

## **BAB II**

### **TINJAUAN PUSTAKA**

Pada bab ini akan menguraikan tinjauan pustaka dari literatur yang dirujuk pada penelitian ini.

#### **2.1 Rekayasa Kebutuhan**

Rekayasa kebutuhan adalah tahap pertama pada rekayasa perangkat lunak serta merupakan bagian dari rekayasa perangkat lunak yang ditujukan buat memecahkan masalah yang berkaitan sama tujuan, fungsi, dan batasan perangkat lunak, dikarenakan rekayasa persyaratan adalah alat dengan tujuan mengumpulkan persyaratan, berfokus pada analisis dan dokumentasi. Selain itu, Rekayasa kebutuhan memiliki pengaruh paling dominan terhadap kemampuan produk perangkat lunak akhir [15]. Menurut (Falih, 2019) tahapan rekayasa kebutuhan adalah tahapan yang penting dalam pengembangan perangkat lunak [17]. Kebanyakan kegagalan dalam mengembangkan perangkat lunak disebabkan oleh spesifikasi kebutuhan yang tidak konsisten, tidak lengkap, dan salah. Jika persyaratan tidak dinyatakan dengan jelas, pengembang perangkat lunak dapat merasakan, menafsirkan, menerapkan, dan mengevaluasinya secara berbeda. Hal ini dapat mengakibatkan perangkat lunak yang tidak sesuai dengan harapan pengguna, berkualitas buruk karena memicu ketidakpuasan di antara para pemangku kepentingan, dan membuang-buang waktu dan uang untuk pengembangan perangkat lunak.

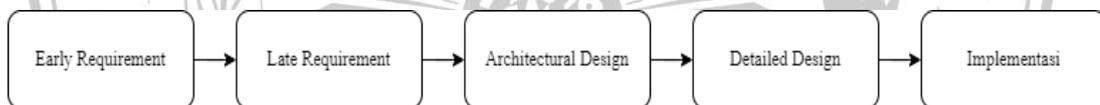
#### **2.2 Goal Oriented Requirement Engineering**

Dalam rekayasa kebutuhan ada salah satu metode yang sering digunakan yaitu Goal Oriented Requirements Engineering (GORE). Gore merupakan pendekatan dalam rekayasa kebutuhan berorientasi pada identifikasi tujuan [7]. Tujuan diubah menjadi requirement. Gore mendefinisikan tujuan sebagai tujuan yang ingin dicapai oleh suatu sistem melalui kerjasama antara agen dengan perangkat lunak dan lingkungan pelanggan [18]. Gore membantu mengidentifikasi kebutuhan dalam bentuk tujuan tingkat tinggi yang harus disertakan dalam perangkat lunak sementara berdasarkan kebutuhan stakeholder. Gore memiliki beberapa keunggulan, antara lain memberikan alasan persyaratan serta dapat menunjukkan struktur yang mudah dimengerti pada pembuatan dokumen persyaratan perangkat lunak [19]. Tujuan dari metode Gore berkaitan dengan aktivitas yang mengarah ke kebutuhan utama perangkat lunak. Terutama mencakup kegiatan-kegiatan, membawa tujuan, penyempurnaan tujuan dan berbagai jenis analisis tujuan, dan tanggung jawab tugas untuk tujuan bagi agen. Ada beberapa metode Gore yaitu NFR, i\*/TROPOS, KAOS, GBRAM [7]. Selain itu juga ada Deriving Tabular Event-Based Specifications from goal oriented requirement model (DTEBS), GBRAM (Goal-Based Requirements Analysis Method), Visual Variability Analysis for goal models (VVA), Goal-Oriented Idea

Generation Method (GOIG), Deriving Operational Software Specifications (DOSS), Agent-Based Tactics for goal-oriented requirements elaboration (A-BT), and goal oriented requirement elicitation based on General System Thinking Heuristics (GSTH) [6].

### 2.3 Tropos

Tropos merupakan salah satu metode dalam identifikasi kebutuhan perangkat lunak yang terdapat pada Gore, yang bertujuan untuk menganalisis dan merancang proses mengembangkan perangkat lunak dari awal hingga akhir. Secara khusus, Tropos mengandalkan gagasan ide untuk mengembangkan model sistem prospektif pada lingkungannya yang disempurnakan dan diperluas secara bertahap dengan menyajikan antarmuka umum untuk berbagi macam aktivitas mengembangkan perangkat lunak dan menyediakan pedoman guna dokumentasi serta evolusi perangkat lunak [7]. Metode Tropos menggunakan dari metode i\*, menambahkan desain arsitektural, desain detail, dan berkembang menjadi metode berorientasi agen dalam proses identifikasi rekayasa kebutuhan. Dibandingkan dengan metode lain untuk mengidentifikasi kebutuhan rekayasa berorientasi tujuan, metode Tropos mengungguli metode lain dalam fase pengembangan sistem [20]. Berdasarkan kajian dari peneliti sebelumnya [7], [14], [15], [20], [21], dalam implementasinya metode Tropos memiliki 5 tahapan yaitu early requirement, late requirement, architectural design, detailed design dan implementasi.

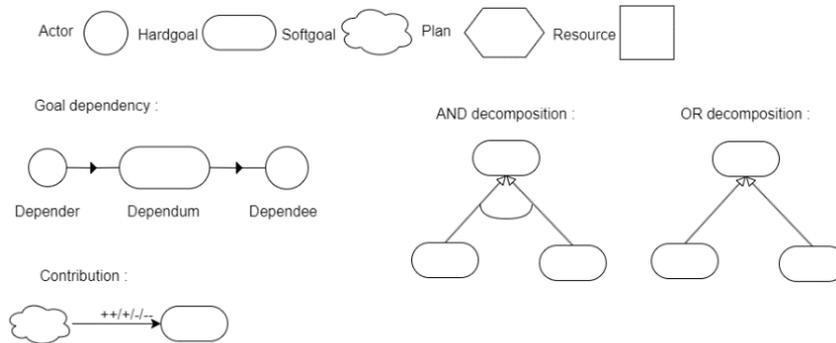


**Gambar 2.1** Proses pengembangan metode Tropos [21].

Terdapat 2 tahapan utama pada metode Tropos yaitu, early requirement dan late requirement. Kedua tahapan ini mempunyai pendekatan konseptual dan metodologi yang serupa [14]. Pada tahap early requirement seluruh stakeholder yang terlibat dalam sistem sekarang akan dilakukan identifikasi serta perannya masing-masing, dan akan digambarkan kedalam bentuk sebuah diagram yang saling terkait [22]. Pertanyaan yang dapat digunakan sebagai panduan awal untuk tahapan ini adalah siapa stakeholder dalam sistem yang sedang berjalan, apa tujuannya, bagaimana satu tujuan dan lainnya saling terkait, dan apa strategi aktor yang saling terkait untuk mencapai tujuan itu [20].

Tahap Late requirement bertujuan untuk mengidentifikasi perubahan domain yang disebabkan oleh perubahan pada sistem saat ini. Fokus dari tahap ini adalah sistem yang akan ditentukan, sehingga pada tahap ini tujuan utama yang diperoleh pada tahap sebelumnya direduksi menjadi beberapa tujuan kecil di bawahnya [20]. Pada tahap ini akan diperkenalkan aktor baru yang akan dideklarasikan dalam sistem. Peran baru ini berarti peran yang dapat digantikan atau dilakukan oleh sistem. Pertanyaan yang dapat digunakan sebagai panduan awal untuk fase

kebutuhan selanjutnya adalah target mana yang berhasil diganti atau ditangani oleh sistem baru ini, dan dependensi mana yang berhasil dipindahkan atau diganti dari aktor utama untuk sistem baru [22]. Pemodelan dalam metode Tropos didasari oleh metamodel atas konseptual dengan notasi-notasinya sebagai berikut [21]:



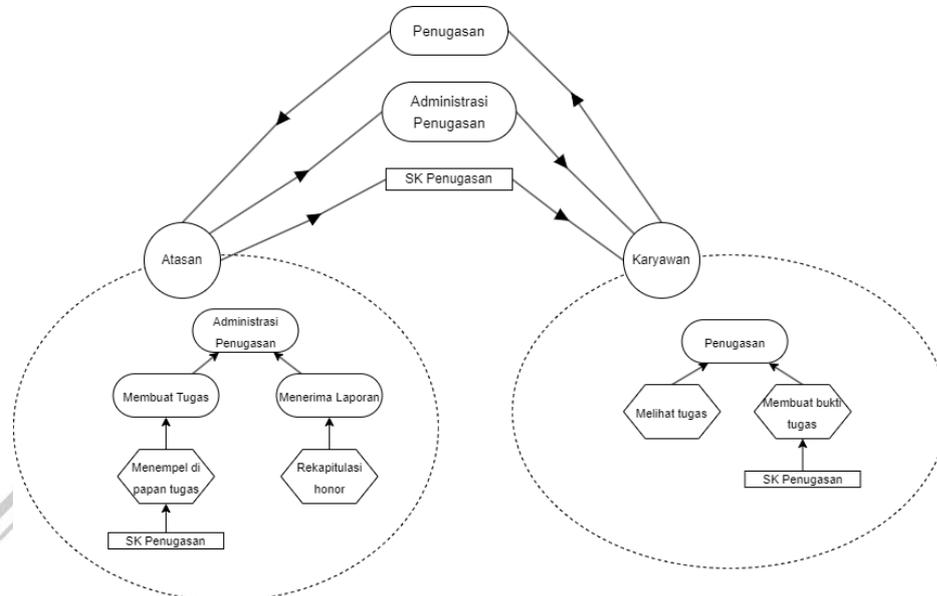
**Gambar 2.2** Notasi pada pemodelan Tropos [21].

1. Goal: terdapat dua jenis yaitu hardgoal dan softgoal. Hardgoal didasarkan pada tujuan actor yang harus dicapai. Sedangkan softgoal dalam mengembangkan perangkat lunak bisa disamakan dengan kebutuhan non-fungsional.
2. Actor: actor dapat berupa agen atau karakter fisik, social atau software. Dalam metode Tropos, actor didefinisikan sebagai karakteristik perilaku dan posisi actor social yang dinyatakan.
3. Rencana: menyatakan bagaimana suatu tujuan akan tercapai. Menerapkan rencana yang artinya bekerja menuju goal.
4. Dependensi: merupakan dua actor yang menunjukkan bahwa satu actor bergantung pada actor lain untuk mencapai tujuan, mengimplementasikan rencana, membutuhkan atau menghasilkan sumber daya. Actor yang bergantung pada actor lainnya disebut dependen, actor yang dianggap dependen disebut dependee, dan objek yang berada dipusat yang saling kebergantungan disebut dependum.
5. Recourse: sumber yang mewakili entitas fisik atau informasi yang diinginkan oleh aktor.

### 2.3.1 Early Requirement

Tahap awal dalam melakukan analisis menggunakan metode Tropos yaitu early requirement. Tahapan ini akan mengidentifikasi stakeholder yang berhubungan dengan sistem dan memodelkannya sebagai aktor sosial yang berkerja sama untuk mencapai tujuan, melaksanakan rencana, dan menyediakan sumber daya. Dengan mendefinisikan dependensi ini secara jelas, selain apa dan bagaimana sistem berfungsi, dimungkinkan untuk menyatakan mengapa dan apa hasil akhirnya untuk memverifikasi bagaimana implementasi akhir cocok dengan persyaratan awal [23]. Tahapan ini menggambarkan diagram yang terdiri atas aktor, goal, dan

sumber daya yang diperlukan guna mencapai tujuan serta hubungan antara komponen [14]. Tahapan ini juga memodelkan kondisi domain saat ini yang berkaitan dengan masalah yang sedang terjadi pada proses bisnis. Berikut contoh model diagram Early Requirement dari hasil penelitian Fadli dkk [15].



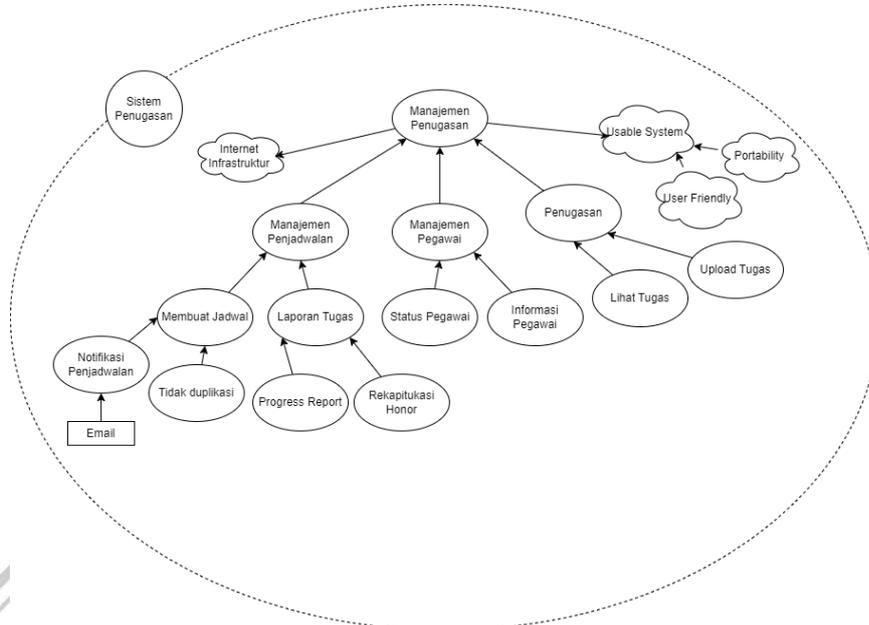
**Gambar 2.3** Contoh Model Early Requirement [15].

Dari model diatas dapat dilihat bahwa pada penelitian Fadli dkk [15], memiliki 2 actor yaitu atasan dan karyawan. Actor atasan memiliki goal “Administrasi Penugasan” didekomposisi menerapkan AND-Decomposition menjadi beberapa subgoal yaitu membuat tugas dan menerima laporan dari subgoal tersebut akan dilakukan tahapan selanjutnya menggunakan plan. Subgoal membuat tugas pertama atasan menempelkan tugas di papan penugasan, yang membuat karyawan mengetahui pekerjaan yang harus dilakukan, karyawan harus datang ke kantor untuk mengetahui tugas yang sudah disediakan. Subgoal membuat laporan karyawan yang sudah menyelesaikan tugas disegerakan membuat bukti laporan yang digunakan guna rekapitulasi honor. Sedangkan karyawan mempunyai goal “Penugasan” yang dipecah jadi beberapa subgoal yang berupa plan untuk mendukung goal yaitu melihat tugas dan membuat bukti tugas. Hubungan antar aktor tersebut yang sudah dijabarkan untuk memenuhi kebutuhan. Setelah pemodelan early requirement selesai, maka dilanjutkan dengan melakukan pemodelan late requirement.

### 2.3.2 Late Requirement

Tahap selanjutnya dalam melakukan analisis menggunakan metode Tropos yaitu late requirement. Tahapan ini akan dilakukan identifikasi tujuan yang lebih detail berdasarkan pada tahap identifikasi kebutuhan sebelumnya, dengan cara memecah tujuan utama jadi subgoal berdasarkan fungsi setiap aktor. Tahap ini fokus pada system-to-be di lingkungannya. Tahap ini dimulai dengan mengenalkan model utama, aktor baru, serta membangun keterikatan terhadap aktor lain.

Keterikatan ini menentukan kebutuhan fungsional dan non-fungsional dari sistem [21]. Berikut contoh model late requirement dari hasil penelitian Fadli dkk [15].



**Gambar 2.4** Contoh Model Late Requirement [15].

Gambar 2.4 adalah sistem penugasan aktor baru yang didapatkan sesuai dengan early requirement dan menunjukkan sekumpulan goal, subgoal, softgoal, dan resource serta perubahan yang mengakomodir sistem yang dibangun. Tujuannya adalah meningkatkan efisiensi operasional komponen program TVRI Jawa Barat. Ada beberapa peran dalam sistem tersebut yaitu admin, atasan dan karyawan. Sistem mempermudah kegiatan penugasan di TVRI Jawa Barat, admin akan diberitahukan melalui email untuk memberikan tugas kepada karyawan. Karyawan yang menyelesaikan tugas akan mengunggah tugas dan mengirimkannya ke atasan melalui sistem tugas. Hasil akhir analisis persyaratan akhir berupa analisis aktor, goal, subgoal, dan kontribusi softgoal beserta fungsionalitas dan non-fungsional di sistem penugasan yang akan dibangun. Untuk lebih jelas dapat dilihat pada Tabel 2.1.

**Tabel 2.1** Contoh Kebutuhan sistem [15].

No	Actor	Goal	Subgoal	Evaluasi
1.	Atasan	Laporan Tugas	<ul style="list-style-type: none"> <li>• Menerima laporan dari pegawai</li> <li>• Rekap laporan</li> </ul>	<ul style="list-style-type: none"> <li>• Menampilkan daftar acara yang telah selesai</li> <li>• Mengunduh bukti dari pegawai</li> </ul>
2.	Pegawai	Penugasan	<ul style="list-style-type: none"> <li>• Melihat Jadwal</li> <li>• Membuat bukti Tugas</li> </ul>	<ul style="list-style-type: none"> <li>• Menampilkan daftar acara yang sedang berlangsung</li> </ul>

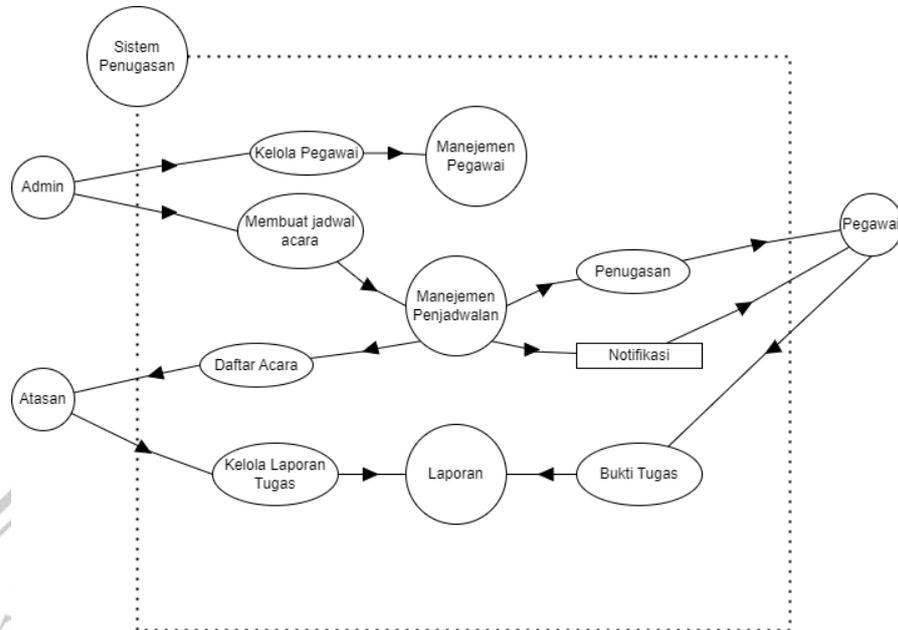
				<ul style="list-style-type: none"> <li>• Submit bukti tugas</li> </ul>
3.	Admin	Buat Tugas	<ul style="list-style-type: none"> <li>• Membuat jadwal</li> <li>• Lihat tugas</li> </ul>	<ul style="list-style-type: none"> <li>• Submit acara yang akan dibuat</li> <li>• Menampilkan daftar acara</li> </ul>
		Manajemen Pegawai	<ul style="list-style-type: none"> <li>• Status pegawai</li> <li>• Informasi pegawai</li> </ul>	<ul style="list-style-type: none"> <li>• Menampilkan bahwa pegawai dalam tugas atau tidak</li> </ul>
4.	Sistem Penugasan	Manajemen Penugasan	<ul style="list-style-type: none"> <li>• Membuat jadwal</li> <li>• Notifikasi penjadwalan</li> <li>• Tidak duplikasi</li> <li>• Laporan</li> <li>• Progress report</li> <li>• Manajemen pegawai</li> <li>• Status pegawai</li> <li>• Lihat tugas</li> <li>• Upload bukti tugas</li> </ul>	<ul style="list-style-type: none"> <li>• Submit bukti tugas</li> <li>• Notifikasi melalui email</li> <li>• Submit bukti tugas</li> <li>• Menampilkan acara sedang berlangsung</li> <li>• Kelola pegawai (CRUD)</li> <li>• Menampilkan bahwa pegawai dalam tugas atau tidak</li> </ul>

Tabel 2.1 merupakan contoh tabel kebutuhan sistem dari hasil penelitian Fadli dkk [15]. Tabel ini akan dijadikan acuan dalam penelitian ini.

### 2.3.3 Architectural Design

Architectural desain dibangun dengan menggambarkan aktor-aktor yang bertugas mengeksekusi tujuan utama dari sistem yang dibangun, dengan tujuan menguraikan kompleksitas sistem yang dibangun, yang dideskripsi dari tujuan tersebut untuk dibuat menjadi komponen-komponen yang lebih kecil atau sub-tujuan untuk membuat desain, implementasi dan manajemen lebih mudah [21]. Tahap ini juga terkait pada detail fungsional agen perangkat lunak pada sistem dan hubungan yang terjadi, berfokus di bagian input-output, sehingga mendefinisikan secara detail bagaimana setiap agen menjalankan rencana untuk mencapai tujuannya [22]. Desain arsitektural menjelaskan sekumpulan tipe agen dan memberikan tugas satu atau lebih fungsi yang berbeda untuk masing-masingnya (penugasan agen). Secara umum, penugasan agen tidak unik dan bergantung pada perancang. Jumlah agen dan kapabilitas yang ditugaskan ke masing-masing agen adalah pilihan yang didorong oleh analisi yang diperluas dari grafik aktor dan cara perancang memandang sistem dalam kaitannya dengan agen [7]. Tahap selanjutnya dalam memodelkan langkah yang lebih detail adalah mengambil spesifikasi input

yang dihasilkan selama tahap desain arsitektural. Tahap ini akan menggunakan diagram aktivitas UML untuk mewakili fungsi dan rencana. Berikut contoh model architectural design.



**Gambar 2.5** Contoh Model Architectural Design [15].

Gambar 2.5 merupakan contoh model architectural design dari hasil studi Fadli dkk, tahun 2019 [15]. Dari model tersebut menampilkan bahwa model tujuan buat beberapa subaktor di sistem penugasan TVRI Jawa Barat terdiri atas, manajemen pegawai, manajemen penjadwalan, dan laporan. Pada studi tersebut setiap subaktor mempunyai analisis tujuan yang bergantung dengan aktor, kapabilitas, dan subaktor. Untuk lebih jelas dapat dilihat pada tabel 2.2.

**Tabel 2.2** Contoh Relasi Antar Actor Capability dan Sub Actor [15].

No	Actor	Capability	Sub Actor
1.	Admin	Kelola Pegawai	Manajemen Pegawai
2.	Admin	Membuat Jadwal	Manajemen Penjadwalan
3.	Atasan	Daftar Acara	Manajemen Penjadwalan
4.	Atasan	Kelola Laporan Tugas	Laporan
5.	Pegawai	Penugasan	Manajemen Penjadwalan
6.	Pegawai	Notifikasi	Manajemen Penjadwalan
7.	Pegawai	Bukti Tugas	Laporan

Tabel 2.2 merupakan contoh tabel relasi antar actor capability dan Sub actor dari hasil penelitian Fadli dkk [15]. Tabel ini akan dijadikan acuan penelitian ini.

### **2.3.4 Detailed Design**

Setelah tahap architectural design sudah dilakukan, selanjutnya membuat detail design dengan mengambil input spesifikasi yang diperoleh dari tahapan sebelumnya. Pada tahap ini akan modelkan kapabilitas, protokol, dan tugas atau rencana ditentukan secara rinci, dan diagram aktivitas UML digunakan untuk merepresentasikan kapabilitas [7]. Diagram aktivitas UML memungkinkan untuk memodelkan kapabilitas (atau set kapabilitas terkait) dari perspektif agen tertentu. Tahap ini didasarkan pada tahap desain arsitektural, dan desain sistem dapat diselesaikan lebih detail. Rencana rinci dengan sub-aktor dan kompetensi yang terkait dengan setiap tujuan dari masing-masing aktor. Dilanjutkan dengan menggunakan diagram UML, yaitu diagram aktivitas dimana masing-masing fungsi dapat dimodelkan dari sudut pandang masing-masing aktor terhadap sub-aktor.

### **2.4 Acceptance Criteria**

Acceptance Criteria adalah requirement yang menjelaskan hasil yang diinginkan dari requirement, fitur, atau user story. Acceptance Criteria merupakan metode yang digunakan untuk menentukan standar produk atau layanan penerima. Metode ini melibatkan pengguna menentukan standar penerima dan memastikan bahwa produk atau layanan yang diproduksi memenuhi kebutuhan dan harapan pengguna [24]. Metode ini terdiri dari beberapa bagian yaitu, given, when, dan then. Given menjelaskan kondisi awal sebelum pengguna melakukan beberapa aksi pada fungsi atau produk. When menjelaskan pengguna mengambil tindakan fungsi atau produk. Then menjelaskan menjelaskan hasil yang diharapkan setelah mengambil fungsi atau tindakan produk [25]. Saat merumuskan acceptance criteria banyak hal harus dipertimbangkan seperti memahami masalah, kebutuhan serta harapan pengguna, memahami tujuan, memahami solusi yang diambil, memahami proses bisnis, mengumpulkan dan mencatat seluruh informasi pendukung, merumuskan, dan terakhir finalisasi [26].

Acceptance Criteria dapat menentukan apa yang diinginkan oleh pengguna akhir, pelanggan, atau klien. Acceptance Criteria juga bisa berupa daftar kriteria untuk menunjukkan bahwa cerita pengguna telah selesai. Acceptance criteria digunakan sebagai pedoman untuk melakukan User Acceptance Testing (UAT) untuk memastikan terpenuhinya persyaratan [27]. Dalam proses identifikasi dan validasi requirement, acceptance criteria penting untuk memastikan bahwa requirement terpenuhi. Acceptance criteria juga dapat membantu mengurangi kebutuhan akan desain ulang dan memberikan dasar yang realistis untuk memperkirakan biaya, risiko, dan jadwal produksi [28].

Acceptance criteria dapat didetailkan dengan menggunakan acceptance test, atau biasa disebut dengan acceptance criteria. Acceptance test akan mengkonfirmasi maksud sebenarnya dari pelanggan. Beberapa asumsi dan

ekspektasi terkait cerita pengguna dapat dikomunikasikan melalui Acceptance test. Dengan Acceptance test, masalah komunikasi antara tim pengembangan dan klien dapat dikurangkan. Pengujian penerimaan menangkap harapan pelanggan dan kemudian mengkomunikasikan harapan tersebut kepada pengembang.

## **2.5 Penelitian Terkait**

Dalam penelitian yang dilakukan oleh Bresciani dkk [23] Tropos memiliki beberapa kelebihan dibandingkan dengan metode rekayasa kebutuhan lainnya, yaitu Tropos dapat menganalisis kebutuhan secara holistik dengan mempertimbangkan aspek-aspek sosial, organisasional, dan lingkungan dari domain dan sistem, Tropos dapat menganalisis kebutuhan secara iteratif dengan memperbaiki dan memperluas model-model kebutuhan seiring dengan perkembangan pengetahuan tentang domain dan sistem, Tropos dapat menganalisis kebutuhan secara konsisten dengan menggunakan notasi grafis yang sama untuk semua tahapan rekayasa kebutuhan, dan Tropos dapat menganalisis kebutuhan secara fleksibel dengan memungkinkan adanya variasi dan alternatif dalam mencapai tujuan-tujuan.

Tropos telah digunakan dalam banyak penelitian pengembangan aplikasi. Seperti yang dilakukan oleh Rinda dkk [7]. Pada penelitian tersebut menganalisis kebutuhan sistem manajemen stok darah pada Palang Merah Indonesia (PMI) Kota Bandung dengan menggunakan metode Tropos, yang dimana terdapat masalah yaitu kesulitannya keluarga donor darah dalam mendapatkan informasi, bentuk informasi yang dibutuhkan ialah jadwal unit donor darah, ketersediaan stok darah, dan pendaftaran donor darah. Tropos digunakan dengan tujuan untuk terfokus terhadap identifikasi kebutuhan di tahapan pemodelan persyaratan awal dan persyaratan akhir. Hasil dari penelitian tersebut didapatkan bahwasannya Tropos yang diterapkan pada sistem manajemen stok darah pada PMI Kota Bandung dapat diterapkan karena menunjukkan bahwa hasil pemodelan Tropos yang memuaskan adalah 3,55% dan untuk hasil 0,45% masih ada kekurangan fungsionalitas yang akan dibangun.

Penelitian lain yang menggunakan Tropos dalam pengembangan aplikasi yaitu yang dilakukan oleh Farhan dkk [14]. Penelitian tersebut bertujuan untuk mengevaluasi proses rekayasa kebutuhan sistem di pengelolaan gudang mobil PT. Istana Bandung Raya Motor dengan menerapkan metode Tropos. Pada PT. Istana Bandung Raya Motor khususnya pada bagian gudang proses stok opname dilakukan masih secara manual, yang membuat terjadi kesalahan komunikasi antara pemangku kepentingan dan kesalahan manusia pada pembuatan laporan. Tropos yang digunakan pada penelitian ini terfokus di proses rekayasa kebutuhan sistem dan guna memahami bagaimana cara mengevaluasi sistem yang dikembangkan sesuai pemodelan. Dari penelitian ini didapatkan hasil bahwasannya Tropos dapat digunakan sebagai metodologi dalam merekayasa kebutuhan sistem pengelolaan

gudang yang membantu dalam menangani proses bisnis stok opname. Pemodelan yang dihasilkan dari tahapan pemodelan Tropos juga diselaraskan dengan tujuan masing-masing stakeholder dalam proses bisnis yang sedang berjalan.

