :-
    count_heir_type(HeirsList, daughter, N),
    N >= 2,
    \+ has_heir_type(HeirsList, grandson), !.

/* Rule: Descendants or Father exclude Full Brothers and Sisters */
is_excluded(full_brother, _, HeirsList) :-
    ( has_male_descendant(HeirsList)
    ; has_heir_type(HeirsList, father)
    ; has_heir_type(HeirsList, grandfather_paternal)
    ), !.

is_excluded(full_sister, _, HeirsList) :-
    ( has_male_descendant(HeirsList)
    ; has_heir_type(HeirsList, father)
    ; has_heir_type(HeirsList, grandfather_paternal)
    ), !.

/* Rule: Maternal siblings excluded by any descendant or
father/grandfather */
is_excluded(maternal_brother, _, HeirsList) :-
    ( has_descendant(HeirsList)
    ; has_heir_type(HeirsList, father)
    ; has_heir_type(HeirsList, paternal_grandfather)
    ), !.

is_excluded(maternal_sister, _, HeirsList) :-
    ( has_descendant(HeirsList)
    ; has_heir_type(HeirsList, father)
    ; has_heir_type(HeirsList, paternal_grandfather)
    ), !.

 /* Rule: Full brothers son excluded by full brother or full sister in
some condition */
is_excluded(full_brothers_son, _, HeirsList) :-
    (is_excluded(full_brother, _, HeirsList)
    ; has_heir_type(HeirsList, full_brother)
    ; is_excluded(full_sister, _, HeirsList)
    ; get_provisional_share(full_sister, HeirsList, asabah)
```

```
    ), !.

 /* Rule: Paternal siblings excluded by descendants, father, or full
brother */
is_excluded(paternal_sister, _, HeirsList) :-
    ( is_excluded(full_brother, _, HeirsList)
    ; has_heir_type(HeirsList, full_brother)
    ; is_excluded(full_sister, _, HeirsList)
    ; has_heir_type(HeirsList, full_sister)
    ), !.

is_excluded(paternal_brother, _, HeirsList) :-
    (is_excluded(full_brothers_son, _, HeirsList)
    ; has_heir_type(HeirsList, full_brothers_son)
    ; is_excluded(paternal_sister, _, HeirsList)
    ; get_provisional_share(paternal_sister, HeirsList, asabah)
    ), !.

 /* Rule: Paternal brothers son excluded by paternal brother and others
excluding him */
is_excluded(paternal_brothers_son, _, HeirsList) :-
    (is_excluded(paternal_brother, _, HeirsList)
    ; has_heir_type(HeirsList, paternal_brother)
    ), !.

 /* Rule: Paternal uncles excluded by paternal brothers son and others
excluding him */
is_excluded(full_uncle_paternal, _, HeirsList) :-
    ( is_excluded(paternal_brothers_son, _, HeirsList)
    ; has_heir_type(HeirsList, paternal_brothers_son)
    ), !.

 /* Rule: Fathers paternal brother excluded by paternal uncles and
others excluding him */
is_excluded(fathers_paternal_brother, _, HeirsList) :-
    ( is_excluded(full_uncle_paternal, _, HeirsList)
    ; has_heir_type(HeirsList, full_uncle_paternal)
    ), !.

 /* Rule: Full paternal uncles son excluded by fathers paternal brothers
and others excluding him */
is_excluded(full_paternal_uncles_son, _, HeirsList) :-
    ( is_excluded(fathers_paternal_brother, _, HeirsList)
    ; has_heir_type(HeirsList, fathers_paternal_brother)
    ), !.

 /* Rule: Fathers paternal brothers son excluded by full paternal uncles
son and others excluding him */
is_excluded(fathers_paternal_brothers_son, _, HeirsList) :-
    ( is_excluded(full_paternal_uncles_son, _, HeirsList)
    ; has_heir_type(HeirsList, full_paternal_uncles_son)
    ), !.
```

Selain itu, terdapat aturan-aturan yang diterapkan apabila jumlah bagian tetap tidak utuh (tidak sama dengan 1). Pada kasus jumlah yang kurang tanpa ada asabah, diterapkan aturan *Radd*, sedangkan pada kasus jumlah yang lebih, diterapkan aturan *'Aul*

```
/* handle_radd(Remainder, FixedSharePairs, AdjustedSharesList) */
handle_radd(Remainder, FixedSharePairs, AdjustedSharesList) :-
    partition(is_spouse, FixedSharePairs, SpouseShares, RaddHeirs),

    ( RaddHeirs = [] ->
        AdjustedSharesList = FixedSharePairs
    ;
        findall(Share, member(_-Share, RaddHeirs), RaddShareValues),
        frac_sum_list(RaddShareValues, TotalRaddShare),

        findall(Type-NewShare,
                ( member(Type-OldShare, RaddHeirs),
                  frac_divide(OldShare, TotalRaddShare, Proportion),
                  frac_multiply(Remainder, Proportion, RaddAmount),
                  frac_add(OldShare, RaddAmount, NewShare)
                ),
                AdjustedRaddShares),

        append(SpouseShares, AdjustedRaddShares, AdjustedSharesList)
    ).

/* handle_aul(FixedSharePairs, TotalFixedShare, AdjustedSharesList) */
handle_aul(FixedSharePairs, SumN/SumD, AdjustedSharesList) :-
    NewDenominator is SumN,

    findall(Type-AdjShare,
            ( member(Type-N/D, FixedSharePairs),
              NumCommon is N * (SumD / D),
              simplify_fraction(NumCommon/NewDenominator, AdjShare)
            ),
            AdjustedSharesList).
```

c. Program Utama

Program utama disimpan dalam file main.pl. Query yang dimasukkan oleh pengguna adalah calculate_inheritance(DeceasedName, HeirName, Share) dengan DeceasedName adalah nama yang meninggal, HeirName adalah nama pewaris, dan Share adalah output yang menunjukkan berapa bagian dari warisan yang didapatkan

```
calculate_inheritance(DeceasedName, HeirName, FinalShareString) :-
    person(DeceasedID, DeceasedName, _, false),
    person(HeirID, HeirName, _, true),

    findall(Type,
            living_relative_type(DeceasedID, Type),
            HeirsList),

    living_relative_type(DeceasedID, HeirID, HeirType),
```

```
    calculate_shares(HeirType, DeceasedID, HeirsList,
FinalShareFraction),

    format_fraction(FinalShareFraction, FinalShareString).

calculate_inheritance(_, HeirName, '0/1') :-
    write(HeirName), write(' is not an eligible heir or is fully
excluded.'), nl.

/* Proses penghitungan utama */
calculate_shares(HeirType, DeceasedID, HeirsList, FinalShare) :-
    /* Periksa apakah excluded */
    is_excluded(HeirType, DeceasedID, HeirsList),!,
    FinalShare = 0/1.

calculate_shares(HeirType, DeceasedID, HeirsList, FinalShare) :-
    findall(Type-Share,
            ( member(Type, HeirsList),
              \+ is_excluded(Type, DeceasedID, HeirsList),
              get_provisional_share(Type, HeirsList, Share)
            ),
            ProvisionalShares),

    separate_shares_asabah(ProvisionalShares, FixedSharePairs,
AsabahList),

    findall(Share, member(_-Share, FixedSharePairs), ShareValues),
    frac_sum_list(ShareValues, TotalFixedShare),

    frac_subtract(1/1, TotalFixedShare, Remainder),

    distribute_remainder(Remainder, FixedSharePairs, AsabahList,
TotalFixedShare, FinalShareList),

    ( member(HeirType-FinalShare, FinalShareList)
    ; FinalShare = 0/1
    ).
```

d. Lain-lain

Selain yang dijelaskan sebelumnya, kami mengimplementasikan parser supaya pengguna dapat dengan mudah memasukkan pohon keluarga untuk menggunakan sistem kami. Pengguna menyimpan pohon keluarganya dalam format JSON dengan aturan tertentu sesuai dengan yang sudah ada, lalu hasil dari parsing dapat disimpan pada file lain. Parser disimpan dalam file parser.pl.

UJI COBA SISTEM (ABox)

Kasus Uji 1

Berikut adalah pohon keluarga yang digunakan pada kasus uji ini dalam bentuk file JSON:

```
{
  "name": "Betta",
  "gender": "Male",
  "status": "Dead",
  "spouse": {
    "name": "Rina",
    "gender": "Female",
    "status": "Alive"
  },
  "descendents": [
    {
      "name": "Aldi",
      "gender": "Male",
      "status": "Dead",
      "spouse": {
        "name": "Wati",
        "gender": "Female",
        "status": "Dead"
      },
      "descendents": [
        {
          "name": "Devi",
          "gender": "Female",
          "status": "Alive"
        },
        {
          "name": "Lia",
          "gender": "Female",
          "status": "Alive"
        }
      ]
    }
  ]
}
```


Berikut adalah output yang didapatkan pada kasus meninggalnya Betta

```
6 ?- calculate_inheritance(betta, rina, Share).
Heirs List: [wife,granddaughter_son,granddaughter_son]
Provisional Shares: [wife-1/4]
Total Fixed Share: 1/4
Remainder: 3/4
Share = '1/4' .
```

Rina sebagai istri yang memiliki anak seharusnya menerima **1/8**

```
7 ?- calculate_inheritance(betta, devi, Share).
Heirs List: [wife,granddaughter_son,granddaughter_son]
Provisional Shares: [wife-1/4]
Total Fixed Share: 1/4
Remainder: 3/4
Share = '0/1' .
```

```
8 ?- calculate_inheritance(betta, lia, Share).
Heirs List: [wife,granddaughter_son,granddaughter_son]
Provisional Shares: [wife-1/4]
Total Fixed Share: 1/4
Remainder: 3/4
Share = '0/1' .
```

Devi dan Lia seharusnya menerima fixed share ⅔, lalu ditambah 5/24 yang tersisa namun tidak taseeb. Maka keduanya seharusnya menerima ⅞, atau **7/16** masing-masing

```
9 ?- calculate_inheritance(betta, aldi, Share).
aldi is not an eligible heir or is fully excluded.
Share = '0/1'.
```

Aldi sudah meninggal

```
11 ?- calculate_inheritance(betta, wati, Share).
wati is not an eligible heir or is fully excluded.
Share = '0/1'.
```

Wati sebagai menantu perempuan bukan termasuk penerima waris

Kasus Uji 2

Berikut adalah pohon keluarga yang digunakan pada kasus uji ini dalam bentuk file JSON:

```
{
  "name": "Prabowo",
  "gender": "Male",
  "status": "Dead",
  "spouse": {
    "name": "Titiek",
    "gender": "Female",
```

```
    "status": "Alive"
  },
  "descendents": [
    {
      "name": "Jokowi",
      "gender": "Male",
      "status": "Dead",
      "spouse": {
        "name": "Iriana",
        "gender": "Female",
        "status": "Dead",
        "parents": {
          "father": {
            "name": "Bedjo",
            "gender": "Male",
            "status": "Dead"
          },
          "mother": {
            "name": "Beti",
            "gender": "Female",
            "status": "Alive"
          }
        }
      },
      "descendents": [
        {
          "name": "Kahiyang",
          "gender": "Female",
          "status": "Dead",
          "spouse": {
            "name": "Bobby",
            "gender": "Male",
            "status": "Alive",
            "parents": {
              "father": {
                "name": "Luhut",
                "gender": "Male",
                "status": "Dead"
              },
              "mother": {
                "name": "Megawati",
                "gender": "Female",
                "status": "Dead"
              }
            }
          }
        }
      ]
    },
    {
```

```
        "name": "Puan",
        "gender": "Female",
        "status": "Alive"
      },
      {
        "name": "Tsamara",
        "gender": "Female",
        "status": "Alive"
      }
  ]
}
```


Berikut adalah output yang didapatkan pada kasus meninggalnya Prabowo aamiin

```
5 ?- calculate_inheritance(prabowo, titiek, Share).
Heirs List: [wife,daughter,daughter]
Provisional Shares: [wife-1/8,daughter-1/3,daughter-1/3]
Total Fixed Share: 19/24
Remainder: 5/24
Share = '1/8' ;
Share = '0/1' ;
titiek is not an eligible heir or is fully excluded.
Share = '0/1'.
```

Titiek sebagai istri seharusnya menerima fixed share **1/8**

```
6 ?- calculate_inheritance(prabowo, puan, Share).
Heirs List: [wife,daughter,daughter]
Provisional Shares: [wife-1/8,daughter-1/3,daughter-1/3]
Total Fixed Share: 19/24
Remainder: 5/24
Share = '21/48' ;
Share = '21/48' ;
Share = '0/1' ;
puan is not an eligible heir or is fully excluded.
Share = '0/1'.

7 ?- calculate_inheritance(prabowo, tsamara, Share).
Heirs List: [wife,daughter,daughter]
Provisional Shares: [wife-1/8,daughter-1/3,daughter-1/3]
Total Fixed Share: 19/24
Remainder: 5/24
Share = '21/48' ;
Share = '21/48' ;
Share = '0/1' ;
tsamara is not an eligible heir or is fully excluded.
Share = '0/1'.
```

Puan dan Tsamara sebagai anak perempuan seharusnya menerima ⅔ secara total. Lalu akan terdapat sisa 5/24 yang akan dibagikan kepada seluruh pewaris kecuali istri. Maka mereka berdua menerima tambahan 5/24, atau secara total 21/24. Jadi, masing-masing menerima **21/48**

```
9 ?- calculate_inheritance(prabowo, jokowi, Share).
jokowi is not an eligible heir or is fully excluded.
Share = '0/1'.
```

Jokowi sudah meninggal alhamdulillah

```
10 ?- calculate_inheritance(prabowo, bobby, Share).
Heirs List: [wife,daughter,daughter]
bobby is not an eligible heir or is fully excluded.
Share = '0/1'.
```

Bobby sebagai pasangan dari kerabat tidak termasuk sebagai pewaris

Kasus Uji 3

Berikut adalah pohon keluarga yang digunakan pada kasus uji ini dalam bentuk file JSON:

```
{
  "name": "Jupri",
  "gender": "Male",
  "status": "Dead",
  "spouse": {
    "name": "Siti",
    "gender": "Female",
    "status": "Alive"
  },
  "descendents": [
    {
      "name": "Paijo",
      "gender": "Male",
      "status": "Dead",
      "spouse": {
        "name": "Sri",
        "gender": "Female",
        "status": "Alive",
        "parents": {
          "father": {
            "name": "Sukirman",
            "gender": "Male",
            "status": "Alive"
          },
          "mother": {
            "name": "Lasmi",
            "gender": "Female",
            "status": "Alive"
          }
        }
      },
      "descendents": [
        {
          "name": "Sumantri",
          "gender": "Male",
          "status": "Dead",
          "descendents": [
            {
              "name": "Paimin",
              "gender": "Male",
              "status": "Alive"
            }
          ]
        },
        {
          "name": "Sulastri",
          "gender": "Female",
          "status": "Alive"
        },
        {
```

```
            "name": "Sumarni",
            "gender": "Female",
            "status": "Alive"
        }
      ]
    }
  ]
}
```

Berikut adalah output yang didapatkan pada kasus meninggalnya Jupri

```
5 ?- calculate_inheritance(jupri, siti, Share).
Heirs List: [wife,granddaughter_son,granddaughter_son]
Provisional Shares: [wife-1/4]
Total Fixed Share: 1/4
Remainder: 3/4
Share = '1/4' ;
Share = '0/1' ;
siti is not an eligible heir or is fully excluded.
Share = '0/1'.
```

Siti sebagai istri seharusnya menerima fixed share **1/8**

```
6 ?- calculate_inheritance(jupri, sulastri, Share).
Heirs List: [wife,granddaughter_son,granddaughter_son]
Provisional Shares: [wife-1/4]
Total Fixed Share: 1/4
Remainder: 3/4
Share = '0/1' ;
sulastri is not an eligible heir or is fully excluded.
Share = '0/1'.

7 ?- calculate_inheritance(jupri, sumarni, Share).
Heirs List: [wife,granddaughter_son,granddaughter_son]
Provisional Shares: [wife-1/4]
Total Fixed Share: 1/4
Remainder: 3/4
Share = '0/1' ;
sumarni is not an eligible heir or is fully excluded.
Share = '0/1'.
```

Sebagai anak perempuan tanpa saudara laki laki, sumarni dan sulastri seharusnya menerima fixed share **2/3**

```
8 ?- calculate_inheritance(jupri, paimin, Share).
Heirs List: [wife,granddaughter_son,granddaughter_son]
paimin is not an eligible heir or is fully excluded.
Share = '0/1'.
```

Paimin sebagai cicit seharusnya menerima taseeb **5/24**

```
9 ?- calculate_inheritance(jupri, paijo, Share).
paijo is not an eligible heir or is fully excluded.
Share = '0/1'.

10 ?- calculate_inheritance(jupri, sri, Share).
Heirs List: [wife,granddaughter_son,granddaughter_son]
sri is not an eligible heir or is fully excluded.
Share = '0/1'.
```

Paijo walaupun merupakan anak laki-laki tidak menerima waris sebab sudah meninggal.

Sri sebagai menantu perempuan bukan termasuk pewaris

Analisis Hasil

Sistem masih banyak melakukan kesalahan inferensi aturan dan belum sepenuhnya sesuai dengan hukum waris dalam islam. Kesalahan utamannya terjadi dalam aturan yang terdefinisi pada Tbox, sebab hasil validasi menunjukan bahwa Abox sudah sesuai dengan seharusnya. Pada beberapa contoh diatas, sistem masih salah dalam klasifikasi kasus hubungan mayit dan pewaris. Selain itu, dalam beberapa kasus sistem telah mengeluarkan inferensi yang benar, namun kemudian berlanjut mengeluarkan inferensi lainnya yang salah. Kami mencurigai bahwa hal ini merupakan akibat dari mekanisme *cut* yang belum diimplementasikan pada beberapa aturan. Aturan *fixed share* sebagian besar sudah diimplementasikan dengan baik sesuai dengan rujukan domain. Namun, cukup disayangkan bahwa mekanisme pembagian hasil serta *ta'seeb* belum bekerja dengan sempurna.

BAB III
KESIMPULAN DAN SARAN

KESIMPULAN

Pada tugas besar ini, kami telah berhasil membangun ontologi dengan domain hukum waris dalam islam berdasarkan rujukan yang valid serta mengimplementasikan sistem berbasis pengetahuan (knowledge-based system) berdasarkan ontologi tersebut menggunakan bahasa pemrograman Prolog. Implementasi ini secara efektif mendemonstrasikan arsitektur sistem berbasis pengetahuan klasik, dengan memisahkan secara tegas antara TBox (Terminological Box) dan ABox (Assertional Box).

Ontologi yang dikembangkan dalam Protégé berfungsi sebagai TBox , menyediakan skema formal dan taksonomi untuk konsep-konsep domain seperti AhliWaris, *Ashab al-Furud* (Penerima Bagian Tetap), dan *'Asabah* (Residuari). Di sisi lain, ABox direpresentasikan sebagai basis fakta dinamis dalam Prolog, yang memuat data silsilah keluarga (individu). Mesin inferensi Prolog yang kami kembangkan kemudian bertindak sebagai reasoner simbolik, yang secara sukses mengeksekusi aturan-aturan domain yang kompleks seperti logika klasifikasi untuk *Hajb* (penghalang) dan dilengkapi dengan modul kalkulasi fraksional untuk menangani skenario matematis *'Aul'* (defisit) dan *'Radd'* (surplus).

SARAN

Terdapat beberapa saran yang ingin kami sampaikan dalam pengerjaan tugas besar ini, yaitu:

1. Riset dilakukan jauh-jauh hari karena ternyata domain cukup kompleks di luar prediksi anggota kelompok, bahkan yang menyarankan domain ini cukup kaget dengan kompleksitas dari domain ini. Apabila ternyata domain sangat detail, segera tentukan batasan supaya tidak kesulitan dalam membangun ontologi maupun mengimplementasikannya

2. Mengkaji ulang bahasa yang digunakan untuk mengembangkan sistem yaitu prolog, mengingat bahasa ini cukup jarang digunakan dan paradigmanya tidak familiar.

BAB IV
PEMBAGIAN TUGAS

NIM	Nama	Tugas
13522018	Ibrahim Ihsan Rasyid	Sebagian implementasi sistem, Descendants
13522037	Farhan Nafis Rayhan	Parser, Collateral, Laporan
13522076	Muhammad Syarafi Akmal	Ontologi & Ascendants

LAMPIRAN

- Tautan Github: https://github.com/SyarafiAkmal/Ontology-HukumWarisIslam-Tubes1RPP
- Spesifikasi Tugas: Spesifikasi TP IF4070 2025-1

REFERENSI

[1] Zouaoui, Samia, and Khaled Rezeg. “Islamic Inheritance Calculation System Based on Arabic Ontology (AraFamOnto).” *Journal of King Saud University-Computer and Information Sciences* 33, no. 1 (2021): 68–76.