CHAPTER 1: THE PROBLEM

1.1 Rationale

Recommender System is a system to recommends a product desired by the user. The Recommender System must manage the user model in the recommendation process so that the recommended products fulfill user needs [1], [2]. The weakness of the recommender system is that at the first interaction the user's needs cannot be detailed, because users rarely state their needs at the beginning of the interaction, thus making the recommended product not in accordance with user preferences. For this reason, conversational recommender system (CRS) research has been develop [3]. CRS is a form of recommender system (knowledge-based) that uses a conversation mechanism to obtain the product needed by the user [2]. The advantage of this CRS is that during the interaction between the user-system, the system can imitate a sales professional to detail the user's needs, so that the recommended product is in accordance with the user's needs [3]. The characteristic of CRS is that the interaction consists of three components, that are generating question, product recommendation, and explanation facility [3].

User requirements on CRS are very important because this will have an impact on the resulting CRS interactions (generating questions and recommendation process). One form of user requirements is product functional requirements and product technical features (individual items) [4]. Product functional requirements are product requirements that are seen in terms of functionality or usability. For example, users need a smartphone that can be used to play games. CRS based on product functional requirements, can accommodate users who do not understand technical features. However, this CRS based on product functional requirements makes the interaction between the user and the system slower [1], [3], [5]–[8]. Meanwhile, product technical features are technical features that exist in the product, for example RAM, primary camera, secondary camera, etc. CRS based on product technical features More effective for products that have few technical features and is implemented for expert users. However, Novice users (users who are not familiar with technical features) have difficulty expressing their needs [9]–[12].

Compound critiquing is a feedback-based approach (critiquing) on the recommender system [2]. This approach performs critiquing by using more than one item of property being

criticized (multi-item property) [11]. Compound critiquing is often used in the domain of high tech products that have complex attributes [8], [9], [11], [13], because it is considered to increase the efficiency of interaction in CRS. When processing generating criticism, the algorithm that is often used is Apriori [11], [14]. Apriori is one of the algorithms for data mining associations rules [14]. In addition to the a priori algorithm, there are other algorithms in association rules, namely FP-Growth and ECLAT [15]. Both algorithms are considered to have better performance than Apriori [15].

By combining product functional requirements and technical features in CRS, it is hoped that it will make it easier for novice users to express their needs and can improve the process of query refinements (detailing user requirements through special mechanisms). We will utilize compound critiquing by using the FP algorithm when generating questions (critiques) of product technical features. So that using this model is expected to increase interaction efficiency and improve performance.

1.2 Problem Statement

Functional requirements-based CRS has been widely developed, because it can accommodate users who do not understand technical features (novice users) and will be best implemented in product domains that have complex and multi-functional features. However, when user needs are still too general or user needs are still not detailed, the system will interact to detail the needs (query refinement). Functional requirements is a mapping of several technical features, so it requires repeated interactions to get specific requirements. This results in inefficient interaction between the user and the system (too many iterations of the user-system). Examples of mapping between functional requirements and technical features are: smartphones that can be used to play games, mapping technical features of RAM (high), internet speed (high), screen size (high), screen technology (high), etc.

1.3 Objective

In this research, we will combine product functional requirements and technical features in user-system interaction. To generate product functional requirements-based questions, we use a semantic reasoning approach (previous research), and to generate technical features questions we use a compound critiquing approach. The product technical features that we

use are technical specifications that are familiar to users. By using this model, it is hoped can improve efficiency of query refinement.

1.4 Hypothesis

CRS is a type of recommender system that have interaction using question-and-answer dialog between the user and the system [5]. This interaction is used to explore user requirements, so this interaction is iterative and realtime until the user's requirements are specific. It requires an efficient CRS to perform query refinement and has fast (based on running time) for generating questions [15], so it can increase user satisfaction. Query refinement is the process for detailing user requirements through special mechanisms, to recommend products according to user needs [16]. To see the efficiency of query refinement is to look at the model's ability to reduce the number of remaining records in each interaction [3]. Algorithm performance is seen when generating questions using running time (/seconds) measurements [15]. The following are the premises and hypotheses that we have compiled:

Premise 1 : Product functional requirements can make it easier for users to express their needs [1].

Premise 2 : Compound critiquing in technical features-based CRS can detail user requirements, so query refinement is more efficient [17][18].

Premise 3 : In generating frequent itemset Fp-Growth has a shorter running time than Apriori[19].

Hypothesis :

1. Combining functional requirements and technical features-based CRS can make it easier for novice users to express their needs and improve query refinement efficiency.
2. Efficient query refinement and fast algorithms (when generating questions) can increase user satisfaction with the interaction mode.
3. Fast (based on running time) for generating questions can provide fast real-time interaction between user-systems.

1.5 Assumption

CRS based on functional requirements and product technical features can increase user satisfaction and improve query refinement efficiency.

1.6 Scope and Delimitation

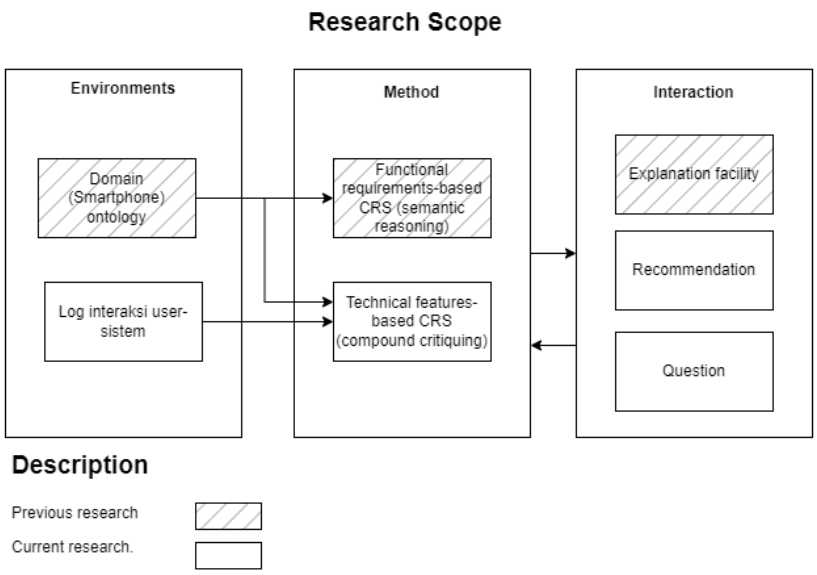

Figure 1.1 Research Scope

1. The dataset used is functional requirements and technical features.
2. The ontology used is the smartphone ontology in 2015.
3. The model evaluation used is running time, query refinement, and user satisfaction.
4. The question generation algorithm used is the FP-Growth algorithm.
5. Interactions in CRS are explanation facility, recommendation, question.

1.7 Importance of the Study

In the compound critiquing domain, the algorithm that is often used for question generation is Apriori. Apriori is an algorithm for generating frequent itemset in data mining. In addition to the Apriori algorithm there are other algorithms such as FP-Growth and Eclat. The FP-Growth and Eclat algorithm has never been implemented before in compound critiquing.