



Tesis-Sidang Akhir - EF235401

**Identifikasi Ontologi Untuk Mendeteksi
Kualitas Produk Perangkat Lunak
Berdasarkan Kualitas Tim Pengembang**

**CHANDRA KIRANA JATU INDRASWARI
6025221053**

Dosen Pembimbing
Dr. Umi Laili Yuhana S.Kom., M.Sc.
Ir. Siti Rochimah, MT., Ph.D.

**PROGRAM MAGISTER
BIDANG KEAHLIAN MANAJEMEN CERDAS INFORMASI
PROGRAM STUDI S2 TEKNIK INFORMATIKA
DEPARTEMEN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI ELEKTRO DAN INFORMATIKA CERDAS
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2024**



Tesis-Sidang Akhir - EF235401

Identifikasi Ontologi Untuk Mendeteksi Kualitas Produk Perangkat Lunak Berdasarkan Kualitas Tim Pengembang

**CHANDRA KIRANA JATU INDRASWARI
6025221053**

**Dosen Pembimbing
Dr. Umi Laili Yuhana S.Kom., M.Sc.
Ir. Siti Rochimah, MT., Ph.D.**

**PROGRAM MAGISTER
BIDANG KEAHLIAN MANAJEMEN CERDAS INFORMASI
PROGRAM STUDI S2 TEKNIK INFORMATIKA
DEPARTEMEN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI ELEKTRO DAN INFORMATIKA CERDAS
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2024**

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN TESIS

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Komputer (M.Kom.)
di
Institut Teknologi Sepuluh Nopember

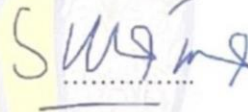
Oleh:
Chandra Kirana Jatu Indraswari
6025221053

Tanggal Ujian: 17 Juli 2024
Periode Wisuda: September 2024

Disetujui oleh:

Pembimbing

1. Dr. Umi Laili Yuhana S.Kom., M.Sc.
NIP. 197512202001122002
2. Ir. Siti Rochimah, MT., Ph.D.
NIP. 196810021994032001

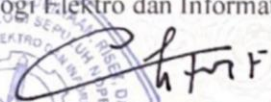


Penguji

1. Dr. Bilqis Amaliah, S.Kom., M.Kom.
NIP. 197509142001122002
2. Ir. Ary Mazharuddin Shiddiqi, S.Kom.,
M.Comp.Sc., Ph.D.
NIP. 198106202005011003
3. Ratih Nur Esti Angraini, S.Kom., M.Sc., Ph.D.
NIP. 198412102014042003



Kepala Departemen Teknik Informatika
Fakultas Teknologi Elektro dan Informatika Cerdas


Prof. Dr. Eng. Chastine Fatichah, S.Kom., M.Kom.
NIP. 197512202001122002



[Halaman ini sengaja dikosongkan]

PERNYATAAN ORISINALITAS

Yang bertanda tangan di bawah ini,

Nama Mahasiswa (NRP) : Chandra Kirana Jatu Indraswari (6025221053)
Dosen Pembimbing 1 (NIP) : Dr. Ir. Umi Laili Yuhana S.Kom., M.Sc.
(197906262005012002)
Dosen Pembimbing 2 (NIP) : Ir. Siti Rochimah, MT., Ph.D.
(196810021994032001)
Program Studi : S2 Teknik Informatika
Departemen : Departemen Teknik Informatika
Fakultas : Fakultas Teknologi Elektro dan Informatika Cerdas

Dengan ini menyatakan bahwa Tesis yang berjudul **“Identifikasi Ontologi Untuk Mendeteksi Kualitas Produk Perangkat Lunak Berdasarkan Kualitas Tim Pengembang”** adalah hasil karya sendiri, bersifat orisinal, dan ditulis dengan mengikuti kaidah penulisan ilmiah.

Apabila di kemudian hari ditemukan ketidaksesuaian dengan pernyataan ini, maka saya bersedia menerima sanksi sesuai dengan ketentuan yang berlaku di Institut Teknologi Sepuluh Nopember.

Surabaya, 20 Juni 2024

Mahasiswa,



Chandra Kirana Jatu Indraswari
NRP. 6025221028

Mengetahui,

Dosen Pembimbing 1,

Dr. Ir. Umi Laili Yuhana S.Kom., M.Sc.
NIP. 197906262005012002

Dosen Pembimbing 2,

Ir. Siti Rochimah, MT., Ph.D.
NIP. 196810021994032001

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Dengan menyebut nama Allah yang Maha Pengasih lagi Maha Penyayang. Segala puji dan syukur kehadirat Allah *Subhanahu wa Ta'ala* atas segala nikmat, rahmat, taufik, serta hidayah-Nya. Shalawat serta salam semoga tetap tercurahkan kepada Nabi Muhammad *Shallallahu 'Alaihi wa Sallam*, keluarga, sahabat, dan para pengikut setianya. *Aamiin*. Atas kehendak Allah, laporan tesis berikut yang berjudul :

"Identifikasi Ontologi Untuk Mendeteksi Kualitas Produk Perangkat Lunak Berdasarkan Kualitas Tim Pengembang"

dapat terlaksana dan diselesaikan dengan penuh tanggung jawab. Laporan tesis ini disusun sebagai salah satu syarat memperoleh gelar Magister Komputer di Departemen Teknik Informatika, Institut Teknologi Sepuluh Nopember.

Tesis ini bertujuan untuk mendeteksi kualitas produk perangkat lunak dengan berdasarkan kualitas dari tim pengembang perangkat lunak dengan menggunakan metode ontologi dan decision tree guna menentukan metode terbaik yang dapat menghasilkan nilai akurasi yang tinggi untuk mempermudah dalam menentukan kualitas produk perangkat lunak.

Penulis menyadari bahwa penyelesaian tesis ini tidak lepas dari dukungan dan bantuan dari berbagai pihak, baik secara langsung maupun tidak langsung. Untuk itu, pada kesempatan ini penulis hendak menyampaikan ucapan terima kasih dan penghargaan yang setinggi-tingginya kepada yang penulis hormati:

1. Ibu Dr. Umi Laili Yuhana S.Kom., M.Sc. selaku Pembimbing I dan Ibu Ir. Siti Rochimah, MT., Ph.D. selaku Pembimbing II, yang telah banyak memberikan bimbingan, arahan, dan dukungan selama penulis menyelesaikan tesis tanpa merasa bosan dan letih.
2. Ibu Dr. Bilqis Amaliah, S.Kom., M.Kom., Bapak Ary Mazharuddin Shiddiqi, S.Kom., M.Comp.Sc., Ph.D., dan Ibu Ratih Nur Esti Anggraini, S.Kom., M.Sc., Ph.D. selaku Tim Penguji yang banyak memberikan masukan, semangat, dan ide yang belum terpikirkan oleh penulis untuk perbaikan tesis ini.
3. Bapak Dr. Ahmad Saikhu, S.Si., MT., selaku Kepala Program Studi Pascasarjana, Departemen Teknik Informatika, Fakultas Teknologi Elektro dan Informatika Cerdas, Institut Teknologi Sepuluh Nopember yang telah memberikan kebijakan program studi serta menyediakan sarana dan prasarana yang baik.
4. Dekan Fakultas Teknologi Elektro dan Informatika Cerdas, Kepala Program Studi Teknik Informatika beserta dengan jajarannya yang telah memberikan dukungan dan bantuan sehingga mendukung kelancaran studi lanjut penulis.
5. Seluruh dosen pascasarjana Departemen Teknik Informatika, atas ilmu dan pengetahuan yang telah diberikan selama masa studi dan seluruh staf serta

laboran di Departemen Teknik Informatika yang telah membantu dan melayani selama proses penyelesaian pendidikan Magister.

6. Orang tua dan teman-teman seperjuangan kelas Mbak Chilla, Mbak Alifi, Mbak Yasinta, Mbak Anita dan serta seluruh teman kuliah di S2 Teknik Informatika ITS atas dukungan, bantuan, dan diskusinya selama menempuh pendidikan di kampus.
7. Segenap pihak yang telah membantu penulis dalam menyelesaikan tesis ini, yang tidak dapat disebutkan semuanya.

Penulis menyadari bahwa penulisan tesis ini masih jauh dari kesempurnaan, oleh karena itu kritik dan saran yang bersifat membangun sangat diharapkan penulis untuk perbaikan dan pengembangan penelitian lebih lanjut. Akhir kata, semoga hasil penelitian yang dituangkan dalam buku tesis ini dapat bermanfaat bagi perkembangan ilmu dan teknologi di kalangan akademisi dan praktisi, serta semua pihak lainnya.

Surabaya, 10 Juni 2024

Penulis,

Chandra Kirana Jatu Indraswari

IDENTIFIKASI ONTOLOGI UNTUK MENDETEKSI KUALITAS PRODUK PERANGKAT LUNAK BERDASARKAN KUALITAS TIM PENGEMBANG

Nama mahasiswa : Chandra Kirana Jatu Indraswari
NRP : 6025221053
Pembimbing I : Dr. Umi Laili Yuhana S.Kom., M.Sc.
Pembimbing II : Ir. Siti Rochimah, MT., Ph.D.

ABSTRAK

Perangkat lunak yang berfungsi dengan baik menjadi parameter kunci bagi tim yang mengadopsi metodologi *Agile* dalam proses pengembangan produk perangkat lunak. Sesuai dengan prinsip *Agile Manifesto*, faktor yang sangat penting dalam menilai kesuksesan proyek pengembangan perangkat lunak berbasis *Agile* adalah kualitas individu dan keseluruhan tim. Namun sayangnya, banyak pengembang perangkat lunak yang belum mengakui pentingnya hal ini. *The Standish Group* mengungkapkan bahwa "tim yang baik" merupakan faktor yang sangat signifikan dalam menentukan berhasil tidaknya suatu proyek perangkat lunak.

Penelitian ini akan mengusulkan prediksi dalam evaluasi yang dilakukan terhadap *Product Quality* (kualitas produk) yang dihasilkan oleh tim dengan komposisi tertentu. Pertama penelitian ini menggunakan penalaran berbasis ontologi tentang kualitas produk yang hendak diestimasi berdasarkan kepribadian dan perannya dalam tim, serta aturan-aturan untuk menerapkan pengetahuan dengan menggunakan SWRL (*Semantic Web Rules Language*). Kedua penelitian dilakukan menggunakan salah satu metode *machine learning* yaitu *Random Forest* dan *Decision Tree* untuk prediksi kualitas produk perangkat lunak. Penelitian ini difokuskan pada mahasiswa program D4 Teknik Informatika PENS di semester 5. Model yang dibangun akan dievaluasi menggunakan *cross validation* dan *confusion matriks* untuk memvalidasi hasil prediksi yang terbaik. Hasil penelitian dengan menggunakan metode ontology mendapatkan nilai akurasi sebesar 70% dan F1-Score sebesar 80% sedangkan menggunakan metode *Random Forest* dan *Decision Tree* mendapatkan nilai akurasi sebesar 66,67% dan F1-Score sebesar 76% dari hasil kedua metode dapat disimpulkan bahwa dengan metode *Ontology* jauh lebih efektif dibandingkan dengan metode *Random Forest* dan *Decision Tree* dalam identifikasi kualitas produk perangkat lunak berdasarkan tim pengembangnya.

Kata kunci: *Product Quality, Agile, Machine Learning, Ontologi*

ONTOLOGY IDENTIFICATION TO DETECT THE QUALITY OF THE SOFTWARE PRODUCT BASED ON THE QUALITY OF THE DEVELOPMENT TEAM

Name : Chandra Kirana Jatu Indraswari
Student Identity Number : 6025221053
Supervisor I : Dr. Umi Laili Yuhana S.Kom., M.Sc.
Supervisor II : Ir. Siti Rochimah, MT., Ph.D.

ABSTRACT

Well-functioning software is a key parameter for teams that adopt Agile methodology in the software product development process. In accordance with the principles of the Agile Manifesto, a very important factor in assessing the success of an Agile-based software development project is the quality of the individuals and the entire team. But unfortunately, many software developers do not yet recognize the importance of this. The Standish Group reveals that a "good team" is a very significant factor in determining the success or failure of a software project.

This research will propose predictions in the evaluation carried out on Product Quality (product quality) produced by a team with a certain composition. Firstly, this research uses ontology-based reasoning about product quality to be estimated based on personality and role in the team, as well as rules for applying knowledge using SWRL (Semantic Web Rules Language). Both studies were carried out using one of the machine learning methods, namely Random Forest and Decision Tree, to predict the quality of software products. This research is focused on students of the D4 Informatics Engineering program at PENS in semester 5. The model built will be evaluated using cross validation and confusion matrices to validate the best prediction results. The results of research using the ontology method get an accuracy value of 70% and an F1-Score of 80%, while using the Random Forest and Decision Tree methods get an accuracy value of 66.67% and an F1-Score of 76%. From the results of both methods it can be concluded that the Ontology method is much more effective than the Random Forest and Decision Tree methods in identifying the quality of software products based on the development team.

Keyword: Product Quality, Agile, Machine Learning, Ontology

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

LEMBAR PENGESAHAN.....	iv
PERNYATAAN ORISINALITAS.....	vi
KATA PENGANTAR	viii
ABSTRAK	x
ABSTRACT	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xv
DAFTAR TABEL	xvi
BAB I PENDAHULUAN	1
1.1. LATAR BELAKANG.....	1
1.2. PERUMUSAN MASALAH.....	3
1.3. TUJUAN PENELITIAN.....	3
1.4. MANFAAT PENELITIAN	4
1.5. KONTRIBUSI PENELITIAN	4
1.6. BATASAN MASALAH	4
BAB II KAJIAN PUSTAKA DAN DASAR TEORI.....	5
2.1 Kualitas Produk Perangkat Lunak.....	5
2.2 Metode Pengembangan <i>Agile</i>	6
2.3 Kerangka Kerja <i>Scrum</i>	7
2.4 <i>Project Management Antipatterns</i>	8
2.5 <i>Personality</i> MBTI	10
2.6 Penalaran Berbasis Ontologi.....	11
BAB III METODOLOGI PENELITIAN.....	15
3.1 Tahapan Penelitihan.....	15
3.1.1 Studi Literatur	15
3.1.2 Perancangan Metode	16
3.1.3 Implementasi Metode.....	20
3.2 Dataset Penelitian	21
3.3 Pemetaan Ontologi.....	24
3.4 Rencana Validasi	25
BAB IV HASIL PENELITIAN DAN PEMBAHASAN.....	29

4.1	Hasil Kinerja Metode Ontologi.....	29
4.2	Hasil <i>Query</i> Pengujian Ontologi pada <i>Protégé</i>	38
4.3	Hasil Kinerja Metode <i>Random Forest</i>	41
4.4	Hasil Pengujian Menggunakan <i>Random Forest</i>	43
4.5	Hasil Kinerja Metode <i>Decision Tree</i>	44
4.6	Hasil Pengujian Menggunakan <i>Decision Tree</i>	46
4.7	Pembahasan Hasil Penelitian.....	47
4.7.1	Pembahasan Hasil <i>Query</i> Menggunakan Ontologi	47
4.7.2	Pembahasan Hasil Kinerja Menggunakan <i>Random Forest</i>	49
4.7.3	Pembahasan Hasil Kinerja Menggunakan <i>Decision Tree</i>	50
4.7.4	Perbandingan Hasil Kinerja Ontologi, <i>Random Forest</i> dan <i>Decision Tree</i> ...	54
BAB V KESIMPULAN DAN SARAN.....		59
5.1	Kesimpulan	59
5.2	Saran.....	60
DAFTAR PUSTAKA		61
BIOGRAFI PENULIS		64

DAFTAR GAMBAR

Gambar 2. 1 Kerangka Kerja <i>Scrum</i>	8
Gambar 2. 2 Dasar <i>Personality</i> MBTI	11
Gambar 2. 3 Contoh Sempel <i>Domain</i> Ontologi	13
Gambar 3. 1 Tahapan Penelitian	15
Gambar 3. 2 Desain Ontologi	17
Gambar 3. 3 <i>Flowchart</i> Pembuatan Ontologi.....	20
Gambar 3. 4 Desain Sistem Penelitian	21
Gambar 3. 5 Dataset	22
Gambar 3. 6 Formula <i>Confusion Matrix</i>	26
Gambar 4. 1 <i>Class Team</i>	29
Gambar 4. 2 <i>Class Team Development</i>	30
Gambar 4. 3 <i>Class Team Dysfunction</i>	31
Gambar 4. 4 <i>Instances</i>	31
Gambar 4. 5 <i>Data Properties</i>	32
Gambar 4. 6 <i>SWRL Rule</i>	33
Gambar 4. 7 Hasil <i>Query</i> Pengujian Ontologi pada <i>Protégé</i>	40
Gambar 4. 8 Hasil Evaluasi Model <i>Random Forest</i>	42
Gambar 4. 9 Visualisasi Hasil Pengujian dengan <i>Random Forest</i>	44
Gambar 4. 10 Hasil Evaluasi Model <i>Decision Tree</i>	45
Gambar 4. 11 Visualisasi <i>Confusion Matrix</i> dengan <i>Decission Tree</i>	47
Gambar 4. 12 Hasil <i>Query</i> Pengujian Ontologi	48
Gambar 4. 13 Hasil Prediksi dengan <i>Random Forest</i>	50
Gambar 4. 14 Hasil Prediksi dengan <i>Decission Tree</i>	52
Gambar 4. 15 Kesimpulan Hasil Prediksi.....	54

DAFTAR TABEL

Tabel 2. 1 Referensi Parameter Relevan	14
Tabel 3. 1 Tabel <i>Confusion Matrix</i>	26
Tabel 4. 1 Hasil Komparasi Menggunakan Tiga Metode.....	56

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

1.1. LATAR BELAKANG

Konsep populer di bidang Teknologi Informasi, kini menjadi salah satu dari 10 topik teratas untuk organisasi dan manajer. *Agility* mengacu pada seberapa mudah dan cepat organisasi dapat merasakan dan merespons lingkungan pasar yang dinamis dan mengubah strategi. Oleh karena itu, topik ini banyak diadopsi dalam berbagai organisasi (K. Eilers dkk., 2022). Hal ini terkonfirmasi dalam *CHAOS report* yang dirilis oleh *Standish Group* yang menunjukkan bahwa proyek *Agile* lebih berhasil daripada proyek *Waterfall*, memiliki lebih sedikit tantangan dan lebih sedikit kegagalan (L. Khoza dkk., 2020). *Agile mindset* merupakan seperangkat sikap, perilaku, dan cara berpikir seseorang yang meningkatkan efektivitas mereka dan tim mereka dalam bekerja mengikuti nilai (*Agile Manifesto*) dan prinsip *Agile* untuk kepentingan pelanggan (J. Miller dkk., 2019).

Faktor '*people*' dalam ketangkasan disorot dalam *Manifesto Agile*, yang merupakan dasar utama untuk pengembangan perangkat lunak *agile* dinyatakan dalam nilai pertamanya "*Individuals and Interactions over Processes and Tools*". Selain itu, beberapa makalah juga telah menunjukkan bahwa aktor merupakan faktor keberhasilan penting untuk ketangkasan (K. Eilers dkk., 2022). Hal ini merupakan sesuatu yang mudah dipahami. Perangkat lunak dikembangkan oleh manusia, dibaca oleh manusia, dan dipelihara oleh manusia. Oleh karena itu, para praktisi harus mengetahui dan meningkatkan faktor manusia yang mempengaruhi anggota tim pengembangan perangkat lunak, aktivitas tim, dan, pada gilirannya akan meningkatkan kualitas produk. Namun, terkadang faktor manusia tidak mendapat perhatian yang sama dengan faktor teknis oleh manajer, anggota tim, dan akademisi di bidang Rekayasa Perangkat Lunak (E. Dutra dkk., 2021). Dalam laporan *CHAOS 2020* yang dirilis oleh *Standish Group*, "*good team*" dinyatakan sebagai faktor penting dalam menentukan keberhasilan sebuah proyek. Oleh karena itu, penilaian terhadap kualitas tim menjadi penting bagi manajemen tim (U. Sa'Adah dkk., 2021).

Kualitas produk perangkat lunak menjadi hal yang sangat penting. Kualitas produk perangkat lunak mencerminkan sejauh mana perangkat lunak tersebut memenuhi kebutuhan pengguna, berfungsi dengan baik, dan memiliki tingkat keandalan yang tinggi. Kualitas produk perangkat lunak harus diukur dikarenakan beberapa hal antara lain adalah kepuasan pengguna, kualitas perangkat lunak yang buruk dapat mengakibatkan pengguna merasa frustrasi dan tidak puas. Ini dapat berdampak negatif pada citra perusahaan dan loyalitas pelanggan (V. Berg dkk., 2020). Selain itu kualitas perangkat lunak yang buruk dapat menyebabkan kerusakan, kerugian finansial, dan bahkan risiko keselamatan dalam beberapa kasus. Oleh karena itu, keandalan perangkat lunak sangat penting. Perangkat lunak yang berkualitas rendah dapat mengurangi efisiensi dan produktivitas pengguna. Ini dapat berdampak pada waktu dan sumber daya yang terbuang. Dan yang terakhir biaya perbaikan perangkat lunak yang kurang berkualitas memerlukan perbaikan yang lebih sering, yang dapat mengakibatkan biaya tambahan bagi organisasi.

Penalaran berbasis ontologi adalah metode yang dapat digunakan untuk memprediksi kualitas produk perangkat lunak yang memiliki kelebihan diantaranya adalah pengetahuan yang mendalam. Ontologi adalah representasi formal dari pengetahuan dalam domain tertentu (Noy, N.F dkk., 2001). Dengan menggunakan ontologi, dapat memiliki pemahaman yang mendalam tentang aspek-aspek yang memengaruhi kualitas perangkat lunak. Selain itu kemampuan untuk menggabungkan pengetahuan, ontologi memungkinkan untuk menggabungkan pengetahuan dari berbagai sumber yang berbeda, termasuk pengetahuan eksplisit dan implisit. Hal ini dapat membantu dalam memahami berbagai faktor yang memengaruhi kualitas produk perangkat lunak. Dan tidak hanya itu, dengan menggunakan penalaran berbasis ontologi, dapat membangun model prediktif yang lebih akurat untuk memprediksi kualitas produk perangkat lunak. Hal ini dapat membantu dalam mengidentifikasi potensi masalah sebelum muncul. Fleksibilitas, ontologi dapat diperbarui dan disesuaikan dengan perubahan dalam domain atau kebutuhan bisnis (Noy, N.F dkk., 2001). Ini membuatnya menjadi metode yang fleksibel untuk memprediksi kualitas perangkat lunak yang terus berubah. Dengan menggunakan metode penalaran berbasis ontologi, dapat meningkatkan pemahaman tentang kualitas produk perangkat lunak dan mengambil langkah yang tepat untuk memastikan bahwa produk perangkat lunak yang

dihasilkan memiliki kualitas yang tinggi, memenuhi harapan pengguna, dan memberikan manfaat yang maksimal bagi organisasi.

Penelitian sebelumnya sudah memprediksi kualitas produk dari hasil pengembangan perangkat lunak mahasiswa berdasarkan kualitas tim. Metrik pengukuran yang digunakan menggabungkan model pengembangan tim Tuckman dan model Disfungsi Tim Lencioni (E. Barkema dkk., 2013). Dalam penelitian tersebut diperoleh kesimpulan adanya korelasi antara kualitas dari tim pengembangan perangkat lunak dengan hasil kualitas produk dimana akurasi hasil prediksi sebesar 80% melalui penelitian dan pengambilan data langsung di lapangan. Penelitian ini mengusulkan model dalam prediksi kualitas produk yang didasarkan pada 2 variabel : tipe kepribadian dan peran dalam tim perangkat lunak menggunakan penalaran berbasis ontologi. Dengan metode yang diusulkan diharapkan dapat meningkatkan akurasi prediksi kualitas produk perangkat lunak.

1.2. PERUMUSAN MASALAH

Berdasarkan latar belakang masalah yang telah diuraikan di atas maka didapatkan beberapa poin masalah yang dapat dibahas dan diteliti pada tesis kali ini yaitu :

1. Bagaimana membangun penalaran berbasis *ontology* berdasarkan kualitas tim guna mengidentifikasi sebuah kualitas produk?
2. Bagaimana merumuskan *rule* yang digunakan untuk mengidentifikasi kualitas produk berdasarkan kualitas tim?
3. Bagaimana mengidentifikasi kualitas produk berdasarkan kualitas tim menggunakan salah satu metode *machine learning*?
4. Bagaimana membandingkan hasil identifikasi kualitas produk berdasarkan kualitas tim menggunakan penalaran berbasis *ontology* dengan *machine learning*?

1.3. TUJUAN PENELITIAN

Tujuan dari penelitian ini adalah membangun penalaran berbasis ontologi untuk mengidentifikasi kualitas produk yang dihasilkan, berdasarkan kombinasi komposisi dari kepribadian dan peran dari tim pengembangan perangkat lunak.

1.4. MANFAAT PENELITIAN

Manfaat dari penelitian ini adalah diharapkan bisa membantu meningkatkan kualitas produk berdasarkan kualitas tim pengembang nya yang dihasilkan melalui serangkaian langkah mitigasi yang direkomendasikan setelah mengetahui kualitas dari tim pengembang perangkat lunak.

1.5. KONTRIBUSI PENELITIAN

Kontribusi dari penelitian ini adalah memberikan metode-metode baru yang dikemas dalam sebuah pustaka untuk melakukan hal-hal sebagai berikut.

1. Metode prediksi kualitas produk perangkat lunak berdasarkan kualitas tim dengan penalaran berbasis ontologi
2. Metode prediksi kualitas produk perangkat lunak berdasarkan kualitas tim dengan menggunakan salah satu metode *machine learning*
3. Membandingkan hasil prediksi kualitas produk terbaik berdasarkan kualitas tim menggunakan metode penalaran berbasis ontologi dengan hasil prediksi menggunakan salah satu metode *machine learning*.

1.6. BATASAN MASALAH

Untuk memfokuskan permasalahan penelitian ini, batasan masalah yang ditentukan adalah sebagai berikut:

1. Penelitian dilakukan menggunakan data dari hasil penelitian yang dilakukan di Politeknik Elektronika Negeri Surabaya (PENS).
2. Penelitian dilakukan kepada mahasiswa yang mengambil mata kuliah Workshop Pengembangan Perangkat Lunak (WPPL) pada semester 5.
3. Data penelitian setiap angkatan terdapat 2 kelas paralel, dengan total jumlah tim RPL sebanyak 28 tim. Data diambil pada mahasiswa di tahun 2020-2022.
4. Data prediksi didasarkan dari aspek kategori yaitu: kepribadian, peran dari tim pengembangan perangkat lunak.
5. Pengamatan kualitas produk perangkat lunak ini terbatas berdasarkan kualitas tim pengembang.

BAB II

KAJIAN PUSTAKA DAN DASAR TEORI

Pada bab ini dijelaskan mengenai pustaka yang digunakan sebagai landasan ilmiah penelitian dan pustaka penelitian yang telah dilakukan sebelumnya terkait penelitian yang akan dilakukan. Kajian pustaka yang dilakukan mencakup kualitas produk, metode pengembangan *agile*, kerangka kerja *scrum*, *project management antipatterns*, *personality MBTI*, prediksi kualitas produk perangkat lunak, parameter kualitas produk, penalaran ontologi.

2.1 Kualitas Produk Perangkat Lunak

Kualitas produk perangkat lunak adalah elemen krusial dan rumit dalam proses pengembangan perangkat lunak yang berdampak pada kemampuan perangkat lunak untuk memenuhi kebutuhan pengguna, beroperasi secara optimal, dan memberikan nilai tambah. Menurut Brown dan koleganya (1998), definisi kualitas perangkat lunak bisa sangat beragam, tetapi umumnya mencakup kemampuan perangkat lunak untuk memenuhi standar kinerja yang ditetapkan, bebas dari cacat atau bug, dan mampu memberikan pengalaman pengguna yang memuaskan.

Karakteristik penting yang digunakan untuk menilai kualitas perangkat lunak meliputi keandalan, efisiensi, keamanan, kemudahan penggunaan, skalabilitas, dan fleksibilitas. Keandalan merujuk pada kemampuan perangkat lunak untuk berfungsi dengan benar dalam berbagai kondisi dan periode waktu yang lama. Efisiensi mengacu pada penggunaan sumber daya yang optimal seperti waktu, memori, dan daya komputasi. Keamanan memastikan bahwa perangkat lunak terlindungi dari ancaman dan kerentanan yang dapat membahayakan data dan operasi.

Kemudahan penggunaan berkaitan dengan seberapa intuitif dan mudah perangkat lunak dapat digunakan oleh pengguna akhir, sementara skalabilitas merujuk pada kemampuan perangkat lunak untuk beradaptasi dengan peningkatan beban kerja atau jumlah pengguna tanpa penurunan kinerja. Fleksibilitas mengacu pada kemampuan perangkat lunak untuk diubah atau ditingkatkan dengan mudah guna memenuhi

kebutuhan atau kondisi baru. Semua karakteristik ini sangat penting dalam mengukur dan memastikan kualitas perangkat lunak.

Kualitas perangkat lunak yang buruk dapat mengakibatkan konsekuensi serius seperti ketidakpuasan pengguna, kerugian finansial, dan bahkan risiko keselamatan. Oleh karena itu, pemahaman yang mendalam tentang konsep kualitas perangkat lunak serta kemampuan untuk mengukur dan memastikan kualitas yang tinggi adalah hal yang sangat penting dalam pengembangan perangkat lunak yang sukses dan berkelanjutan. Sa'Adah dan koleganya (2021) menekankan pentingnya hal ini dalam konteks modern pengembangan perangkat lunak, di mana standar dan harapan kualitas terus meningkat. (U. Sa'Adah dkk., 2021).

2.2 Metode Pengembangan Agile

Dunia saat ini sedang berada pada era VUCA (*Volatility, Uncertainty, Complexity, Ambiguity*). Kondisi ini menuntut kelincahan (*agility*) agar bisa segera beradaptasi dengan keadaan yang cepat berubah, tidak bisa diprediksi, kompleks dan tidak mudah diinterpretasi (A. Sidky dkk., 2007). *Agility* sebagai sebuah konsep mengacu pada kemampuan organisasi, tim, atau proyek untuk bereaksi terhadap perubahan yang terjadi pada mereka. Dalam pengertian umum, kelincahan dapat didefinisikan sebagai "kemampuan untuk bereaksi dan beradaptasi dengan perubahan yang diharapkan dan tidak terduga dalam lingkungan yang dinamis secara konstan dan cepat, dan untuk menggunakan perubahan tersebut (jika mungkin) sebagai keuntungan" (G. C. Flores dkk., 2021). Dalam pengembangan perangkat lunak, metode *Agile* telah terbukti menjadi alat yang ampuh ketika tujuannya adalah untuk membangun model bisnis yang sukses dan menguntungkan. Ketika sebuah perusahaan perlu dengan cepat memenuhi kebutuhan pasar dan pelanggan.

Agile telah terbukti jauh lebih efektif dibandingkan dengan proses tradisional yang kaku dan penuh upacara. Sejak munculnya Manifesto Agile pada tahun 2001, yang dirumuskan oleh Kent Beck dan rekan-rekannya, prinsip dan praktik metodologi Agile telah menjadi serangkaian praktik yang sangat populer dalam industri perangkat lunak. Agile menggantikan metode pengembangan perangkat lunak tradisional yang cenderung kaku dan berat selama dekade terakhir. Penggunaan Agile dalam rekayasa perangkat

lunak telah menjadi area penelitian yang luas, yang diilustrasikan oleh banyaknya literatur yang tersedia (A. I. Böhmer dkk., 2016).

Berbagai manfaat telah dilaporkan terkait penggunaan Agile dalam beberapa bidang penting. Agile mendorong kolaborasi yang lebih baik dengan pelanggan, memperbaiki proses kerja untuk menangani cacat, dan mempromosikan belajar melalui pemrograman berpasangan. Selain itu, Agile menekankan pentingnya berpikir ke depan untuk manajemen, menjaga fokus pada pekerjaan saat ini bagi insinyur, dan memberikan estimasi yang lebih akurat (A. I. Böhmer dkk., 2016). Pendekatan ini menawarkan fleksibilitas dan responsivitas yang lebih besar terhadap perubahan kebutuhan dan lingkungan proyek, yang sangat berharga dalam industri perangkat lunak yang dinamis.

Studi-studi tentang pengembangan Agile sering kali menyoroti faktor manusia sebagai tema yang berulang. Faktor-faktor seperti dinamika tim, koordinasi antar tim, dan keterlibatan pelanggan memainkan peran kunci dalam keberhasilan pengembangan Agile. Pengaruh interaksi manusia ini terhadap proses Agile sangat signifikan, karena metodologi Agile sangat bergantung pada komunikasi yang efektif dan kolaborasi. Dengan demikian, memahami dan mengelola aspek-aspek manusia ini menjadi krusial dalam penerapan dan pengembangan metodologi Agile yang sukses.

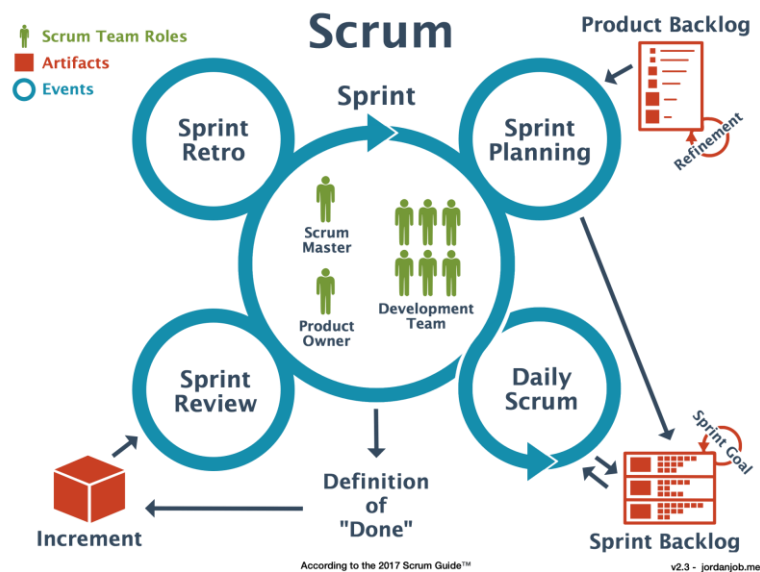
2.3 Kerangka Kerja *Scrum*

Scrum adalah sebuah kerangka kerja yang ringan dan fleksibel yang dirancang untuk membantu individu, tim, dan organisasi menghasilkan nilai melalui solusi yang adaptif untuk menangani masalah-masalah yang kompleks (K. Schwaber dkk., 2020). Dalam Scrum, sebuah kelompok terdiri dari orang-orang dengan keterampilan dan keahlian yang beragam, yang secara kolektif dapat melakukan pekerjaan yang diperlukan. Mereka juga dapat berbagi atau memperoleh keterampilan tambahan sesuai kebutuhan. Keberhasilan penerapan Scrum sangat bergantung pada kemampuan anggota tim untuk menguasai lima nilai inti Scrum: Komitmen, Fokus, Keterbukaan, Rasa Hormat, dan Keberanian.

Penerapan Scrum membutuhkan seorang Scrum Master yang bertanggung jawab untuk menciptakan dan memelihara lingkungan kerja yang optimal. Dalam lingkungan ini, Product Owner bertugas untuk mengatur dan mengurutkan pekerjaan yang

diperlukan untuk menyelesaikan masalah kompleks ke dalam Product Backlog. Tim Scrum kemudian memilih sebagian pekerjaan dari backlog ini dan mengubahnya menjadi nilai yang meningkat atau Increment selama periode yang disebut Sprint. Proses ini berulang setiap Sprint, di mana tim dan pemangku kepentingan memeriksa hasil yang telah dicapai dan membuat penyesuaian untuk Sprint berikutnya.

Secara keseluruhan, Scrum adalah metode yang mendorong kolaborasi, komunikasi, dan adaptasi berkelanjutan. Dengan berfokus pada nilai-nilai inti dan peran yang jelas, tim Scrum dapat bekerja lebih efektif dalam lingkungan yang dinamis dan berubah-ubah. Gambar 2.2 dalam dokumen asli memberikan gambaran visual tentang kerangka kerja Scrum, menggambarkan alur kerja dan interaksi antara berbagai elemen dan peran dalam Scrum.



Gambar 2. 1 Kerangka Kerja *Scrum*

2.4 Project Management Antipatterns

Project Management Antipatterns adalah salah satu cabang keahlian dalam *Project Management* yang berfokus pada keahlian manajer proyek IT dalam melakukan manajemen perorangan (*people management*) dan manajemen lingkungan kerja (*environmental management*) dari tim yang bekerja dalam perusahaan pengembangan

software (G. C. Flores dkk., 2021). Sebuah proyek memerlukan seorang pimpinan yang disebut sebagai *Project Manager*. Dalam dunia manajemen proyek di industri IT, *Project Manager* adalah orang yang bertanggung jawab terhadap kesuksesan perencanaan dan eksekusi suatu proyek pengembangan perangkat lunak yang berfokus pada 3 peran: *people management*, *software process management*, dan *external relationship*. *People management* masuk ke dalam salah satu bagian dari *Project Management* dan menempati urutan keempat dari sepuluh penyebab utama kegagalan *software development*. Buruknya keahlian dalam bidang manajerial yang dimiliki oleh seorang *Project Manager* dalam industri IT dapat menjadi salah satu penyebab utama kegagalan tersebut. Selain itu, permasalahan mengenai staf atau anggota tim juga menjadi penyebab lainnya. Tim pengembang perangkat lunak adalah sekumpulan individu yang melakukan aktivitas pengembangan yang melibatkan banyak aktivitas manusia (*human-centric*) yang penuh dengan ketidakpastian (G. C. Flores dkk., 2021).

Ketidakpastian dalam konteks tim mencakup berbagai aspek kepribadian dan keahlian yang dimiliki oleh setiap anggotanya (A. C. D. Batista dkk., 2020). Setiap anggota tim memiliki kepribadian dan keahlian yang berbeda, yang mencakup hard skill dan soft skill. Hard skill biasanya merujuk pada keahlian teknis seperti pemrograman serta desain visual dan pengalaman pengguna (UI/UX) dari produk perangkat lunak yang dikembangkan. Sementara itu, soft skill mencakup keterampilan sosial seperti kepribadian, kemampuan manajerial, dan kemampuan bekerja dalam tim (Y. Lindsjorn dkk., 2018).

Keahlian teknis atau hard skill mencakup berbagai kemampuan spesifik seperti pengkodean program dan desain UI/UX. Penguasaan hard skill ini penting untuk memastikan bahwa perangkat lunak yang dikembangkan berkualitas tinggi dan memenuhi kebutuhan pengguna. Sementara itu, soft skill berfokus pada aspek sosial dan interaksi antar anggota tim. Keterampilan ini meliputi kemampuan untuk berkomunikasi secara efektif, mengelola konflik, serta membangun dan memelihara hubungan kerja yang baik dalam tim.

Pada ranah soft skill, manajemen tim dan pembangunan budaya kerja yang positif sangat penting. Kompetensi dalam manajemen proyek, terutama dalam mengelola setiap individu dalam tim dan membangun teamwork yang efektif, memainkan peran krusial. *Project Management Antipatterns*, yang mencakup praktik-praktik terbaik untuk

menghindari kesalahan umum dalam manajemen proyek, sangat penting dalam konteks ini (M. Hoegl dkk., 2001). Keterampilan manajerial yang baik dan kemampuan untuk mengembangkan budaya tim yang kuat dapat mengurangi ketidakpastian dan meningkatkan kinerja keseluruhan tim.

2.5 Personality MBTI

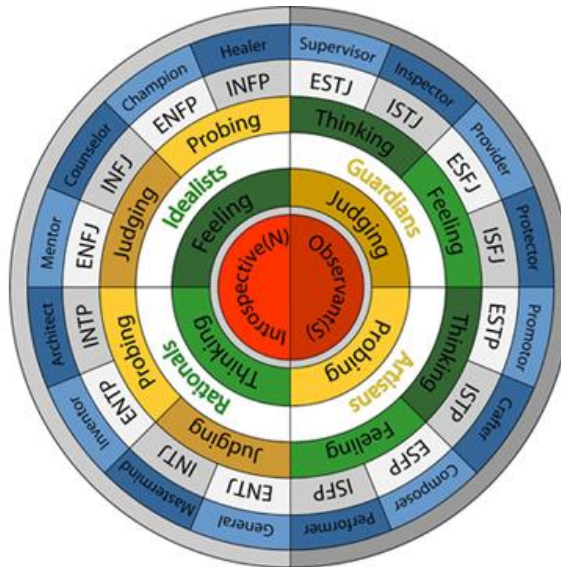
MBTI (Myers-Briggs Type Indicator) adalah psikotes yang dikembangkan untuk mengukur preferensi psikologis seseorang dalam memahami dunia dan membuat keputusan. Tes ini diciptakan oleh Isabel Briggs Myers sejak tahun 1940. MBTI dirancang untuk mengukur kecerdasan individu, bakat, dan tipe kepribadian seseorang. MBTI adalah salah satu instrumen yang paling banyak digunakan untuk tujuan ini (David Keirsej dkk., 1998). Tes ini telah diperbarui dan divalidasi secara ketat selama lebih dari tujuh puluh tahun, menjadikannya alat yang andal dalam berbagai konteks.

MBTI didasarkan pada teori tipe kepribadian dari Carl Gustav Jung, yang pertama kali menulis tentang hal ini dalam bukunya "Psychological Types" pada tahun 1921. Tujuan utama MBTI adalah membuat teori tipe psikologis yang dijelaskan oleh Carl Jung dapat dipahami dan digunakan dalam kehidupan sehari-hari. Tes ini membantu individu memahami karakter dan preferensi psikologis mereka, yang dapat diterapkan dalam berbagai aspek kehidupan, termasuk pengembangan pribadi dan profesional. Hingga saat ini, MBTI adalah tes kepribadian yang paling banyak digunakan di dunia, selain tes Enneagram.

Selain itu, MBTI sering digunakan untuk mengevaluasi kepribadian karyawan di perusahaan, dengan tujuan menempatkan mereka pada posisi yang memungkinkan potensi mereka berkembang secara optimal. Berdasarkan teori tipologi yang diusulkan oleh Carl Gustav Jung dalam bukunya "Psychological Types", yang diterbitkan pada tahun 1921, MBTI mengidentifikasi empat fungsi psikologis utama yang digunakan manusia dalam menjalani kehidupan: sensasi (sensation), intuisi (intuition), perasaan (feeling), dan pemikiran (thinking). Dasar dari kepribadian MBTI ini dapat dijelaskan lebih lanjut dalam ilustrasi pada Gambar 2.3.

MBTI memberikan wawasan mendalam mengenai cara individu berinteraksi dengan dunia dan orang-orang di sekitar mereka. Alat ini menyediakan kerangka kerja yang bermanfaat untuk memahami kekuatan dan kelemahan pribadi, serta bagaimana

seseorang dapat berkontribusi secara efektif dalam tim dan organisasi. Dengan pemahaman yang lebih baik tentang tipe kepribadian, MBTI memungkinkan individu dan organisasi untuk memanfaatkan perbedaan dan keragaman kepribadian. Hal ini membantu dalam mencapai tujuan bersama dengan lebih efektif, dengan menghargai dan mengintegrasikan berbagai tipe kepribadian dalam lingkungan kerja atau sosial.



Gambar 2. 2 Dasar *Personality* MBTI

2.6 Penalaran Berbasis Ontologi

Ontologi adalah representasi formal yang digunakan untuk menggambarkan entitas, konsep, dan hubungan dalam suatu *domain* pengetahuan. Ontologi membantu mengorganisir informasi dalam struktur hierarki, memungkinkan komputer untuk memahami makna dan konteks data (Noy dkk., 2001). Dalam penelitian ini, ontologi menjadi dasar untuk penalaran komputasional yang memungkinkan sistem untuk mengambil keputusan dan melakukan inferensi. Penalaran berbasis ontologi memanfaatkan struktur ontologi untuk melakukan penalaran komputasional. Prinsip dasar penalaran ini adalah konsistensi dan inferensi. Sistem komputer menggunakan ontologi untuk memastikan konsistensi antara data yang berbeda dan kemudian melakukan inferensi untuk menghasilkan informasi baru atau menyimpulkan informasi yang tidak eksplisit terdapat dalam data. Salah satu keunggulan penalaran berbasis

ontologi adalah kemampuannya dalam mengatasi perbedaan format dan interpretasi dalam berbagai sumber informasi. Dengan memiliki ontologi yang terdefinisi dengan baik, sistem komputer dapat mengintegrasikan data dari berbagai sumber yang heterogen. Selain itu, penalaran berbasis ontologi juga memungkinkan sistem untuk mengeksplorasi hubungan yang kompleks antara entitas dalam ontologi (Noy dkk., 2001).

Penalaran berbasis ontologi memiliki beragam aplikasi praktis. Misalnya, dalam bidang temu kembali informasi semantik, sistem dapat memberikan hasil pencarian yang lebih relevan dengan memahami makna konsep dalam ontologi. Dalam analisis data besar, ontologi dapat membantu dalam pemrosesan dan penggalian data yang lebih efisien. Selain itu, dalam pengembangan sistem cerdas di berbagai industri, penalaran berbasis ontologi dapat digunakan untuk meningkatkan kualitas pengambilan keputusan. Penting untuk mencatat bahwa dalam konteks penelitian ini, penalaran berbasis ontologi memiliki peran kunci dalam mengembangkan model atau sistem yang mampu mengatasi permasalahan tertentu (Noy dkk., 2001). Penelitian ini akan memanfaatkan konsep penalaran berbasis ontologi sebagai salah satu landasan teoritis utama untuk merancang dan mengimplementasikan solusi yang akan diuji dalam penelitian empiris. Dengan memahami konsep dasar dan manfaat dari penalaran berbasis ontologi, penelitian ini dapat menghasilkan kontribusi yang signifikan dalam pemahaman dan aplikasi teknologi berbasis ontologi.

Membuat asumsi yang jelas tentang *domain* yang mendukung implementasi ini memungkinkan fleksibilitas dalam mengubahnya jika pengetahuan tentang *domain* berubah. Penciptaan kode keras yang mengasumsikan karakteristik dunia dalam bahasa pemrograman dapat menyulitkan pencarian dan pemahaman, serta menjadi sulit untuk dimodifikasi, terutama bagi individu yang tidak memiliki keahlian dalam pemrograman. Selain itu, memiliki spesifikasi pengetahuan *domain* yang tegas juga bermanfaat untuk pengguna yang baru dalam *domain* tersebut, membantu mereka memahami makna istilah-istilah yang digunakan dalam *domain* tersebut (A. I. Böhmer dkk., 2016). Berikut adalah contoh simple *domain* ontologi ditunjukkan pada Gambar 2.4



Gambar 2. 3 Contoh Sempel *Domain Ontologi*

Pengetahuan domain sering kali diorganisir menggunakan ontologi, yang merupakan praktik umum dalam pengetahuan operasional. Ontologi memungkinkan penjelasan tugas dengan mengonfigurasi produk dari komponennya sesuai dengan spesifikasi yang diperlukan, serta mengimplementasikan program yang melakukan konfigurasi ini secara independen dari produk dan komponen tersebut (McGuinness dan Wright, 1998). Misalnya, ontologi dapat dikembangkan untuk komponen dan karakteristik PC, kemudian algoritma yang sesuai dapat diterapkan untuk mengonfigurasi PC sesuai kebutuhan. Selain itu, algoritma yang sama juga dapat digunakan untuk mengonfigurasi elevator jika ontologi komponen elevator diimplementasikan pada program tersebut (Rothenfluh et al., 1996).

Sering kali, pembuatan ontologi dalam domain tertentu bukanlah tujuan utama. Proses ini lebih mirip dengan mendefinisikan sekumpulan data dan strukturnya yang kemudian digunakan oleh program-program lain. Metode pemecahan masalah, aplikasi independen domain, dan agen perangkat lunak menggunakan ontologi dan basis pengetahuan yang dibangun dari ontologi sebagai sumber data mereka. Sebagai contoh, dalam pengembangan ontologi silsilah keluarga, informasi implisit dapat diperoleh dari data yang sudah terdefinisi atau aturan yang telah dibangun sebelumnya.

Contoh lainnya adalah penentuan hubungan paman dalam silsilah keluarga menggunakan aturan SWRL (Semantic Web Rule Language) dengan tools seperti Protégé. Dalam kasus ini, ontologi tidak hanya menyediakan struktur data tetapi juga mendukung implementasi algoritma yang memungkinkan inferensi otomatis, mengungkapkan informasi baru yang tidak secara eksplisit dinyatakan dalam data asli.

Berikut merupakan tabel beberapa referensi yang relevan dengan penelitian yang akan dilakukan berdasarkan parameter yang digunakan, untuk detailnya bisa informasi ditunjukkan pada Tabel 2.1 dibawah ini.

Tabel 2. 1 Referensi Parameter Relevan

No	Judul Paper	Jurnal/Conference	Parameter
1	Generating Team Quality Formula to Predict Product Quality in Software Engineering Projects of College Students	The 13th International Conference on Information & Communication Technology and System (ICTS 2021), Oktober 2021	Team Development Team Dysfunction Product Quality (UI/UX, FE, BE, Web, Mobile, Dekstop)
2	Classifying Composition of Software Development Team Using Machine Learning Techniques	International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM) 2022, November 2022	Team Role MBTI Personality Gender Team Performance
3	Achieving agility and quality in product development – an empirical study of hardware startups	Journal of Systems and Software	Investments Planning Design Agile Methods
4	Teamwork Quality and the Success of Innovative Projects: A Theoretical Concept and Empirical Evidence	Organization Science	Communication Coordination Balance of Member Mutual Support Effort & Cohesion
5	Teamwork quality and project success in software development: A survey of agile development team	Journal of Systems and Software	Communication Coordination Balance of Member Mutual Support Effort & Cohesion

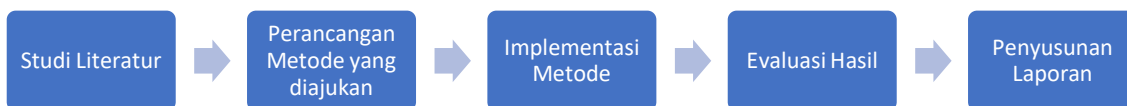
BAB III

METODOLOGI PENELITIAN

Pada bab ini terdiri dari dua bagian penjelasan yaitu rancangan penelitian dan detail rancangan Penelitian. Rancangan penelitian merupakan gambaran besar proses dalam penelitian. Sedangkan detail rancangan penelitian akan menjelaskan secara lebih detail setiap proses yang ada dalam penelitian.

3.1 Tahapan Penelitian

Pada bagian ini akan dibahas mengenai tahapan rancangan penelitian yang akan diperlihatkan berdasarkan ilustrasi. Seperti pada ilustrasi akan terdapat studi literatur, perancangan metode yang akan diajukan, implementasi metode, evaluasi hasil, dan penyusunan laporan. Setiap step dalam ilustrasi akan dijelaskan pada sub bab berikutnya. Berikut adalah ilustrasi tahapan penelitian yang ditunjukkan pada Gambar 3.1 berikut.



Gambar 3. 1 Tahapan Penelitian

3.1.1 Studi Literatur

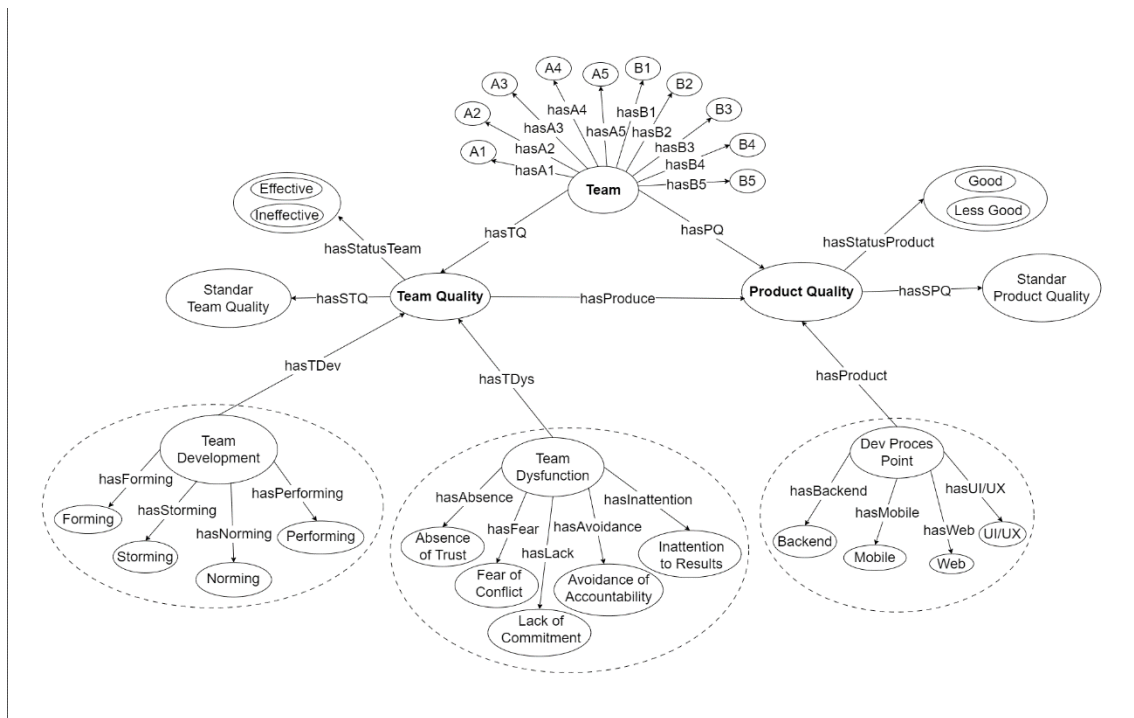
Studi literatur dilakukan dengan tujuan untuk mendapatkan konsep dasar serta sebagai referensi terkait tinjauan pustaka yang berhubungan dengan penelitian, baik dalam perancangan sistem, implementasi metode, dan evaluasi hasil yang didapatkan. Literatur yang digunakan terdiri dari buku, jurnal, artikel ilmiah, serta materi yang didapatkan dari internet.

3.1.2 Perancangan Metode

Proses ini menjelaskan tentang perancangan metode yang akan diajukan untuk memprediksi kualitas produk yang dihasilkan, berdasarkan kombinasi komposisi dari kepribadian dan peran dari tim pengembangan perangkat lunak dengan menggunakan penalaran berbasis ontologi dan *machine learning*. Kemudian hasil akhirnya adalah akan dilakukan perbandingan hasil terbaik dari dua metode tersebut.

Proses pengolahan data dengan menggunakan ontologi nantinya akan dilakukan menggunakan *tools* yang bernama *Protégé* yang merupakan sebuah aplikasi perangkat lunak yang menggunakan ontologi sebagai basisnya untuk mengelola pengetahuan. Ini dirancang untuk membantu para ahli pengetahuan dalam merancang dan membangun ontologi, memodelkan akuisisi pengetahuan, dan mengintegrasikan domain pengetahuan. *Protégé* memungkinkan visualisasi hierarki subkelas dalam bentuk pohon, mendukung multiple inheritance, dan menetapkan *class* "*THING*" sebagai *root* dalam hirarki kelas yang terbentuk. Dengan demikian, *Protégé* menyediakan proses terintegrasi untuk konseptualisasi dasar pengetahuan dan memperluas arsitektur sistem untuk memfasilitasi pemodelan pengetahuan yang lebih sederhana dan mudah dipahami melalui perubahan visual lingkungan.

Berikut gambar dibawah ini merupakan *design* ontologi yang akan dibangun dengan *classes team* yang terdiri dari kurang lebih 10 tim per tahun, lalu *team quality* yang memiliki *sub-class team development* dan *team dysfunction*, dan terakhir *class* ketiga yaitu *product quality*. *Class team development* sendiri terdiri dari empat *sub-class* yaitu *forming*, *storming*, *norming* dan *performing*. *Class team dysfunction* memiliki *sub-class* yaitu *absence of trust*, *fear of conflict*, *lack of commitment*, *avoidance of accountability* dan *inattention to results*. *Class product quality* memiliki *sub-class* yaitu *UI/UX*, *Backend*, *Frontend* dan *Mobile* dikarenakan pada WPPL Tahun 2022 setiap tim tidak mendevlop untuk versi desktop maka disamaratakan untuk Tahun 2020 dan 2021 tidak terdapat parameter *desktop* guna mempermudah mengolah data dan membandingkan hasilnya design ontologi secara lengkap bisa dilihat pada Gambar 3.2 dibawah ini.



Gambar 3. 2 Desain Ontologi

Untuk mengidentifikasi kontribusi dari masing-masing entitas kepada kualitas produk berdasarkan desain ontologi yang ditunjukkan dalam diagram, berikut adalah penjelasan detailnya:

a) Team :

Team A1-B5 merupakan aspek-aspek atau anggota tim yang berkontribusi pada kualitas tim.

b) Team Quality :

a) Team Development

- Forming merupakan tahap pembentukan tim, yang penting untuk memulai dan mengarahkan tim menuju tujuan.
- Storming merupakan tahap konflik dalam tim, yang harus dikelola untuk menjaga tim tetap efektif.
- Norming merupakan tahap penetapan norma dalam tim, yang membantu tim bekerja lebih efektif.

- Performing merupakan tahap dimana tim bekerja dengan kinerja maksimal, langsung berkontribusi pada hasil yang baik.

b) Team Dysfunction

- Absence of Trust merupakan tahap penilaian faktor ketiadaan kepercayaan yang dapat merusak kolaborasi dan produktivitas dalam tim.
- Fear of Conflict merupakan tahap penilaian faktor ketakutan akan konflik yang bisa menghambat komunikasi yang jujur dan terbuka.
- Lack of Commitment merupakan tahap penilaian faktor kurangnya komitmen yang dapat mengakibatkan rendahnya dedikasi dan usaha dalam tim.
- Avoidance of Accountability merupakan tahap penilaian dalam faktor penghindaran tanggung jawab yang bisa menyebabkan penurunan kualitas hasil.
- Inattention to Results merupakan tahap penilaian faktor ketidakpedulian terhadap hasil yang bisa menyebabkan kegagalan dalam mencapai tujuan yang ditetapkan.

c) Product Quality

c) Dev Process Point

- Backend merupakan aspek pengembangan backend yang berpengaruh pada performa dan stabilitas produk.
- Mobile merupakan aspek pengembangan aplikasi mobile yang berpengaruh pada aksesibilitas dan pengalaman pengguna.
- Web merupakan aspek pengembangan web yang berpengaruh pada interaktivitas dan kecepatan akses.

- UI/UX merupakan aspek desain antarmuka dan pengalaman pengguna yang sangat mempengaruhi kepuasan pengguna akhir.

d) Standar Team Quality

Mencakup standar yang harus dipenuhi oleh tim untuk dianggap memiliki kualitas yang baik.

e) Standar Product Quality (Standar Kualitas Produk)

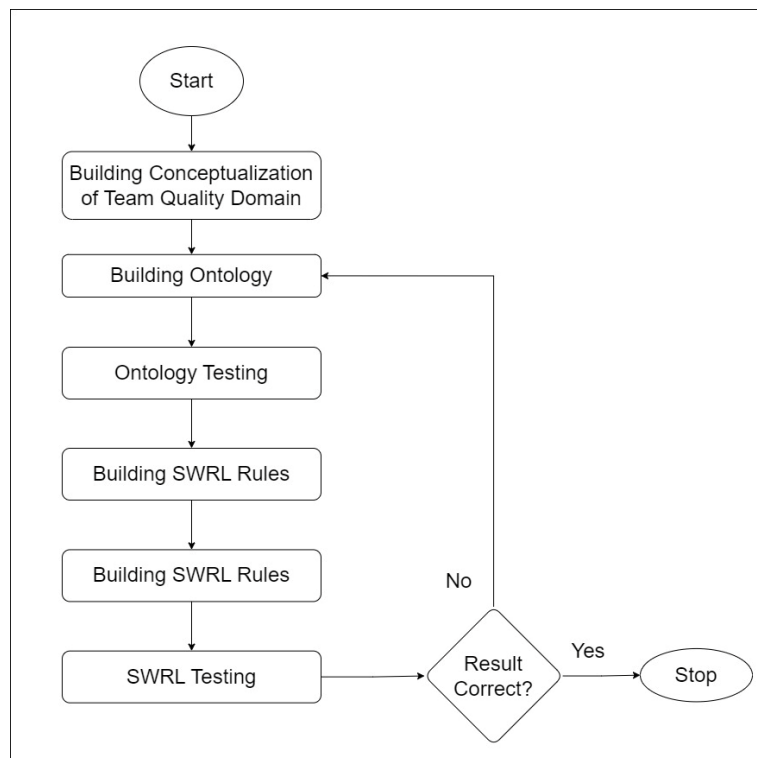
Mencakup standar yang harus dipenuhi oleh produk untuk dianggap memiliki kualitas yang baik.

Kontribusi Entitas terhadap Kualitas Produk :

1. Tim melalui anggotanya (A1-B5) secara langsung mempengaruhi kualitas tim yang kemudian berdampak pada kualitas produk.
2. Kualitas Tim yang baik (melalui pengembangan tim dan manajemen difungsi tim) akan meningkatkan efektivitas dan efisiensi dalam pengembangan produk, yang akhirnya berkontribusi pada kualitas produk.
3. Dev Proses Point memastikan bahwa setiap aspek pengembangan produk (backend, mobile, web, UI/UX) dikelola dengan baik, sehingga menghasilkan produk yang berkualitas tinggi.
4. Standar Kualitas Tim dan Standar Kualitas Produk memberikan panduan dan tolok ukur untuk memastikan bahwa tim dan produk memenuhi harapan dan standar industri yang berlaku.

Secara keseluruhan, entitas-entitas ini saling berinteraksi dan berkontribusi untuk memastikan bahwa produk yang dihasilkan memenuhi standar kualitas yang tinggi. Ontologi ini membantu dalam memahami dan mengelola hubungan antara kualitas tim dan kualitas produk secara sistematis dan terstruktur.

Pembuatan ontologi dimulai dengan langkah pertama yaitu menggambarkan secara konseptual domain kualitas tim dengan menggunakan pengetahuan yang telah dikumpulkan dari penelitian sebelumnya. Setelah itu, ontologi kualitas tim dibangun berdasarkan konseptualisasi tersebut. Kemudian, beberapa aturan *Semantic Web Rule Language (SWRL)* didefinisikan untuk melakukan penalaran terhadap ontologi yang telah dibuat. Rincian lebih lanjut mengenai alur penelitian pada tahap ini dapat dilihat pada Gambar 3.11 dibawah ini.

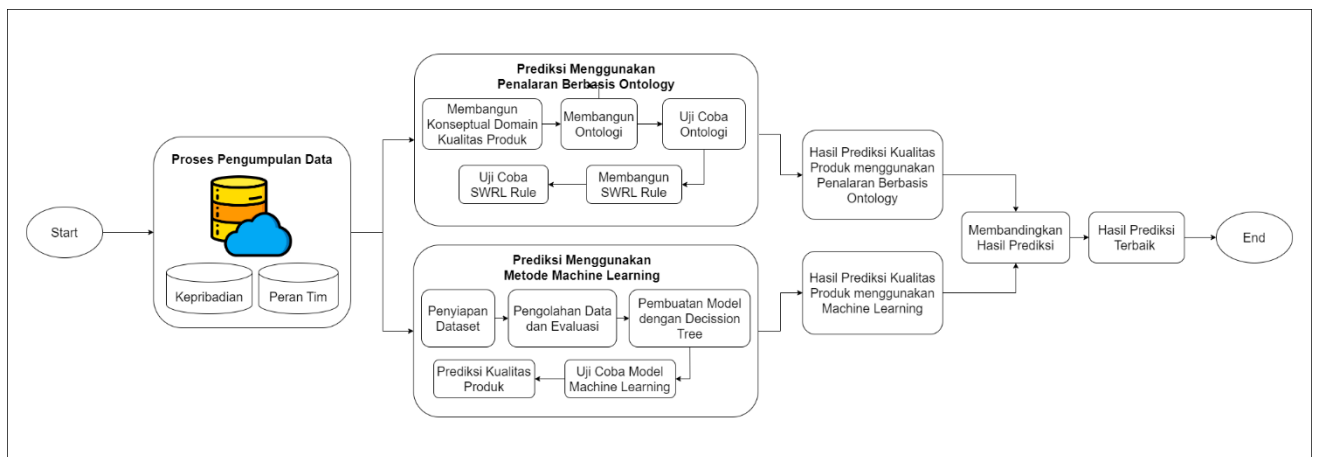


Gambar 3. 3 Flowchart Pembuatan Ontologi

3.1.3 Implementasi Metode

Proses ini akan menjelaskan implementasi metode yang akan dikerjakan tujuan akhir dari seluruh rangkaian penelitian ini adalah membangun penalaran berbasis ontologi untuk memprediksi kualitas produk yang dihasilkan, setelah itu dibandingkan dengan hasil prediksi menggunakan *machine learning* dan dari dua metode tersebut didapatkan hasil yang terbaik, prediksi tersebut diproses berdasarkan kombinasi komposisi dari kepribadian dan peran dari tim pengembangan perangkat lunak. Prediksi kualitas produk berdasarkan kualitas tim

yang sudah diselesaikan dilakukan di penelitian sebelumnya, selanjutnya adalah membangun ontologi nya. Langkah pertama adalah membangun konseptualisasi dari domain kualitas tim berdasarkan seluruh pengetahuan yang telah didapatkan dari penelitian sebelumnya. Setelah itu, akan dibangun ontologi kualitas tim berdasarkan konseptualisasi yang telah didapatkan. Selanjutnya didefinisikan beberapa aturan SWRL (*Semantic Web Rule Language*) yang akan digunakan untuk melakukan penalaran terhadap ontologi yang telah dibuat. Alur penelitian atau *design system* selengkapnya ditunjukkan pada Gambar 3.2



Gambar 3. 4 Desain Sistem Penelitian

3.2 Dataset Penelitian

Objek penelitian ini adalah mahasiswa dari Program Studi D4 Teknik Informatika Politeknik Elektronika Negeri Surabaya. Sebagai sebuah politeknik, model kurikulum masih memberlakukan sistem paket, belum sepenuhnya mengadopsi sistem SKS. Sehingga, secara default mahasiswa di setiap kelasnya akan selalu bersama selama perkuliahan dari semester 1 sampai dengan semester 8. Penelitian ini dilakukan kepada mahasiswa yang mengambil mata kuliah Workshop Pengembangan Perangkat Lunak (WPPL) di semester 5. Di angkatan terdapat 2 kelas paralel, dengan total jumlah mahasiswa kurang lebih 57 orang setiap tahun nya yang terdiri dari 8-10 tim RPL. Data diambil pada mahasiswa Teknik informatika PENS di tahun 2020-2022 yang bisa dilihat pada Gambar 3.5 berikut.

Tim	Team Development					Team Dysfunction					Normalized	Status Team Quality	Dev Proses Point					Status Product Quality	
	Forming	Storming	Norming	Performing	Kesimpulan level	Absence of Trust	Fear of Conflict	Lack of Commitment	Avoidance of Accountability	Inattention to Results			Total	UI/UX	Backend	Web	Mobile		Average
B1-2020	28	26	29	28	3	7	7	7	6	8	35	67	Ineffective	86	90	68	80	81	Good
B2-2020	25	25	26	27	4	6	6	6	6	6	30	95	Effective	96	89	60	85	83	Good
B3-2020	26	25	29	33	4	8	7	8	7	8	38	98	Effective	93	91	83	78	86	Good
B4-2020	27	25	29	32	4	8	8	8	7	7	38	98	Effective	70	90	70	78	77	Less Good
B5-2020	24	23	32	37	4	8	9	9	7	8	41	99	Effective	99	90	83	88	90	Good
A1-2021	25	23	33	36	4	8	8	8	8	7	39	98	Effective	90	84	93	78	86	Good
A2-2021	25	25	29	34	4	9	8	9	7	8	41	99	Effective	90	78	89	76	83	Good
A3-2021	27	22	32	36	4	9	8	8	8	8	41	99	Effective	83	86	87	70	81	Good
A4-2021	26	23	31	34	4	8	8	9	7	7	39	98	Effective	82	74	64	73	73	Less Good
A5-2021	29	27	31	34	4	8	8	8	7	7	38	98	Effective	81	82	74	72	77	Less Good
B1-2021	26	23	31	34	4	9	9	8	7	8	41	99	Effective	83	87	87	82	85	Good
B2-2021	25	21	32	38	4	9	9	9	7	7	41	99	Effective	90	91	89	91	90	Good
B3-2021	24	21	34	37	4	9	8	8	8	8	41	99	Effective	84	88	91	92	89	Good
B4-2021	26	26	33	35	4	9	9	8	7	8	41	99	Effective	86	87	90	90	88	Good
A1-2022	25	27	28	31	4	7	7	8	6	7	35	97	Effective	85	83	88	92	87	Good
A2-2022	21	20	26	27	4	8	8	9	7	8	40	98	Effective	86	85	95	91	89	Good
A3-2022	28	26	30	32	4	8	8	9	8	8	41	99	Effective	0	82	83	89	64	Less Good
A4-2022	28	25	28	31	4	7	7	7	6	7	34	97	Effective	89	83	87	89	87	Good
B1-2022	26	25	31	36	4	8	8	9	8	8	41	99	Effective	87	85	89	90	88	Good
B2-2022	26	24	28	31	4	7	8	8	7	7	37	98	Effective	81	75	81	88	81	Good
B3-2022	26	29	27	27	2	7	6	7	6	8	34	96	Ineffective	87	59	85	9	60	Less Good
B4-2022	26	26	27	28	4	6	7	7	6	7	33	96	Effective	80	77	87	90	83	Good

Gambar 3. 5 Dataset

Tabel yang Anda unggah menampilkan berbagai metadata yang digunakan dalam dataset untuk mengevaluasi status kualitas tim dan produk. Berikut adalah penjelasan dari setiap metadata yang terdapat dalam dataset:

1. **Tim** merupakan pengenal unik untuk setiap tim yang dievaluasi, terdiri dari kode dan tahun. Dataset mencakup tim dari berbagai tahun (2020, 2021, 2022), dengan identifikasi seperti B1-2020, A1-2021, dll.

Terdapat total 22 entri, dengan distribusi sebagai berikut:

- 5 tim dari tahun 2020
- 6 tim dari tahun 2021
- 11 tim dari tahun 2022

2. **Team Development** yang terdiri dari :

- a) Forming merupakan nilai menunjukkan tahap awal pembentukan tim.
- b) Storming merupakan nilai yang menunjukkan fase konflik dan penyesuaian antar anggota tim.
- c) Norming merupakan nilai yang menunjukkan fase di mana tim mulai bekerja sama lebih baik.
- d) Performing merupakan nilai yang menunjukkan fase di mana tim bekerja dengan efisien dan efektif.
- e) Kesimpulan Level merupakan nilai total dari tahap perkembangan tim.

Nilai pada kolom "Forming", "Storming", "Norming", dan "Performing" berkisar antara 23 hingga 34. Kesimpulan level berkisar antara 4 hingga 8, yang menandakan berbagai tingkat perkembangan tim.

3. **Team Dysfunction** yang terdiri dari :

- a) **Absence of Trust** merupakan nilai yang menunjukkan tingkat ketidakpercayaan di dalam tim.
- b) **Fear of Conflict** merupakan nilai yang menunjukkan ketakutan anggota tim untuk menghadapi konflik.
- c) **Lack of Commitment** merupakan nilai yang menunjukkan kurangnya komitmen anggota tim terhadap tujuan.
- d) **Avoidance of Accountability** merupakan nilai yang menunjukkan penghindaran tanggung jawab oleh anggota tim.
- e) **Inattention to Results** merupakan nilai yang menunjukkan ketidakpedulian anggota tim terhadap hasil akhir.
- f) **Total** merupakan nilai total dari disfungsi tim.

Nilai pada kolom "Absence of Trust", "Fear of Conflict", "Lack of Commitment", "Avoidance of Accountability", dan "Inattention to Results" berkisar antara 2 hingga 9. Total nilai dysfunction berkisar antara 35 hingga 44. Nilai yang dinormalisasi berkisar antara 35 hingga 99.

4. **Normalized** merupakan nilai total dari disfungsi tim yang dinormalisasi.

5. **Status Team Quality** merupakan status klasifikasi kualitas tim berdasarkan nilai-nilai di atas. Terdapat dua kategori: "Ineffective" dan "Effective". Berdasarkan data, 3 tim dikategorikan sebagai "Ineffective" dan sisanya, 19 tim, sebagai "Effective".

6. **Dev Process Point** merupakan skor dari berbagai aspek proses pengembangan yang berbeda:
- a) UI/UX merupakan skor untuk aspek antarmuka pengguna dan pengalaman pengguna. Skor UI/UX berkisar antara 74 hingga 90.
 - b) Backend merupakan skor untuk aspek pengembangan backend. Skor Backend berkisar antara 69 hingga 89.
 - c) Web merupakan skor untuk pengembangan web. Skor Web berkisar antara 68 hingga 89.
 - d) Mobile merupakan skor untuk pengembangan aplikasi mobile. Skor Mobile berkisar antara 70 hingga 89.
 - e) Average merupakan nilai rata-rata dari semua skor di atas. Nilai rata-rata keseluruhan berkisar antara 76 hingga 88.
7. **Status Product Quality** merupakan klasifikasi kualitas produk berdasarkan skor proses pengembangan dengan status Good dan Less Good. Ada dua kategori: "Good" dan "Less Good". Berdasarkan data, 17 tim dikategorikan sebagai "Good" dan 5 tim sebagai "Less Good".

Dataset ini mencakup evaluasi menyeluruh dari berbagai aspek perkembangan dan disfungsi tim, serta proses pengembangan produk, yang memungkinkan analisis mendalam terhadap kualitas tim dan produk yang dihasilkan.

3.3 Pemetaan Ontologi

Pemetaan ontologi dalam konteks mendeteksi kualitas produk perangkat lunak berdasarkan kualitas tim pengembang merupakan pendekatan yang memanfaatkan struktur formal untuk merepresentasikan pengetahuan terkait. Ontologi yang merupakan suatu model yang menjelaskan konsep, hubungan, dan entitas dalam suatu domain pengetahuan. Dalam penelitian ini, ontologi digunakan untuk menggambarkan keterkaitan antara karakteristik kualitas produk perangkat lunak dengan kemampuan dan kualitas tim pengembang yang terlibat dalam proses pengembangan perangkat lunak.

Pertama, ontologi yang mencakup aspek kualitas produk perangkat lunak, seperti *team development*, *team dysfunction* dan *product quality*. Setiap aspek ini dapat diuraikan dalam ontologi sebagai entitas-entitas yang saling terkait. Selanjutnya, ontologi juga mencakup dimensi kualitas tim pengembang, seperti keahlian teknis, kolaborasi, dan manajemen proyek. Hubungan antara aspek-aspek kualitas produk dan dimensi kualitas tim pengembang dapat dijelaskan melalui properti dan relasi dalam ontologi.

Kedua, dengan menggunakan ontologi ini dapat melakukan analisis terhadap hubungan antara kualitas produk perangkat lunak dan kualitas tim pengembang. Misalnya, sistem dapat mengevaluasi sejauh mana keahlian teknis tim pengembang berkontribusi terhadap keandalan atau kinerja produk perangkat lunak. Pemetaan ontologi memberikan landasan formal untuk mengukur, menganalisis, dan merancang tindakan perbaikan untuk meningkatkan kualitas produk dengan memperhatikan faktor kualitas tim pengembang.

Ketiga, pemetaan ontologi ini juga memungkinkan mengadopsi pendekatan berbasis pengetahuan untuk mendukung pengambilan keputusan. Dengan memahami ontologi, dapat memberikan rekomendasi untuk pengembangan lebih lanjut, pelatihan tim, atau pengaturan ulang prioritas proyek untuk meningkatkan kualitas produk secara holistik. Penerapan ontologi dalam mendeteksi kualitas produk perangkat lunak ini mengintegrasikan informasi tentang tim pengembang ke dalam proses evaluasi kualitas, menciptakan pemahaman yang lebih komprehensif dan kontekstual dalam pengambilan keputusan pengembangan perangkat lunak.

3.4 Rencana Validasi

Untuk mengevaluasi kinerja model penalaran berbasis ontologi pada dataset yang digunakan untuk pengolahan data dengan tujuan melakukan deteksi ataupun prediksi yang terkategori sebagai proses klasifikasi, maka evaluasi kinerja model dilakukan dengan metode *confusion matrix* dan *cross validation*. *Confusion matrix* adalah tabel yang digunakan untuk memvisualisasikan dan merangkum kinerja algoritma klasifikasi. Ada empat nilai yang dihasilkan di dalam tabel *confusion matrix*, di antaranya *True Positive* (TP), *False Positive* (FP), *False Negative* (FN),

dan *True Negative* (TN). Ilustrasi tabel *confusion matrix* dapat dilihat pada Gambar 3.3 dibawah ini.

Tabel 3. 1 Tabel *Confusion Matrix*

		Nilai Aktual	
		Positive	Negative
Nilai Prediksi	Positive	TP	FP
	Negative	FN	TN

Keterangan:

- *True Positive* (TP) : Jumlah data yang bernilai Positif dan diprediksi benar sebagai Positif.
- *False Positive* (FP) : Jumlah data yang bernilai Negatif tetapi diprediksi sebagai Positif.
- *False Negative* (FN) : Jumlah data yang bernilai Positif tetapi diprediksi sebagai Negatif.
- *True Negative* (TN) : Jumlah data yang bernilai Negatif dan diprediksi benar sebagai Negatif.

Confusion matrix digunakan untuk menghitung nilai akurasi, *presicion*, *recall*, dan *F1-score* atau *F-measure*. Keempat metode evaluasi tersebut sangat bermanfaat untuk mengukur performa dari *classifier* atau algoritma *machine learning* yang digunakan untuk melakukan prediksi, yang masing-masing ditunjukkan pada persamaan 3.1 sampai dengan persamaan 3.4 berikut ini.

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad (3.1)$$

$$Precision = \frac{TP}{TP+FP} \quad (3.2)$$

$$Recall = \frac{TP}{TP+FN} \quad (3.3)$$

$$F1-score = \frac{2 \times Precision \times Recall}{Precision+Recall} \quad (3.4)$$

Gambar 3. 6 Formula *Confusion Matrix*

Akurasi adalah metrik yang digunakan untuk mengukur seberapa baik sebuah model dalam memprediksi data dengan benar. Ini dihitung dengan membagi jumlah prediksi benar (baik positif maupun negatif) dengan total jumlah data dalam dataset. Dengan kata lain, akurasi menunjukkan persentase dari total data yang diklasifikasikan dengan benar oleh model. Akurasi sangat berguna ketika data seimbang antara kategori positif dan negatif, namun mungkin kurang efektif ketika data tidak seimbang.

Precision adalah metrik yang digunakan untuk mengukur ketepatan prediksi positif yang dilakukan oleh sistem. Precision menunjukkan seberapa akurat model dalam mengidentifikasi objek yang sebenarnya positif dari semua prediksi positif yang dihasilkan. Ini dihitung dengan membagi jumlah prediksi positif yang benar dengan total jumlah prediksi positif. Precision menjadi penting terutama dalam situasi di mana biaya dari kesalahan positif (false positives) tinggi, seperti dalam diagnosis penyakit.

Recall adalah metrik yang menunjukkan kemampuan model dalam menemukan semua instance positif dalam dataset. Recall dihitung dengan membagi jumlah prediksi positif yang benar dengan jumlah total instance positif yang sebenarnya dalam dataset. Recall penting ketika tujuan utama adalah menangkap semua kasus positif, meskipun itu berarti menerima beberapa kesalahan positif. Ini sangat relevan dalam aplikasi di mana missing positif (false negatives) dapat berdampak besar, seperti dalam deteksi kanker.

F1-Score adalah metrik yang menggabungkan precision dan recall ke dalam satu nilai harmonis untuk memberikan gambaran yang seimbang tentang kinerja model. F1-Score dihitung sebagai harmonik rata-rata dari precision dan recall, sehingga memberikan keseimbangan antara keduanya. Ini sangat berguna ketika kita ingin mempertimbangkan keduanya, precision dan recall, secara bersamaan dalam evaluasi model. Metode cross-validation (CV) adalah pendekatan statistik yang sering digunakan untuk mengevaluasi kinerja model atau algoritma, di mana data dipisahkan menjadi dua subset: data training dan data testing. Model dilatih menggunakan subset training dan dievaluasi menggunakan subset testing, untuk memastikan bahwa model memiliki performa yang baik pada data yang tidak terlihat sebelumnya.

[Halaman ini sengaja dikosongkan]

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

Bab ini membahas mengenai hasil penelitian yang bertujuan untuk mengidentifikasi mendeteksi kualitas produk perangkat lunak berdasarkan kualitas tim pengembang dengan menggunakan metode ontologi, *random forest* dan *decision tree*. Kemudian hasil pengujian dihitung nilai akurasi guna menentukan metode yang terbaik untuk mencapai tujuan tersebut. Selain itu, hasil pengujian model berdasarkan penerapan juga menjadi bagian dari pembahasan berikut.

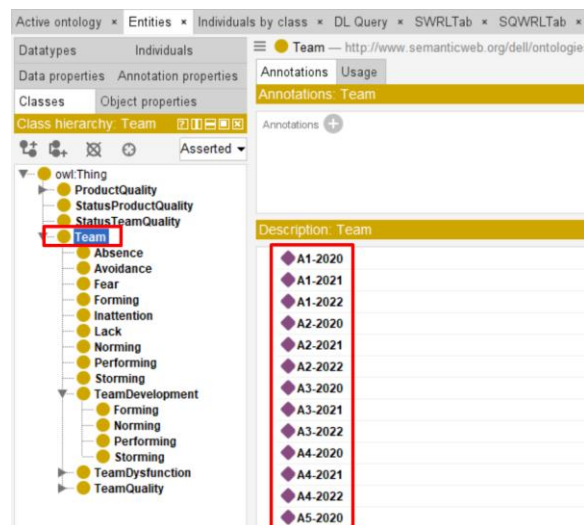
4.1 Hasil Kinerja Metode Ontologi

a. Pendeklarasian *Class*

Dalam proses mendeklarasikan sebuah kelas dalam *tools Protégé* terdapat beberapa tahapan untuk tahap pertama adalah mendeklarasikan semua *class* yang akan digunakan untuk membangun ontologi yang terdiri dari :

- *Class Team*

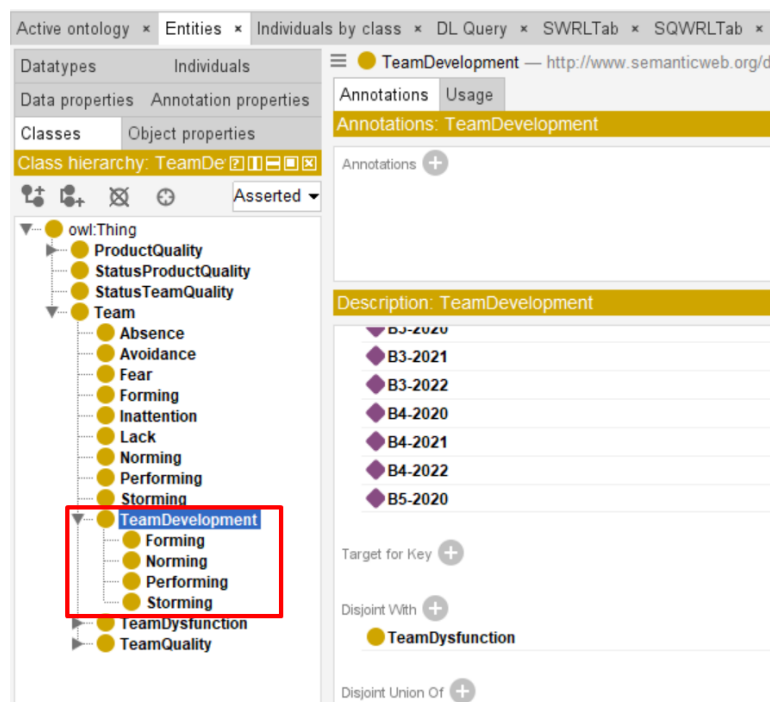
Dalam *class team* memiliki *instances* sebanyak 28 team yang terdiri dari 10 team pada tahun 2020, 10 team pada tahun 2021 dan 8 team pada tahun 2022 yang bisa dilihat pada Gambar 4.1 dibawah ini.



Gambar 4.1 *Class Team*

- *Class Team Development*

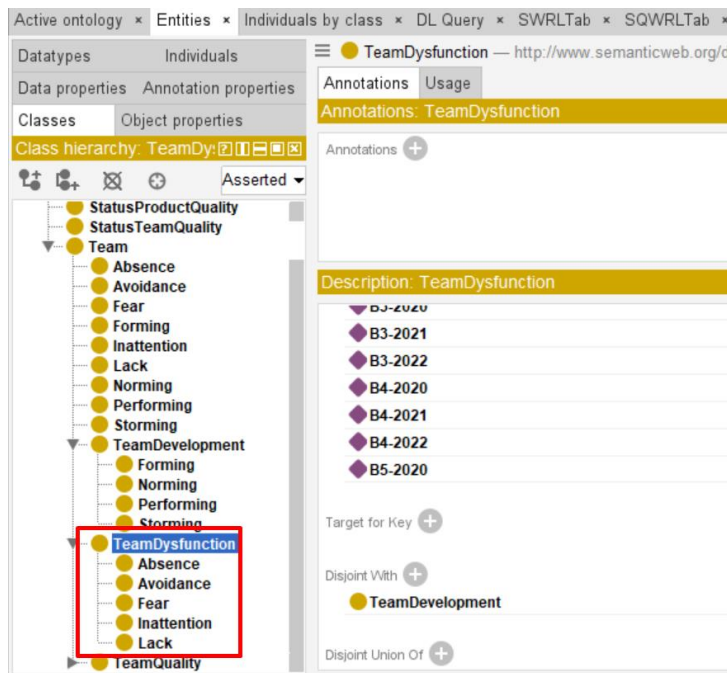
Pada tahapan ini mendefinisikan *class team development* yang memiliki *sub-class forming, norming, storming* dan *performing, team development* sendiri *disjoint* dengan *team dysfunction* dikarenakan untuk menentukan nilai *team quality*. Pada *class team development* juga terdapat *instances* dari semua team yang dihitung dari tahun 2020-2022 yang bisa dilihat pada Gambar 4.2 dibawah ini.



Gambar 4. 2 Class Team Development

- *Class Team Dysfunction*

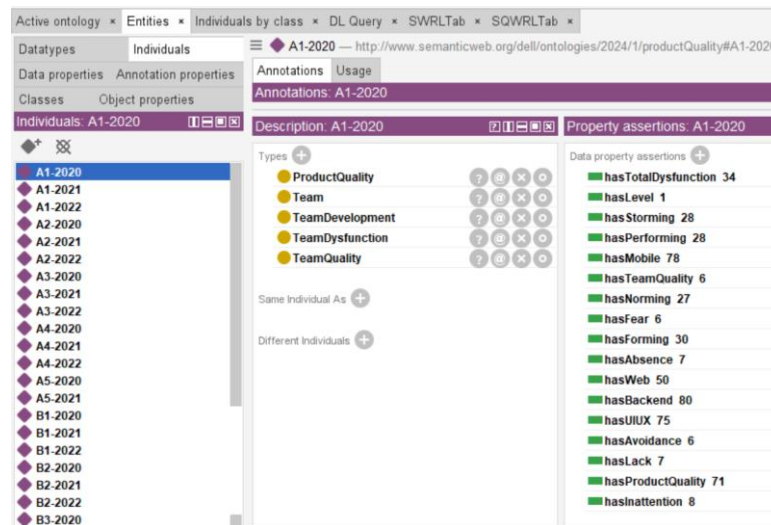
Tahap ini mendefinisikan *class team dysfunction* yang memiliki *sub-class absence of trust, fear of conflict, lack of commitment, inattention to result* dan *avoidance of accountability*. *Team Dysfunction* juga *disjoint* dengan *team development* guna menentukan nilai *team quality*. Didalam nya juga memiliki *instances* dari keseluruhan jumlah tim yang bisa dilihat pada gambar berikut ini Gambar 4.3.



Gambar 4.3 Class Team Dysfunction

b. Pendeklarasian *Instances*

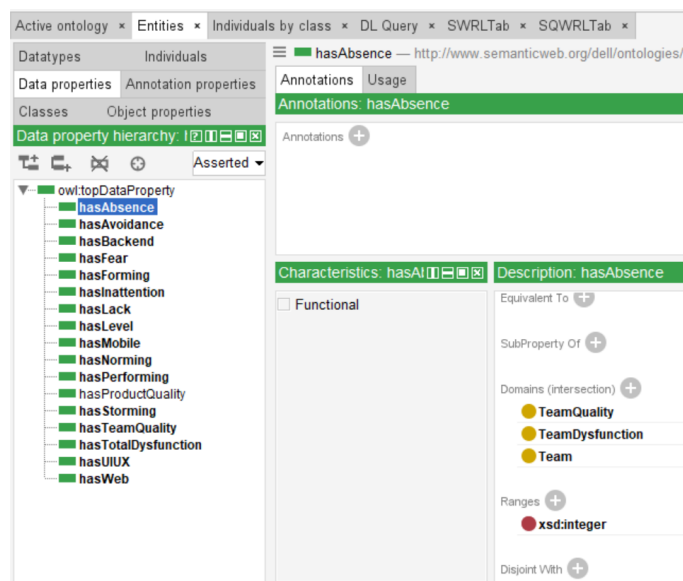
Tahap ini menentukan *instances* sehingga setiap tim per tahun nya dideklarasikan satu per satu. Salah satu contohnya pada tim A1-2020 yang didalam nya memiliki data *properties* yang meliputi *sub-class* dari *team development*, *team dysfuction*, dan *product quality* yang terlihat pada Gambar 4.4. dibawah ini.



Gambar 4.4 Instances

c. Pendeklarasian Data *Properties*

Pada tahapan ini mendeklarasikan data *properties* (relasi) yang dibutuhkan untuk menghubungkan antara *class* dan *instances* yang sudah di deklarasikan sebelumnya. Data *properties* yang dibuat mencangkup tiga domain yaitu *team quality*, *team dysfunction* dan *team*. Data *properties* ini digunakan untuk menampung keseluruhan nilai yang dimiliki oleh *class team* yang terlihat pada Gambar 4.5 dibawah ini.



Gambar 4. 5 Data *Properties*

d. Pembuatan SWRL *Rule*

Beberapa jenis aturan telah dibuat untuk mendukung analisis dalam konteks pengembangan tim. Aturan pertama digunakan untuk menentukan level pada *team development*, memungkinkan evaluasi yang lebih terperinci terhadap kemajuan dan kinerja tim. Selanjutnya, aturan kedua digunakan untuk menghitung total nilai dari *team dysfunction*, memberikan pemahaman yang lebih baik tentang kendala yang mungkin dihadapi oleh tim. Selain itu, terdapat aturan ketiga untuk menghitung dan mengkategorikan *team quality*, yang membantu dalam menilai kualitas keseluruhan dari sebuah tim. Begitu pula dengan aturan keempat untuk menghitung dan mengkategorikan *product quality*,

yang membantu mengukur kualitas produk yang dihasilkan oleh tim. Aturan terakhir, pembuatan *query* hasil keseluruhan menggunakan aturan-aturan ini memungkinkan untuk mendapatkan pemahaman yang lebih komprehensif dan *actionable* terkait dengan proses dan hasil dari pengembangan tim tersebut.

Ada beberapa macam aturan yang dibuat dalam penelitian ini, antara lain:

1. Penentuan level pada *team development*
2. Menghitung total nilai dari *team dysfunction*
3. Menghitung dan mengkategorikan *team quality*
4. Menghitung dan mengkategorikan *product quality*
5. Pembuatan *query* hasil keseluruhan

Name	Query	Comment
Query Table	productQuality:Team(?t) ^ productQuality:hasLevel(?l ?level) ^ productQuality:hasTotalDysfunction(?t ?totalDysfunction) ^ productQuality:hasTeamQuality(?t ...	Query Table
Rule 1 - Forming	productQuality:Team(?t) ^ productQuality:hasForming(?t ?forming) ^ productQuality:hasPerforming(?t ?performing) ^ productQuality:hasNorming(?t ?norming) ^ ...	Menentukan Level Forming
Rule 1 - Norming	productQuality:Team(?t) ^ productQuality:hasForming(?t ?forming) ^ productQuality:hasStorming(?t ?storming) ^ productQuality:hasNorming(?t ?norming) ^ ...	Menentukan Level Norming
Rule 1 - Performing	productQuality:Team(?t) ^ productQuality:hasForming(?t ?forming) ^ productQuality:hasStorming(?t ?storming) ^ productQuality:hasNorming(?t ?norming) ^ ...	Menentukan Level Performing
Rule 1 - Storming	productQuality:Team(?t) ^ productQuality:hasForming(?t ?forming) ^ productQuality:hasStorming(?t ?storming) ^ productQuality:hasNorming(?t ?norming) ^ ...	Menentukan Level Storming
Rule 2 - Team Dysfunction	productQuality:Team(?t) ^ productQuality:hasAbsence(?t ?absence) ^ productQuality:hasFear(?t ?fear) ^ productQuality:hasAvoidance(?t ?avoidance) ^ prod...	Menghitung Total Team Dysfunction
Rule 3 - Effective	productQuality:Team(?t) ^ productQuality:hasTeamQuality(?t ?teamQuality) ^ swrb:greaterThanOrEqualTo(?teamQuality, 80) -> productQuality:hasStatusTeam...	Menentukan Status Team Quality
Rule 3 - Ineffective	productQuality:Team(?t) ^ productQuality:hasTeamQuality(?t ?teamQuality) ^ swrb:lessThan(?teamQuality, 80) -> productQuality:hasStatusTeamQuality(?t ...	Menentukan Status Team Quality
Rule 3 - Team Quality	productQuality:Team(?t) ^ productQuality:hasLevel(?l ?level) ^ productQuality:hasTotalDysfunction(?t ?totalDysfunction) ^ swrb:multiply(?kesimpulanLevel, ...	Menghitung Team Quality
Rule 4 - Good	productQuality:Team(?t) ^ productQuality:hasProductQuality(?t ?productQuality) ^ swrb:greaterThanOrEqualTo(?productQuality, 80) -> productQuality:hasStatu...	Menentukan Product Quality
Rule 4 - Less Good	productQuality:Team(?t) ^ productQuality:hasProductQuality(?t ?productQuality) ^ swrb:lessThan(?productQuality, 80) -> productQuality:hasStatusProductQu...	Menentukan Product Quality
Rule 4 - Product Quality	productQuality:Team(?t) ^ productQuality:hasUIUX(?t ?uiux) ^ productQuality:hasBackend(?t ?backend) ^ productQuality:hasWeb(?t ?web) ^ productQuality:...	Menghitung Product Quality

Gambar 4. 6 SWRL Rule

a. Penentuan Level pada *Team Development*

Tahapan ini merupakan pembuatan SWRL *rule* yang bertujuan menentukan level pada *team development* yang terdiri dari level 1-4, dimana level satu mempresentasikan level *forming*, level dua mempresentasikan level *storming*, level tiga mempresentasikan level *norming*, dan terakhir level empat mempresentasikan level *performing*.

- Penentuan Level *Forming*

Level *forming* ditentukan dari nilai *forming* harus lebih besar dari *storming*, *norming*, dan *performing* kemudian dikategorikan menjadi level 1.

Berikut SWRL *rule* untuk Level 1 *forming*.

```
Team(?t) ^ hasStorming(?t, ?storming) ^ hasPerforming(?t, ?performing) ^
hasNorming(?t, ?norming) ^ hasForming(?t, ?forming) ^
swrlb:greaterThanOrEqualTo(?forming, ?performing) ^
swrlb:greaterThanOrEqualTo(?forming, ?norming) ^
swrlb:greaterThanOrEqualTo(?forming, ?storming) ->
hasLevel(?t, 1)
```

- Penentuan Level *Storming*

Level *storming* ditentukan dengan membandingkan dengan nilai lainnya, ketika nilai *storming* pada setiap timnya melebihi nilai *forming*, *norming*, *performing* sehingga dikategorikan pada level 2.

Berikut SWRL *rule* penentuan Level 2 *Storming*.

```
Team(?t) ^ hasForming(?t, ?forming) ^ hasStorming(?t, ?storming) ^
hasNorming(?t, ?norming) ^ hasPerforming(?t, ?performing) ^
swrlb:greaterThanOrEqualTo(?storming, ?forming) ^
swrlb:greaterThanOrEqualTo(?storming, ?norming) ^
swrlb:greaterThanOrEqualTo(?storming, ?performing) ->
hasLevel(?t, 2)
```

- Penentuan Level *Norming*

Level *norming* ditentukan dengan membandingkan dengan nilai lainnya, ketika nilai *storming* pada setiap timnya melebihi nilai *forming*, *storming*, *performing* sehingga dikategorikan pada level 3.

Berikut SWRL *rule* penentuan Level 3 *Norming*.

```
Team(?t) ^ hasForming(?t, ?forming) ^ hasStorming(?t, ?storming) ^
hasNorming(?t, ?norming) ^ hasPerforming(?t, ?performing) ^
swrlb:greaterThanOrEqualTo(?norming, ?forming) ^
swrlb:greaterThanOrEqualTo(?norming, ?storming) ^
```

```
swrlb:greaterThanOrEqual(?norming, ?performing) ->
hasLevel(?t, 3)
```

- Penentuan Level *Performing*

Terakhir dalam menentukan level performing ditentukan dengan membandingkan dengan nilai lainnya, ketika nilai *storming* pada setiap timnya melebihi nilai *forming*, *storming*, *norming* sehingga dikategorikan pada level 4.

Berikut SWRL rule penentuan Level 4 *Performing*.

```
Team(?t) ^ hasForming(?t, ?forming) ^ hasNorming(?t, ?norming) ^
hasStorming(?t, ?storming) ^ hasPerforming(?t, ?performing) ^
swrlb:greaterThanOrEqual(?performing, ?forming) ^
swrlb:greaterThanOrEqual(?performing, ?storming) ^
swrlb:greaterThanOrEqual(?performing, ?norming) ->
hasLevel(?t, 4)
```

b. Menghitung Total Nilai dari *Team Dysfunction*

Total *team dysfunction* untuk setiap tim ditentukan dengan menjumlahkan skor dari setiap kategori, yang meliputi *Absence of Trust*, *Fear of Conflict*, *Lack of Commitment*, *Avoidance of Accountability*, dan *Inattention to Results*.

Perhitungan total *team dysfunction* untuk sebuah tim tertentu diilustrasikan di bawah ini, mengikuti rumus:

$$Total\ Dysfunction = Absence + Fear + Avoidance + Lack + Inattention \quad (4.1)$$

Berikut SWRL rule nya :

```
Team(?t) ^ hasAbsence(?t, ?absence) ^ hasFear(?t,
?fear) ^ hasAvoidance(?t, ?avoidance) ^ hasLack(?t,
```

```
?lack) ^ hasInattention(?t, ?inattention) ^
swrlb:add(?total, ?absence, ?fear, ?avoidance, ?lack,
?inattention) -> hasTotalDysfunction(?t, ?total)
```

c. Menghitung dan Mengkategorikan *Team Quality*

Perhitungan total kualitas tim melibatkan sebuah rumus normalisasi. Rumus ini mengalikan 100 dengan kesimpulan level yang didapatkan dari pengolahan nilai *team development*, menambahkan hasilnya dengan total *team dysfunction* dikurangi 115, kemudian membaginya dengan 330, dan akhirnya dikalikan dengan 100, sesuai dengan rumus berikut:

$$f(\alpha, \beta) = \frac{((100\alpha + \beta) - 115)}{330} \times 100 \quad (4.2)$$

Berikut untuk SWRL rule formula diatas :

```
Team(?t) ^ hasLevel(?t, ?level) ^ hasTotalDysfunction(?t,
?totalDysfunction) ^
swrlb:multiply(?kesimpulanLevel, ?level, 100) ^
swrlb:add(?total, ?kesimpulanLevel, ?totalDysfunction) ^
swrlb:subtract(?minus, ?total, 115) ^
swrlb:multiply(?hasil, ?minus, 100) ^
swrlb:divide(?teamQuality, ?hasil, 330) ->
hasTeamQuality(?t, ?teamQuality)
```

Setelah menemukan nilai rata-rata kualitas produk, kategorisasi dilakukan sebagai berikut:

Kualitas Tim ≥ 80 : "Effective" (1)

Kategori *Team Quality* – Effective

```
Team(?t) ^ hasTeamQuality(?t, ?teamQuality) ^
swrlb:greaterThanOrEqual(?teamQuality, 80) ->
```

<i>hasStatusTeamQuality(?t, effective)</i>
--

Kualitas Tim < 80 : "Ineffective" (2)

Kategori <i>Team Quality – Ineffective</i>
--

<i>Team(?t) ^ hasTeamQuality(?t, ?teamQuality) ^ swrlb:lessThan(?teamQuality, 80) -> hasStatusTeamQuality(?t, ineffective)</i>

d. Menghitung dan Mengkategorikan *Product Quality*

Untuk menghitung nilai kualitas tim, gunakan rumus di bawah ini:

$$Product\ Quality = \frac{UIUX+Backend+Web+Mobile}{4} \quad (4.3)$$

Berikut untuk SWRL rule formula diatas :

<i>Team(?t) ^ hasUIUX(?t, ?uiux) ^ hasBackend(?t, ?backend) ^ hasWeb(?t, ?web) ^ hasMobile(?t, ?mobile) ^ swrlb:add(?total, ?uiux, ?backend, ?web, ?mobile) ^ swrlb:divide(?productQuality, ?total, 4) -> hasProductQuality(?t, ?productQuality)</i>

Setelah menemukan hasil kualitas tim dengan normalisasi, langkah selanjutnya adalah mengkategorikannya, seperti berikut:

Kualitas Produk >= 80 : "Good" (3)

Status <i>Product Quality – Good</i>

<i>Team(?t)^hasTQ(?t,?TQ)^hasSNTQ(?t,?SNTQ)^ swrlb:greaterThanOrEqual(?TQ,?SNTQ)-> hasProductQuality(?t,"Good")</i>
--

Kualitas Produk < 80 : “Less Good” (4)

Status <i>Product Quality</i> – Less Good

<i>Team</i> (?t)^ <i>hasTQ</i> (?t,?TQ)^ <i>hasSNTQ</i> (?t,?SNTQ)^ <i>swrlb:lessThan</i> (?TQ,?SNTQ)-> <i>hasProductQuality</i> (?t,"Less Good")

e. Pembuatan *Query* Hasil Keseluruhan

Pertanyaan digunakan untuk menampilkan nilai properti, yang dalam penelitian ini merupakan hasil perhitungan. Pertanyaan dijalankan menggunakan perintah *swrl:selectDistinct*(*?*). Istilah “*distinct*” bertujuan untuk menghindari kelebihan. Dalam pertanyaan ini, objek dari *Team*, Kesimpulan *Level*, Nilai *Team Dysfunction*, Nilai *Team Quality*, Status *Team Quality*, Nilai *Product Quality*, and Status *Product Quality* ditampilkan.

Berikut SWRL untuk menampilkan hasil query tabel keseluruhan nilai:

<i>Team</i> (?t) ^ <i>hasLevel</i> (?t, ?level) ^ <i>hasTotalDysfunction</i> (?t, ?totalDysfunction) ^ <i>hasTeamQuality</i> (?t, ?teamQuality) ^ <i>hasStatusTeamQuality</i> (?t, <i>?statusTeamQuality</i>) ^ <i>hasProductQuality</i> (?t, ?productQuality) ^ <i>hasStatusProductQuality</i> (?t, ?statusProductQuality) -> <i>swrl:selectDistinct</i> (?t, ?level, ?totalDysfunction, ?teamQuality, <i>?statusTeamQuality</i> , ?productQuality, ?statusProductQuality)

4.2 Hasil *Query* Pengujian Ontologi pada *Protégé*

Pengujian dilakukan dalam beberapa tahapan sebagai berikut :

1. Memasukkan Skor Untuk Setiap Tim

Termasuk nilai-nilai untuk menilai tahap-tahap dalam pengembangan tim seperti *forming*, *norming*, *performing*, dan *storming*. Selain itu, memasukkan nilai untuk mengevaluasi *team development*, *team dysfunction*, dan *dev process point*. Seluruh data dimasukkan ke dalam *editor* individu di *Protégé* untuk setiap *class* dan *instance* nya.

2. Menentukan *Level Team Development*

Dalam pengembangan tim, terdapat 4 aspek: *forming*, *norming*, *storming*, dan *performing*. Untuk menentukan kesimpulan levelnya adalah dengan menemukan hasil skor tertinggi dari semua aspek, kemudian ditentukan pada tingkat mana aspek tersebut berada.

3. Menghitung *Team Dysfunction*

Disfungsi tim dihitung sesuai dengan aturan nomor 2. Contoh perhitungan ini dapat dilihat pada tim A1 dengan menjumlahkan semua nilai dari *forming*, *norming*, *storming*, dan *performing*. Gambar 4.7 menunjukkan hasil *query* dari pengujian ontologi pada *Protégé*, yang akan secara otomatis terisi di *editor* individu ketika SWRL disfungsi tim dijalankan.

4. Menghitung Nilai dan Menentukan Status *Team Quality*

Pada tahap ini, nilai kualitas tim dihitung menggunakan rumus normalisasi, kemudian dikategorikan berdasarkan nilai yang diperoleh. Jika skor melebihi 80, tim masuk dalam kategori "*Effective*", sedangkan jika skor di bawah 80, tim masuk dalam kategori "*Ineffective*".

5. Menghitung nilai dan Menentukan Status *Product Quality*

Pada tahap ini, skor rata-rata kualitas produk untuk setiap tim dihitung, kemudian dikategorikan berdasarkan skor yang diperoleh dengan nilai ambang batas 80. Jika skor rata-rata kualitas produk melebihi 80, masuk dalam kategori "*Good*", sedangkan jika skor di bawah 80, masuk dalam kategori "*Less Good*". Untuk hasil *query* tabel hasil bisa dilihat pada Gambar 4.7 dibawah ini.

Name	Query	Comment				
SQWRL Queries	OWL 2 RL	Query Table				
t	level	totalDysfunction	teamQuality	statusTeamQuality	productQuality	statusProductQuality
productQuality:A1-2020	1	34	6	productQuality:ineffective	71	productQuality:lessgood
productQuality:A1-2021	4	39	98	productQuality:effective	86	productQuality:good
productQuality:A1-2022	4	35	97	productQuality:effective	87	productQuality:good
productQuality:A2-2020	3	34	66	productQuality:ineffective	84	productQuality:good
productQuality:A2-2021	4	41	99	productQuality:effective	83	productQuality:good
productQuality:A2-2022	4	40	98	productQuality:effective	89	productQuality:good
productQuality:A3-2020	4	39	98	productQuality:effective	86	productQuality:good
productQuality:A3-2021	4	41	99	productQuality:effective	82	productQuality:good
productQuality:A3-2022	4	41	99	productQuality:effective	64	productQuality:lessgood
productQuality:A4-2020	4	33	96	productQuality:effective	79	productQuality:lessgood
productQuality:A4-2021	4	39	98	productQuality:effective	73	productQuality:lessgood
productQuality:A4-2022	4	34	97	productQuality:effective	87	productQuality:good
productQuality:A5-2020	4	31	96	productQuality:effective	79	productQuality:lessgood
productQuality:A5-2021	4	38	98	productQuality:effective	77	productQuality:lessgood
productQuality:B1-2020	3	35	67	productQuality:ineffective	81	productQuality:good
productQuality:B1-2021	4	41	99	productQuality:effective	85	productQuality:good
productQuality:B1-2022	4	41	99	productQuality:effective	88	productQuality:good
productQuality:B2-2020	4	30	95	productQuality:effective	83	productQuality:good
productQuality:B2-2021	4	41	99	productQuality:effective	90	productQuality:good
productQuality:B2-2022	4	37	98	productQuality:effective	81	productQuality:good
productQuality:B3-2020	4	38	98	productQuality:effective	86	productQuality:good
productQuality:B3-2021	4	41	99	productQuality:effective	89	productQuality:good
productQuality:B3-2022	2	34	36	productQuality:ineffective	60	productQuality:lessgood

Gambar 4. 7 Hasil *Query* Pengujian Ontologi pada *Protégé*

Tahap ini dilakukan dengan membandingkan standar *product quality* dan total *team quality*. Untuk tim A1-2021 yang memiliki nilai *product quality* 86 setelah dibandingkan dengan standar *product quality* pada persamaan 3 dan 4 maka tim A1-2021 mendapatkan prediksi *product quality* “Good” sama dengan yang terlihat pada gambar 4.7 diatas. Dari analisa hasil pengujian seperti yang dijelaskan sebelumnya, terlihat bahwa rancangan ontologi sudah dapat memberikan penilaian dan idetifikasi status *product quality*, sesuai dengan aturan yang dibuat.

$$Accuracy = \frac{True\ Positif + True\ Negatif}{True\ Positif + False\ Positif + False\ Negatif + True\ Negatif} = \frac{17}{17 + 6} = 0,7037037037$$

$$Precision = \frac{True\ Positif}{True\ Positif + False\ Positif} = \frac{17}{17 + 6} = 0,7391304348$$

$$Recall = \frac{True\ Positif}{True\ Positif + False\ Negatif} = \frac{17}{17 + 2} = 0,8947368421$$

$$F1\ Score = \frac{2 \times Precision \times Recall}{Precision + Recall} = \frac{2 \times 0,7391304348 \times 0,8947368421}{0,7391304348 + 0,8947368421} = 0,8095238095$$

Hasil dari *confusion matrix* menunjukkan bahwa model klasifikasi memiliki akurasi sebesar 0,7037, yang berarti sekitar 70,37% dari semua prediksi yang dilakukan oleh model adalah benar. Akurasi ini memberikan gambaran umum tentang seberapa sering model membuat prediksi yang tepat, baik itu untuk kelas positif maupun negatif. Namun, akurasi saja tidak selalu memberikan gambaran lengkap tentang kinerja model, terutama jika data tidak seimbang antara kelas positif dan negatif.

Precision sebesar 0,7391 menunjukkan bahwa dari semua prediksi yang dianggap positif oleh model, 73,91% di antaranya benar-benar positif. *Precision* adalah metrik yang penting dalam situasi di mana biaya kesalahan positif tinggi, misalnya dalam diagnosa medis di mana *false positive* dapat menyebabkan pengobatan yang tidak perlu. Nilai *precision* yang tinggi menunjukkan bahwa model cukup efektif dalam meminimalkan jumlah *false positive*, sehingga sebagian besar dari prediksi positif adalah benar.

Recall atau sensitivitas sebesar 0,8947 mengindikasikan bahwa model mampu mengidentifikasi 89,47% dari semua kasus positif yang sebenarnya. *Recall* sangat penting dalam konteks di mana deteksi positif yang benar sangat kritis. *F1 Score*, yang merupakan rata-rata harmonis dari *precision* dan *recall*, bernilai 0,8095. *F1 Score* menggabungkan kedua metrik ini untuk memberikan satu nilai yang mencerminkan keseimbangan antara ketepatan (*precision*) dan kelengkapan (*recall*). Dengan *F1 Score* yang tinggi, model ini menunjukkan performa yang baik dalam klasifikasi, dengan keseimbangan yang kuat antara mengidentifikasi semua kasus positif dan menghindari prediksi positif palsu.

4.3 Hasil Kinerja Metode *Random Forest*

Hasil dari *confusion matrix* menggunakan model *Random Forest* untuk tugas klasifikasi biner. Dalam tabel dibawah, bisa dilihat metrik seperti *precision*, *recall*, *f1-score*, dan support untuk setiap kelas (0 dan 1). *Precision* adalah rasio prediksi positif yang benar terhadap total prediksi positif. *Recall* adalah rasio prediksi positif yang benar terhadap total aktual positif, sedangkan *f1-score* adalah rata-rata harmonis dari *precision* dan *recall*, yang memberikan gambaran seimbang dari kedua metrik tersebut.

Untuk kelas 0, *precision* tercatat sempurna yaitu 1.000, artinya setiap prediksi positif untuk kelas 0 adalah benar. Namun, *recall* untuk kelas ini sangat rendah, yaitu

hanya 0.250, menunjukkan bahwa model hanya berhasil mengidentifikasi 25% dari semua kasus aktual kelas 0. *F1-score* untuk kelas ini adalah 0.400, yang mencerminkan ketidakseimbangan antara *precision* yang sangat tinggi dan *recall* yang sangat rendah.

Sebaliknya, untuk kelas 1, *precision* adalah 0.625, menunjukkan bahwa ada beberapa prediksi positif yang salah untuk kelas ini. Namun, *recall* untuk kelas 1 sangat tinggi, mencapai 1.000, yang berarti model berhasil mengidentifikasi semua kasus aktual kelas 1 dengan benar. *F1-score* untuk kelas ini adalah 0.769, menunjukkan performa yang lebih seimbang dibandingkan kelas 0.

Secara keseluruhan, akurasi model tercatat sebesar 66.67%. Akurasi ini menunjukkan bahwa sekitar dua pertiga dari semua prediksi yang dibuat oleh model adalah benar. Namun, akurasi saja tidak selalu memberikan gambaran lengkap tentang performa model, terutama jika ada ketidakseimbangan kelas dalam dataset. Oleh karena itu, metrik seperti *macro avg* dan *weighted avg f1-score* memberikan wawasan tambahan tentang keseimbangan performa model di berbagai kelas.

Kesimpulannya, model *Random Forest* ini menunjukkan performa yang sangat baik dalam mengidentifikasi kelas 1 (dengan *recall* sempurna), namun kurang efektif dalam mengidentifikasi kelas 0 (dengan *recall* yang sangat rendah). Meskipun akurasi totalnya cukup baik, perbaikan lebih lanjut diperlukan untuk meningkatkan deteksi kelas 0 agar model lebih seimbang dan efektif secara keseluruhan. Berikut hasil confusion matrix menggunakan *Random Forest* bisa dilihat melalui gambar 4.10 dibawah ini.

```
Random Forest Classification Report:
              precision    recall  f1-score   support
0               1.000000    0.250000    0.400000     4.000000
1               0.625000    1.000000    0.769231     5.000000
accuracy               0.666667    0.666667    0.666667     9.000000
macro avg               0.812500    0.625000    0.584615     9.000000
weighted avg           0.791667    0.666667    0.605128     9.000000
Akurasi Random Forest Model: 66.67%
```

Gambar 4. 8 Hasil Evaluasi Model *Random Forest*

4.4 Hasil Pengujian Menggunakan *Random Forest*

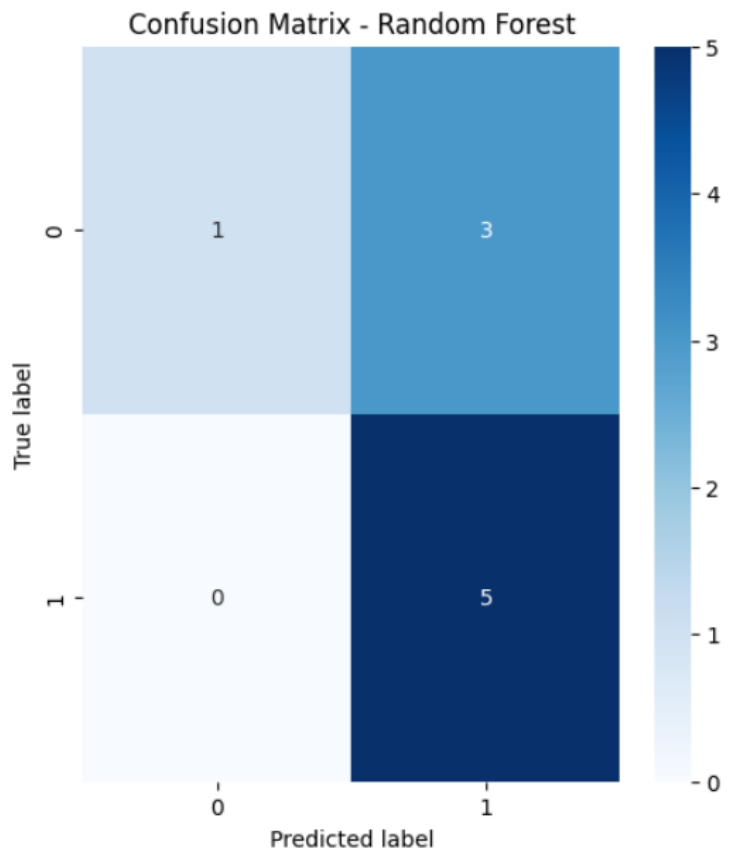
Visualisasi dibawah merupakan *confusion matrix* dari model *Random Forest* untuk tugas klasifikasi biner. *Matrix* ini menggambarkan performa prediksi model dalam bentuk matriks 2x2, yang memudahkan untuk melihat distribusi prediksi benar dan salah di antara kelas-kelas. Sumbu y merepresentasikan label sebenarnya (*True label*), sedangkan sumbu x menunjukkan label prediksi (*Predicted label*).

Dari *confusion matrix* tersebut, dapat dilihat bahwa untuk kelas 0 (baris pertama), model hanya berhasil memprediksi 1 *instance* dengan benar sebagai kelas 0 (*True Negative*), tetapi salah memprediksi 3 *instance* lainnya sebagai kelas 1 (*False Positive*). Hal ini menunjukkan bahwa model cenderung kurang efektif dalam mendeteksi kelas 0, seperti yang juga tercermin dari nilai *recall* yang rendah untuk kelas ini dalam laporan sebelumnya.

Untuk kelas 1 (baris kedua), model berhasil memprediksi semua 5 *instance* dengan benar sebagai kelas 1 (*True Positive*) dan tidak membuat kesalahan prediksi untuk kelas ini (*False Negative* = 0). Ini berarti bahwa model sangat efektif dalam mendeteksi kelas 1, yang juga didukung oleh nilai *recall* yang sempurna (1.000) untuk kelas ini dalam laporan sebelumnya.

Secara keseluruhan, *confusion matrix* ini menguatkan hasil dari laporan klasifikasi, dimana model memiliki bias dalam mendeteksi kelas 1 dengan lebih baik dibandingkan kelas 0. Jumlah kesalahan prediksi yang lebih banyak terjadi pada kelas 0, yang menyebabkan ketidakseimbangan dalam performa model. Akurasi keseluruhan model adalah 66.67%, yang dihitung dari jumlah prediksi benar (1 untuk kelas 0 dan 5 untuk kelas 1) dibagi dengan total *instance* (9).

Kesimpulannya, visualisasi ini menunjukkan bahwa model *Random Forest* memiliki kekuatan dalam mendeteksi kelas 1 tetapi kelemahan signifikan dalam mendeteksi kelas 0. Untuk meningkatkan performa model secara keseluruhan, perlu dilakukan upaya untuk meningkatkan *recall* dan *f1-score* untuk kelas 0, misalnya dengan teknik penyeimbangan data seperti oversampling atau undersampling, atau dengan mengoptimalkan *hyperparameters* dari model *Random Forest*. Berikut hasil visualisasi *confusion matrix* menggunakan metode *Random Forest* bisa dilihat melalui gambar 4.9.



Gambar 4. 9 Visualisasi Hasil Pengujian dengan *Random Forest*

4.5 Hasil Kinerja Metode *Decision Tree*

Hasil *confusion matrix* klasifikasi untuk model *Decision Tree* pada tugas klasifikasi biner. Dalam laporan ini, dapat dilihat metrik evaluasi seperti *precision*, *recall*, *f1-score*, dan *support* untuk masing-masing kelas (0 dan 1). *Precision* adalah rasio prediksi positif yang benar terhadap total prediksi positif. *Recall* adalah rasio prediksi positif yang benar terhadap total aktual positif, sedangkan *f1-score* adalah rata-rata harmonis dari *precision* dan *recall*, memberikan gambaran seimbang dari kedua metrik tersebut.

Untuk kelas 0, *precision* tercatat sempurna yaitu 1.000, artinya setiap prediksi positif untuk kelas 0 adalah benar. Namun, *recall* untuk kelas ini sangat rendah, yaitu hanya 0.250, menunjukkan bahwa model hanya berhasil mengidentifikasi 25% dari semua kasus aktual kelas 0. *F1-score* untuk kelas ini adalah 0.400, yang mencerminkan ketidakseimbangan antara *precision* yang sangat tinggi dan *recall* yang sangat rendah.

Sebaliknya, untuk kelas 1, *precision* adalah 0.625, menunjukkan bahwa ada beberapa prediksi positif yang salah untuk kelas ini. Namun, *recall* untuk kelas 1 sangat tinggi, mencapai 1.000, yang berarti model berhasil mengidentifikasi semua kasus aktual kelas 1 dengan benar. *F1-score* untuk kelas ini adalah 0.769, menunjukkan performa yang lebih seimbang dibandingkan kelas 0.

Secara keseluruhan, akurasi model tercatat sebesar 66.67%. Akurasi ini menunjukkan bahwa sekitar dua pertiga dari semua prediksi yang dibuat oleh model adalah benar. Namun, seperti telah disebutkan sebelumnya, akurasi saja tidak selalu memberikan gambaran lengkap tentang performa model, terutama jika ada ketidakseimbangan kelas dalam dataset. Oleh karena itu, metrik seperti *macro avg* dan *weighted avg f1-score* memberikan wawasan tambahan tentang keseimbangan performa model di berbagai kelas.

Kesimpulannya, model *Decision Tree* ini menunjukkan performa yang sangat baik dalam mengidentifikasi kelas 1 (dengan *recall* sempurna), namun kurang efektif dalam mengidentifikasi kelas 0 (dengan *recall* yang sangat rendah). Meskipun akurasi totalnya cukup baik, perbaikan lebih lanjut diperlukan untuk meningkatkan deteksi kelas 0 agar model lebih seimbang dan efektif secara keseluruhan. Sama seperti model *Random Forest* sebelumnya, teknik penyeimbangan data dan optimasi *hyperparameters* dapat diterapkan untuk meningkatkan performa model. Berikut hasil confusion matrix menggunakan *Decision Tree* bisa dilihat melalui gambar 4.10 dibawah ini.

Decision Tree Classification Report:				
	precision	recall	f1-score	support
0	1.000000	0.250000	0.400000	4.000000
1	0.625000	1.000000	0.769231	5.000000
accuracy	0.666667	0.666667	0.666667	0.666667
macro avg	0.812500	0.625000	0.584615	9.000000
weighted avg	0.791667	0.666667	0.605128	9.000000
Akurasi Decision Tree Model: 66.67%				

Gambar 4. 10 Hasil Evaluasi Model *Decision Tree*

4.6 Hasil Pengujian Menggunakan *Decision Tree*

Visualisasi *confusion matrix* dari model *Decision Tree* untuk tugas klasifikasi biner. Hasil *Confusion matrix* menggambarkan jumlah prediksi yang benar dan salah untuk setiap kelas, yang diwakili dalam bentuk matriks 2x2. Sumbu y merepresentasikan label sebenarnya (*True label*), sedangkan sumbu x menunjukkan label prediksi (*Predicted label*).

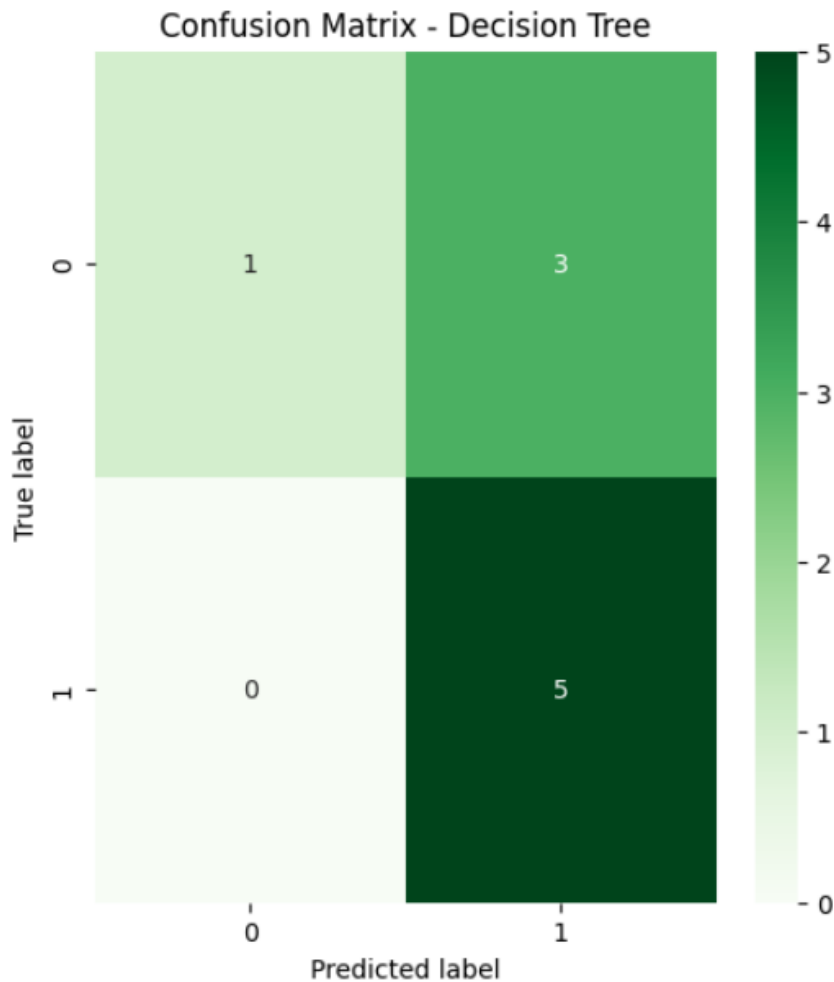
Untuk kelas 0 (baris pertama), model hanya berhasil memprediksi 1 *instance* dengan benar sebagai kelas 0 (*True Negative*), tetapi salah memprediksi 3 *instance* lainnya sebagai kelas 1 (*False Positive*). Ini menunjukkan bahwa model memiliki kesulitan dalam mendeteksi kelas 0, yang tercermin dalam nilai *recall* yang rendah untuk kelas ini (0.250). *Precision* untuk kelas 0 tinggi (1.000) karena semua prediksi untuk kelas 0 benar, tetapi karena jumlah *instance* yang terdeteksi sangat sedikit, *f1-score* menjadi rendah (0.400).

Sebaliknya, untuk kelas 1 (baris kedua), model berhasil memprediksi semua 5 *instance* dengan benar sebagai kelas 1 (*True Positive*) dan tidak membuat kesalahan prediksi untuk kelas ini (*False Negative* = 0). Ini menunjukkan bahwa model sangat efektif dalam mendeteksi kelas 1, seperti yang tercermin dari nilai *recall* yang sempurna (1.000). *Precision* untuk kelas 1 adalah 0.625, menunjukkan beberapa prediksi kelas 1 yang salah, namun *f1-score* untuk kelas ini tetap cukup tinggi (0.769), mencerminkan keseimbangan yang lebih baik antara *precision* dan *recall*.

Secara keseluruhan, akurasi model adalah 66.67%, yang menunjukkan bahwa dua pertiga dari semua prediksi yang dibuat oleh model adalah benar. Namun, akurasi ini perlu dilihat dalam konteks ketidakseimbangan kelas yang ada, di mana model jauh lebih baik dalam mendeteksi kelas 1 dibandingkan kelas 0. Hal ini juga terlihat dari *macro average f1-score* yang lebih rendah (0.584615), yang mengindikasikan bahwa performa model tidak seimbang di antara kedua kelas.

Kesimpulannya, model *Decision Tree* menunjukkan performa yang sangat baik dalam mendeteksi kelas 1 tetapi memiliki kekurangan signifikan dalam mendeteksi kelas 0. Meskipun *precision* untuk kelas 0 sangat tinggi, rendahnya *recall* menunjukkan bahwa model gagal mendeteksi sebagian besar *instance* dari kelas ini. Untuk meningkatkan performa keseluruhan model, khususnya dalam mendeteksi kelas 0, diperlukan upaya tambahan seperti penyeimbangan data, optimasi *hyperparameter*, atau

penggunaan metode *ensemble* yang dapat membantu mengatasi kelemahan ini dan membuat model lebih seimbang dan efektif. Berikut hasil visualisasi *confusion matrix* menggunakan metode *Decision Tree* bisa dilihat melalui gambar 4.11.



Gambar 4. 11 Visualisasi *Confusion Matrix* dengan *Decision Tree*

4.7 Pembahasan Hasil Penelitian

4.7.1 Pembahasan Hasil *Query* Menggunakan Ontologi

Ontologi yang dibuat dalam penelitian ini menunjukkan hubungan antara kelas tim dengan properti yang dimilikinya. Penelitian ini melibatkan lima aturan rancangan utama, yaitu: aturan untuk menentukan level dalam *team development*, menghitung total nilai dari *team dysfunction*, menghitung dan mengategorikan *team quality*, menghitung dan mengategorikan

product quality, dan membuat *query* untuk hasil keseluruhan. Setiap aturan ini dirancang untuk membantu dalam penilaian berbagai aspek kinerja tim dan produk.

Dari hasil pengujian yang dilakukan menggunakan ontologi dengan *tools Protégé*, terlihat bahwa rancangan ontologi dan aturan yang dibuat cukup akurat dalam mengidentifikasi kualitas produk berdasarkan kualitas tim. Keakuratan hasil ini didukung oleh nilai akurasi sebesar 70.37%, menunjukkan bahwa metode ontologi yang digunakan dapat diandalkan untuk keperluan evaluasi ini.

Selain itu, hasil *query* keseluruhan yang ditunjukkan pada gambar 4.12 memperlihatkan efektivitas rancangan ontologi dalam menghasilkan *output* yang diinginkan. Hasil pengujian ini tidak hanya menegaskan validitas aturan yang dirancang, tetapi juga menunjukkan bahwa ontologi dapat digunakan sebagai alat yang berguna untuk mengidentifikasi dan mengevaluasi kualitas produk berdasarkan berbagai indikator kinerja tim. Dengan demikian, penelitian ini berhasil menunjukkan bahwa pendekatan ontologi merupakan metode yang efektif dan akurat untuk analisis kualitas dalam konteks pengembangan tim dan produk. Hasil *query* secara keseluruhan bisa dilihat gambar 4.12 berikut ini.

↑	level	totalDysfunction	teamQuality	statusTeamQuality	productQuality	statusProductQuality
productQuality:A1-2020	1	34	6	productQuality:ineffective	71	productQuality:lessgood
productQuality:A1-2021	4	39	98	productQuality:effective	86	productQuality:good
productQuality:A1-2022	4	35	97	productQuality:effective	87	productQuality:good
productQuality:A2-2020	3	34	66	productQuality:ineffective	84	productQuality:good
productQuality:A2-2021	4	41	99	productQuality:effective	83	productQuality:good
productQuality:A2-2022	4	40	98	productQuality:effective	89	productQuality:good
productQuality:A3-2020	4	39	98	productQuality:effective	86	productQuality:good
productQuality:A3-2021	4	41	99	productQuality:effective	82	productQuality:good
productQuality:A3-2022	4	41	99	productQuality:effective	64	productQuality:lessgood
productQuality:A4-2020	4	33	96	productQuality:effective	79	productQuality:lessgood
productQuality:A4-2021	4	39	98	productQuality:effective	73	productQuality:lessgood
productQuality:A4-2022	4	34	97	productQuality:effective	87	productQuality:good
productQuality:A5-2020	4	31	96	productQuality:effective	79	productQuality:lessgood
productQuality:A5-2021	4	38	98	productQuality:effective	77	productQuality:lessgood
productQuality:B1-2020	3	35	67	productQuality:ineffective	81	productQuality:good
productQuality:B1-2021	4	41	99	productQuality:effective	85	productQuality:good
productQuality:B1-2022	4	41	99	productQuality:effective	88	productQuality:good
productQuality:B2-2020	4	30	95	productQuality:effective	83	productQuality:good
productQuality:B2-2021	4	41	99	productQuality:effective	90	productQuality:good
productQuality:B2-2022	4	37	98	productQuality:effective	81	productQuality:good
productQuality:B3-2020	4	38	98	productQuality:effective	86	productQuality:good
productQuality:B3-2021	4	41	99	productQuality:effective	89	productQuality:good
productQuality:B3-2022	2	34	36	productQuality:ineffective	60	productQuality:lessgood

Gambar 4. 12 Hasil Query Pengujian Ontologi

4.7.2 Pembahasan Hasil Kinerja Menggunakan *Random Forest*

Hasil prediksi status kualitas menggunakan model *Random Forest (RF)*. Dalam kolom "*Predicted Quality Status (RF)*", kita melihat serangkaian angka yang menunjukkan apakah status kualitas diprediksi baik (1) atau buruk (0). Dari delapan baris data yang ditampilkan, kita bisa melihat bahwa model *Random Forest* memprediksi tujuh kali status kualitas baik (1) dan satu kali status kualitas buruk (0). Jika kita membandingkan prediksi yang dibuat oleh *Random Forest* dengan kolom "*Actual Quality Status*", kita bisa mengevaluasi seberapa akurat model ini. Dari gambar 4.13, bahwa model *Random Forest* berhasil memprediksi dengan tepat tujuh dari delapan kali. Satu kesalahan prediksi terjadi pada baris keempat, di mana model memprediksi status kualitas buruk (0) sementara status yang sebenarnya adalah baik (1).

Analisis ini menunjukkan bahwa model *Random Forest* memiliki tingkat akurasi yang cukup tinggi dalam memprediksi status kualitas. Hal ini menandakan bahwa *Random Forest* adalah model yang andal untuk digunakan dalam prediksi status kualitas berdasarkan parameter yang telah diberikan. Keakuratan prediksi yang dihasilkan oleh model ini membuktikan efektivitasnya dalam mengidentifikasi status kualitas dengan cukup tepat. Parameter-parameter yang digunakan dalam model ini terbukti relevan dan efektif dalam menentukan status kualitas, yang menjadi alasan utama mengapa model *Random Forest* mampu memberikan prediksi yang hampir akurat.

Secara keseluruhan, fokus pada kotak merah di gambar ini menyoroti keunggulan model *Random Forest* dalam melakukan prediksi. Gambar tersebut memperlihatkan kemampuan model ini dalam memberikan hasil yang hampir akurat, yang semakin memperkuat keandalan dan efektivitas model ini dalam penggunaan praktis. Dengan hasil identifikasi yang baik, model *Random Forest* dapat diandalkan untuk digunakan dalam berbagai situasi di mana prediksi status kualitas diperlukan. Hasil identifikasi menggunakan metode *Random Forest* dapat dilihat secara rinci pada gambar 4.13.

	td_Forming	td_Storming	td_Norming	td_Performing	tf_Absence of Trust	\
0	27	25	29	32	32	8
1	26	23	31	34	34	8
2	24	23	32	37	37	8
3	28	26	30	32	32	8
4	30	28	27	28	28	7
5	25	25	29	34	34	9
6	25	21	32	38	38	9
7	24	21	34	37	37	9
8	27	22	32	36	36	9

	Ftf_ear of Conflict	tf_Lack of Commitment	tf_Avoidance of Accountability	\
0	8	8	8	7
1	8	8	9	7
2	9	9	9	7
3	8	8	9	8
4	6	6	7	6
5	8	8	9	7
6	9	9	9	7
7	8	8	8	8
8	8	8	8	8

	tf_Inattention to Results	Predicted Quality Status (RF)	\
0	7	1	1
1	7	1	1
2	8	1	1
3	8	1	1
4	8	0	0
5	8	1	1
6	7	1	1
7	8	1	1
8	8	1	1

	Predicted Quality Status (DT)	Actual Quality Status
0	1	0
1	1	0
2	1	1
3	1	0
4	0	0
5	1	1
6	1	1
7	1	1
8	1	1

Gambar 4. 13 Hasil Prediksi dengan *Random Forest*

4.7.3 Pembahasan Hasil Kinerja Menggunakan *Decision Tree*

Hasil prediksi status kualitas dari model *Decision Tree* (DT) diilustrasikan dengan kotak merah, yang memperlihatkan perbandingan antara prediksi model dengan status kualitas aktual dari tim-tim yang dievaluasi. Setiap baris data menampilkan prediksi model *Decision Tree* terhadap status kualitas suatu tim, yang kemudian disandingkan dengan status kualitas aktualnya untuk mengevaluasi keakuratan prediksi. Pada baris pertama, model *Decision Tree* memprediksi status kualitas tim dengan nilai 1

(kualitas tinggi), sedangkan status kualitas aktual tim tersebut adalah 0 (kualitas rendah). Hal ini menunjukkan ketidakakuratan model dalam prediksi untuk tim ini.

Pada baris kedua hingga kelima, model Decision Tree berhasil memprediksi status kualitas tim dengan benar, di mana setiap prediksi status kualitas bernilai 1 sesuai dengan status kualitas aktual mereka. Ini menandakan bahwa dalam beberapa kasus, model Decision Tree mampu memberikan prediksi yang tepat dan konsisten dengan kondisi nyata. Keberhasilan dalam prediksi ini menunjukkan potensi model untuk diandalkan dalam menilai status kualitas, meskipun masih terdapat ruang untuk perbaikan dalam beberapa kasus lainnya.

Pada baris keenam, model Decision Tree memprediksi status kualitas tim dengan nilai 0 (kualitas rendah), yang sesuai dengan status kualitas aktual tim tersebut. Keberhasilan prediksi ini menunjukkan akurasi model dalam beberapa kasus di mana prediksi nilai rendah ternyata tepat. Hal ini semakin menggarisbawahi keandalan model Decision Tree dalam memberikan prediksi yang sesuai dengan kondisi sebenarnya, meskipun ada beberapa prediksi yang tidak akurat pada baris-baris lainnya.

Baris ketujuh hingga kesembilan juga menunjukkan keberhasilan model Decision Tree dalam memprediksi status kualitas tim, di mana semua prediksi status kualitas bernilai 1 sesuai dengan status kualitas aktual mereka. Secara keseluruhan, model Decision Tree menunjukkan performa prediksi yang cukup baik dengan mayoritas prediksinya sesuai dengan status kualitas aktual dari tim-tim yang dievaluasi. Namun, terdapat beberapa kasus seperti pada baris pertama di mana prediksi model tidak akurat. Keakuratan model ini dalam memprediksi status kualitas tim dapat ditingkatkan dengan penyesuaian lebih lanjut dan analisis mendalam terhadap faktor-faktor yang mempengaruhi prediksi.

Untuk meningkatkan akurasi prediksi model Decision Tree dalam memprediksi status kualitas tim, diperlukan penyesuaian lebih lanjut dan analisis mendalam terhadap faktor-faktor yang mempengaruhi prediksi. Walaupun model ini telah menunjukkan performa yang cukup baik dengan mayoritas prediksinya sesuai dengan status kualitas aktual dari tim yang dievaluasi, beberapa kasus seperti pada baris pertama menunjukkan ketidakakuratan yang perlu diperbaiki. Pendekatan yang lebih komprehensif dalam memilih dan menyaring parameter yang digunakan, serta optimalisasi model melalui teknik-teknik seperti pruning, bisa membantu meningkatkan keandalan prediksi. Selain

itu, penggunaan data tambahan atau fitur baru yang relevan dapat memberikan informasi yang lebih kaya dan meningkatkan kemampuan model dalam mengidentifikasi status kualitas secara lebih akurat. Ke depan, dengan peningkatan ini, model Decision Tree memiliki potensi untuk menjadi alat yang lebih andal dalam menilai dan memprediksi status kualitas tim, seperti yang ditunjukkan di gambar 4.14.

	td_Forming	td_Storming	td_Norming	td_Performing	tf_Absence of Trust	\
0	27	25	29	32	32	8
1	26	23	31	34	34	8
2	24	23	32	37	37	8
3	28	26	30	32	32	8
4	30	28	27	28	28	7
5	25	25	29	34	34	9
6	25	21	32	38	38	9
7	24	21	34	37	37	9
8	27	22	32	36	36	9

	Ftf_ear of Conflict	tf_Lack of Commitment	tf_Avoidance of Accountability	\
0	8	8	8	7
1	8	8	9	7
2	9	9	9	7
3	8	8	9	8
4	6	6	7	6
5	8	8	9	7
6	9	9	9	7
7	8	8	8	8
8	8	8	8	8

	tf_Inattention to Results	Predicted Quality Status (RF)	\
0	7	7	1
1	7	7	1
2	8	8	1
3	8	8	1
4	8	8	0
5	8	8	1
6	7	7	1
7	8	8	1
8	8	8	1

	Predicted Quality Status (DT)	Actual Quality Status
0	1	0
1	1	0
2	1	1
3	1	0
4	0	0
5	1	1
6	1	1
7	1	1
8	1	1

Gambar 4. 14 Hasil Prediksi dengan Decission Tree

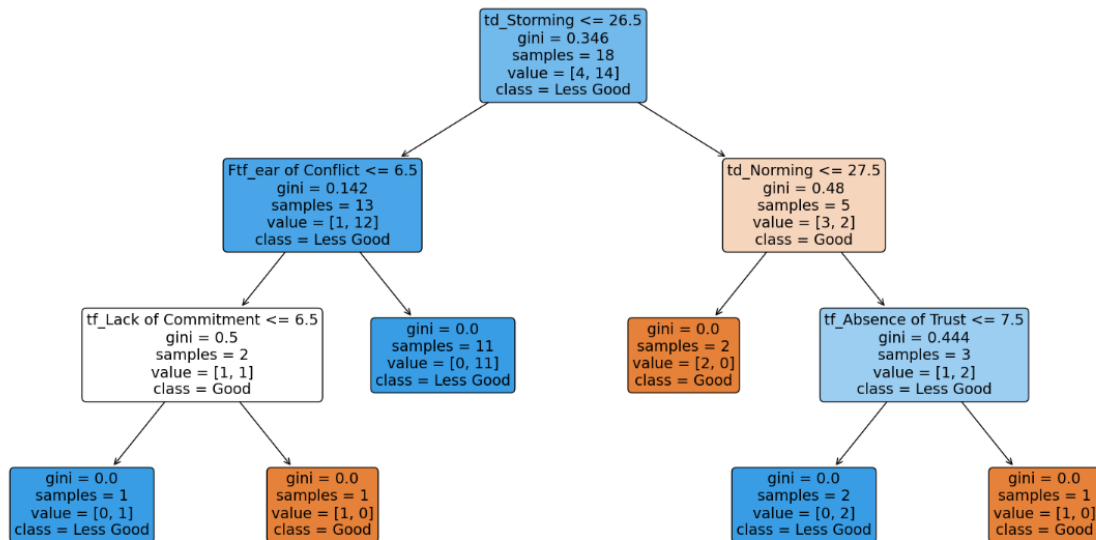
Visualisasi pada sebuah pohon keputusan yang digunakan untuk mengklasifikasikan suatu set data menjadi dua kelas: "Good" dan "Less Good". Pohon keputusan ini dibangun berdasarkan berbagai fitur seperti "Storming", "Norming", "Fear of Conflict", "Lack of Commitment", dan "Absence of Trust". Setiap simpul dalam pohon keputusan ini mewakili suatu kondisi yang memisahkan data berdasarkan nilai dari fitur tersebut.

Dimulai dari simpul akar, pohon ini pertama kali membagi data berdasarkan nilai fitur "Storming". Jika nilai "Storming" kurang dari atau sama dengan 26.5, data tersebut diproses ke simpul kiri, sementara jika lebih dari 26.5, diproses ke simpul kanan. Pada simpul kiri, data lebih lanjut dibagi berdasarkan nilai "Fear of Conflict". Jika nilainya kurang dari atau sama dengan 6.5, data akan dibagi lagi berdasarkan fitur "Lack of Commitment". Pada simpul "Lack of Commitment", data dibagi menjadi dua kelas yang sama: satu untuk "Good" dan satu lagi untuk "Less Good". Ini menunjukkan bahwa pada tingkat ini, tidak ada dominasi yang jelas dari satu kelas atas yang lainnya. Namun, jika "Fear of Conflict" lebih dari 6.5, maka data tersebut otomatis diklasifikasikan sebagai "Less Good" tanpa perlu pembagian lebih lanjut.

Di cabang kanan pohon keputusan, data dengan nilai "Storming" lebih dari 26.5 akan dibagi lagi berdasarkan nilai fitur "Norming". Jika nilai "Norming" kurang dari atau sama dengan 27.5, data tersebut akan diklasifikasikan sebagai "Good". Namun, jika nilai "Norming" lebih dari 27.5, data tersebut kemudian akan dibagi lagi berdasarkan nilai fitur "Absence of Trust". Jika nilai "Absence of Trust" kurang dari atau sama dengan 7.5, data tersebut akan diklasifikasikan sebagai "Less Good", sedangkan jika lebih dari 7.5, data tersebut akan diklasifikasikan sebagai "Good". Pohon keputusan ini menggambarkan bagaimana berbagai faktor yang diukur dapat mempengaruhi klasifikasi data. Visualisasi ini memudahkan pemahaman proses pembagian data dan menunjukkan bagaimana setiap fitur mempengaruhi hasil akhir.

Setiap simpul dan cabang pada pohon ini memperlihatkan logika pengambilan keputusan yang jelas, membantu dalam interpretasi hasil klasifikasi secara keseluruhan. Pohon keputusan ini menggambarkan bagaimana berbagai faktor yang diukur dapat mempengaruhi klasifikasi data. Visualisasi ini memudahkan pemahaman proses pembagian data dan menunjukkan bagaimana setiap fitur mempengaruhi hasil akhir. Setiap simpul dan cabang pada pohon ini memperlihatkan logika pengambilan

keputusan yang jelas, membantu dalam interpretasi hasil klasifikasi secara keseluruhan. Hasil lengkap dari pohon keputusan ini dapat dilihat pada gambar 4.15 di bawah ini.



Gambar 4.15 Kesimpulan Hasil Prediksi

4.7.4 Perbandingan Hasil Kinerja Ontologi, *Random Forest* dan *Decision Tree*

Hasil dari confusion matrix yang disajikan dalam tabel berikut ini memperlihatkan perbandingan kinerja antara dua metode klasifikasi, yakni Ontologi dan Decision Tree, dengan menggunakan empat metrik utama: Akurasi, Recall, Presisi, dan F1 Score. Akurasi mengukur proporsi prediksi yang benar dari keseluruhan prediksi yang dilakukan oleh model. Recall, atau sensitivitas, menilai kemampuan model dalam mengidentifikasi semua kasus positif yang benar. Presisi mengevaluasi sejauh mana prediksi positif yang dihasilkan oleh model benar-benar akurat. F1 Score, yang merupakan rata-rata harmonis dari Recall dan Presisi.

1. *Accuracy* merupakan persentase prediksi yang benar terhadap total data.

Hasil:

- *Ontologi* = 0,7037037037 (70.37%)
- *Random Forest* = 0,625 (62.5%)
- *Decision Tree* = 0,625 (62.5%)

Ontologi memiliki akurasi yang lebih tinggi dibandingkan dengan *Decision Tree*, yang berarti *Ontologi* membuat lebih banyak prediksi yang benar secara keseluruhan.

2. *Recall* (*sensitivity* atau *true positive rate*) merupakan persentase dari kasus positif yang benar-benar terdeteksi sebagai positif.

Hasil:

- *Ontologi* = 0,7391304348 (73.91%)
- *Random Forest* = 1,000000 (100%)
- *Decision Tree* = 1,000000 (100%)

Decision Tree berhasil mendeteksi semua kasus positif (*Recall* = 100%), sementara *Ontologi* memiliki *recall* yang lebih rendah.

3. *Precision* merupakan persentase dari prediksi positif yang benar-benar positif.

Hasil:

- *Ontologi* = 0,8947368421 (89.47%)
- *Random Forest* = 0,769231 (76.92%)
- *Decision Tree* = 0,769231 (76.92%)

Ontologi memiliki presisi yang lebih tinggi dibandingkan dengan *Decision Tree*, artinya ketika *Ontologi* memprediksi positif, prediksi tersebut lebih sering benar.

4. *F1 Score* merupakan harmonic mean dari presisi dan *recall*, memberikan gambaran tentang keseimbangan antara presisi dan *recall*.

Hasil:

- *Ontologi* = 0,8095238095 (80.95%)
- *Random Forest* = 0,666667 (66.67%)
- *Decision Tree* = 0,666667 (66.67%)

Ontologi memiliki *F1 Score* yang lebih tinggi dibandingkan dengan *Decision Tree*, menunjukkan bahwa *Ontologi* memiliki keseimbangan yang lebih baik antara presisi dan *recall*.

Tabel 4. 1 Hasil Komparasi Menggunakan Tiga Metode

	Ontologi	Random Forest	Decision Tree
<i>Accuracy</i>	0,7037037037	0,625000	0,625000
<i>Recall</i>	0,7391304348	1,000000	1,000000
<i>Precision</i>	0,8947368421	0,769231	0,769231
<i>F1 Score</i>	0,8095238095	0,666667	0,666667

Hasil dari *confusion matrix* dalam tabel di atas menunjukkan perbandingan kinerja antara dua metode klasifikasi, yaitu *Ontologi*, *Decision Tree*, dan *Random Forest* berdasarkan empat metrik utama: Akurasi, *Recall*, Presisi, dan *F1 Score*. Akurasi *Ontologi* sebesar 70.37% lebih tinggi dibandingkan dengan *Decision Tree* yang sebesar 62.5%, menunjukkan bahwa *Ontologi* membuat lebih banyak prediksi yang benar secara keseluruhan.

Dalam hal *recall*, *Decision Tree* memiliki nilai yang sempurna sebesar 100%, yang berarti metode ini berhasil mendeteksi semua kasus positif. Namun, hal ini dicapai dengan mengorbankan presisi, yang hanya sebesar 76.92%. Sebaliknya, *Ontologi* memiliki *recall* sebesar 73.91% tetapi dengan presisi yang jauh lebih tinggi, yaitu 89.47%. Ini menunjukkan bahwa ketika *Ontologi* memprediksi positif, prediksi tersebut lebih sering benar, sementara *Decision Tree* lebih sering menghasilkan false positives meskipun berhasil mendeteksi semua kasus positif.

F1 Score memberikan gambaran tentang keseimbangan antara presisi dan *recall*. Ontologi memiliki *F1 Score* sebesar 80.95%, lebih tinggi dibandingkan *Decision Tree* yang hanya 66.67%. Ini menunjukkan bahwa Ontologi memiliki keseimbangan yang lebih baik antara mendeteksi kasus positif dan meminimalkan *false positives*. Dengan demikian, Ontologi lebih efektif dalam memberikan prediksi yang benar secara konsisten, sementara *Decision Tree* lebih baik dalam mendeteksi semua kasus positif tetapi dengan risiko lebih tinggi menghasilkan prediksi positif yang salah.

[Halaman ini sengaja dikosongkan]

BAB V

KESIMPULAN DAN SARAN

Penelitian ini melakukan identifikasi menggunakan metode ontologi yang bertujuan untuk mendeteksi *product quality* berdasarkan *team quality*, selain itu diproses juga menggunakan metode *random forest* dan *decision tree* dan diakhir dibandingkan metode mana yang terbaik dengan menghasilkan nilai akurasi yang paling tinggi. Pembuatan sistem ini dapat membantu para peneliti untuk menentukan kualitas produk perangkat lunak berdasarkan dengan kualitas dari tim pengembangnya.

5.1 Kesimpulan

Penelitian yang dilakukan untuk mengidentifikasi *product quality* berdasarkan *team quality* menggunakan ontologi, dan metode *machine learning* yaitu *random forest* dan *decision tree* sudah diselesaikan dengan hasil yang dicapai pada setiap skenario uji coba. Kesimpulan yang didapat sebagai berikut :

1. Penelitian ini berhasil merancang ontologi dengan akurasi sebesar 70.37%, yang lebih tinggi dibandingkan dengan metode Random Forest dan Decision Tree yang masing-masing memiliki akurasi sebesar 62.5%. Hal ini menunjukkan bahwa ontologi lebih andal dalam memberikan prediksi yang benar secara keseluruhan ketika digunakan untuk mengidentifikasi kualitas produk berdasarkan kualitas tim.
2. Ontologi memiliki presisi yang lebih tinggi sebesar 89.47% dibandingkan Random Forest dan Decision Tree yang hanya sebesar 76.92%. Ini menunjukkan bahwa prediksi positif yang dihasilkan oleh ontologi lebih sering benar dibandingkan dengan metode lainnya, sehingga ontologi lebih efektif dalam merumuskan aturan yang tepat untuk mengidentifikasi kualitas produk.
3. Pengujian menggunakan metode Random Forest dan Decision Tree menunjukkan performa yang sangat baik dalam mendeteksi semua kasus positif dengan recall sebesar 100%. Namun, hal ini dicapai dengan mengorbankan presisi, yang berarti banyak prediksi positif yang sebenarnya salah.

4. Ontologi memiliki F1 Score yang lebih tinggi yaitu 80.95% dibandingkan dengan Random Forest dan Decision Tree yang hanya sebesar 66.67%. Ini menunjukkan bahwa ontologi memiliki keseimbangan yang lebih baik antara presisi dan recall, membuatnya lebih efektif dalam memberikan prediksi yang benar secara konsisten.

5.2 Saran

Berdasarkan hasil penelitian yang didapatkan, berikut adalah saran pengembangan yang dapat dilakukan untuk perbaikan penelitian :

1. Berdasarkan hasil penelitian, disarankan untuk menggunakan Ontologi sebagai metode utama dalam kasus di mana keseimbangan antara mendeteksi kasus positif dan meminimalkan *false positives* adalah penting. Ontologi menunjukkan kinerja yang lebih seimbang dan konsisten.
2. Implementasi metode *Decision Tree* dapat digunakan dalam situasi di mana recall yang tinggi sangat penting, seperti dalam aplikasi yang memerlukan deteksi semua kasus positif tanpa menghiraukan jumlah *false positives*.
3. Pengembangan dan peningkatan untuk meningkatkan performa *Decision Tree*, disarankan untuk mengembangkan metode tambahan atau teknik *tuning* yang dapat meningkatkan presisi tanpa mengorbankan recall yang tinggi. Metode seperti *pruning*, penggunaan ensemble methods, atau peningkatan kualitas data pelatihan dapat dipertimbangkan.
4. Implementasi dan uji coba pada saat sebelum implementasi penuh, uji coba kedua metode pada data yang lebih luas dan beragam sangat disarankan. Hal ini untuk memastikan bahwa hasil yang diperoleh konsisten dan dapat diandalkan dalam berbagai kondisi yang mungkin dihadapi dalam aplikasi nyata.

Dengan mempertimbangkan kesimpulan dan saran ini, diharapkan penggunaan metode yang tepat dapat meningkatkan kinerja sistem klasifikasi secara keseluruhan, memberikan hasil yang lebih akurat dan andal.

DAFTAR PUSTAKA

- K. Eilers, C. Peters, and J. M. Leimeister, "Why the agile mindset matters," *Technol Forecast Soc Change*, vol. 179, Jun. 2022, doi: 10.1016/j.techfore.2022.121650.
- L. Khoza and C. Marnewick, "Waterfall and agile information system project success rates-a South African perspective," *South African Computer Journal*, vol. 32, no. 1, pp. 43–73, Jul. 2020, doi: 10.18489/sacj.v32i1.683.
- J. Miller and P. Gaida, "On the agile mindset of an effective team - An industrial opinion survey," in *Proceedings of the 2019 Federated Conference on Computer Science and Information Systems, FedCSIS 2019*, Sep. 2019, pp. 841–849. doi: 10.15439/2019F198.
- E. Dutra, B. Diirr, and G. Santos, "Human Factors and their Influence on Software Development Teams - A Tertiary Study," in *ACM International Conference Proceeding Series*, Sep. 2021, pp. 442–451. doi: 10.1145/3474624.3474625.
- U. Sa'Adah, M. B. A. Rasyid, S. Rochimah, and U. L. Yuhana, "Generating Team Quality Formula to Predict Product Quality in Software Engineering Project of College Students," in *Proceedings of 2021 13th International Conference on Information and Communication Technology and System, ICTS 2021*, 2021, pp. 106–111. doi: 10.1109/ICTS52701.2021.9607916.
- A. Sidky, J. Arthur, and S. Bohner, "A disciplined approach to adopting Agile practices: The agile adoption framework," *Innov Syst Softw Eng*, vol. 3, no. 3, pp. 203–216, 2007, doi: 10.1007/s11334-007-0026-z.
- G. C. Flores, A. M. Moreno, and L. Peters, "Agile and Software Project Management Antipatterns: Clarifying the Partnership," *IEEE Softw*, vol. 38, no. 5, pp. 39–47, 2021, doi: 10.1109/MS.2020.3001030.
- M. Leppänen, "A Comparative Analysis of Agile Maturity Models," *Information Systems Development*, vol. 35, pp. 329–343, 2013, doi: 10.1007/978-1-4614-4951-5_27.
- H. Looks, J. Fangmann, J. Thomaschewski, M. J. Escalona, and E. M. Schön, "Towards a Standardized Questionnaire for Measuring Agility at Team Level," *Lecture Notes in Business Information Processing*, vol. 419 LNBIP, no. June, pp. 71–85, 2021, doi: 10.1007/978-3-030-78098-2_5.

- V. Berg, J. Birkeland, A. Nguyen-Duc, I. O. Pappas, and L. Jaccheri, “Achieving agility and quality in product development - an empirical study of hardware startups,” *Journal of Systems and Software*, vol. 167, 2020, doi: 10.1016/j.jss.2020.110599.
- P. A. Laplante and C. J. Neill, *Antipatterns: Identification, refactoring, and management*. 2005.
- D. Settas and I. Stamelos, “Resolving complexity and interdependence in software project management antipatterns using the dependency structure matrix,” *Studies in Computational Intelligence*, vol. 150, pp. 205–217, 2008, doi: 10.1007/978-3-540-70561-1_15.
- R. Charikleia, P. Timo, B. Eleni, and S. Ioannis, “Software Project Management Antipatterns in Students’ Projects,” *The 20th SQM 2012 conference*, pp. 173-187, 2012.
- S. Hönel, “Technical Reports Compilation: Detecting the Fire Drill anti-pattern using Source Code,” no. June, 2021, [Online].
- M. Hoegl and H. G. Gemuenden, “Teamwork Quality and the Success of Innovative Projects: A Theoretical Concept and Empirical Evidence,” *Organization Science*, vol. 12, no. 4, pp. 435–449, 2001, doi: 10.1287/orsc.12.4.435.10635.
- Y. Lindsjörn, D. I. K. Sjøberg, T. Dingsøy, G. R. Bergersen, and T. Dybå, “Teamwork quality and project success in software development: A survey of agile development teams,” *Journal of Systems and Software*, vol. 122, pp. 274–286, 2016, doi: 10.1016/j.jss.2016.09.028.
- E. Weimar, A. Nugroho, J. Visser, A. Plaat, M. Goudbeek, and A. P. Schouten, “The Influence of Teamwork Quality on Software Team Performance,” *ArXiv*, pp. 1–33, 2017.
- Y. Lindsjörn, G. R. Bergersen, T. Dingsøy, and D. I. K. Sjøberg, “Teamwork quality and team performance: Exploring differences between small and large agile projects,” *Lecture Notes in Business Information Processing*, vol. 314, pp. 267–274, 2018, doi: 10.1007/978-3-319-91602-6_19.
- A. C. D. Batista, R. M. C. R. de Souza, F. Q. B. da Silva, L. D. A. Melo, and G. Marsicano, “Teamwork quality and team success in software development: A non-exact replication study,” *International Symposium on Empirical Software Engineering and Measurement*, 2020, doi: 10.1145/3382494.3410632.

- W. J. Brown, R. C. Malveau, T. J. Mowbray, and J. Wiley, "AntiPatterns: Refactoring Software, Architectures, and Projects in Crisis," *Crisis*, vol. 3, no. 4, pp. 281–284, 1998, [Online]. Available: <http://www.amazon.com/dp/0849329949>
- Noy, Natalya F. McGuinness, Deborah L., " Ontology Development 101: A Guide to Creating Your First Ontology"
- Umi Sa'adah, Umi Laili Yuhana , Maulidan Bagus Afridian Rasyid, Siti Rochimah, " Generating Team Quality Formula to Predict Product Quality in Software Engineering Project of College Students"
- Umi Laili Yuhana, Umi Sa'adah, Siti Rochimah, Maulidan Bagus Afridian Rasyid, Chandra Kirana Jatu Indraswari "Classifying Composition of Software Development Team Using Machine Learning Techniques"
- E. Barkema and J. W. Moran, "Scoring The Tuckman Team Maturity Questionnaire Electronically," 2013. [Online]. Available: http://en.wikipedia.org/wiki/Tuckman's_stages_of_group_development, accessed 5/19/2013 <http://www.nwlink.com/~donclark/leader/teamsuv.html>, accessed 10/1/2013
- A. I. Böhmer and U. Lindemann, "Agile Innovation - Challenges while implementing agile approaches within complex mechatronic processes of large corporations.," *Proceedings of ISPIM Conferences*, no. December, pp. 1–14, 2016.

BIOGRAFI PENULIS



Nama : Chandra Kirana Jatu Indraswari
Tempat Lahir : Tulungagung
Tanggal Lahir : 31 Maret 2000
Alamat : RT.02 RW.01 Desa Batokan, Kec. Ngantru,
Kab. Tulungagung
Institusi : Institut Teknologi Sepuluh Nopember
Email : 6025221053@student.its.ac.id

A. Riwayat Pendidikan

- SD Mojoagung (2006-2012)
- SMP Negeri 1 Kedungwaru (2012-2015)
- Multimedia - SMK Negeri 1 Boyolangu (2015-2018)
- D4 Teknik Informatika - Politeknik Elektronika Negeri Surabaya (2018-2022)
- S2 Teknik Informatika - Institut Teknologi Sepuluh Nopember (2022-2024)

B. Kegiatan Penelitian

- Stimasi Kualitas Produk Berdasarkan Kepribadian dan Peran Dalam Tim Pengembangan Perangkat Lunak Berbasis Agile di Perguruan Tinggi Vokasi (Penelitian Disertasi Dosen PENS Bu Umi Sa'adah) pada tahun 2023-2024.

C. Publikasi Ilmiah Studi Magister

- Classifying Composition of Software Development Team Using Machine Learning Techniques on International Conference of Computer Engineering, Network, and Intelligent Multimedia (CENIM) di Surabaya Tahun 2022.
- Ontology Identification to Detect The Quality of Software Products Based on Team Quality on International Seminar on Intelligent Technology and Its Applications (ISITIA) di Lombok (Online) Tahun 2024.