

*A machining process is very dependent on the model created. The more complicated the model, the greater the design difficulty and the greater the machining process. Reduced production costs can help a company increase profits. A focus on production cost can be achieved in a number of ways, the first of which is by replacing materials or changing the design. It is better to reduce product costs during the design stage than during the manufacturing stage. The main objective of this research is to develop an application that can recognize features in a CAD program and calculate the complexity index of shapes in real time. In this study, the prismatic features and slab features classified by Jong-Yun Jung were used. The feature recognition method applied in this study is a hybrid of the rule-based and graph-based methods, which uses the STL file developed by Sunil and Pande to obtain all the information needed. Then, the results are extracted from feature recognition data and are used to calculate the product complexity index of the model being studied. This study applied the product complexity index, following the model developed earlier by El Maraghy. Validation is performed by comparing the software count with the complexity index calculated with the STEP method by Hendri and Sholeh et al. This research develops a program that recognizes features in CAD software and calculates the index complexity of shapes in real time. This will allow designers to calculate the expected complexity value during the design process. As a result, the estimated production cost can be seen early on. Finally, this software is tested for calculating the index values for the complexity of a combined features model. The use of eight slots and eight pockets as a benchmark scoring for shape produces a more accurate product complexity index*

**Keywords:** features, feature recognition, complexity index, STL file, CAD, manufacturing process

# DEVELOPMENT OF PRODUCT COMPLEXITY INDEX IN 3D MODELS USING A HYBRID FEATURE RECOGNITION METHOD WITH RULE-BASED AND GRAPH-BASED METHODS

**Hendri Dwi Saptioratri Budiono**

Doctor of Mechanical Engineering,  
Associate Professor\*

E-mail: hendri@eng.ui.ac.id

**Finno Ariandiyudha Hadiwardoyo**

Bachelor of Engineering

E-mail: finno.hadiwardoyo@gmail.com

\*Department of Mechanical Engineering  
Universitas Indonesia

Kampus Baru UI, Depok, Indonesia, 16424

Received date 05.04.2021

Accepted date 21.05.2021

Published date 10.06.2021

**How to Cite:** Budiono, H. D. S., Hadiwardoyo, F. A. (2021). Development of product complexity index in 3D models using a hybrid feature recognition method with rule-based and graph-based methods. *Eastern-European Journal of Enterprise Technologies*, 3 (1 (111)), 47–61. doi: <https://doi.org/10.15587/1729-4061.2021.227848>

## 1. Introduction

Recently, the market evolved and became more competitive, causing companies to have to work to maintain their competitive advantage. The main factor that influences competitive advantage is product cost [1]. The previous work indicates that estimating the cost of future products as soon as possible is necessary. The ability to influence the final cost decreases as work on a project advances because of the increased cost of modification [2]. The price of a product is closely related to the cost of its production. Lower production costs make cheaper final prices. A company can increase its profits by reducing its products' production costs. An emphasis on production cost can be created in several ways; first by replacing the materials or changing the design. Reducing product costs is better at the design stage than at the manufacturing stage. Designers can modify a design to obtain proper performance at a reasonable cost if the manufacturing cost can be reliably estimated during the design stage [3]. In the design stage, proactive decision-making and elimination of mistakes were performed using a cost modeling

system for lean product and process development [4]. In this way, production costs will be lowered, which also leads to the probability of improving the product.

The value of the complexity index of the product must be taken into account to estimate the overall cost of production [5]. The value of the complexity can be obtained if the features of the part are already known and identified. The more features that are contained in the part, the greater its complexity, resulting in a costly manufacturing process [6]. The complexity also significantly affects the quality, energy consumption, and overall process efficiency [7]. Production times can be shortened if the product structure is simplified by altering the design features [8]. Recently, more efficient design and development of products are demanded by the industry [9]. If a company decides to change the design of its product, its production cost also changes. However, an estimation of the production cost is only possible if the design is complete. This can be inefficient because when a design exhibits greater complexity than a previous one, it is then constrained to be altered to reduce the production cost. Changing a design can be difficult for designers.

Obtaining the value of the product's complexity index on the CAD models can be done by utilizing the feature identification method. Then, all of the information can be calculated to obtain a value for the products' complexity index [10]. These values for the complexity index can be used as a reference for determining the costs of the production of the model.

Therefore, the study is devoted to overcoming the scientific problem, that is, the development of the application, which is able to automatically recognize features in the design and calculate the complexity index of shapes in real time.

## 2. Literature review and problem statement

The paper [11] presents an explanation of Facet Model 3D, which represents a form or model in 3D space, forming a solid discretization. The facet of the model is formed from a collection of 2D form fields commonly known as facets. This program produces a model-based 3D triangulation triangle or triangular mesh, which is a finite element modeling where flat square elements can be represented as two coinciding triangles. 3D facet models can be stored in a .stl file that contains vertex, edge, and face information. This file format is commonly used in CAD programs.

The paper [12] provides the definition of feature recognition (FR), it is a geometric model process in a CAD system to find portions of the model that match the characteristics of interest. Feature identification is a method that automatically identifies the model's features as designed in CAD and extracts all information for the feature. Automatic feature extraction in the CAD model exhibits the effect of reducing lead time from the design analysis process [5]. The paper [13] proposed an intelligent FR methodology (IFRM) to develop an FR system to communicate with a range of CAD/CAM systems. An extraction of manufacturing features from engineering CAD models is being developed by [14] using a computer-based FR process.

The study of feature-based design (FBD) has been underway for several years. Definitions of features appear in areas such as feature taxonomies, mapping, validation, constraints in FBD, interfaces, information-storage schemes, and the object-oriented approach [15]. Interactive feature definition is used to facilitate the definition of features by non-programming experts and has been applied to a prototype of a re-design support system [16]. The term feature demonstrates different meanings, depending on where the term is used. Many articles defined features. According to [17], a feature is any entity that is part of the design, engineering, or manufacture of a product. [18] considers a feature to designate a unique area on the part's surface. However, many definitions are representatives of engineered geometry, assembly, or other aspects of manufacturing.

The paper [19] classified features according to their cutting type, finding four feature classes. The first type is rotational features, created using a turning machine. The second is prismatic features, which mostly use a slab machine or a face-milling machine. The third is slab features, which use an end mill machine. The last is revolving features, which use a drilling machine and a reaming machine. Fig. 1 presents each of those classes.

These features can be identified automatically using software programmed to identify features. The previous research developed a method of identifying features using the hybrid method [20]. This method forms a combination of graph-based and rule-based methods. Graph-based methods are a means of identification that generates features by utilizing

a face adjacency graph (FAG) from CAD models. An FAG constitutes a relationship between neighboring faces. An example of an FAG can be seen in Fig. 2.

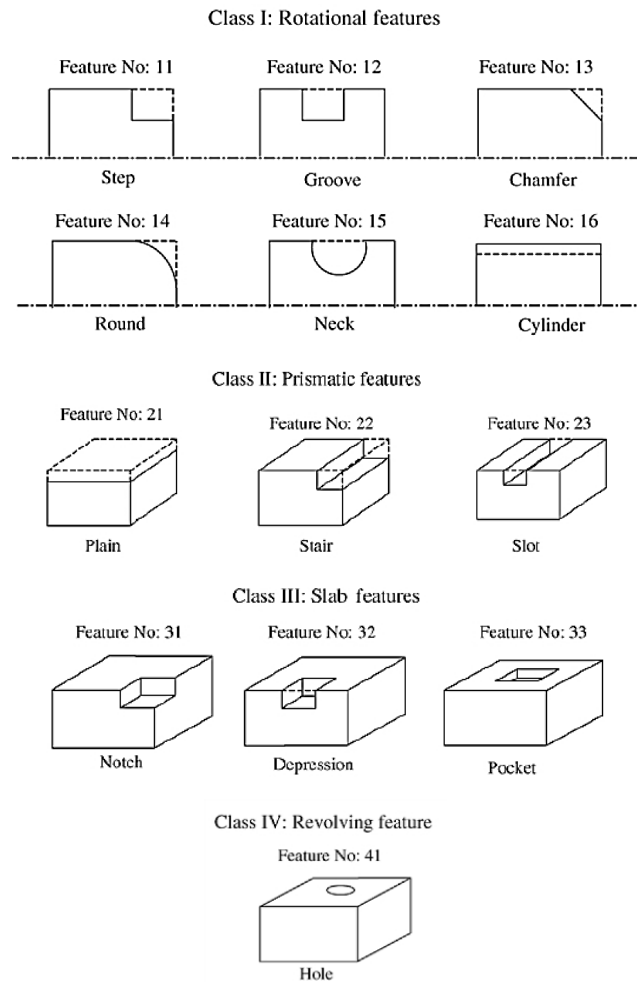


Fig. 1. Terms and classification of features [19]

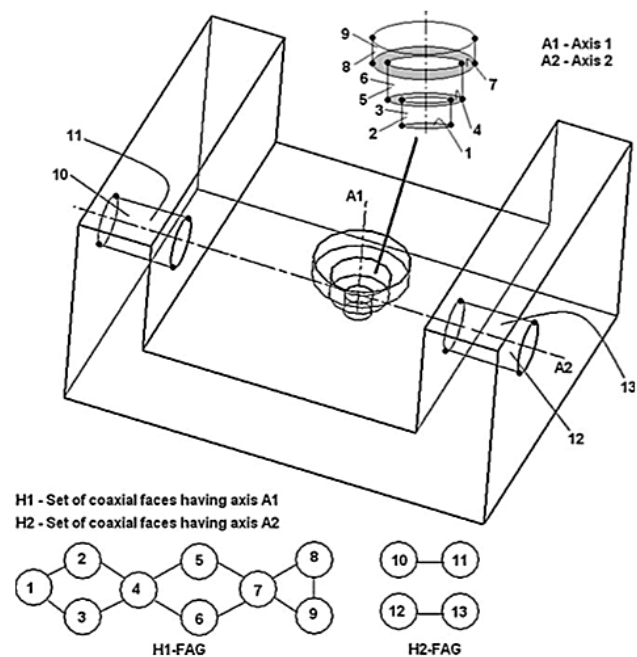


Fig. 2. Face Adjacency Graph [20]

A feature can be identified through a focus on its face adjacency and can engage a set of rules to define a certain feature; hence, a feature FAG (FFAG) is identified. The first step of identifying an FFAG is identifying the type of face, that is, a plane, cylinder, or curve. The plane face is called a base explicit feature graph (BEFG), and a cylinder or curve face is called a no-BEFG (NEFG). One unique feature can be defined through setting a certain rule in the parent-children relationship in the BEFG and NEFG.

Product complexity is affected by information and volume [21]. This information consists of the basic material used to make the product, the product design, its specifications, and its components or parts (Fig. 3, 4). This can be narrowed down to features and specifications. Features are the forms to generate, and specifications are the desired qualities associated with the generated features. The product complexity index is denoted as  $CI_{product}$ .

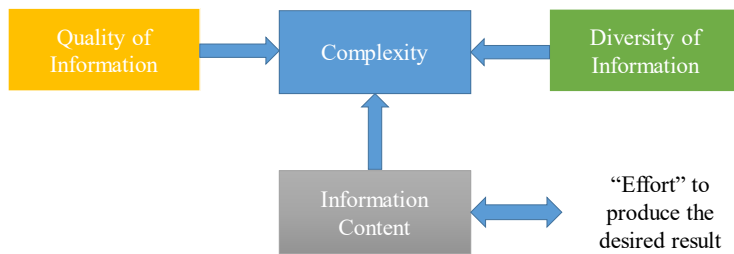


Fig. 3. Complexity Elements [5]

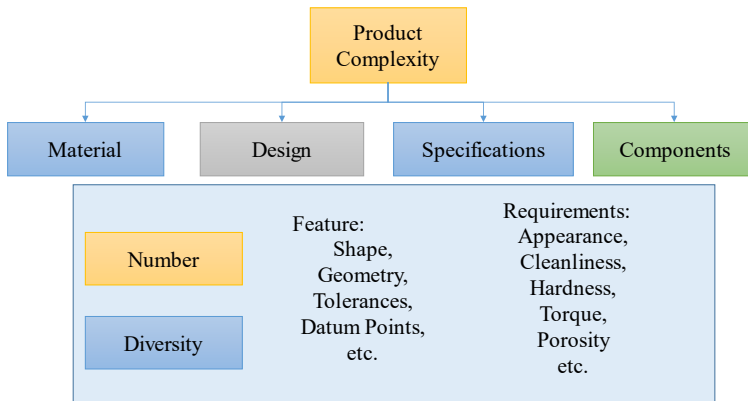


Fig. 4. Product Complexity Elements [5]

The compression factor and the size of the information of entropy  $H$  as expressed by (1) are used to represent the number of information elements:

$$H = \log_2 N + 1, \quad (1)$$

where  $N$  – the total amount of information.

Uniqueness scale or diversity ratio ( $D_R$ ) is defined as the ratio of differing information among the total information, as expressed in (2):

$$D_R = \frac{n}{N}, \quad (2)$$

where  $n$  – the amount of unique information.

The relative complexity coefficient ( $C_j$ ) has been introduced to capture the information content. The complexity of the feature or task increases with the effort expended (i.e., increased necessary stages or tools). The matrix method is

used to determine the relative complexity coefficient [21]. The complexity of the products is represented by the product complexity index ( $CI_{product}$ ) and is a function of product information/entropy ( $H_{product}$ ), product diversity ratio ( $D_{R,product}$ ), and relative product complexity coefficient ( $C_{j,product}$ ). The value of the relative product complexity coefficient ( $C_{j,product}$ ) is based on the general principles of manufacturing and depends on the type of the process or volume. Its value increases with the effort required to produce the final component of the product.

The product complexity index ( $CI_{product}$ ) is a combination of the relative ratios of diversity and complexity and is expressed using entropy information, as shown in (3):

$$CI_{product} = D_{R,product} + C_{j,product} \cdot H_{product},$$

$$CI_{product} = \frac{n}{N} + C_{j,product} \cdot \log_2 N + 1. \quad (3)$$

The product complexity coefficient  $C_{j,product}$  is defined as:

$$C_{j,product} = \sum_{f=1}^F x_f \cdot C_{f,feature}, \quad (4)$$

where  $C_{f,feature}$  – the coefficient of the relative complexity features;  $x_f$  – the percentage of the  $x$ -th factor related to dissimilar features.

$$C_{f,feature} = \frac{F_N \cdot F_{CF} + S_N \cdot S_{CF}}{F_N + S_N}, \quad (5)$$

where  $F_N$  – the number of features;  $F_{CF}$  – the feature complexity factor;  $S_N$  – the number of specifications;  $S_{CF}$  – the specification complexity factor.

$$F_{CF} = \frac{\sum_{j=1}^J factor\_level_j}{J}, \quad (6)$$

where  $J$  – the number of categories that affect the feature,  $factor\_level_j$  – the factor for the  $j$ -th category.

$$S_{CF} = \frac{\sum_{k=1}^K factor\_level_k}{K}, \quad (7)$$

where  $K$  – the number of specifications that affect the feature,  $factor\_level_k$  – the factor for the  $k$ -th specification.

From the description of the literature, it can be seen that at the early stage of design, cost estimation is not possible without feature recognition that can be used to determine the level of complexity of a product design based on the machining process. Therefore, it is necessary to develop a CAD application that can automatically calculate the complexity index.

### 3. The aim and objectives of the study

The aim of this study is to develop an application that is able to recognize features in a CAD program and calculate the complexity index of shapes in real time. To achieve this aim, the following objectives are accomplished:

- to calculate the complexity index using the STEP Method;
- to develop an application to identify prismatic and slab features automatically from designs made in CAD (com-

plexity index calculations using a hybrid feature-recognition method).

**4. Materials and Methods**

The development methods used on this research are described below.

**4.1. Software Design and Rules**

Fig. 5 presents a flow diagram for the procedure of creating the software, which is divided into two phases. The first phase is the identification of all features of the model, and the second is the calculation of the values of the product complexity index. Then, the software can be called the FR Complexity Index (FRCI).

The method used to recognize the features was developed by [20]. This is a hybrid method that is both graph-based and rule-based. Four stages are present in the process of introducing the model's features. These stages are as follows: (1) reading the STL file, (2) segmentation of the region according to the normal force, (3) formation of FAG, and (4) identification of features using certain rules.

An STL file is a CAD data format that represents a model in the form of triangular facets. Each facet exhibits a normal

force that points outward from the model and three vertices (the *x*, *y*, and *z* coordinates).

The part model with the STL format exhibits certain limitations, as follows:

- absence of topological information: the information on connectivity between the facets is not available in the STL file, and all facet data are randomly arranged;
- errors in STL files: STL files often contain errors when opened, including numerical errors, vertex-to-vertex output errors, errors in facet orientation, non-manifold conditions, and others.

This step is the first step necessary to take to obtain the information needed for each subsequent one. The coordinates and direction of the normal force obtained from the facets are necessary for the next stage.

Parsing is done at this stage to separate the facets. This process resembles numbering each facet to establish its identity. The algorithm of the process of STL parsing can be seen in Fig. 6. Reading of this STL file can only be done with STL files created in CATIA applications.

This model's segmentation into regions can be done by looking for facets with the same normal force and then grouping them together. By grouping facets according to their normal style, finding neighboring facets to establish a region becomes easier [22].

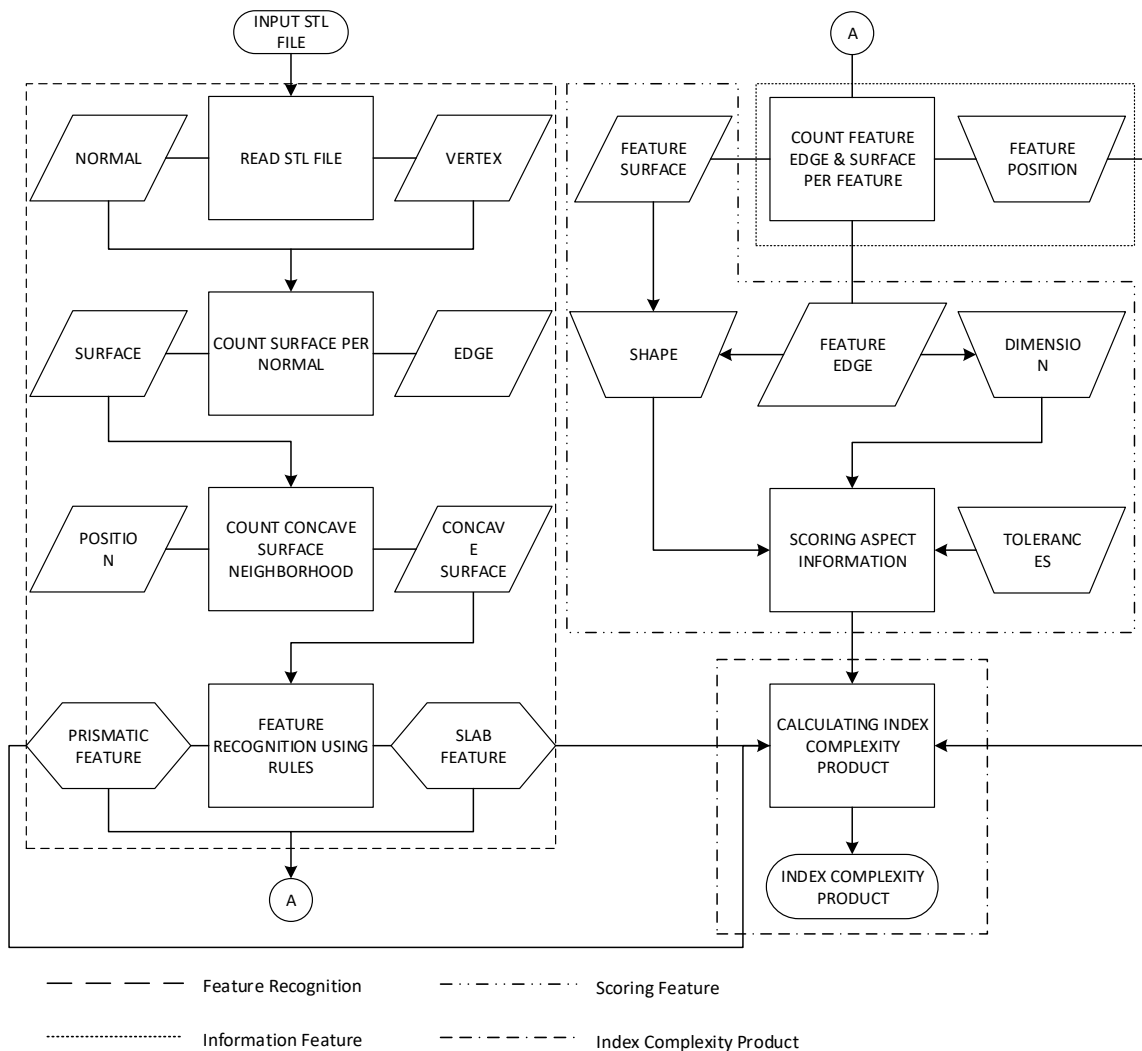


Fig. 5. Software Flow Diagram

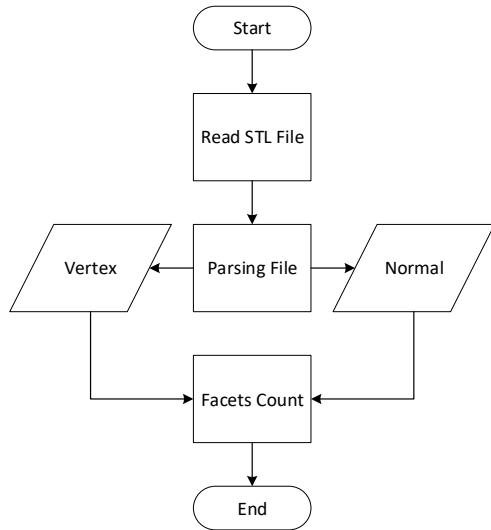


Fig. 6. STL Parsing Algorithm

This segmentation of model regions in the STL is carried out to obtain surface information from the model. A region is a set of neighboring facets. After the grouping is performed, checking the relationship between neighboring facets becomes necessary. The algorithm of this process can be seen in Fig. 7.

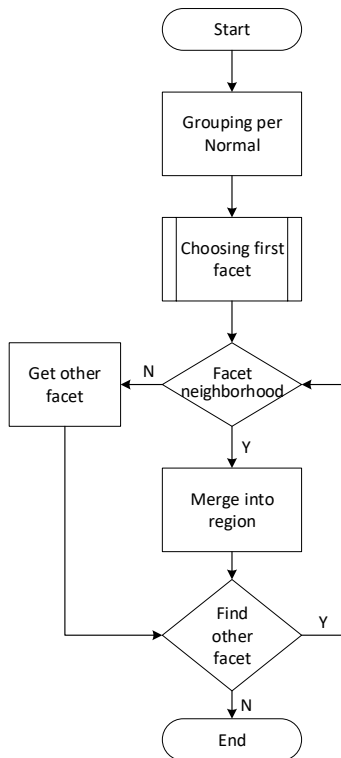


Fig. 7. Region Segmentation Algorithm

The first selection of a facet is made randomly, and the selected facet is checked for neighboring relationships to form a region. This is performed using information on the facet, namely the edge, such that the edge is obtained using information on two joined vertices. If the facets exhibit the same edge information as the neighboring facets that have been checked, then the facets are neighbors and can be grouped

into a region. Then, the neighboring relationship is checked between the region and other facets.

If the facet is a neighbor of the region, the facet can be registered in the region being checked. This process is carried out for regions with other normal forces until no more facets remain in the STL file being tested. In this study, the surface of the region is flat and not curved.

The purpose of establishing an FAG is to find a neighboring relationship between one surface and other surfaces to form an angle of 90°, a basin, an angle of 270°, or a convex form [12]. An illustration of this stage can be seen in Fig. 8.

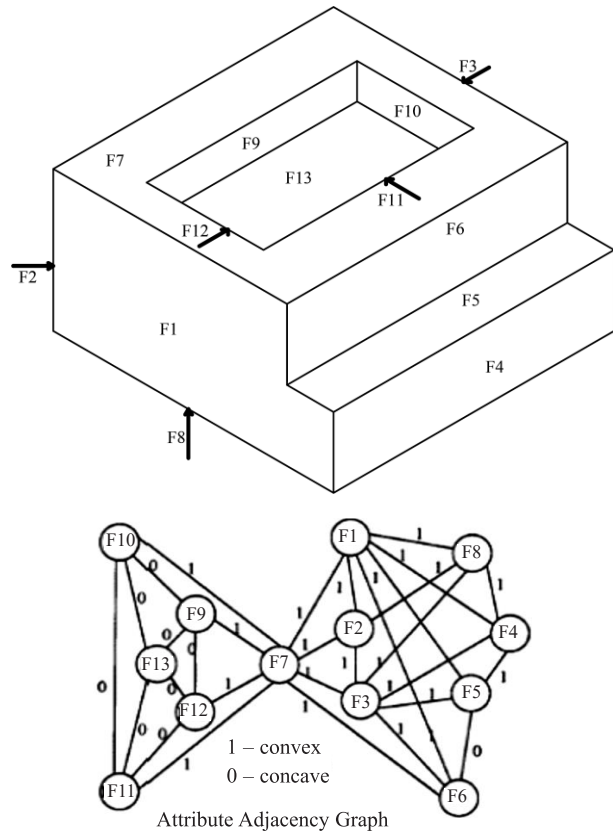


Fig. 8. Relationship Between Surfaces [12]

To obtain this information, each surface is given an image point that provides information on the surface position to be tested for its neighboring relationship with other surfaces. Fig. 9 provides an overview of the algorithms used in this method.

This positioning utilizes coordinate image information; the midpoint of this position determination marks the midpoint of the CAD model that has been created. Taking the extreme points from the  $x$ ,  $y$ , and  $z$  axes and then the midpoints for each axis, we get point 0 from the CAD model. After each surface demonstrates an image point, one surface is taken randomly, and its proximity to other surfaces is checked.

The way to check this is to use normal force information from the surface being checked and the surface of its neighbors. The rule governing whether a surface is a neighboring surface to another demonstrating a concave profile is as follows:

- the checked surface demonstrates a smaller coordinate value than the neighboring surface, according to the normal force of the checked surface;
- the neighboring surface demonstrates a smaller coordinate value than the surface that is checked, according to the normal force of the neighboring surface;



– this neighboring surface is a neighboring surface and forms a concave profile only when conditions a and b are met, such that if at least one of the two is not met, then the surface is convex or does not produce an angle of 90°.

Information from surfaces that demonstrate a neighboring concave relationship can be used to identify the model's

features. Surfaces that exhibit a neighboring relationship with other surfaces that are convex or produce an angle of 270° will be separated or eliminated; thus, only the concave surface can be identified in its features, according to the neighboring relationship. Thus, this FAG becomes an FFAG, in which a BEFG and an NEFG are present.

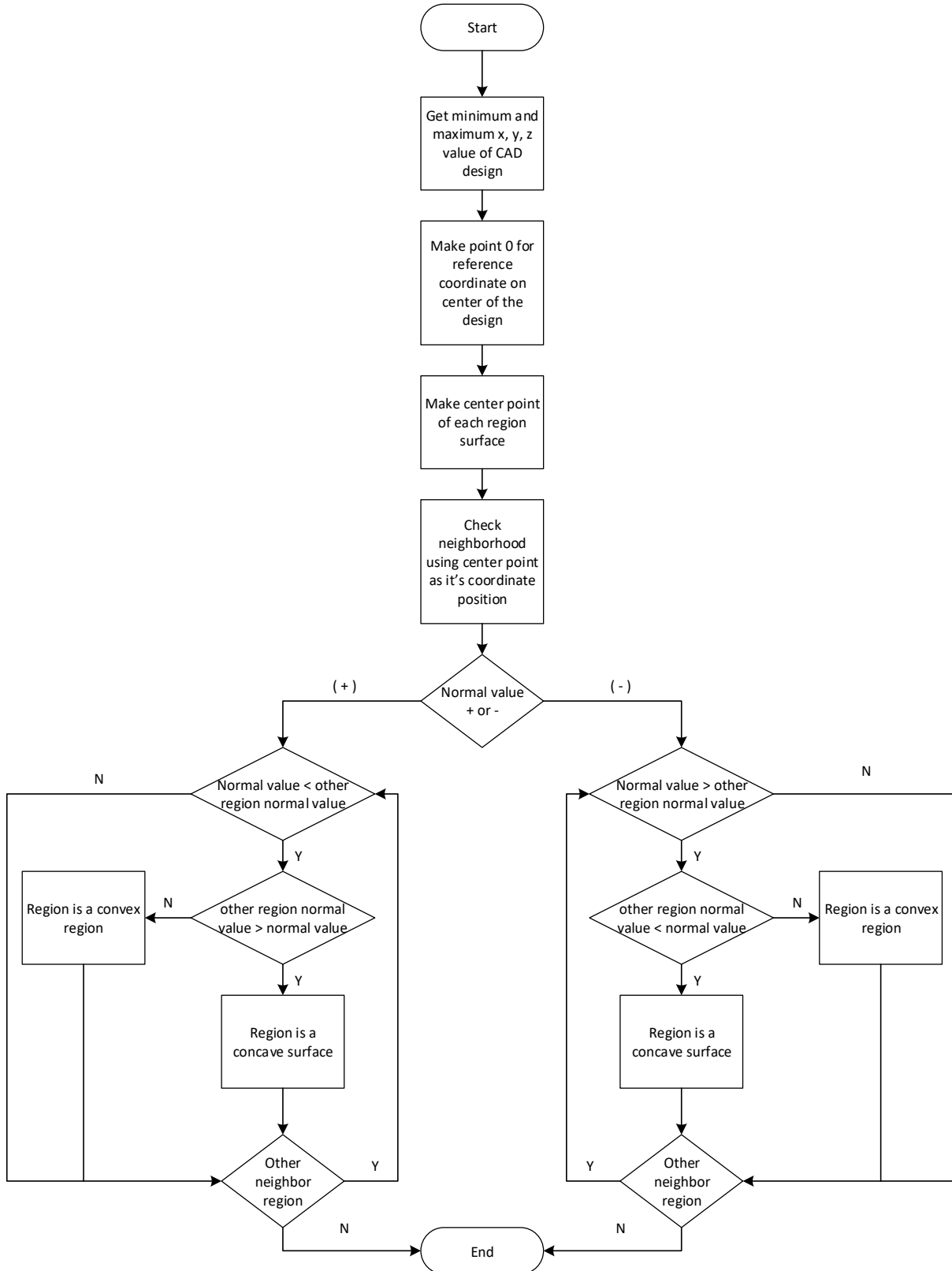


Fig. 9. Concave and Convex Algorithm

In this study, only BEFG is discussed, which is a feature graph that incorporates a data surface as the base. No NEFG is discussed here because this type of graph is a feature graph that exhibits a flat base surface or one with a curved shape as the surface of its base where curvature is not detected on this software.

By developing the rule-based concept that is simplified to be applied on each facet. From the FAG, we can obtain BEFG, for which specific rules can be given to determine the model's features. Among the features to be identified are the features within the prismatic and slab categories.

Following the feature classification made by [19], it appears that the prismatic feature incorporates two types of features, which are included in the prismatic feature category, namely, the stair feature and the slot feature. The BEFG for each feature is shown in Fig. 10. The rules for the stair feature rules are very simple, such that the checked surface has only one neighboring surface. In comparison, the slot rule exhibits a base surface that is crushed by the neighboring surface.

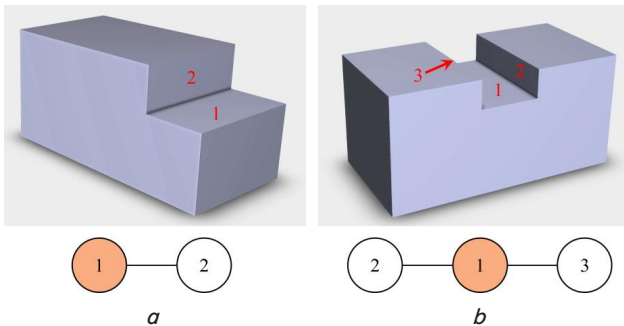


Fig. 10. Prismatic feature category: *a* – Graph Stair Feature; *b* – Graph Slot Feature

Three features are included in the slab feature category: a depression feature, a notch feature, and a pocket feature. The BEFG for each feature is shown in Fig. 11.

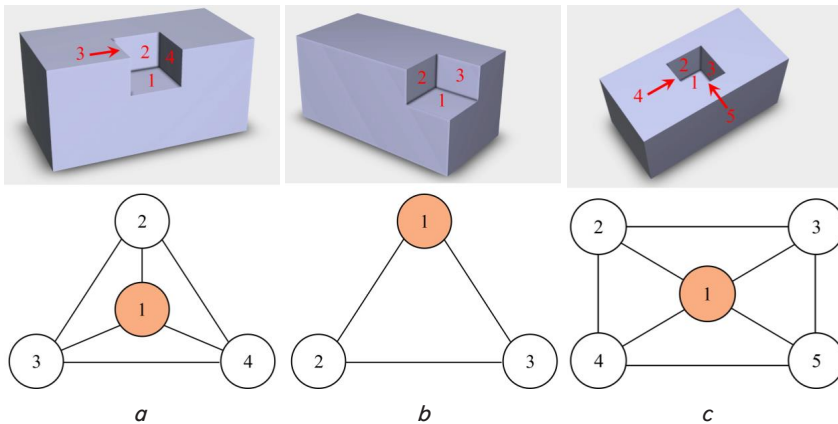


Fig. 11. Slab Feature: *a* – Graph Depression Feature; *b* – Graph Notch Feature; *c* – Graph Pocket Feature

The BEFG of the notch feature is triangular, such that the ground surface exhibits two neighboring surfaces, and these two neighbors are neighbors to each other. The BEFG shape of the depressed feature is similar to the BEFG shape of the notch feature. The difference here is that the base surface exhibits three neighboring surfaces where one of the neighbors also exhibits three neighboring sur-

faces, and the remainder exhibit two neighboring surfaces. The base surface exhibits four neighboring surfaces for the pocket feature, each of which exhibits three other neighboring surfaces.

#### 4. 2. Calculating the Product Complexity Index

To obtain the value for the product complexity index of a CAD model, additional information from the CAD models is needed. This information includes the following:

- 1) the type of features contained in the CAD model;
- 2) the form of the features;
- 3) the geometrical features;
- 4) tolerance.

All of this information is obtained from the FR process of the CAD model. The information obtained can then be weighted according to each aspect.

In this stage, the information obtained is the total amount of information drawn from the model according to its features ( $N$ ). If the number of features in the model is known, the  $N$  value can be obtained in a multiple of 15 from the number of features. Apart from knowing the number of features, knowing the type of each feature is also important. The weighting of the features' shape and dimensional values depends on the types of features in the model because the weighting calculations are carried out in each feature. This information is needed to obtain the amount of unique information ( $n$ ); the value of  $n$  can only be obtained after the following is known:

- 1) what types of features are present and how many features are the same;
- 2) the dimensions of each feature;
- 3) the position of the features.

The shape value of a model depends on the number of its features. The information needed to obtain the feature's shape value is rooted in how large the number of surfaces are present that form the feature and how many edges it exhibits. The weighting is based on the feature category. This feature includes prismatic and slab features.

For weighting on the prismatic feature, the slot feature can be used as a reference because it exhibits a higher number of faces and edges than a stair. In this study, the weighting reference for prismatic features uses the slot feature with a total of 4 features and their multiples, up to 16 features in one model. Fig. 12 provides an overview of the shape-weighting algorithm, with reference to four slots and four pockets.

For weighting on the slab feature, the pocket feature can be used as a reference because it exhibits a greater number of surfaces and edges than other slab features. In this study, the weighting reference for the slab feature uses a pocket feature with 4 features and their multiples, up to 16 features in one model.

The weighting of the feature geometry is drawn from the length ( $L$ ) and depth ( $D$ ) dimensions of each feature. The weighting is based on the given feature category. The categories of this feature are prismatic and slab features. Dimensional weighting is different from feature weighting, which uses the feature with the most information as a reference for weighting.

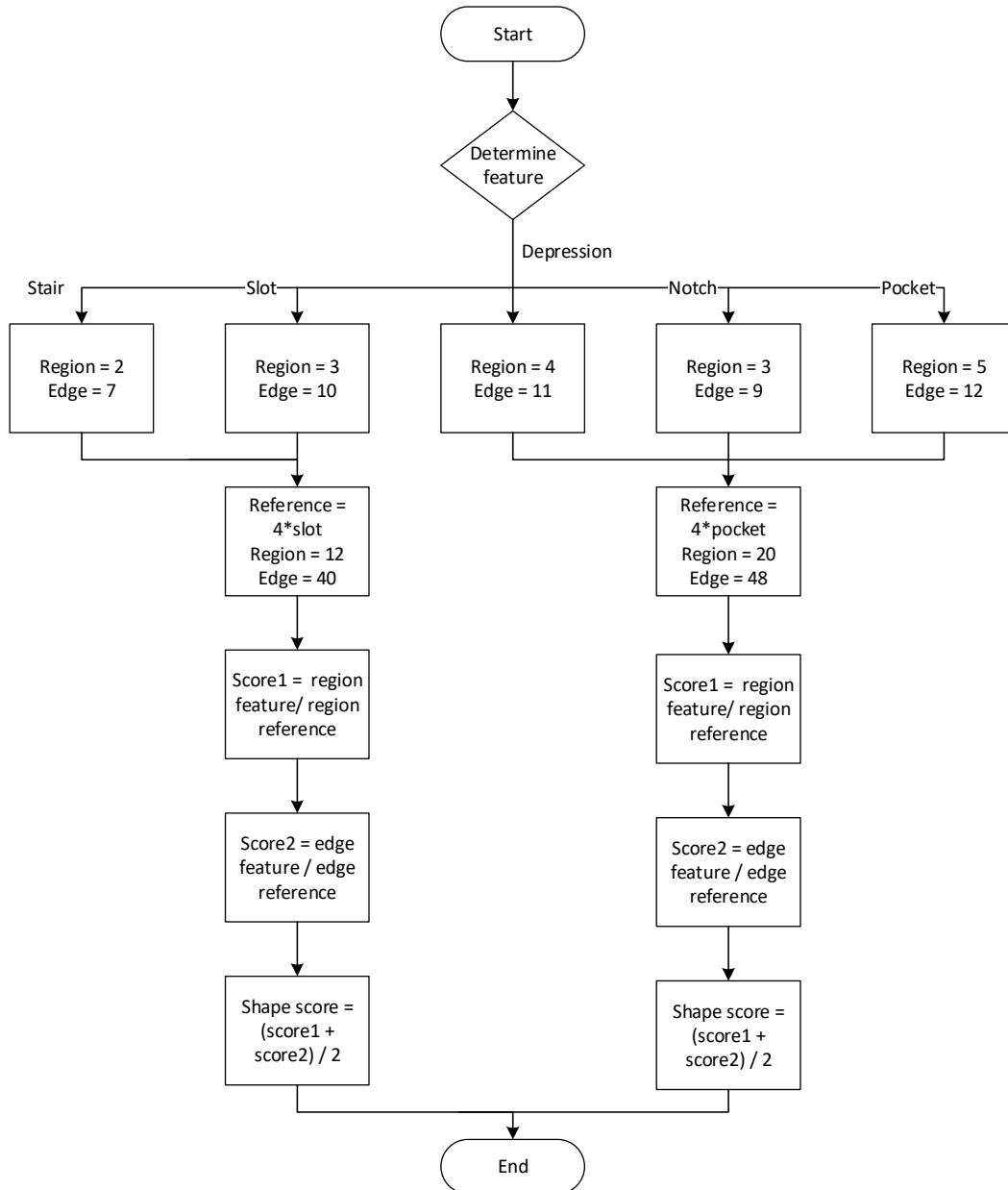


Fig. 12. Shape Score Algorithm

Each feature exhibits an algorithm for determining  $L$ , width, and  $D$ .

The algorithm used to determine the weighting for each feature dimension can be seen in Fig. 13. In the prismatic feature category, dimensional weighting is only performed if a model exhibits a simultaneous stair feature and slot feature. Thus, the highest-dimensional value is sought, and then this value is used as a reference for the highest value, namely 1; if the number is worth more than 1, then the value must be 1. This applies to the slab feature category; weighting with the reference is only performed if more than one type of feature is present in the model's category of slab features. If only one feature is present for each feature category, then the weighting does not require a reference, and if the dimensional value of the feature is more than 1, then it must be changed to 1.

The tolerance value is taken from Table 1, which indicates the allowable variations in relation to the level of accuracy and the dimensions of the features based on ISO 2768 [23].

The level of precision is the maximum weighted value for tolerance. The accuracy number is the smallest value from Table 1, so it must be changed to the highest one. To do this, the number must be raised to the power of  $-1$  ( $n^{-1}$ ), giving it a value like that in Table 2. To obtain the weighted value, the precise value of each size class can be used as a divisor for each of the allowed variation values, namely, the accurate, medium, and rough values. Thus, a weighted value is obtained in Table 3.

For now, the tolerance value used in this software is moderate for each class.

The method of calculating the values for the product complexity index uses (1)–(7). The information needed to calculate the feature complexity factor ( $F_{cf}$ ) is the values for feature shape, feature dimensions, and tolerance, divided by the number of aspects, namely three (6). After obtaining the  $F_{cf}$  value of the features, we investigate  $C_{f, feature}$  in terms of the number of features in the model (5). Up to this point, the calculations are carried out per feature type.



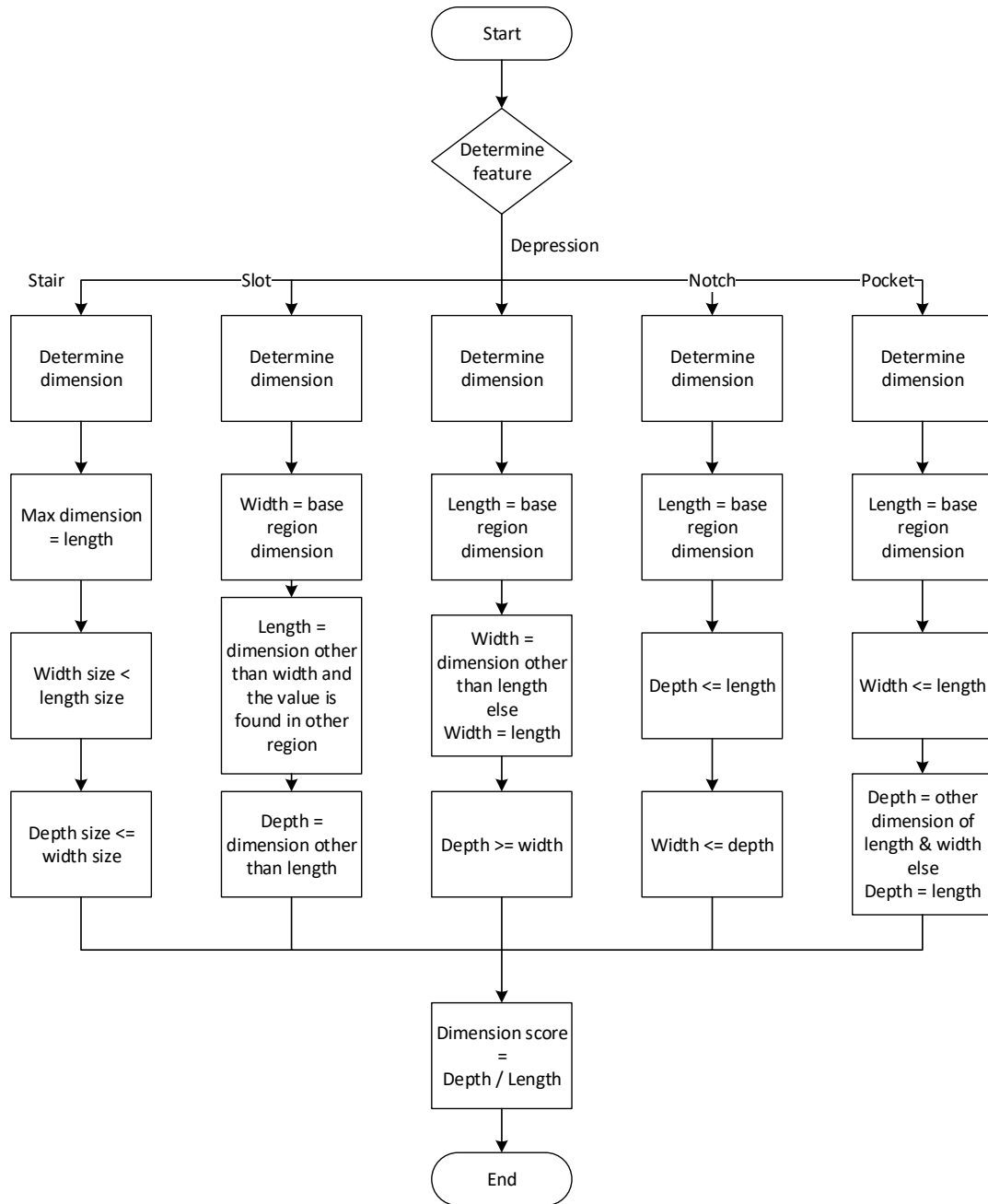


Fig. 13. Dimension Score Algorithm

Table 1

Permissible Variation

Nominal Size (mm)		0.5–3	3–6	6–30	30–120	120–315	315–1,000	1,000–2,000
Permissible variation	Precise	±0.05	±0.05	±0.1	±0.15	±0.2	±0.3	±0.5
	Medium	±0.1	±0.1	±0.2	±0.3	±0.5	±0.8	±1.2
	Rough	–	±0.2	±0.5	±0.8	±1.2	±2	±3

Table 2

Tolerance After the Power of  $-1 (n^{-1})$

Nominal Size (mm)		0.5–3	3–6	6–30	30–120	120–315	315–1,000	1,000–2,000
Permissible variation	Precise	20.00	20.00	10.00	6.67	5.00	3.33	2.00
	Medium	10.00	10.00	5.00	3.33	2.00	1.25	0.83
	Rough	–	5.00	2.00	1.25	0.83	0.50	0.33

Table 3

Weighting Tolerance

Nominal Size (mm)		0.5–3	3–6	6–30	30–120	120–315	315–1,000	1,000–2,000
Permissible variation	Precise	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	Medium	0.50	0.50	0.50	0.50	0.40	0.38	0.42
	Rough	–	0.25	0.20	0.19	0.17	0.15	0.17

After obtaining  $C_{f,feature}$ ,  $C_{j,product}$  can be obtained by adding up the values for all of the  $C_{f,feature}$  values for all types of features in the model, multiplied by the percentage of the number of features that appear in the model. The  $H_{product}$  and  $D_{R,product}$  values are obtained using the information on  $N$  and  $n$  obtained at the initial weighting stage. To obtain the  $H_{product}$  and  $D_{R,product}$  values, (1)–(2) are used. After obtaining the  $H_{product}$ ,  $D_{R,product}$ , and  $C_{j,product}$  values, the value for  $CI_{product}$  is obtained using Equation 3, which gives the sum of  $D_{R,product}$  and  $C_{j,product}$ , multiplied by  $H_{product}$ .

no final determination can be made because the information on the points is arranged arbitrarily and arranged by number coding in the STEP file. Therefore, the dimensions can be taken directly from the CAD software, although the dimensional information can be obtained from the STEP file. The information obtained is presented in Table 4.

Table 4

Information on Pocket Feature

Information	Number
Advance face	5
Edge curve	12
Dimension	Value (mm)
Length	20
Width	20
Depth	20

5. Results of application development

5.1. Validation

In this section, the calculation of the basic feature complexity index in the product's feature is carried out using the STEP method and calculations with the FRCI application.

5.1.1. Complexity Index Calculations Using the STEP Method

The calculation of the complexity index using the STEP method is conducted for one of the slab features, shown in Fig. 14. The calculations performed using this method are divided into three stages, namely (1) information gathering for weighting, (2) phase weighting, and (3) calculating the value of the product complexity index [10].

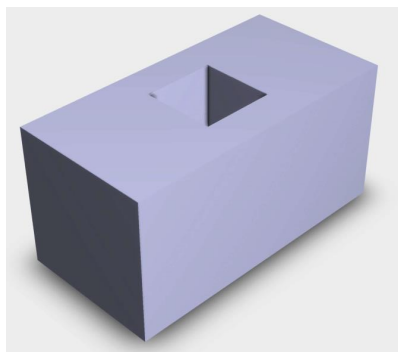


Fig. 14. Pocket Feature

The information required to obtain the product complexity index value is the model's surface, edge, and dimensions. This information can be obtained by opening the STEP file and observing the number of advance faces and edge curves in the model. For data retrieval from the STEP file, the number of advance faces to be calculated must be reduced by 6 advance faces, and the number of edge curves must be reduced by 12. This is because a cube exhibits 6 sides and 12 edges, a fact that is used to calculate the complexity value of a model using only its features.

The dimensions of the model can also be obtained from the coordinate point information on the edge curve. However,

Table 5 shows the information that is obtained in the initial stage of the calculation for the STEP method. The information obtained is used for the values of  $N$  (number) and  $n$  (diversity).

Table 5

Number ( $N$ ) and Diversity ( $n$ ) Values

Description	Number	Diversity
Length	1	1
Tolerance	2	2
Width	1	1
Tolerance	2	2
Depth	1	1
Tolerance	2	2
Feature	1	1
Tolerance	2	2
X values	1	1
Y values	1	1
Z values	1	1
Sum	15	15

The weighting for the form value, dimensional value, and tolerance value is performed after receiving the information obtained in the first step. Weighted-shape values are obtained using advance face and edge curve information from the calculated features and compared using references in relation to their feature categories. In this case, the comparison reference uses four pockets according to the reference for the slab feature. The weighting for the form values is given in Table 6.

Table 6

Weightings of the Form Values

Description	Pocket	4 pockets
Surface	5	20
Score	0.25	1
Edge	12	48
Score	0.25	1
Shape Score	0.25	1

The information used to calculate the weighted dimension value includes  $L$  and  $D$  of the features under study. For the weighted comparison of this feature's dimensions using dimensions from other features, other features are obtained within the same model. If no other features are present in the same category in the model, then the weighted value is  $D/L$ . The weighting of the dimensional value exhibits a maximum value of 1. If the calculation result exhibits a value of more than 1, then it must be adjusted to 1 because the maximum value for this weighting is 1. The weighting of the dimensional values of this model can be seen in Table 7.

Table 7

Weightings of the Dimensional Values

Description	Pocket
Length	20
Width	20
Depth	20
Score	1
Geometry Score	1

The value of tolerance as shown in Table 8 is taken to be at a moderate level, so the weighted value is 0.50. The next step is to find the values for  $F_{CF}$ ,  $C_{f,feature}$ ,  $C_{j,product}$ ,  $H_{product}$ ,  $D_{R,product}$ , and  $CI_{product}$ . The results for each component are given in Tables 9–11 using (1)–(7).

Table 11 shows that the value of the model's product complexity index with a pocket feature is 6.333.

Table 8

Weightings of the Tolerance Values

Nominal Size (mm)		6–30
Permissible Variation	Precise	1.00
	Moderate	0.50
	Rough	0.20

Table 9

Calculation of  $F_{CF}$

Description	Features $J=3$				Sum $D$	$D/J$
	Number	Diversity				
		Shape	Geometry	Tolerances		
Pocket	1	0.25	1	0.5	1.75	0.583
Sum	1					

Table 10

Calculation of  $C_{f,feature}$  and  $C_{j,product}$

Feature	% Feature	$D/J$	C.Feature	C.Relative
Pocket	1	0.58333	0.58333	0.58333
Sum	1			0.58333

Table 11

Calculation of  $CI_{product}$

Description	
$n$	15
$N$	15
$D$ . Product	1
$H$ . Product	4
$C$ . Index	6.333

### 5. 1. 2. Complexity Index Calculations Using a Hybrid Feature Recognition Method

The calculation of the product complexity index value in this section is done with FRCI software. This software is very simple and can provide complete information for calculating the product complexity index value of the computed STL file. FRCI software is also very easy to operate. The operating steps are as follows:

- press the Open STL button;
- select the STL file to calculate its complexity index value;
- FRCI directly calculates the complexity index value of its products;
- the dimensions and feature weighting can be viewed by pressing the detail button for each feature type;
- to view a 3D image from an STL file, press the render button.

An overview of the FRCI software can be seen in Fig. 15.

The calculation of the product complexity index using a hybrid FR method is achieved in the same way as the calculation using the STEP method. The difference here lies in the reference value for the weighting value for the pocket. In this method, the calculation used four reference-weighting values, namely, the pocket reference value at 4 pockets, 8 pockets, 12 pockets, and 16 pockets. Table 12 and Fig. 16 provide an overview of the results of the calculation of the product complexity index values using different reference values.

The data above indicate that the difference value of the product complexity index between each pocket reference at 4–8 pockets is 0.167, at 8–12 pockets, it is 0.055, and at 12–16 pockets, it is 0.028. The difference in the complexity index value at 4–8 pockets is the largest, and the differences in the value between other pockets are not significant. Thus, the value of the four reference pockets is less appropriate for use as a reference because it has a much higher delta value for the product complexity index than others. As a benchmark, it is more appropriate to use 8 pockets or 12 pockets, as these exhibit a much smaller delta for the value of the product complexity index between them.

The results of calculation validation are shown in Fig. 17 that the software performs a valid calculation for the value of the product complexity index on the pocket feature, with a product complexity index value of 6.333 that exhibits the same value as the calculation results obtained using the STEP method in Table 11.

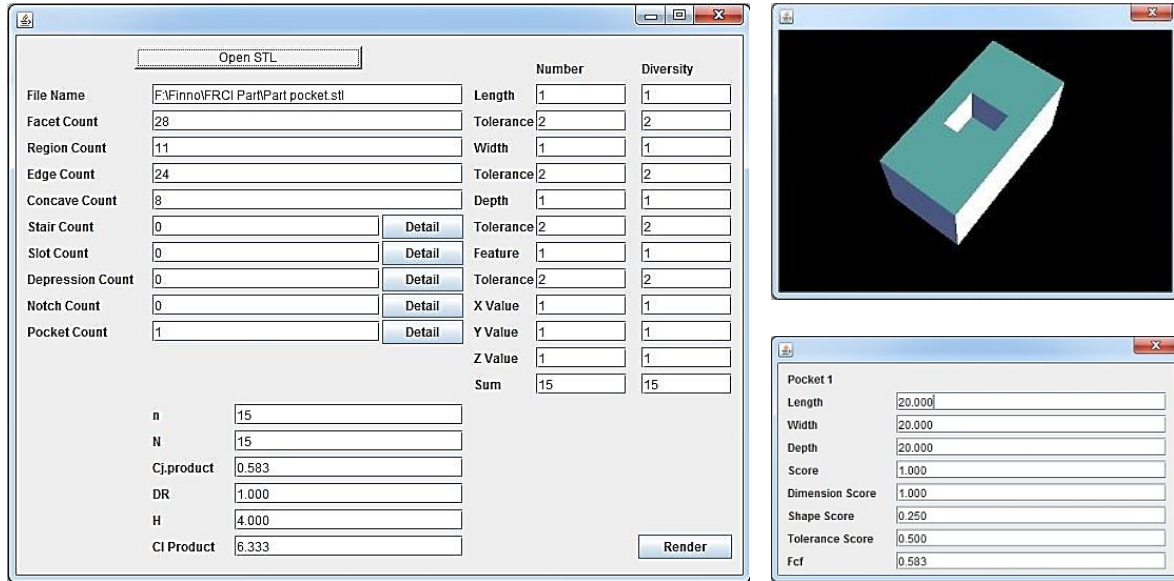


Fig. 15. FRCI Software Display

Table 12

Results for Each Reference Value

Reference	Product Complexity Index Values
4 Pockets	6.333
8 Pockets	6.166
12 Pockets	6.111
16 Pockets	6.083

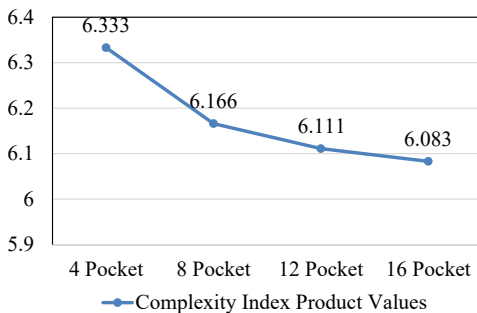


Fig. 16. Values of the Product Complexity Index for Each Reference

5. 2. Implementation of FRCI on various features

The model used for validation adopts a combined features model, as shown in Fig. 18.

FRCI software is used for this validation, with a reference of eight slots and eight pockets for the weighting value benchmark of shape features.

The two pockets as shown in Fig. 18 are detected by the software, and it is shown that they share a dimension.

Fig. 19 shows the calculated product complexity index of combined features from the STL file, and Fig. 20 shows the render of the STL model. In Fig. 19, it is shown that *N* (number) and *n* (diversity) exhibit different values. This is because certain types of features exhibit a number of more than 1 and exhibit the same dimensions.

Details on the dimensions can be seen in Fig. 21, 22.

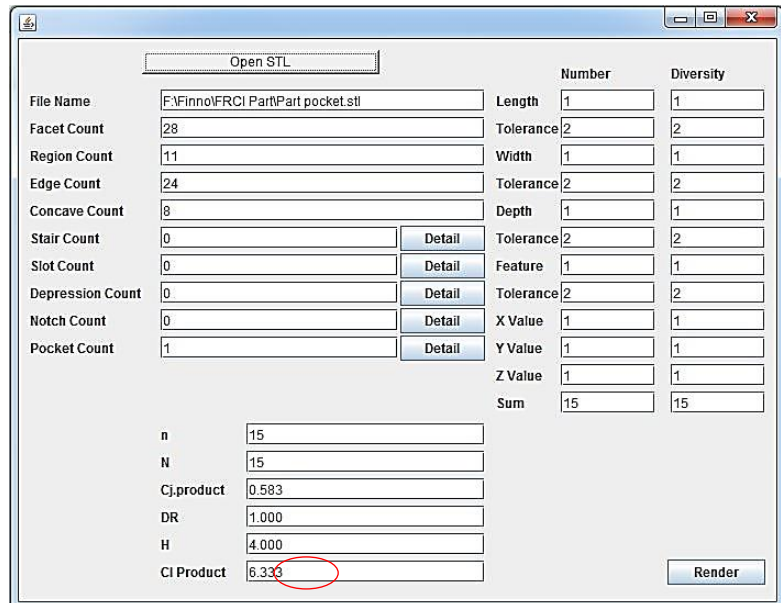


Fig. 17. Results of the Calculation Using the Four Pockets Reference

In Fig. 22, it can be seen that two pockets are detected on the model, each of which exhibits the same dimensions. This is what distinguishes *N* and *n*, and this will then affect  $D_{R,product}$  and  $H_{product}$ , so that the value of the product complexity index of the model with combined features becomes 8.081. Thus, the software can calculate and detect features of the model with combined features.

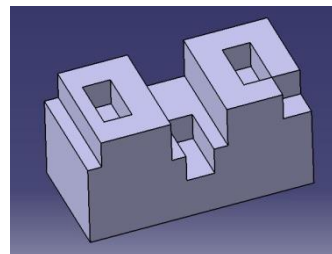


Fig. 18. Combined Feature

Open STL

	Number	Diversity
File Name	F:\Finno\FRCI Part\Part gabungan.stl	
Facet Count	94	Length 6
Region Count	29	Tolerance 12
Edge Count	74	Width 6
Concave Count	27	Tolerance 12
Stair Count	1	Depth 6
Slot Count	1	Tolerance 12
Depression Count	1	Feature 6
Notch Count	1	Tolerance 12
Pocket Count	2	X Value 6
		Y Value 6
		Z Value 6
		Sum 90
		Diversity 76

n	76
N	90
Cj.product	0.397
DR	0.844
H	6.508
CI Product	8.081

Render

Fig. 19. Calculation of Product Complexity Index of the Combined Feature Using FRCI Software for the Reference of Eight Slots and Eight Pockets

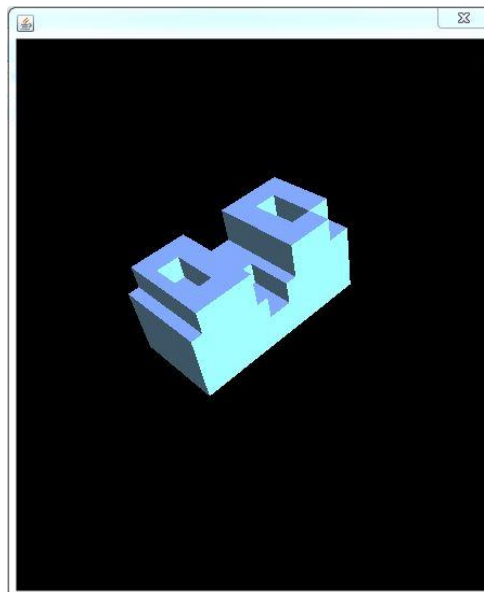


Fig. 20. Render of the Combined Feature STL Model

Stair 1	
Length	40.000
Width	5.000
Depth	10.000
Score	0.250
Dimension Score	0.500
Shape Score	0.085
Tolerance Score	0.500
Fcf	0.362

Slot 1	
Length	40.000
Width	15.000
Depth	20.000
Score	0.500
Dimension Score	1.000
Shape Score	0.125
Tolerance Score	0.500
Fcf	0.542

Fig. 21. Information and Calculation of  $C_{f \text{ feature}}$  in Prismatic Feature



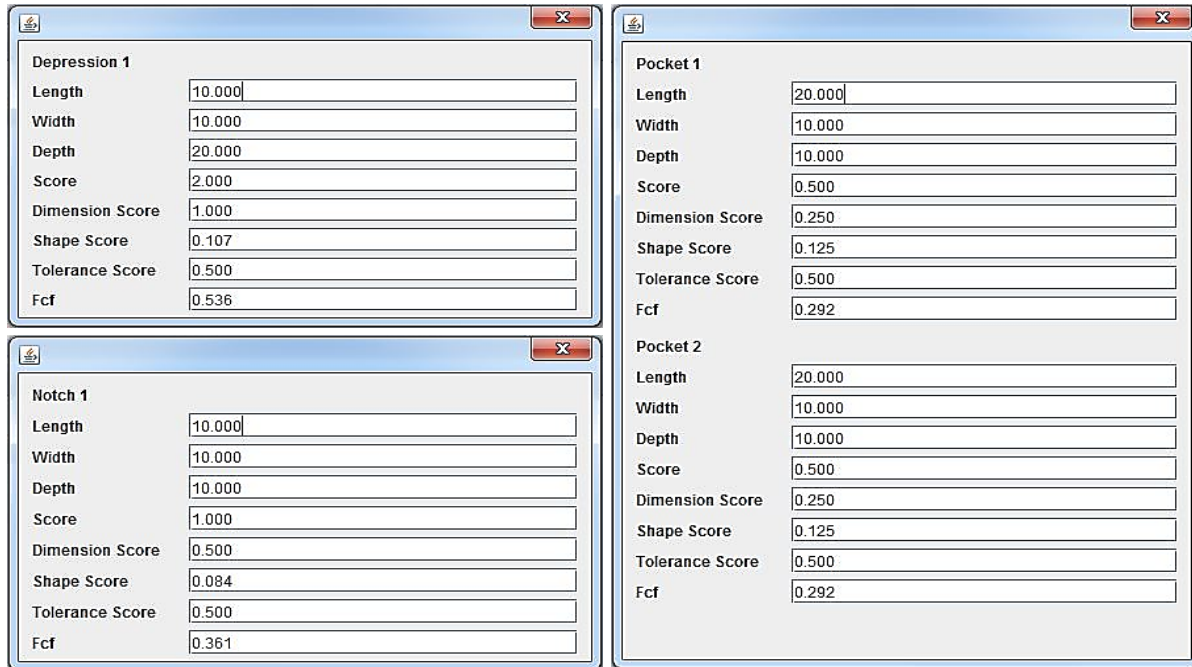


Fig. 22. Information and Calculation of  $C_{f\ feature}$  in Slab Feature

### 6. Discussion of the research results of application development

In section 5. 1, it is proven that the FRCI application is valid for calculating the complexity index of the basic features, which in this research was tested on the pocket feature. The results show that FRCI is proven to be valid as calculated using the STEP method. From section 5. 2, we tried to calculate the complexity index for a design that has various features. The result of this implementation is that FRCI has the ability to correctly calculate the complexity index of the existing designs without any errors.

This method is applicable to CAD applications and can give developers the flexibility to develop CAD products. It can be used as a plug-in or add-in to a CAD application. As a plug-in, it is very useful for designers; it will respond to CAD programs when used. When a user creates a model running this plug-in, the model created by this user is provided with a complexity index value. When the user changes the model, the complexity index of the created model also changes.

This idea exhibits promise for the manufacturing world; if the product complexity index value is known as early as possible, the estimated production cost can be known sooner. Design is the most likely place for reductions in production cost; therefore, knowing the value for product complexity early is important in the design stage.

With the existing limitations, this study is focused only on discussing the efforts to create automatic feature recognition software using STL files. This method is to get information from the CAD model that has been made.

Therefore, the considerations in making this software pay attention to:

- 1) the software can only detect prismatic features and slab features;
- 2) the software can only read models with a flat surface, the curved surface cannot be read by this software;

3) the software can only read STL files from CATIA CAD programs;

4) the software does not calculate the value for the complexity of the material;

5) the software can only identify features with a 90° angle.

The first disadvantage experienced during this research is related to programming problems, what platforms are available and which ones are most often used so that 3rd party applications can be integrated into add-on applications for CAD applications that are often used in the industry.

The second disadvantage is the curved surface. In this research, the shape must be 90° (not free degree) to recognize surface 1 and other surfaces. To recognize a surface, the base rule is from grouping triangles with the same normal directional force. That is why curved surface is quite difficult because the normal force direction from triangulation for 1 curved surface is different. This can be overcome by developing a rule if the difference is on a certain degree from the other triangles, the group of triangles are still considered the same surface.

For future research, the development of an application that can detect the complexity of curved surfaces is highly recommended considering that the surface of industrial products is increasingly varied.

### 7. Conclusions

1. The validation results show that the complexity index value generated by the FRCI application has the same value as the STEP method. This study indicates that the FRCI application can calculate the value of the product complexity index based on the features that exist in the product model.

2. The software can detect and identify features existing in the design, namely, the slab feature and the prismatic feature. The shape of a feature reflects the level of complexity and

machining time that will be used to create the feature. The more complex a feature, the longer the machining time. The longer the machining time, the higher the cost. Therefore, the complexity index can be associated with cost.

---

### Acknowledgments

---

This research is supported by PPI Q2 2021 Universitas Indonesia.

---

### References

1. Roy, R., Souchoroukov, P., Shehab, E. (2011). Detailed cost estimating in the automotive industry: Data and information requirements. *International Journal of Production Economics*, 133 (2), 694–707. doi: <https://doi.org/10.1016/j.ijpe.2011.05.018>
2. Farineau, T., Rabenasolo, B., Castelain, J. M., Meyer, Y., Duverlie, P. (2001). Use of Parametric Models in an Economic Evaluation Step During the Design Phase. *The International Journal of Advanced Manufacturing Technology*, 17 (2), 79–86. doi: <https://doi.org/10.1007/s001700170195>
3. Shehab, E. M., Abdalla, H. S. (2001). Manufacturing cost modelling for concurrent product development. *Robotics and Computer-Integrated Manufacturing*, 17 (4), 341–353. doi: [https://doi.org/10.1016/s0736-5845\(01\)00009-6](https://doi.org/10.1016/s0736-5845(01)00009-6)
4. Wasim, A., Shehab, E., Abdalla, H., Al-Ashaab, A., Sulowski, R., Alam, R. (2012). An innovative cost modelling system to support lean product and process development. *The International Journal of Advanced Manufacturing Technology*, 65 (1-4), 165–181. doi: <https://doi.org/10.1007/s00170-012-4158-4>
5. Budiono, H. D. S., Kiswanto, G., Soemardi, T. P. (2014). Method and Model Development for Manufacturing Cost Estimation during the Early Design Phase Related to the Complexity of the Machining Processes. *International Journal of Technology*, 5 (2), 183. doi: <https://doi.org/10.14716/ijtech.v5i2.402>
6. Fischer, P., Heingärtner, J., Duncan, S., Hora, P. (2020). On part-to-part feedback optimal control in deep drawing. *Journal of Manufacturing Processes*, 50, 403–411. doi: <https://doi.org/10.1016/j.jmapro.2019.10.019>
7. Mali, R. A., Gupta, T. V. K., Ramkumar, J. (2021). A comprehensive review of free-form surface milling– Advances over a decade. *Journal of Manufacturing Processes*, 62, 132–167. doi: <https://doi.org/10.1016/j.jmapro.2020.12.014>
8. Boothroyd, G., Dewhurst, P., Knight, W. A. (2010). *Product Design for Manufacture and Assembly*. CRC Press, 712. doi: <https://doi.org/10.1201/9781420089288>
9. Limon-Leyva, P. A., Balvántin, A. J., Diosdado-De-la-Peña, J. A., Figueroa-Díaz, R. A., Rojas-Mancera, E., Ramirez, V. A. (2020). Parametric optimization of roll-hemming process in oblique planes with linear and non-linear trajectories. *Journal of Manufacturing Processes*, 50, 123–131. doi: <https://doi.org/10.1016/j.jmapro.2019.12.019>
10. Budiono, H. D. S., Sholeh, M., Kiswanto, G., Soemardi, T. P. (2014). Application of Semi Automatic Model of Product Complexity Index Calculation by Identification and Recognition of Geometric Features Information. *Applied Mechanics and Materials*, 493, 576–582. doi: <https://doi.org/10.4028/www.scientific.net/amm.493.576>
11. Kresnha, P. (2010). Feature Recognition Model Facet 3D using Fuzzy Geometry Based on Drive Point on Multiaxis CAM System. Universitas Indonesia.
12. Shah, J. J., Anderson, D., Kim, Y. S., Joshi, S. (2000). A Discourse on Geometric Feature Recognition From CAD Models. *Journal of Computing and Information Science in Engineering*, 1 (1), 41–51. doi: <https://doi.org/10.1115/1.1345522>
13. Abouel Nasr, E. S., Kamrani, A. K. (2006). A new methodology for extracting manufacturing features from CAD system. *Computers & Industrial Engineering*, 51 (3), 389–415. doi: <https://doi.org/10.1016/j.cie.2006.08.004>
14. Jones, T. J., Reidsema, C., Smith, A. (2006). Automated Feature Recognition System for supporting conceptual engineering design. *International Journal of Knowledge-Based and Intelligent Engineering Systems*, 10 (6), 477–492. doi: <https://doi.org/10.3233/kes-2006-10606>
15. Shahin, T. M. M. (2008). Feature-Based Design – An Overview. *Computer-Aided Design and Applications*, 5 (5), 639–653. doi: <https://doi.org/10.3722/cadaps.2008.639-653>
16. Salomons, O. W., van Slooten, F., Jonker, H. G., van Houten, F. J. A. M., Kals, H. J. J. (1995). Interactive feature definition. *IFIP Advances in Information and Communication Technology*, 144–160. doi: [https://doi.org/10.1007/978-0-387-34834-6\\_8](https://doi.org/10.1007/978-0-387-34834-6_8)
17. Sreevalsan, P. C., Shah, J. J. (1992). Unification of form feature definition methods. *Proceedings of the IFIP WG 5.2 Working Conference on Intelligent Computer Aided Design*, 83–106.
18. Pratt, M. J., Wilson, P. R. (1985). Requirements for support of form features in a solid modelling system. Final Report, CAM-I Report R-85-ASPP-01.
19. Jung, J.-Y. (2002). Manufacturing cost estimation for machined parts based on manufacturing features. *Journal of Intelligent Manufacturing*, 13, 227–238. doi: <https://doi.org/10.1023/A:1016092808320>
20. Sunil, V. B., Agarwal, R., Pande, S. S. (2010). An approach to recognize interacting features from B-Rep CAD models of prismatic machined parts using a hybrid (graph and rule based) technique. *Computers in Industry*, 61 (7), 686–701. doi: <https://doi.org/10.1016/j.compind.2010.03.011>
21. ElMaraghy, W. H., Urbanic, R. J. (2003). Modelling of Manufacturing Systems Complexity. *CIRP Annals*, 52 (1), 363–366. doi: [https://doi.org/10.1016/s0007-8506\(07\)60602-7](https://doi.org/10.1016/s0007-8506(07)60602-7)
22. Sunil, V. B., Pande, S. S. (2008). Automatic recognition of features from freeform surface CAD models. *Computer-Aided Design*, 40 (4), 502–517. doi: <https://doi.org/10.1016/j.cad.2008.01.006>
23. ISO 2768. General Geometrical Tolerances and Technical Drawings. Available at: <https://www.plianced.com/compliance-wiki/iso-2768-general-geometrical-tolerances-and-technical-drawings/>