CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Refer to IEEE standard; definition of software development process is concerned primarily with the production aspect of software development, as opposed to the technical aspect, such as software tools. These processes exist primarily for supporting the management of software development, and are generally skewed toward addressing business concerns. Many software development processes can be run in a similar way to general project management processes. A clone occurs when a code fragment is identical to another code fragment according to some basic criteria. These criteria may be syntactical, semantical, or both of them (El-Matarawy, El-Ramly, & Bahgat, 2013). In software project management, it is common to reuse some code fragments by copying with or without minor modifications for use in different programs or maintained by the same entity.

Most of the software systems consist of a large number of identical code segments. These identical code segments are known as code clones. According to previous research, a software system consists of about 7% to 23% of cloned code (Yuan & Guo, 2011). There are four types of code clone which are type I, type II,
type III and type IV. Below is the diagram to show the difference of four types of code clone.

![Diagram showing types of code clone](image)

**Figure 1.1:** The Types of Code Clone

For type I for above figure, these two fragments are textually after removing the whitespace and comments. Next, for the type II of code clone in above figure shows that the two code segments change a lot in their shape, variable names and value assignments. However, the syntactic structure is still similar in both segments. Type III in the figure of types of code clone above shows that the two fragments and from the corresponding difference, all the original statements are used directly or after being changed in their identifiers or literals with one insertion in the first line, making this code fragment as Type III of code clone. Without this inserted statement, this copied fragment could be a Type II code clone. For type IV of code clone in the figure above shows that from the semantics point of view both the code fragments
are similar in their functionality and termed as Type IV semantic clones although one is a simple code fragment and another is a recursive function with no structural similarities between the statements of the two fragments.

Clones are considered harmful in software maintenance and should be removed or detected at least. However, it would have been much better if there is no clone at all in the developed system so that we would not have to think about neither removal nor detection of clones. The idea is to use a clone detection tool in the normal development process to avoid cloning in the software right from the beginning. There are two ways of how to use a clone detection tool in the development process for avoiding clones. One way is the preventive control where a new function is added to the system only after being confirmed that this new function is not a clone to any existing one or there are specific reasons of adding that function as a clone to the system. The other way is the problem mining where any modification to a function must be consistently propagated to all of its similar functions in the system. Therefore, no clones are created unnecessarily, and the probability of update anomalies is reduced significantly (Roy & Cordy, 2007).

1.2 PROBLEM STATEMENT

Over the last decade many techniques and tools for software clone detection have been proposed. This includes textual approaches and semantic approaches. Most of them are oriented to a specific computer language and they range from high precision to low precision, and from high recall to low recall (El-Matarawy et al., 2013). There are five approaches that have been used in code clone detection which are text-based approach, token-based approach, tree-based approach, metrics-based detection approach and program dependency graph-based (PDG-based) detection approach. The figures below explain of all the code clone detection approach: