Available at https://journalenrichment.com/index.php/jr/

Enrichment: Journal of Multidisciplinary Research and Development

Decision Support System For Determining Food Menu Using Analytical Hierarchy Process (Ahp) Method

Ninda Maulina[1*], Wina Witanti[2], Agus Komarudin[3]

[1, 2, 3] Universitas Jenderal Achmad Yani, Indonesia

*Email: nindamaulina@gmail.com

ARTICLE INFO

Keywords:
decision support systems, food menu, Analytical Hierarchy Processes, decision makers

ABSTRACT

Food menus in the culinary and restaurant industries often involve various complex features, such as food origin, food category, diet category, ingredients, and time. This study aims to develop a Decision Support System (DSS) based on the AHP method to assist chefs, restaurant managers and food business owners in compiling a diverse menu, meeting nutritional needs, taking into account certain preferences and limitations, and creating a pleasant dining experience. The Analytic Hierarchy Process (AHP) method can be used as a tool in making more effective and structured decisions. The results of this study indicate that the Analytic Hierarchy Process (AHP) method succeeded in producing relative weights for each criterion and sub-criteria, thus enabling priority in preparing food menus. In testing, this system is able to provide the best recommendations based on global priority values for certain types of food, which are expected to increase the variety and quality of food menus and meet consumer preferences and needs. Through experiments conducted using a Decision Support System, a decision model is formed that determines the priority for the weight of all criteria and alternatives. The results show preferences in the process of determining food menus by producing Cold Coffee with a value of 0.30 (29%), Biscuit Dough Donuts with a value of 0.25 (25%), Chicken Dimsum with a value of 0.21 (21%), Succotash with a a value of 0.15 (15%), and Toffee Banana with a value of 0.10 (10%).

INTRODUCTION

The importance of food for the life of every individual is undeniable, as food provides various essential nutrients such as fats, proteins, carbohydrates, minerals, vitamins, and so on. In addition, there are also substances added to food, either intentionally or unintentionally, that can affect the quality of such food (Soekarta & Modok, 2018). The food menu acts as a guide for individuals in choosing the type of food to be consumed. The meal menu may include a variety of dishes offered, including appetizers, main courses, side dishes, snacks and desserts. The meal menu can be tailored to suit various preferences, such as the type of meal (e.g. Asian or European food), special dietary needs (e.g., vegetarian, non-vegetarian, diabetic friendly, etc.), and availability of ingredients. Food menus can be found in various places, such as restaurants, cafes, canteens, hotels, or even in the context of family at home.

With the increasing number of modifications to food menu dishes today, now the search for food menus is easier with various modifications and variations of dishes that can be accessed via the Internet (Salsabella, 2014). However, the difficulty of choosing a menu, especially outside food menus in the midst of a variety of food choices, makes it difficult for food processors to make the right choice. Especially for those who are new to certain types of food, determining the menu is a challenge. The process of searching for menus on digital media is currently still done manually by comparing the available menus with needs. (Meilani & Wardana, 2020) Therefore, a system is needed that can produce food menus based on ingredients, food menu categories, food origin categories (Asian and European), manufacturing time, and lifestyle / diet type (vegetarian, non-vegetarian, eggetarian) according to user needs. The Analytical Hierarchy Process (AHP) method is an effective solution in the development of

decision support systems, because this method can provide rankings based on global weights to help make accurate and appropriate decisions. (Valentino et al., 2021) The decision support system in determining the food menu has been carried out in previous studies, one of which is determining the menu of a restaurant requires several things, such as market needs (food trends), food ingredients, food prices, food attractiveness and preparation process time. With the existence of a restaurant menu selection decision support system, it can help restaurant owners choose the menu that will be used by the restaurant quickly and precisely (Ismanto et al., 2020). With the decision support system for choosing food menu recipes with the Analytical Hierarchy Process (AHP) method, users are helped in selecting menu recipes and are expected to achieve good results (Tanjung, 2017).

From several previous studies or research, it is known that many decision support systems have been implemented for determining food menus in various methods. This study focuses on how the Analytical Hierarchy Process (AHP) method can be implemented in facilitating the decision-making process of determining food menus where there is relative weight on each different criterion, and identifying appropriate food menu alternatives. The purpose of this study is to implement AHP to create a decision support system that can facilitate the determination of food menus by giving relative weight to each different criterion as well as identifying appropriate alternative menus.

In this study, there are problem limits so that they can focus on the problem so that the preparation is more directed and the results are easy to understand. The limitation is that this system is focused on determining the menu of food related to typical cuisines from the Asian and European regions. The criteria used to evaluate the food menu are limited to those related to cooking time, type of food (appetizer, main course, side dish, dessert). User preferences include dietary preferences, or other special dietary needs, food origin (Asian and European), and also foodstuffs. The data obtained is English, because the food listed is an external food. This system does not include how to make the selected food.

The output obtained from this study is a system of determining food menus based on specific criteria. This system will generate the weight of each criterion and rank food alternatives that fit those criteria. The benefit of this research is to produce appropriate food menu alternatives so that they can determine the menu based on the availability of food ingredients, and preference needs to increase efficiency in the food menu selection process. So that managers / chefs can design menus that are more varied and according to preferences. In addition, there are contributions to the development of science and technology, especially in the field of decision support systems and Analytical Hierarchy Process (AHP) methods. A decision support system can assist the manager in selecting alternative food menus based on certain predetermined criteria. In addition, this research can be a contribution in the application of the Analytical Hierarchy Process (AHP) method in the food sector by determining the relative weight of relevant criteria, and choosing the most suitable food menu alternatives based on these relative weights.

METHOD

In the calculation process of the Analytical Hierarcy Process method, there are several stages, which include the stages of calculating the pairwise comparison matrix, priority weight matrix, and matrix consistency. This system flowchart shows the flow of decision making in determining the recommended food menu based on predetermined criteria and subcriteria. The following flowchart of the design of this system can be seen in Figure 1.

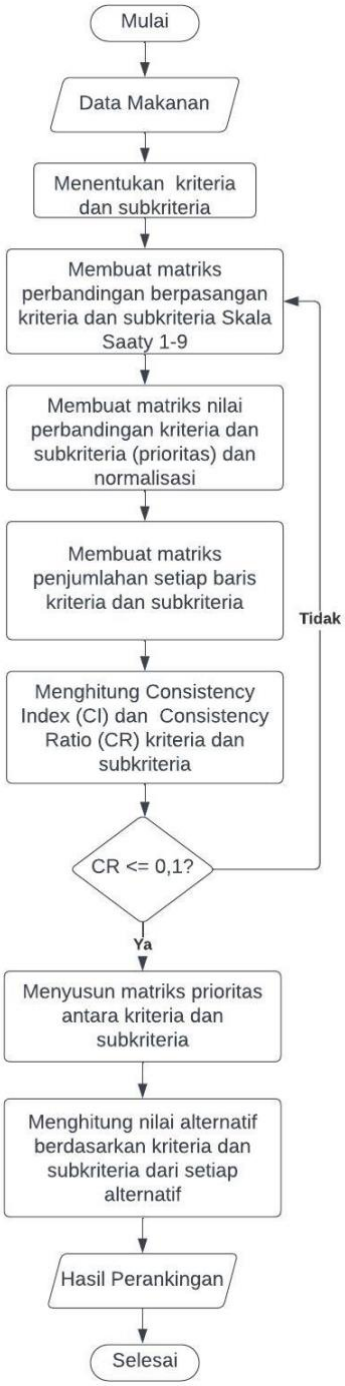


Figure 1. Flowchart System Planning

In designing a system based on the AHP method for decision making is identifying alternatives to be evaluated (Parameswari et al., 2022) (Siregar & Rahayu, 2018) (Prawira & Amin, 2022). For example, an alternative can be a list of menus to choose from. There are 6 (six) alternative food menus used such as Toffe Banana, Cold Coffee, Biscuit Dough Donut, Succotash, Chicken Dimsum. Method calculations in this system include run_method functions that are part of a program or application that may function to run certain analysis methods with given criteria parameters as input arguments. Source Code for run_method process function can be seen below:

```
     function run_method($kriteria){

        $show       = $this->show;

                    $kategori         =    $this->data_model->get_data_method('kategori',
['kriteria'=>$kriteria]);

                     $alternatif    =    $this->data_model->get_data_method('alternatif',
['kriteria'=>$kriteria]);

        #fase kategori

        if ($show) {

            echo "Data Kategori<br>";

            $this->show_table($kategori['data'], $kategori['tot_col']);

        }

        $normalisasi= $this->normalisasi($kategori['data'], $kategori['tot_col']);

        if ($show) {

            echo "Normalisasi & Bobot Prioritas<br>";

            $this->show_table($normalisasi['hasil'], [], $normalisasi['max']);

        }

         $konsisten  = $this->konsisten_matrik($normalisasi['hasil'], $kategori['data'],
$normalisasi['max']);



        #fase alternatif

        if ($show) {

            echo "<br><br>Data Alternatif<br>";

            foreach ($alternatif['data'] as $key => $value) {

                echo "Data ".ucwords($key)."<br>";

                $this->show_table($value, $alternatif['tot_col'][$key]);

            }

        }

        $normalisasi_alternatif = array();

        foreach ($alternatif['data'] as $key => $value) {

                        $normalisasi_alternatif[$key]      =   $this->normalisasi($value,
$alternatif['tot_col'][$key]);

        }

        if($show){

            foreach ($alternatif['data'] as $key => $value) {

                echo "Normalisasi & Bobot Prioritas<br>";

                           $this->show table($normalisasi alternatif[$key]['hasil'],  [],
$normalisasi_alternatif[$key]['max']);

            }

        }

        $perangkingan = $this->perangkingan($normalisasi, $normalisasi_alternatif);



        return ['perangkingan' => $perangkingan, 'konsisten' => $konsisten];

     }
```

To determine the consistency of the comparison matrix, multiplication is performed between each element in the comparison matrix column A with the priority weight of criterion A, the element in the comparison matrix column B with the priority weight of criterion B, and so on. Next, the multiplication result of each row is summed and divided by the sum of those rows with the corresponding priority weights, as seen in Table 3.4.

Table 1. Matrix Consistency Calculation

Criterion	Origin	Menu Categories	Healthy Category	Time	Number of Ingredients	Sum	CM
Origin	0,14	0,10	0,13	0,10	0,30	0,772	5,517
Menu Categories	0,28	0,21	0,19	0,16	0,30	1,146	5,458
Healthy Category	0,07	0,07	0,06	0,10	0,03	0,335	5,186
Time	0,70	0,63	0,32	0,49	0,50	2,639	5,438
Number of Ingredients	0,05	0,07	0,19	0,10	0,10	0,508	5,066
			Sum				26,664

A konsisten_matrik function is a part of a program or application that may be used to calculate the consistency of a normalized matrix. The source code for the matrix-consistent process function can be seen below:

```
function konsisten_matrik($data_normal, $data_asli, $max){

        $bp = array(); #bp = bobot prioritas

        $ci = 0; #konsisten index

        $no = 1;

        #ambil data bobot prioritas

        foreach ($data_normal as $key => $value) {

            $bp[$no++] = $value[count($value)-1];

        }

        #tambah konsisten matrik per baris

        $no = 1;

        foreach ($data_normal as $key => $value) {

            $km = 0;

            for ($i=1; $i < count($data_asli[$key]); $i++) {

                $km += number_format($data_asli[$key][$i] * $bp[$i], $this->nf);

            }

            $data_normal[$key][] = number_format($km/$bp[$no], $this->nf);

            $ci += number_format($km/$bp[$no], $this->nf);

            $no++;

        }

        $kolom_kategori = $no-1;

          $ci = $ci/($kolom_kategori == 0 ? 1 : $kolom_kategori); #ambil nilai rata-rata
konsistensi metrik

        $ci = number_format(($ci - $kolom_kategori) / ($kolom_kategori - 1), $this->nf); #rumus
ci

        $cr = number_format($ci / ($this->saaty[$kolom_kategori == 0 ? 1 : $kolom_kategori] ==
0 ? 1 : $this->saaty[$kolom_kategori == 0 ? 1 : $kolom_kategori]), $this->nf); #rumus cr

        $konsisten = $cr >= 0 && $cr <= 0.1 ? 'konsisten' : 'tidak konsisten';
```

The average value of this calculation results describes the maximum principal eigenvalue (λmax). Next, look for the CI (Consistency Index) obtained by Formula (1).

Sum of result values = 29,901

n (number of criteria) = 5

$\lambda\ maks$ = $(\frac{26,664}{5})$

= 5,333

CI = $\frac{\lambda\ maks-n}{n-1}$ …….. (1)

CI = $\frac{5,333-5}{5-1}$

= $\frac{0,333}{4}$

=0,083

Calculate the Consistency Ratio with the formula:

CR = $\frac{CI}{IR}$ …….. (2)

CR = $\frac{0,083}{1,12}$

CR = 0,074

The Random Consistency Index is used to compare the consistency of a comparison matrix provided by a user with a random comparison matrix. This helps in assessing how consistent and reliable comparisons are.

Table 2. Random Consistensy Index

N	1	2	3	4	5	6	7	8	9	10
IR	0.00	0.00	0.58	1.90	1.12	1.24	1.32	1.41	1.45	1.49

From the calculation of CI and RI, a CR value of 0.074 was obtained. In the AHP method, if the CR value is less than 0.1, it indicates that the comparison matrix is considered consistent. In this case, since the CR value is 0.074 which is less than 0.1, it can be concluded that the comparison given for the criterion is already consistent.

RESULTS AND DISCUSSION

Testing is carried out on a decision support system for determining food menus. Testing of this system is carried out using black box testing, there is a testing technique known as equivalence partitioning, where the test divides several types of data from the test case that is executed and then executes the existing modules or units.

Black box testing on decision support systems (DSS) with the AHP (Analytic Hierarchy Process) method is a type of testing where testers focus on testing the functionality and features of the application based on external specifications without the need to know or understand the details of the internal implementation of the system. In the context of AHP, black box testing will focus on testing the final result and interaction with the user without having to know or pay attention to how the comparison matrix is actually calculated or how weight is given to alternatives and criteria. The examiner simply examines the inputs provided, the decision-making process, and the outputs produced. The testing stages of Black Box Testing software include several activities consisting of determining test objectives, determining test result categories, test scenarios, test implementation, evaluation of test results.

The purpose of testing is to ensure that the functions that have been created have been carried out properly and correctly, besides that if there are functions that are not suitable can be immediately corrected and returned according to user needs. Testing is done to make it easier for system developers to fix existing errors.

Table 3. Purpose of Testing

No	Use Case	Purpose
1	Add criteria data	Test the software's ability to add new criteria data.
2	Change criteria data	Test the software's ability to change criteria data that has been stored in the database.
3	Delete criteria data	Test the software's ability to delete category data stored in the database.
4	Scale category data	Test the software's ability to input category data values for comparison calculations.
5	Add category data	Test the software's ability to add new category data.
6	Change category data	Test the software's ability to change category data that has been stored in the database.
7	Clear category data	Test the software's ability to delete category data stored in the database.
8	Alternate data scales	Test the software's ability to input alternative data values for comparison calculations.
9	Add alternate data	Test the software's ability to add new alternative data.
10	Change alternate data	Test the software's ability to modify alternative data that has been stored in the database.
11	Clear alternate data	Test the software's ability to delete alternative data stored in the database.
12	Add assessment data	Test the software's ability to add new assessment data.
13	Change assessment data	Test the software's ability to modify assessment data that has been stored in the database.
14	Delete assessment data	Test the software's ability to delete assessment data stored in the database.
15	Method	Test the software's ability to compare processes, and see ranking results

The category of quality testing of the system created is divided into two categories, namely appropriate, it is said to be appropriate if the system tested for quality and function in accordance with planning objectives. Not Conforming, it is said to be non-conforming if the system tested for quality and functionality is not in accordance with planning purposes. Test scenarios on criteria data can be seen in Table 4.

Table 4. Criteria Data Testing Scenarios

Function Name	Feature Name	Test Code	Test Cases
Criteria Data	Add Criteria Data	KU01	• User performs "Add Data" action • Validate data input on form • Selecting the "Add" button • Save data into a database View successfully saved data
	Change Criteria Data	KU02	• The user performs the "change" action by pressing "Change" • Make changes to the data and select the save button • Data successfully converted into database • View successfully changed data
	Clear Criteria Data	KU03	• The user performs the "Delete" action by pressing the delete button • The message "sure to delete this?" appears • Press the "ok" button when the correct data is deleted • Data successfully erased

Test scenarios on alternative data can be seen in Table 4.

Table 5. Alternative Data Testing Scenarios

Function Name	Feature Name	Test Code	Test Cases
Alternative Data	Alternative Data Scale	KU08	• User performs "Input Data" action • Validate data input on columns • Selecting the "Save" button

Function Name	Feature Name	Test Code	Test Cases
			• Save data into a database • View successfully saved data
	Add Alternate Data	KU09	• User performs "Add Data" action • Validate data input on form • Selecting the "Add" button • Save data into a database View successfully saved data
	Change Category Data	KU10	• The user performs the "change" action by pressing "Change" • Make changes to the data and select the save button • Data successfully converted into database • View successfully changed data
	Clear Category Data	KU11	• The user performs the "Delete" action by pressing the delete button • The message "sure to delete this?" appears • Press the "ok" button when the correct data is deleted • Data successfully erased

Test scenarios on alternative data can be seen in Table 5.

Table 5. Assessment Data Test Scenarios

Function Name	Feature Name	Test Code	Test Cases
Assessment Data	Add Assessment Data	KU12	• User performs "Add Data" action • Validate data input on form • Selecting the "Add" button • Save data into a database View successfully saved data
	Change Assessment Data	KU13	• The user performs the "change" action by pressing "Change" • Make changes to the data and select the save button • Data successfully converted into database • View successfully changed data
	Delete Assessment Data	KU14	• The user performs the "Delete" action by pressing the delete button • The message "sure to delete this?" appears • Press the "ok" button when the correct data is deleted • Data successfully erased

The test scenario on the method calculation result data can be seen in table 6.

Table 6. Test Data Scenario Method Calculation Results

Function Name	Feature Name	Test Code	Test Cases
Method	View Methods	KU15	• User selects calculation criteria • Display a table of data from method calculations

Tests are carried out to compare the suitability of the built system with the system designed so as to produce appropriate and non-conforming test results, software testing can be seen in table 7.

Table 7. Software Testing

No	Feature	Expected output	Results obtained	Result
1	Add Criteria Data	Users can add criteria data to the database	Save new data added	APPROPRIATE
2	Change Criteria Data	Users can change the criteria data contained in the database	Save data changes	APPROPRIATE
3	Clear Criteria Data	Users can delete criteria data contained in the database	Delete data from within a database	APPROPRIATE

No	Feature	Expected output	Results obtained	Result
4	Category Data Scale	User inputs category data values for the comparison process	Store the input category data value	APPROPRIATE
5	Add Category Data	Users can add category data to the database	Save new data added	APPROPRIATE
6	Change Category Data	Users can change the category data contained in the database	Save data changes	APPROPRIATE
7	Clear Category Data	Users can delete category data contained in the database	Delete data from within a database	APPROPRIATE
8	Alternative Data Scale	User inputs alternative data values for the comparison process	Store input alternate data values	APPROPRIATE
9	Add Alternate Data	Users can add alternative data to the database	Save new data added	APPROPRIATE
10	Change Alternate Data	Users can change alternative data contained in the database	Save data changes	APPROPRIATE
11	Clear Alternate Data	Users can delete alternative data contained in the database	Delete data from within a database	APPROPRIATE
12	Add Assessment Data	Users can add assessment data to the database	Save new data added	APPROPRIATE
13	Change Assessment Data	Users can change the assessment data contained in the database	Save data changes	APPROPRIATE
14	Delete Assessment Data	Users can delete assessment data contained in the database	Delete data from within a database	APPROPRIATE
15	Method	The system performs calculations and displays menu ranking results.	Can calculate and display ranking results	APPROPRIATE

Furthermore, the conclusion of the black box testing that has been carried out on the Table with a number of functions as many as 15 of them are look at data (criteria, categories, and alternatives), add data (criteria, categories, and alternatives), change data (criteria, categories, and alternatives), delete data (criteria, categories and alternatives), scale data (categories and alternatives), method calculations. Then it can be calculated the percentage of conformity of functions in the system as follows:

Number of Test Codes = 15 Test Codes
Test Code with Corresponding Result = 15 Test Code
Test Code with Mismatched Result = 0 Test Code
Percentage
Percentage = (((number of test codes-test codes do not match))/((number of test codes))×100%=((15-0))/((15)) ×100%
=100%

From the results of the calculation of the system conformity function, it can be concluded that the software testing that has been carried out using blackbox testing can already run in accordance with the specifications that have been set with a percentage of 100% meeting the specifications.

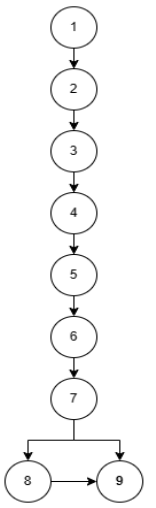
Whitebox testing is a test case design method that generates test cases using a procedural design control structure. Whitebox testing is intended to uncover errors in functional requirements without jeopardizing the internal workings of the software. As shown in the following Whitebox testing table: White box testing on criteria assessment can be seen in Table 8.

Table 8. White Box Assessment Criteria

Number/Node	Syntax

Syntax	`function penilaian($kriteria=""){`
	`$data['config'] = $this->config_model->get_config();`
	`$data['kriteria'] = $this->data_model->get_data('kriteria');`
	`$data['select'] = $kriteria;`
	`$data['hasil'] = $this->general();`
	`$data['ranking'] = $this->ranking($data['hasil']['all']);`
	`// print_r($data['hasil']['hasil']);die();`
	`if(!$this->show)`

Number/Node	Syntax
	$this->load->view('metode/show', $data); }
1	function penilaian($kriteria=""){
2	$data['config'] = $this->config_model->get_config();
3	$data['kriteria'] = $this->data_model->get_data('kriteria');
4	$data['select'] = $kriteria;
5	$data['hasil'] = $this->general();
6	$data['ranking'] = $this->ranking($data['hasil']['all']);
7	if(!$this->show)
8	$this->load->view('metode/show', $data);
9	}



V(G) = *E – N + 2*

V(G) = 9 – 9 + 2 -> There are 2 paths

- 1-2-3-4-5-6-7-8-9
- 1-2-3-4-5-6-7-9

Test Cases	Expected Output	Output Results
1-2-3-4-5-6-7-8-9	Succeed	Succeed
1-2-3-4-5-6-7-9	Fail	Fail

White box testing on the criteria calculation can be seen in Table 9.

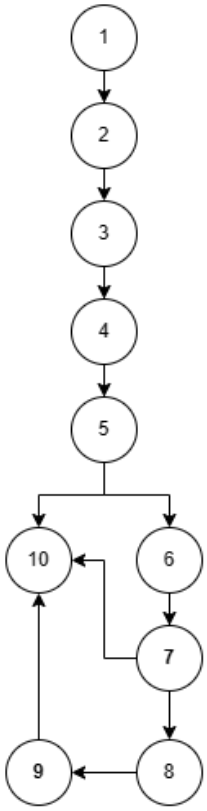
Table 9. White Box Calculate Criteria

Number/Node	Syntax

Syntax	function hitung($kriteria=""){
	$data['config'] = $this->config_model->get_config();
	$data['kriteria'] = $this->data_model->get_data('kriteria');
	$data['select'] = $kriteria;
	$data['hasil'] = $kriteria != "" ? $this->run_method($kriteria) : [];
	$data['ranking'] = $kriteria != "" ? $this->ranking($data['hasil']['perangkingan']) : [];
	if(!$this->show) $this->load->view('metode/show', $data);
	}

1	function hitung($kriteria=""){
2	$data['config'] = $this->config_model->get_config();
3	$data['kriteria'] = $this->data_model->get_data('kriteria');
4	$data['select'] = $kriteria;
5	$data['hasil'] = $kriteria != "" ?

Number/Node	Syntax
6	`$this->run_method($kriteria) : [];`
7	`$data['ranking']  = $kriteria != "" ?`
8	`$this->ranking($data['hasil']['perangkingan']) : [];`
9	`if(!$this->show) $this->load->view('metode/show', $data);`
10	`}`



V(G) = *E – N + 2*
V(G) = 11 – 10 + 2 -> There are 3 paths

- 1-2-3-4-5-6-7-8-9-10
- 1-2-3-4-5-6-7-10
- 1-2-3-4-5-10

Test Cases	Expected Output	Output Results
1-2-3-4-5-6-7-8-9-10	Succeed	Succeed
1-2-3-4-5-6-7-10	Fail	Fail
1-2-3-4-5-10	Fail	Fail

White box testing in run_method can be seen in Table 10.

Table 10. White Box Run_Method

Number/Node	Syntax
Syntax	`function run_method($kriteria){` `$show = $this->show;` `$kategori = $this->data_model->get_data_method('kategori', ['kriteria'=>$kriteria]);` `$alternatif = $this->data_model->get_data_method('alternatif', ['kriteria'=>$kriteria]);` `#fase kategori` `if ($show) {` `echo "Data Kategori ";` `$this->show_table($kategori['data'], $kategori['tot_col']);` `}` `$normalisasi= $this->normalisasi($kategori['data'], $kategori['tot_col']);` `if ($show) {`

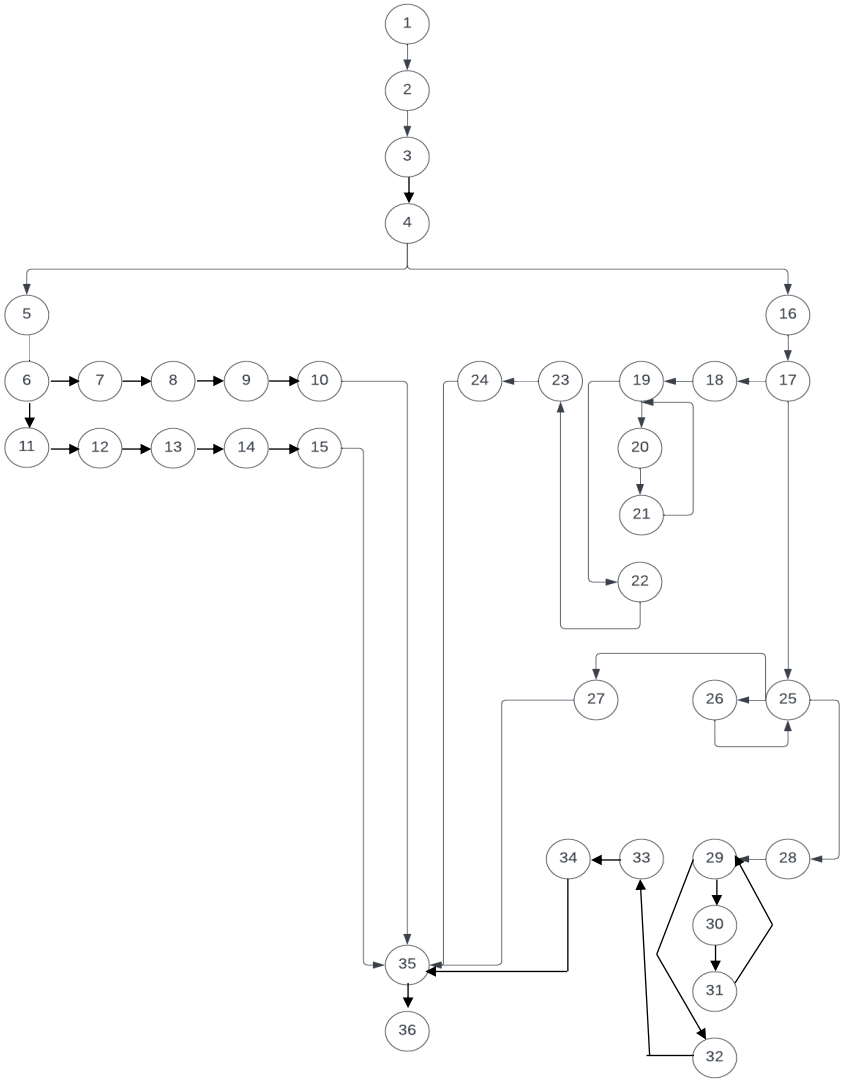
```
                        echo "Normalisasi & Bobot
Prioritas<br>";
                        $this->show_table($normalisasi['hasil'],
[], $normalisasi['max']);
                }
                $konsisten    = $this-
>konsisten_matrik($normalisasi['hasil'], $kategori['data'],
$normalisasi['max']);

                #fase alternatif
                if ($show) {
                        echo "<br><br>Data Alternatif<br>";
                        foreach ($alternatif['data'] as $key =>
$value) {
                                echo "Data ".ucwords($key)."<br>";
                                $this->show_table($value,
$alternatif['tot_col'][$key]);
                        }
                }
                $normalisasi_alternatif = array();
                foreach ($alternatif['data'] as $key => $value)
{
                        $normalisasi_alternatif[$key]     =
$this->normalisasi($value, $alternatif['tot_col'][$key]);
                }
                if($show){
                        foreach ($alternatif['data'] as $key =>
$value) {
                                echo "Normalisasi & Bobot
Prioritas<br>";
                                $this-
>show_table($normalisasi_alternatif[$key]['hasil'], [],
$normalisasi_alternatif[$key]['max']);
                        }
                }
                $perangkingan = $this-
>perangkingan($normalisasi, $normalisasi_alternatif);

                return ['perangkingan' => $perangkingan,
'konsisten' => $konsisten];
        }
1       function run_method($kriteria){
2               $show         = $this->show;
3               $kategori     = $this->data_model-
>get_data_method('kategori', ['kriteria'=>$kriteria]);
4               $alternatif   = $this->data_model-
>get_data_method('alternatif', ['kriteria'=>$kriteria]);
5               #fase kategori
6               if ($show) {
7                       echo "Data Kategori<br>";
8                       $this->show_table($kategori['data'],
$kategori['tot_col']);
9               }
10                              $normalisasi= $this-
  >normalisasi($kategori['data'], $kategori['tot_col']);
11              if ($show) {
12                      echo "Normalisasi & Bobot
Prioritas<br>";
13                      $this-
>show_table($normalisasi['hasil'], [],
$normalisasi['max']);
14              }
15              $konsisten    = $this-
>konsisten_matrik($normalisasi['hasil'], $kategori['data'],
$normalisasi['max']);
16              #fase alternatif
17              if ($show) {
18                      echo "<br><br>Data Alternatif<br>";
19                      foreach ($alternatif['data'] as $key =>
```

```
$value) {
                                    echo "Data
".ucwords($key)."<br>";
                                    $this->show_table($value,
$alternatif['tot_col'][$key]);
                            }
                    }
                    $normalisasi_alternatif = array();
                    foreach ($alternatif['data'] as $key =>
$value) {
                            $normalisasi_alternatif[$key]     =
$this->normalisasi($value, $alternatif['tot_col'][$key]);
                    }
                    if($show){
                            foreach ($alternatif['data'] as $key =>
$value) {
                                    echo "Normalisasi & Bobot
Prioritas<br>";
                                    $this-
>show_table($normalisasi_alternatif[$key]['hasil'], [],
$normalisasi_alternatif[$key]['max']);
                            }
                    }
                    $perangkingan = $this-
>perangkingan($normalisasi, $normalisasi_alternatif);

                    return ['perangkingan' => $perangkingan,
'konsisten' => $konsisten];
            }
```



V(G) = *E – N + 2*

V(G) = 39 – 36 + 2 -> There are 5 paths

- 1-2-3-4-5-6-7-8-9-10-35-36
- 1-2-3-4-5-6-11-12-13-14-15-35-36
- 1-2-3-4-16-17-18-19-20-21-19-22-23-24-35-36
- 1-2-3-4-16-17-25-26-25-27-35-36
- 1-2-3-4-16-17-25-28-29-30-31-29-32-33-34-35-36

Test Cases	Expected Output	Output Results
1-2-3-4-5-6-7-8-9-10-35-36	Succeed	Succeed
1-2-3-4-5-6-11-12-13-14-15-35-36	Succeed	Succeed
1-2-3-4-16-17-18-19-20-21-19-22-23-24-35-36	Succeed	Succeed
1-2-3-4-16-17-25-26-25-27-35-36	Succeed	Succeed
1-2-3-4-16-17-25-28-29-30-31-29-32-33-34-35-36	Succeed	Succeed

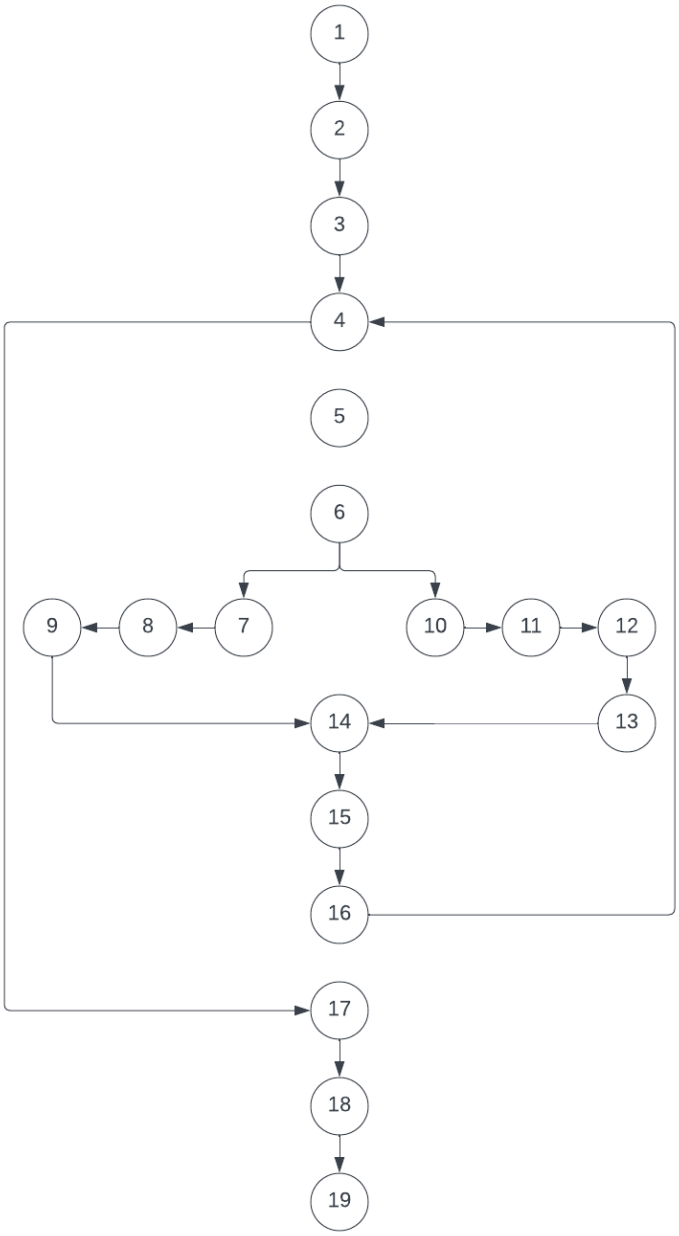
White box testing on normalization can be seen in Table 11.

Table 11. White Box Normalization

Numor/Node	Syntax

Syntax	function normalisasi($data, $sub_data){
	$result = array();
	$max = 0;
	foreach ($data as $key => $value) {
	$total = 0;
	for ($i=0; $i < count($value); $i++) {
	if($i == 0) {
	$result[$key][$i] = $key;
	}else {
	$result[$key][$i] = number_format($value[$i]/$sub_data[$i], $this->nf);
	$total += number_format($value[$i]/$sub_data[$i], $this->nf);
	}
	}
	#bobot prioritas
	$result[$key][$i] = number_format($total/(count($value)-1), $this->nf); #count(val) = jumlah kategori, -1 karena ada nama kategori
	#ambil bobot tertinggi
	$max = $max > $result[$key][$i] ? $max : $result[$key][$i];
	}
	return ['hasil'=>$result, 'max'=>$max];
	}

1	function normalisasi($data, $sub_data){
2	$result = array();
3	$max = 0;
4	foreach ($data as $key => $value) {
5	$total = 0;
6	for ($i=0; $i < count($value); $i++) {
7	if($i == 0) {
8	$result[$key][$i] = $key;
9	}
10	else {
11	$result[$key][$i] = number_format($value[$i]/$sub_data[$i], $this->nf);
12	$total += number_format($value[$i]/$sub_data[$i], $this->nf);
13	}
14	}
15	#bobot prioritas $result[$key][$i] = number_format($total/(count($value)-1), $this->nf); #count(val) = jumlah kategori, -1 karena ada nama kategori
16	#ambil bobot tertinggi $max = $max > $result[$key][$i] ? $max : $result[$key][$i];
17	}
18	return ['hasil'=>$result, 'max'=>$max];
19	}



V(G) = *E – N + 2*

V(G) = 19 – 19 + 2 -> There are 2 paths

- 1-2-3-4-5-6-7-8-9-14-15-16-4-17-18-19
- 1-2-3-4-5-6-10-11-12-13-14-15-16-4-17-18-19

Test Cases	Expected Output	Output Results
1-2-3-4-5-6-7-8-9-14-15-16-4-17-18-19	Succeed	Succeed
1-2-3-4-5-6-10-11-12-13-14-15-16-4-17-18-19	Succeed	Succeed

White box testing on normalization can be seen in Table 12.

Table 12. White Box Consistency Matrix

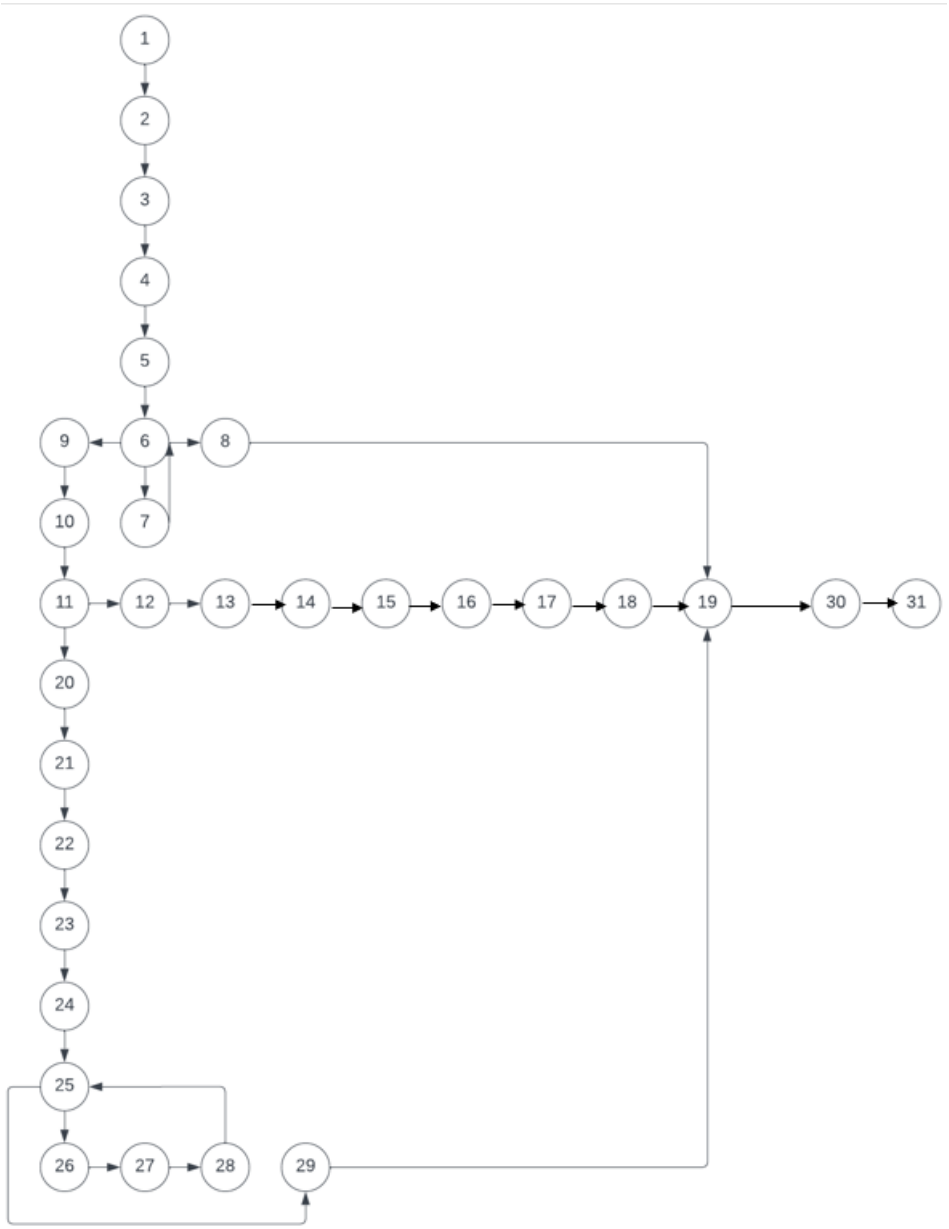
Number/Node	Syntax

Syntax	`Function konsisten_matrik($data_normal, $data_asli, $max){`
	`$bp = array(); #bp = bobot prioritas`
	`$ci = 0; #konsisten index`
	`$no = 1;`
	`#ambil data bobot prioritas`
	`foreach ($data normal as $key => $value) {`

```
                                $bp[$no++] = $value[count($value)-1];
                        }
                        #tambah konsisten matrik per baris
                        $no = 1;
                        foreach ($data_normal as $key => $value) {
                                $km = 0;
                                for ($i=1; $i < count($data_asli[$key]);
$i++) {
                                        $km +=
number_format($data_asli[$key][$i] * $bp[$i], $this->nf);
                                }
                                $data_normal[$key][] =
number_format($km/$bp[$no], $this->nf);
                                $ci += number_format($km/$bp[$no], $this-
>nf);
                                $no++;
                        }
                        $kolom_kategori = $no-1;
                        $ci = $ci/($kolom_kategori == 0 ? 1 :
$kolom_kategori); #ambil nilai rata-rata konsistensi metrik
                        $ci = number_format(($ci - $kolom_kategori) /
($kolom_kategori - 1), $this->nf); #rumus ci
                        $cr = number_format($ci / ($this-
>saaty[$kolom_kategori == 0 ? 1 : $kolom_kategori] == 0 ? 1 :
$this->saaty[$kolom_kategori == 0 ? 1 : $kolom_kategori]), $this-
>nf); #rumus cr
                        $konsisten = $cr >= 0 && $cr <= 0.1 ? 'konsisten' :
'tidak konsisten';

                        if($this->show){
                                echo "Konsistensi Metrik<br>";
                                $this->show_table($data_normal, [], $max,
true);
                                echo "<b>".ucwords($konsisten)."</b><br>";
                        }
                        return ['data_konsisten'=>$data_normal,
'konsisten'=>$konsisten, 'cr' => $cr];
                }
1               function konsisten_matrik($data_normal, $data_asli, $max){
2                       $bp = array(); #bp = bobot prioritas
3                       $ci = 0; #konsisten index
4                       $no = 1;
5                       #ambil data bobot prioritas
6                       foreach ($data_normal as $key => $value) {
7                               $bp[$no++] = $value[count($value)-1];
8                       }
9                       #tambah konsisten matrik per baris
10                                              $no = 1;
11                      foreach ($data_normal as $key => $value) {
12                              $km = 0;
13                              for ($i=1; $i < count($data_asli[$key]);
$i++) {
14                                      $km +=
number_format($data_asli[$key][$i] * $bp[$i], $this->nf);
15                              }
16                              $data_normal[$key][] =
number_format($km/$bp[$no], $this->nf);
17                              $ci += number_format($km/$bp[$no], $this-
>nf);
18                              $no++;
19                      }
20                      $kolom_kategori = $no-1;
21                      $ci = $ci/($kolom_kategori == 0 ? 1 :
$kolom_kategori); #ambil nilai rata-rata konsistensi metrik
22                      $ci = number_format(($ci - $kolom_kategori) /
($kolom_kategori - 1), $this->nf); #rumus ci
23                      $cr = number_format($ci / ($this-
>saaty[$kolom_kategori == 0 ? 1 : $kolom_kategori] == 0 ? 1 :
$this->saaty[$kolom_kategori == 0 ? 1 : $kolom_kategori]), $this-
>nf); #rumus cr
24                      $konsisten = $cr >= 0 && $cr <= 0.1 ? 'konsisten' :
'tidak konsisten';
25                      if($this->show){
26                              echo "Konsistensi Metrik<br>";
27                              $this->show_table($data_normal, [], $max,
true);
28                              echo "<b>".ucwords($konsisten)."</b><br>";
```

```
29                          }
30                          return ['data_konsisten'=>$data_normal,
                'konsisten'=>$konsisten, 'cr' => $cr];
31                  }
```



V(G) = *E – N + 2*
V(G) = 32 – 31 + 2 -> There are 3 paths

- 1-2-3-4-5-6-7-6-8-30-31
- 1-2-3-4-5-6-9-10-11-12-13-14-15-16-17-18-19-30-31
- 1-2-3-4-5-6-9-10-11-20-21-22-23-24-25-26-27-28-25-29-30-31

Test Cases	Expected Output	Output Results
1-2-3-4-5-6-7-6-8-30-31	Succeed	Succeed
1-2-3-4-5-6-9-10-11-12-13-14-15-16-17-18-19-30-31	Succeed	Succeed
1-2-3-4-5-6-9-10-11-20-21-22-23-24-25-26-27-28-25-29-30-31	Succeed	Succeed

CONCLUSION

In the Food Menu Determination Decision Support System using the Analytical Hierarchy Process (AHP) method, the conclusion that can be drawn is that AHP is an effective method in overcoming the complexity of decision making in determining food menus. This method allows users to give relative weight to each criterion and alternative, resulting in an objective and accountable ranking. This Decision Support System assists users in choosing a food menu that suits their preferences and needs. By considering relevant criteria, such as health, menu category, availability of ingredients, and others.

Thus, the Food Menu Determination Decision Support System using the Analytical Hierarchy Process (AHP) method provides significant assistance in decision making related to food menu selection based on established criteria. The results of evaluation and testing show that this system can produce food menu recommendations that match user preferences and predetermined criteria. In addition, the use of the Analytical Hierarchy Process (AHP) method in the Decision Support System provides clarity in the relative ranking of food alternatives.

Suggestions for the system are expected to be further developed so that they get better results, while development suggestions are to develop a more intuitive and easy-to-use user interface. This will ensure users can easily enter criteria, give weights, and see food menu recommendations clearly. Integrate with external resources, such as nutrition databases or food information, to provide more detailed information about nutritional value, food composition, or specific dietary restrictions. This will provide additional benefits for users who want to pay attention to nutritional aspects in food menu selection.

REFERENCES

Ismanto, J., Sarjan, M., & Qashlim, A. (2020). Sistem Pendukung Keputusan Pemilihan Menu Makanan Pada Rumah Makan Menggunakan Metode AHP. *Journal Peqguruang*, *2*(1), 103–109.

Meilani, B. D., & Wardana, A. W. (2020). Sistem Pendukung Keputusan Penentuan Resep Makanan Berdasarkan Bahan Makanan Menggunakan Metode Topsis. *Network Engineering Research Operation*, *5*(1), 15–23.

Parameswari, P. L., Astuti, I., & Ariestya, W. W. (2022). Implementasi Metode Ahp Pada Sistem Pendukung Keputusan Pariwisata Jawa Timur. *Jurnal Teknoinfo*, *16*(1), 40–45.

Prawira, M. A., & Amin, R. (2022). Sistem Pendukung Keputusan Pemilihan Karyawan Terbaik Pada PT. Citra Prima Batara Dengan Metode AHP. *Jurnal Teknik Komputer*, *8*(1), 89–97.

Salsabella, A. (2014). Sistem Pendukung Keputusan Penentuan Resep Masakan Berdasarkan Ketersediaan Bahan Makanan Menggunakan Metode Simple Additive Weighting (SAW) Berbasis Web. *JUSTIN (Jurnal Sistem Dan Teknologi Informasi)*, *2*(3), 110–117.

Siregar, Y. H., & Rahayu, S. (2018). Sistem Pendukung Keputusan Pemilihan Menu Makanan bagi Anak dengan Metode Analitycal Hierarchy Process (AHP). *(JurTI) Jurnal Teknologi Informasi*, *2*(1), 24–31.

Soekarta, R., & Modok, M. (2018). Aplikasi Rekomendasi Resep Masakan Khas Indonesia bagian Timur Menggunakan Metode Collaboration Collective Intelligence dan Slope One. *Insect (Informatics and Security): Jurnal Teknik Informatika*, *4*(1), 20–23.

Tanjung, D. H. (2017). Pemilihan Objek Wisata Di Sumatera Utara Dengan Metode Analytical Hierarchy Process

(AHP). *Seminar Nasional Informatika (SNIf)*, *1*(1), 592–597.

Valentino, V. H., Setiawan, H. S., Saputra, A., Haryanto, Y., & Putra, A. S. (2021). Decision support system for thesis session pass recommendation using AHP (analytic hierarchy process) method. *International Journal of Educational Research and Social Sciences (IJERSC)*, *2*(1), 215–221.