TSABIT
Journal of Computer Science
https://journal.ilmubersama.com/index.php/tsabit

Click here and write your Article Category

Implementation of the C4.5 Decision Tree Algorithm for the Analysis and Diagnosis of Laptop Damage at Mitra Kreasi Computer Medan

Taufik Rizky Pasaribu [1], Khairuddin Nasution [2], Mhd. Zulfansyuri Siambaton [3]

[1,2,3] *Department of Informatics Engineering, Faculty of Engineering, Universitas Islam Sumatera Utara, Medan, 20217, North Sumatra, Indonesia*

ARTICLE INFORMATION

Received: February 00, 00
Revised: March 00, 00
Available Online: April 00, 00

KEYWORDS

Laptop Damage; Diagnosis; C4.5 Decision Tree; Decision Support System; Web-Based Application

CORRESPONDENCE

Phone: +6282276587245
E-mail: taufikrizky012@gmail.com

ABSTRACT

A laptop is a portable computing device that is widely relied upon because of the mobility it offers, particularly for users who spend much of their time outside a fixed office. Frequent and intensive use, however, raises the risk of damage to both hardware and supporting components. At present, the diagnosis of laptop damage is still largely performed manually by technicians based on experience, a practice that is prone to misdiagnosis and tends to take a relatively long time. This research therefore aims to implement the C4.5 Decision Tree algorithm to analyze and diagnose laptop damage based on the symptoms that appear. The data used in this study consist of laptop service history records comprising several damage-symptom attributes, such as whether the laptop powers on, the condition of the screen display, the charging indicator, fan noise, and overheating condition. The C4.5 algorithm was used to build a decision tree by calculating entropy, information gain, and gain ratio values in order to determine the best attribute for the classification process. Calculations on the training data produced a total entropy value of 2.75, with Attribute 1 (the laptop turning off immediately when the charger is unplugged) and Attribute 3 (a BSOD with a MEMORY_MANAGEMENT code) sharing the highest information gain of 0.81125, making either eligible to serve as the root node; this research adopted Attribute 1 as the root. The resulting decision tree was then implemented into a web-based diagnostic application built with PHP and MySQL, and black box testing confirmed that the login, data management, C4.5 modeling, and damage-classification features all functioned as expected. The system is expected to help technicians and customers at Mitra Kreasi Computer Medan obtain faster, more structured, and more objective damage diagnoses.

INTRODUCTION

A laptop is a computer device that can easily be carried anywhere, and this mobility is often the primary reason a person chooses to own one, particularly for those who spend much of their working time away from a fixed office. No user wants their laptop to break down, yet damage frequently occurs unexpectedly, whether because of a specific cause or because the device was already in poor condition when purchased. When the damage is too complex to be repaired independently, bringing the laptop to a service center remains the safest and most reliable solution for the user[2]. Laptop damage can be caused by a range of factors, including the age of the device, improper use, disturbances in the electrical system, or damage to internal components[4]. The process of analyzing and diagnosing laptop damage generally still depends on the experience of the technician, which tends to take a relatively long time and carries the risk of an inaccurate diagnosis. This, in turn, can delay the repair process and increase servicing costs[3].

Along with the advancement of computing technology, data mining algorithms can be used to support the analysis and decision-making process[1]. One algorithm that can be applied to analyze and diagnose laptop damage is the C4.5 Decision Tree algorithm. This algorithm is capable of processing damage-symptom data and producing a decision tree

https://doi.org/10.12345/xxxxx

that is easy to understand and that can be translated directly into diagnostic rules[5]. The algorithm works on the concept of data mining by extracting information from past decisions and forming it into a pattern represented as a decision tree. The process begins with the processing of training data, after which the attribute with the largest priority is identified and assigned as the root of the tree; this step is repeated by searching for the largest gain until the net comparison condition is satisfied and the tree stops growing[6]. The C4.5 algorithm is used to build this decision tree and to represent and model the results of exploring important data, so that the knowledge or information contained in the data becomes easier to recognize.

Mitra Kreasi Computer Medan is one of the laptop repair service providers that handles such damage on a daily basis. Damage is caused by many factors, including the operating environment, which involves extreme temperatures, humidity, dust, and water exposure[8]. The age of the device or frequent dropping also contributes, as do user errors such as installing untrusted applications or failing to update software. Physical damage from impact or heavy pressure, as well as software issues such as bugs, application conflicts, or malware, are likewise common[7]. The aging of the device, which causes the gradual degradation of electronic components over time, adds further complexity. The large number of contributing factors makes the diagnostic process time-consuming, which can reduce customer satisfaction. Understanding this complexity is important for improving the speed and accuracy of diagnosis, so that customers' laptops can be repaired more quickly and efficiently[9].

Based on this background, the present study addresses two research questions: how to design and build a web-based laptop damage-classification application using PHP and MySQL, and how to apply the C4.5 Decision Tree algorithm in the process of analyzing laptop damage[10]. The study is scoped to laptop damage data drawn from service history, with attributes limited to hardware-related damage symptoms, the C4.5 algorithm as the sole classification method, and an output in the form of a damage-diagnosis category. The web-based prediction application was developed using PHP 8.3, Visual Studio Code 1.124.1, HTML5, JavaScript (ES2025), MySQL, and XAMPP 7.4.33[11]. The research aims to build a web-based laptop damage-prediction application and to apply the Decision Tree algorithm to predict laptop damage through that website, with the expectation that the resulting application can be implemented at Mitra Kreasi Computer Medan and ease the development of similar PHP- and MySQL-based systems in the future.

METHOD

Research Site, Instruments, and Data Collection

This research was carried out at Mitra Kreasi Computer, a laptop repair business located on Jalan Ismaliyah, Medan, North Sumatra. The hardware used to develop the system and collect data consisted of a computer with an Intel Core i5-7200U CPU at 2.50–2.70 GHz, 8 GB of RAM, and the Windows 10 operating system, while the software used comprised the Windows operating system, Visual Studio Code 1.124.1, XAMPP 7.4.33, and the PHP 8.3, HTML5, and JavaScript (ES2025) programming languages.

Data were collected through three complementary methods. Direct observation was conducted by reviewing how the technicians at Mitra Kreasi Computer carry out laptop repairs. A literature study was carried out by drawing on books, journals, and articles relevant to the theory and procedures needed to build the application, which served as the theoretical foundation for the system design. Finally, interviews were held directly with the staff and technicians of Mitra Kreasi Computer Medan to obtain first-hand information about the types of damage typically encountered, their symptoms, causes, and the solutions normally applied.

Problem-Solving Procedure

The research procedure began with determining the topic and objective of the study, namely the analysis and diagnosis of laptop damage using the C4.5 Decision Tree algorithm. The problem was then identified as the fact that diagnosis at Mitra Kreasi Computer Medan was still performed manually, which made the process time-consuming and limited the accuracy of the resulting analysis. Data were subsequently collected through observation, interviews, and documentation, covering the types of laptop damage, their symptoms, causes, and the repair solutions applied, all of which served as the raw material for the study.

The collected data were then analyzed to determine the attributes, symptoms, and damage classes to be used in the C4.5 calculation, after which the data were processed using the C4.5 algorithm itself, including the entropy and gain calculations needed to construct the decision tree and arrive at a damage diagnosis. Once the analysis was complete, a web-based diagnostic system was built using the programming language and database that had been determined in advance, and the resulting system was tested to confirm that its functions operated correctly and that the diagnosis it produced matched the available laptop-damage data. The final stage presented the output of the system in the form of a damage diagnosis and a corresponding repair solution based on the symptoms selected by the user, after which the researcher drew conclusions from the study and offered recommendations for future system development.

C4.5 Algorithm Procedure

The implementation of the C4.5 algorithm in this study followed a fixed sequence of steps. The process began by accepting the training data, which contained the history of laptop damage together with the associated symptoms, damage types, and repair solutions obtained from observation and documentation at the research site, and which served as the basis from which the algorithm learned. The system then accepted the damage-symptom attribute data, with each symptom represented as a binary value: 1 if the symptom was present and 0 if it was not.

Next, the system calculated the total entropy across the entire training dataset to measure the overall uncertainty or diversity of the data, followed by the entropy of each individual attribute based on its 0 and 1 values, in order to gauge each symptom's influence on the classification of laptop-damage type. From these entropy values, the system calculated the information gain of every attribute by comparing the total entropy with the attribute's entropy, with the gain indicating how strongly a given attribute could separate the data into distinct damage classes. The system then compared the gain values across all attributes and selected the attribute with the highest gain as the best attribute, since it offered the strongest classification ability; that attribute was assigned as the root node of the decision tree, and the tree's branches were formed according to the possible values of that attribute, namely "yes" or "no" (1 or 0).

This sequence, calculating entropy for each remaining attribute, deriving gain, and selecting the best attribute, was repeated until all attributes had been evaluated; if any attribute remained unprocessed, the procedure returned to the entropy-calculation step, and once every attribute had been accounted for, the decision tree was considered complete. The resulting tree was then used as the basis for diagnosing laptop damage and for recommending the corresponding repair solution.

Dataset

The dataset used in this study was drawn from the laptop repair records at Mitra Kreasi Computer Medan and comprised three main components. The first was a catalog of 53 laptop units across several brands, including Acer, Asus, Axioo, HP, and Lenovo, with each entry specifying its series or type, RAM capacity, storage capacity, and operating system, for example an Acer Aspire 3 A315-58 with 8 GB of RAM, 512 GB of storage, and Windows 11 Home.

The second component was a set of 15 damage-symptom questions used as the attributes in the C4.5 calculation, covering symptoms such as the laptop turning off immediately when the charger is unplugged, random characters appearing while typing, a BSOD with a MEMORY_MANAGEMENT code, a cracked LCD panel caused by impact, Bluetooth that cannot be activated, an exploded motherboard capacitor, the laptop restarting on its own without apparent cause, an NVMe SSD not being recognized in its M.2 slot, the laptop being completely unresponsive, overheating accompanied by spontaneous shutdowns, an unreadable OS partition, a device not being recognized in Device Manager, a complete absence of display output, dust accumulation that clogs the cooling fan, and the CPU not being detected in the BIOS.

The third component was a set of 14 damage categories together with their corresponding repair solutions, including battery damage requiring battery replacement, keyboard or touchpad damage requiring component replacement, RAM/SDIMM errors requiring replacement or cleaning of the memory slot, LCD/LED screen damage requiring panel replacement, USB/LAN/Wi-Fi connectivity problems requiring replacement of the Wi-Fi card or USB port, motherboard damage requiring replacement or reballing, interference with the charger or DC adapter requiring adapter or DC-jack replacement, hard-drive problems requiring HDD/SSD replacement together with data restoration, hinge damage requiring hinge replacement and casing repair, processor-fan damage requiring fan replacement and new thermal paste, a malfunctioning Windows/OS installation requiring reinstallation or repair, driver-installation problems requiring driver reinstallation or firmware updates, and processor damage requiring replacement or reflow. A subset of eight training records drawn from this dataset was used as the worked example for the manual entropy and gain calculations presented in this study.

Manual Entropy and Gain Calculation

Using the eight-record training subset, each of the 15 attributes was first coded as 1 (the symptom is present) or 0 (the symptom is absent) for every record, after which the total entropy of the dataset was calculated. The training subset contained two classes, with a 2:6 split between them, giving a total entropy of:

$$Entropy(Total) = -(2/8 \times \log_2(2/8)) - (6/8 \times \log_2(6/8)) = 2.75 \quad (1)$$

The entropy of each attribute was then calculated separately for its "yes" (value 1) and "no" (value 0) subsets. For Attribute 1, the subset with value 1 was perfectly homogeneous, giving an entropy of 0, while the subset with value 0 (six records) gave an entropy of 2.5849625. Attribute 2 produced an entropy of 0 for value 1 and 2.75 for value 0, meaning it carried no separating power on this subset. Attribute 3 produced an entropy of 1 for value 1 (two records evenly split between classes) and 2.2516 for value 0 (six remaining records). The remaining attributes, 4 through 14, were calculated in the same manner; Attribute 15, for instance, produced an entropy of 0 for value 1 and 2.75 for value 0, identical to Attribute 2. The complete set of entropy values for all 15 attributes is summarized in Table 1.

Table 1. Entropy Values for Each Attribute Based on the Training Subset

Attribute	**Entropy (value = 1)**	**Entropy (value = 0)**
Attribute 5	0	2.75
Attribute 6	0	2.406304
Attribute 7	0	2.75
Attribute 8	0	2.807355
Attribute 9	0	2.75
Attribute 10	0	2.75
Attribute 11	0	2.521641
Attribute 12	0	2.521641
Attribute 13	0	2.75
Attribute 14	0	2.521641
Attribute 15	0	2.75

The information gain of each attribute was then obtained by subtracting the weighted average of its branch entropies from the total entropy. For Attribute 1, this gave Gain = 2.75 − [(2/8 × 0) + (6/8 × 2.585)] = 2.75 − 0 − 1.93875 = 0.81125. For Attribute 2, both branches were weighted against the full entropy of 2.75, giving Gain = 2.75 − 0 − 2.75 = 0. For Attribute 3, Gain = 2.75 − (2/8 × 1) − (6/8 × 2.2516) = 2.75 − 0.25 − 1.6887 = 0.81125, identical to Attribute 1. The remaining attributes were calculated using the same procedure, and the complete results are summarized in Table 2.

Table 2. Information Gain Values for Each Attribute

Attribute	**Information Gain**
Attribute 1	0.81125
Attribute 2	0
Attribute 3	0.81125
Attribute 4	0.543564
Attribute 5	0
Attribute 6	0.644484
Attribute 7	0
Attribute 8	0.293564
Attribute 9	0
Attribute 10	0
Attribute 11	0.543564
Attribute 12	0.543564

Attribute 13	0
Attribute 14	0.543564
Attribute 15	0

Ranking the attributes by gain value showed that Attributes 1 and 3 shared the highest gain of 0.81125, followed by Attribute 6 (0.644483808), then Attributes 4, 11, 12, and 14 (each 0.543564443), and Attribute 8 (0.293564443); the remaining attributes contributed a gain of 0 and therefore had no influence on the classification at this stage. Because Attributes 1 and 3 were tied for the highest gain, either could in principle serve as the root node; this study selected Attribute 1 as the root.

RESULTS AND DISCUSSION

Decision Tree Structure

Based on the gain values obtained in Section 3.5, Attribute 1, the laptop turning off immediately when the charger is unplugged, was selected as the root node with the highest gain of 0.81125. If the answer to this attribute is "yes," the system identifies battery damage and recommends battery replacement; if the answer is "no," the diagnosis proceeds to Attribute 3. At Attribute 3, a BSOD with a MEMORY_MANAGEMENT code, a "yes" answer leads the system to conclude that there is a RAM/SDIMM error, recommending replacement or cleaning of the memory slot, while a "no" answer moves the process to Attribute 6.

At Attribute 6, an exploded motherboard capacitor, a "yes" answer indicates motherboard damage requiring replacement or reballing, whereas a "no" answer leads to Attribute 4, a cracked LCD panel; a "yes" answer here results in a diagnosis of LCD/LED screen damage requiring panel replacement, while a "no" answer continues to Attribute 11. At Attribute 11, a malfunctioning Windows/OS installation, a "yes" answer leads to a recommendation of OS reinstallation or repair, and a "no" answer proceeds to Attribute 12, driver-installation problems, where a "yes" answer results in a recommendation to reinstall the driver or update the firmware, and a "no" answer moves on to Attribute 14.

At Attribute 14, processor-fan damage, a "yes" answer leads to a recommendation of fan replacement together with new thermal paste, while a "no" answer moves to Attribute 8, hard-drive problems; a "yes" answer here results in a recommendation to replace the HDD/SSD and restore the data, while a "no" answer places the case into a separate or special damage category requiring further inspection of other laptop components. Taken together, this branching structure mirrors the diagnostic logic that a technician would normally follow, moving from the most common and most informative symptoms toward more specific or less frequent ones.

System Design

The system was designed around a single administrator actor with full access to its features, as captured in the use case diagram. After authenticating through the login feature, the administrator is directed to a dashboard that presents general information about the system. From there, the administrator can manage laptop specification data, linked to brand and series information used to identify the unit under diagnosis, and manage the laptop data used as both training and testing data for the classification process. A dedicated C4.5 modeling feature allows the administrator to trigger the construction of the decision tree from the available training data, during which the system calculates entropy and gain to determine the best attribute for each node; the resulting tree is then used in the damage-classification feature, where the administrator inputs the relevant symptoms and the system matches them against the decision tree to produce a damage classification and its corresponding solution. A logout feature is provided to end the session securely.

The login process itself is described by a sequence diagram involving the administrator, the login interface, a login controller, the database, and the dashboard page. The administrator enters a username and password, which the login interface forwards to the login controller; the controller in turn requests that the database validate the submitted credentials against the stored account data. If the credentials match, the database returns a successful response and the dashboard page is displayed; if they do not match, the system returns a login-failure message to the user.

The overall workflow is further described by an activity diagram, in which the administrator's access to the main page triggers the display of the login interface; after the credentials are entered, the system validates them through a decision node, returning the administrator to the login page if the data is invalid or granting access to the dashboard if validation succeeds. The administrator then carries out a series of management activities, covering laptop data, question data, training data, and damage data in sequence, before instructing the system to perform the damage classification, the result of which is displayed back to the administrator as the final output. The underlying database structure is represented by a class diagram comprising eight main classes: Users, Brand, Series, Laptop Data, Question, Training Data, Damage, and Tree. The Users class supports administrator authentication, while Brand, Series, and Laptop Data store the specification information for the laptop units under study. Brand and Series are linked to Laptop Data to capture the relationship between unit-level data, the laptop-data attribute of the Laptop Data table is linked to the corresponding attribute in the

Training Data table to supply the data used for training the system, and the damage attribute in the Damage table is linked to the corresponding attribute in the Training Data table to determine the diagnosis produced from the training data.

At the database level, this design was implemented through eight tables: a Laptop Data table storing each laptop's specification text; a Training Data table storing the laptop reference together with 65 binary symptom columns (Mkc01 through Mkc65) and the resulting damage type and solution; a Damage table storing each damage type and its solution; a Brand table; a Users table storing the administrator's username, password, and full name; a Question table storing each question's unique code and text; a Tree table storing the constructed decision tree, including parent–child node relationships, the attribute and its value, the resulting decision, the solution, a leaf-node flag, and the corresponding gain value; and a Series table storing the laptop series. This structure allowed the system to persist both the raw service data and the decision tree generated from it.

Implementation

The login page was implemented with a simple, functional layout, opening with the greeting "Halo! Mari kita mulai" ("Hello! Let's get started") and two input fields for the username and password, with a prominent purple "Masuk" ("Sign in") button used to trigger authentication. Once logged in, the administrator is taken to a dashboard that serves as the operational control center, presenting a sidebar with navigation to both the physical-data menus, such as laptop specifications and laptop data, and the technical menus related to the machine-learning workflow, such as training data, C4.5 modeling, and damage classification. The main dashboard area displays summary statistics in card form, indicating that the knowledge base held 53 laptop records, 65 question records, 14 damage types, and 55 training records, alongside a weather widget for the Medan area and a footer identifying the system's developer.

The brand-management page presents a table of laptop brands, namely Acer, Asus, Axioo, HP, and Lenovo, with columns for the record number, brand name, and an action column containing a yellow edit button and a red delete button, alongside a green "Add Data" button for registering a new brand. The series-management page follows the same layout for managing each brand's series or type, for example Acer's Aspire 3 A315-58, Aspire 5 A514-54, Aspire 5 A515-56, Extensa 15 EX215-54, Nitro 5 AN515-56, and Nitro 5 AN515-57, while the laptop-data page lists each unit together with its RAM, storage, and operating system, with a delete action and an "Add Data" button for new entries.

The question-management page presents each symptom together with its unique code, such as MKC01 or MKC02, and statements like "the laptop turns off immediately when the charger is unplugged," "random characters appear while typing," or "the motherboard capacitor exploded," with an edit action for each entry. The damage-management page similarly lists each damage type together with its recommended solution, such as a driver-installation problem resolved by reinstalling the driver or updating the firmware, keyboard or touchpad damage resolved by component replacement, and processor-fan damage resolved by fan replacement and new thermal paste.

The training-data page lists each laptop entry alongside action buttons for viewing details, editing, and deleting, with an "Add Data" button for new records; one entry observed in the implementation, an Acer Aspire 3 A315-58 with 8 GB of RAM, 512 GB of storage, and Windows 11 Home, appeared across six separate rows, consistent with its use as part of the training dataset that the algorithm learns from. The C4.5 modeling page presents a table with columns for the record number, attribute (symptom), value, decision, gain, and node type, where, for example, attribute MKC21 with a "yes" value produced the decision "processor-fan damage" with a gain of 0.2306 and a leaf-node type, while attribute MKC06 produced "motherboard damage" and attribute MKC42 produced "hinge damage." Attributes with a "no" value were marked as internal nodes, indicating further branching in the tree. Two buttons at the top of this page, a green "Start Mining Process" and a red "Delete Mining Results," allow the administrator to run the C4.5 algorithm on the training data or clear a previously generated model.

The damage-classification page provides a diagnostic form consisting of a dropdown for selecting the laptop unit to be examined and a checklist of coded symptoms (MKC01, MKC02, and so on) covering conditions such as the laptop turning off when the charger is unplugged, a BSOD with a MEMORY_MANAGEMENT code, a cracked LCD panel from impact, or an undetected NVMe SSD in its M.2 slot. After the administrator selects the unit and the relevant symptoms, the system processes the input against the previously constructed decision tree to produce a diagnosis. One observed result page showed the outcome for an Acer Aspire 5 A515-56 with 8 GB of RAM, 512 GB of storage, and Windows 11 Home: based on the symptom MKC01 ("the laptop turns off immediately when the charger is unplugged"), the system traced the decision tree to a diagnosis of battery damage and recommended battery replacement, while also displaying the full list of symptoms considered and their selected status. Finally, the logout feature presents a confirmation dialog with the message "Apakah Anda yakin? Sesi Anda akan diakhiri dan kembali ke halaman Login" ("Are you sure? Your session will end and you will be returned to the login page"), together with a blue "Yes, log out" button to confirm the action and a red "Cancel" button to remain on the current page.

Black Box Testing

Black box testing was carried out to verify the functional behavior of the system without inspecting its internal code structure. The scenarios covered authentication with correct and incorrect credentials; adding, editing, and deleting brand, series, laptop, question, damage, and training data; running and clearing the C4.5 mining process; performing a damage classification both with and without the required inputs supplied; viewing classification results; and logging out, including canceling the logout action. The summary of the test scenarios and their outcomes is presented in Table 3.

Table 3. Black Box Testing Scenarios and Results

No.	Module/Feature	Test Scenario	Test Data	Expected Result	Status
1	Login	Login with correct username and password	Sign in with valid credentials	System displays the dashboard page	Success
2	Login	Login with an incorrect username	Sign in with an invalid username	System displays a login-failure message	Success
3	Login	Login with an incorrect password	Sign in with an invalid password	System displays a login-failure message	Success
4	Brand Data	Adding a new brand	Brand: MSI	Brand data is saved and shown in the table	Success
5	Brand Data	Editing brand data	Asus changed to ASUS	Brand data is successfully updated	Success
6	Brand Data	Deleting brand data	Select a brand record	System removes the record from the table	Success
7	Series Data	Adding a new laptop series	Acer Swift Go 14	Series data is saved	Success
8	Laptop Data	Adding new laptop data	New laptop specification	Laptop data is saved in the table	Success
9	Question Data	Editing question data	Question MKC65 replaced	Question data is successfully changed	Success
10	Damage Data	Editing damage data and its solution	New damage type and solution	Data is successfully changed	Success
11	Training Data	Adding training data	Selected symptoms and resulting damage	Training data is saved	Success
12	Training Data	Editing training data	Edit an existing record	Training data is successfully updated	Success
13	Training Data	Deleting training data	Select a training record	Training data is successfully deleted	Success
14	C4.5 Modeling	Running the mining process	Click "Start Mining Process"	System produces entropy, gain, and the decision tree	Success
15	C4.5 Modeling	Deleting the mining results	Click "Delete Mining Results"	Modeling results are deleted	Success
16	Damage Classification	Diagnosing by selecting a laptop and a symptom	Laptop: Acer Aspire 5; Symptom: MKC01	System displays the diagnosis and solution	Success
17	Damage Classification	Diagnosing without selecting a symptom	Laptop selected, symptom left empty	System displays a validation message	Success
18	Damage Classification	Diagnosing without selecting a laptop	Symptom selected, laptop left empty	System displays a validation message	Success
19	Diagnosis Result	Viewing the classification result detail	Diagnosis result data	System displays the damage type, solution, and selected symptoms	Success
20	Logout	Logging out by confirming "Yes, log out"	Click "Yes, log out"	System ends the session and returns to the login page	Success

21	Logout	Canceling the logout action	Click "Cancel"	System remains on the active page	Success

All of the tested scenarios were recorded as successful, indicating that the authentication mechanism, the CRUD operations across the brand, series, laptop, question, damage, and training-data modules, the C4.5 mining process, the input validation on the classification form, and the session-termination flow each behaved as expected.

CONCLUSION

This study has produced a web-based application, built with PHP and MySQL, that implements the C4.5 Decision Tree algorithm to analyze and diagnose laptop damage based on entropy and information gain calculations across the damage-symptom attributes; the results show that Attribute 1 and Attribute 3 shared the highest gain value of 0.811278124 and were therefore the most influential factors in constructing the decision tree, while Attributes 2, 5, 7, 9, 10, 13, and 15 shared the lowest gain value of 0 and had the least influence on classification, demonstrating that the system can determine the most dominant damage types and produce a structured basis for diagnosis. For future development, it is recommended that Mitra Kreasi Computer Medan periodically update its dataset and technical symptoms to keep pace with newer laptop models, that subsequent research compare C4.5 with other classification methods such as Random Forest, Naive Bayes, or Support Vector Machine to improve diagnostic accuracy, and that the system eventually be extended to a mobile platform for more convenient self-diagnosis.

ACKNOWLEDGMENT

The author gratefully acknowledges the guidance and supervision of Khairuddin Nasution, S.T., M.Kom and Mhd. Zulfansyuri Siambaton, ST., M.Kom throughout the research process, and expresses appreciation to the Department of Informatics Engineering, Faculty of Engineering, Universitas Islam Sumatera Utara, for providing the institutional support necessary to complete this work.

REFERENCES

Journal Article

[1] Aziz, N. (2022). Analisis perancangan sistem informasi (N. S. Wahyuni, Ed.; 1st ed.). Widina Media Utama.

[2] Dewi, S. P., Nurwati, N., & Rahayu, E. (2022). Penerapan data mining untuk prediksi penjualan produk terlaris menggunakan metode K-Nearest Neighbor. Building of Informatics, Technology and Science, 3(4), 639–648.

[3] Faderubun, A., & Maskhuliah, P. (2023). Program aplikasi computer. Scientia Jurnal Ilmiah Sain dan Teknologi, 1(1), 52–58.

[4] Hartati, T., Anastia, N., & Widyastuti, R. (2021). Penerapan SDLC model waterfall pada rancang bangun SI-PKP Direktorat Jenderal ILMATE Kementerian Perindustrian Jakarta. Remik, 6(1).

[5] Hari Utami, F. (2022). Aplikasi pelayanan antrian pasien menggunakan metode FCFS menggunakan PHP dan MySQL. Jurnal Media Infotama, 18(1), 341–139.

[6] Ichsan, A., Zulherry, A., Lubis, T. A., & Shahnaz, B. A. Z. (2025). Utilization of Mobile Applications to Speed Up The Search for Android-Based Index Places. IJATCoS: Indonesian Journal of Applied Technology. Computer and Science, 2(1).

[7] Nasrullah, A. H. (2021). Implementasi algoritma decision tree untuk klasifikasi produk laris. Jurnal Ilmiah Ilmu Komputer, 7(2), 45–51. https://doi.org/10.35329/jiik.v7i2.203

[8] Nasution, I. K. (2025). Penerapan algoritma decision tree untuk meningkatkan ketepatan klasifikasi kerusakan alat berat berbasis website (Studi kasus di Grand Indo Perkasa, Penambangan Batu Bara Sumatera Selatan) [Undergraduate thesis, Universitas Islam Sumatera Utara].

[9] Nasution, M. A. A. H., Siswanto, S., & Suryana, E. (2023). Rancangan media pembelajaran berupa aplikasi augmented reality berbasis Android. Jurnal Media Infotama, 19(2), 528–537.

[10] Noviantoro, A., Silviana, A. B., Fitriani, R. R., & Permatasari, H. P. (2022). Rancangan dan implementasi aplikasi sewa lapangan badminton wilayah Depok berbasis web. Jurnal Teknik dan Science, 1(2), 88–103.

[11] Prasetya, A. F., Sintia, & Putri, U. L. D. (2022). Perancangan aplikasi rental mobil menggunakan diagram UML (Unified Modelling Language). Jurnal Ilmiah Komputer Terapan dan Informasi, 1(1), 14–18.

[12] Putri, S. J., Putri, D. G. P., & Nugraha, W. H. (2024). Analisis komparasi pada teknik black box testing (Studi kasus: website LARS). Journal of Internet and Software Engineering, 5(1).
[13] Resman, K. I. K., Gunadnya, I. B. P., & Budisanjaya, I. P. G. (2021). Rancang bangun sistem informasi e-commerce penjualan ayam pedaging berbasis website di Kabupaten Manggarai Barat, NTT. Jurnal BETA (Biosistem dan Teknik Pertanian), 9(1), 21.
[14] Sagala, A. A. H., & Yahfizham, Y. (2024). Analisis pengenalan konsep algoritma pemrograman matematika pada kehidupan sehari-hari. Morfologi: Jurnal Ilmu Pendidikan, Bahasa, Sastra dan Budaya, 2(1), 1–16.
[15] Sidiq, C. M., Faqih, A., & Dwilestari, G. (2024). Algoritma decision tree C4.5 digunakan untuk mengklasifikasikan data stroke. JATI (Jurnal Mahasiswa Teknik Informatika), 8(2), 1869–1874.
[16] Sihombing, M. M., & Basri, M. (2024). Decision support system memprediksi kerusakan handphone berbasis web menggunakan algoritma C4.5 di Pioneer Service. CESS (Journal of Computer Engineering, System and Science), 9(2), 449.
[17] Tiwari, U. K., & Kumar, S. (2017). Components integration-effect graph: A black box testing and test case generation technique for component-based software. Springer.
[18] Winanjar, J., & Susanti, D. (2021). Rancang bangun sistem informasi administrasi desa berbasis web menggunakan PHP dan MySQL. Prosiding SNAST, 97–105.
[19] Yuhefizar. (2021). Perancangan website sebagai media informasi dan promosi batik khas Kabupaten Kulonprogo. AMIK BSI Yogyakarta, 3(2).
[20] Zulherry, A., Sari, I.P., & Basri, M. (2025). Perancangan Aplikasi Monitoring Kehadiran Pegawai Menggunakan RFID. sudo Jurnal Teknik Informatika 4 (4), 378-384