Designing a Child Behavior Disorder Expert System Database Using the Database Life Cycle Method

Nurhayati[1]*, Eni Nur Rahmawati[2], Imanuel Dwi Anand Sinar Putra[3], Dimas Rizky Maulana[4]
Universitas Duta Bangsa Surakarta[1,2,3,4]
* Correspondence Email: nurhayati@udb.ac.id

A BSTRACT

Behavioral disorders in children are complex issues that require careful handling in order to avoid affecting their future development. An expert system has been developed to assist parents, teachers, and health professionals in diagnosing and treating various types of behavioral disorders in children. A well-designed database is essential for an effective expert system, ensuring integrity, availability, and optimal performance. This research aims to design a database for an expert system for behavioral disorders in children using the database life cycle method. The study employs relational database design, including conceptual, logical, and physical design and implementation in MySQL DBMS. The results of the study indicate that the database design approach, including conceptual, logical, and physical databases, can produce an efficient database that adheres to the principles of a relational database.

KEYWORDS System , expert , database, dblc , relational

INTRODUCTION

The rapid development of information technology has opened up great opportunities in various fields, including health and psychology. One area that can utilize information technology is the development of expert systems to identify and analyze behavioral disorders in children. Behavioral disorders in children are quite complex problems, requiring special attention because they can affect the child's mental and social development in the future. Therefore, a system is needed that can assist experts in diagnosing and providing appropriate treatment recommendations.

Expert systems are one solution that can be used to help diagnose behavioral disorders in children more quickly and efficiently. This system can imitate the ability of an expert to analyze data and provide solutions based on previously programmed knowledge. However, for an expert system to function properly, a strong and structured database design is required.

However, database designing is not a simple task. Many organizations face challenges such as inconsistent data, difficulties in integrating data from various sources, and a lack of flexibility in responding to changing business needs (Wijaya et al., 2021). Without proper design, databases can become a source of problems, such as data redundancy, difficulties in maintenance, and slow system performance (Setiyadi, 2021).

To overcome these challenges, the Database Life Cycle (DBLC) method offers a systematic and structured approach to database development. DBLC includes a series of stages ranging from planning, needs analysis, design, and implementation, to maintenance (Triandini, 2022). The use of the DBLC method in database design for a child behavioral disorder expert system is very important to ensure that the data used is accurate, easily accessible, and can be updated as needed.

The background of this research is to develop a reliable and well-structured database to support the expert system of child behavioral disorders. By using the Database Life Cycle method, it is expected that the expert system developed can provide accurate diagnosis and appropriate treatment recommendations.

RESEARCH METHOD

This research uses descriptive research methods. The database design methodology that the author uses is DBLC (Database Life Cycle), namely a method that explains the life cycle of a database (Samidi & Hidayat, 2023). This DBLC will continue to return to its starting point because a database that will be created will require improvements according to developments. The process in DBLC is divided into three stages, namely conceptual, logical, and physical (Anwar & Purnama, 2022)

RESULTS AND DISCUSSION

The techniques used in database design are divided into three stages, namely conceptual database design, logical database design, and physical database design.

A. Conceptual Database Design

Conceptual database design involves based model creation information organization, without considering aspect planning physique. This process started with building a conceptual data model, without taking implementation details into account. The conceptual design will show an entity and relationship based on the process desired by the user (Bahry et al., 2022). Determining entities and relationships is carried out by taking data to be used as material for analyzing the needs of the expert system for child behavioral disorders that is being built. Conceptual database design results are presented in Table 1 Identification Type Entity.

Table 1. Identification Type Entity

Entity Name	Entity Description	Activity
Admin	Contains information regarding expert system admin data	Grouping admin data based on admin ID, admin username and admin password
User	Contains information about expert system user data	Grouping user data based on ID user, user name, date of birth, age, gender, telephone number,

Entity Name	Entity Description	Activity
		address, username and password user
Info_perilaku	Containing information about child behavioral disorders displayed on the system expert	Grouping information child behavioral disorders includes information ID, date information, title information, information, images
Penyakit	Containing information about type type child behavioral disorders who can decided by the system expert.	Grouping type type child behavioral disorders includes disease ID, name disease, causes and management
Gejala	Containing information about the symptoms that appear through observations, selected symptoms will do search results type child behavioral disorders on the system expert	Grouping symptom child behavioral disorders includes symptom ID and name symptom
Basis_pengetahuan	Contains the rules used for expert system knowledge management.	Grouping management knowledge system expert includes base ID and disease ID
Basis_detail	Contains detailed rules and symptom confidence values used for expert system knowledge management	Grouping details knowledge base rules with applying certainty factors includes symptom ID, measure of belief and disbelief.
Telusur_gejala	Containing history results search child behavioral disorders on the system expert based on entered symptoms	Grouping history search symptom includes search ID, user ID, date search, disease id and value
Detail_telusur	Contains a history of symptoms selected when searching in the expert system.	Grouping history selected symptoms includes search ID and symptom ID
Nakes	Grouping of health workers used during the consultation process in the expert system	Grouping of health workers includes health worker ID, health worker name, profession, health worker user, health worker password, and health worker photo.
Konsultasi	Grouping of consultation data carried out between users and health worker in the system expert	Grouping consultation data includes consultation ID, results search, consultation input date and consultation status

Entity Name	Entity Description	Activity
Balas_konsultasi	Grouping reply to data on consultations carried out between users and power health in the system expert	Grouping reply data consultation with power health includes reply ID, delivery, date reply, reply, reply status consultation
Faskes	Grouping of data on health facilities providing treatment and therapy services for children's behavioral disorders	Grouping of health facility data includes ID, name, type, province, city, address, telephone number, photo, and description of the health facility.

The purpose of identifying entities and their relationships is to establish significant relationships between previously identified entity types. Entity and relationship relationships are presented in Figure 2.

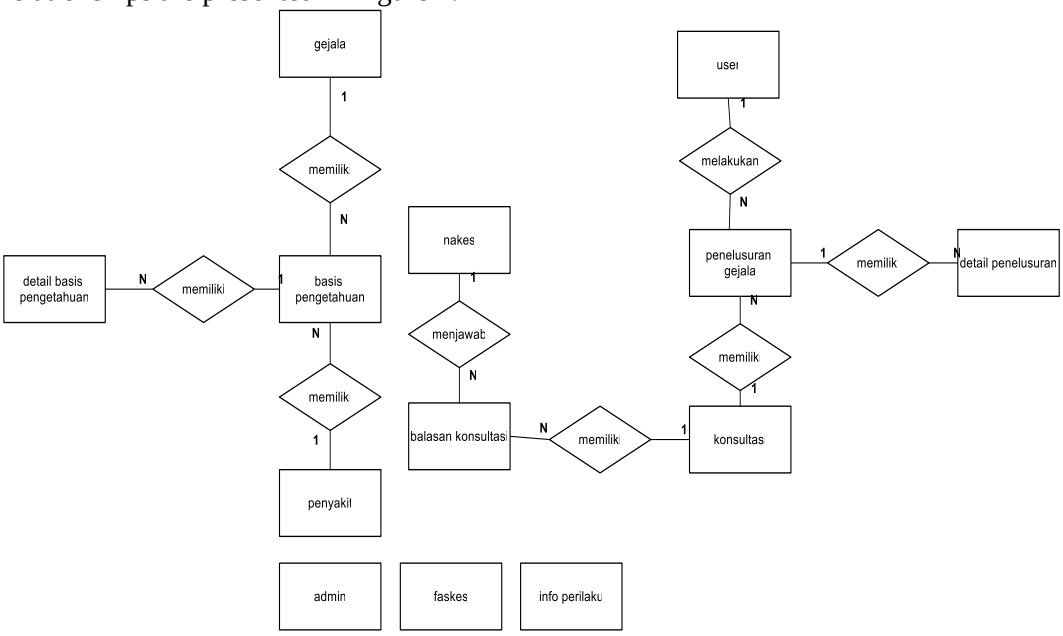

Figure 1. Relationships entities and relationships

The next stage is to determine the domain of the entities and attributes that have been defined in the previous stage. The attributes and domains of the child behavioral disorder expert system are presented in Table 2.

Table 2 Attributes and Domains

Entity Name	Attribute - Domain
Admin	id_admin (varchar) user_admin (varchar) pass_admin (varchar)
User	id_user (int) nama_user (varchar) tgl_lahir (date) umur (int) jk enum('L','P') notelp (varchar) alamat (varchar) username (varchar) pass_user (varchar)
Info_perilaku	id_info (int) tgl_info (datetime) judul_info (varchar) info (text) gambar (varchar)
Penyakit	id_penyakit (varchar) nama_penyakit (varchar) penyebab (text) tata_laksana (text)
Gejala	id_consult (int) id_telusur (int) id_user (int) id_nakes (varchar) hasil_telusur (varchar) pertanyaan (text) tgl_input (datetime) dibaca enum('Y','T')
Basis_pengetahuan	id_basis (varchar) id_penyakit (varchar)
Basis_detail	id_basis (varchar) id_gejala (varchar) MB (double) MD (double)
Telusur_gejala	id_telusur (int) id_user (int) tgl_telusur (datetime) id_penyakit (varchar) nilai (double)
Detail_telusur	id_telusur (int) id_gejala (varchar)

Entity Name	Attribute - Domain
Nakes	id_nakes (varchar) nama_nakes (varchar) profesi (varchar) user_nakes (varchar) pass_nakes (varchar) foto (varchar)
Konsultasi	id_consult (int) id_telusur (int) id_user (int) id_nakes (varchar) hasil_telusur (varchar) pertanyaan (text) tgl_input (datetime) dibaca enum('Y','T')
Balas_konsultasi	id_balas (int) id_consult (int) dari (enum('nakes','user')) tgl_balas (datetime) jawaban (text) dibaca (enum('Y','T'))
Faskes	id_faskes (varchar) nama_faskes (varchar) jenis_faskes (varchar) provinsi_faskes (varchar) kota_kab_faskes (varchar) alamat_faskes (varchar), no_telp (varchar), foto_faskes (varchar) desk_faskes (text)

B. Logical Database Design

Logical database design is a stage in database development that aims to convert a conceptual data model into a more detailed data model that can be implemented in a database management system (DBMS)(Imbulpitiya et al., 2021). Relationship analysis has been carried out before being mapped into the physical database design. The logical data model is presented in Figure 2.

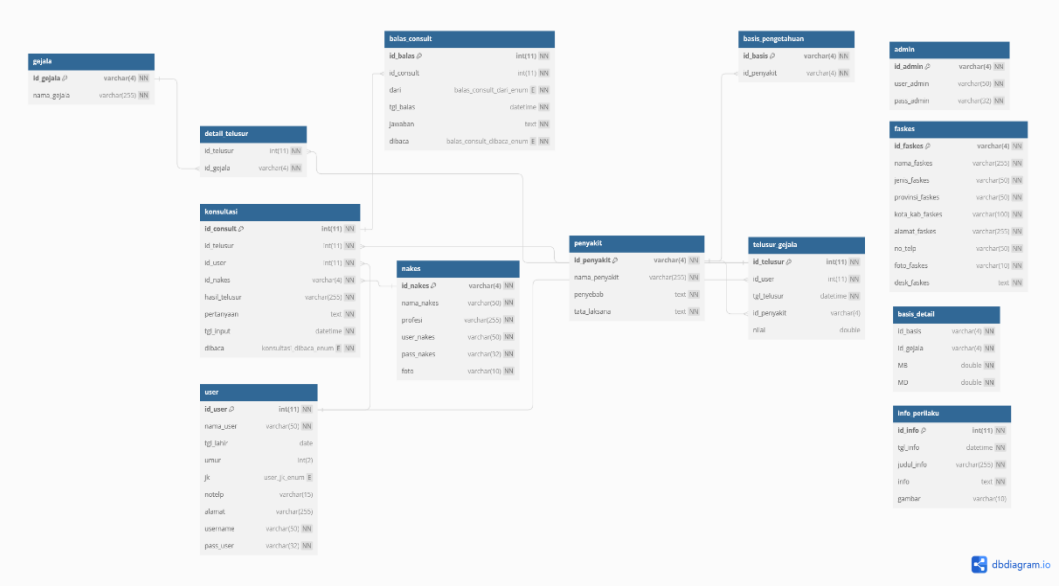

Figure 2 Logical Data Model

C. Physical Database Design

Physical database design is a stage for implementing the results of logical design into physical storage that is adjusted to the database management system (DBMS) software used (Septiawati & Sauda, 2020). The expert system of child behavioral disorders uses MySQL DBMS software, the selection of this DBMS considers that MySQL is compatible with the PHP programming language applied to the expert system. The structure of the tables is shown in the following table description:

Table 3 Structure table Admin

Fields	Type	Size	Information
id_admin	varchar	4	NOT NULL, Primary Key
user_admin	varchar	50	NOT NULL
pass_admin	varchar	30	NOT NULL

Table 4 Structure table User

Fields	Type	Size	Information
id_user	int	11	NOT NULL, Primary Key
nama_user	varchar	50	NOT NULL
tgl_lahir	date		NULL
umur	int	2	NULL
jk	enum	'L','P'	NULL
notelp	varchar	15	NULL
alamat	varchar	255	NULL
username	varchar	50	NOT NULL
pass_user	varchar	30	NOT NULL

Table 5 Structure table Info_perilaku

Fields	Type	Size	Information
id_info	int	11	NOT NULL, Primary Key
tgl_info	datetime		NOT NULL
judul_info	varchar	255	NOT NULL

Fields	Type	Size	Information
info	text		NOT NULL
gambar	varchar	10	NULL

Table 6 Structure table Penyakit

Fields	Type	Size	Information
Id_penyakit	Varchar	4	NOT NULL, Primary Key
Nama_penyakit	Varchar	255	NOT NULL
Penyebab	Text		NOT NULL
Tata_laksana	Text		NOT NULL

Table 7 Structure table Gejala

Fields	Type	Size	Information
Id_gejala	Varchar	4	NOT NULL, Primary Key
Nama_gejala	Varchar	255	NOT NULL

Table 8 Structure table Basis Pengetahuan

Fields	Type	Size	Information
Id_basis	Varchar	4	NOT NULL, Primary Key
Id_penyakit	Varchar	4	NOT NULL, Foreign Key Tabel penyakit

Table 9 Structure table Basis Detail

Fields	Type	Size	Information
Id_basis	Varchar	4	NOT NULL, Foreign key tabel Basis Pengetahuan
Id_gejala	Varchar	4	NOT NULL, Foreign Key Tabel gejala
MB	Double		NOT NULL
MD	Double		NOT NULL

Table 10 Structure table Telusur_gejala

Fields	Type	Size	Information
Id_telusur	Int	11	NOT NULL, Primary Key
Id_user	Int	11	NOT NULL, Foreign Key Tabel user
Tgl_telusur	datetime		NOT NULL
Id_penyakit	Varchar	4	NULL, Foreign Key Tabel Penyakit
Nilai	Double		NULL

Table 11 Structure table Detail_telusur

Fields	Type	Size	Information
id_telusur	Int	11	NOT NULL, Foreign key tabel telusur_gejala
Id_gejala	Varchar	4	NOT NULL, Foreign Key Tabel gejala

Table 12 Structure table Faskes

Fields	Type	Size	Information
id_faskes	verchar	4	NOT NULL, Primary Key
name_faskes	varchar	255	NOT NULL
Health_facility type	Varchar	50	NOT NULL
Province_faskes	Varchar	50	NOT NULL
Kota_kab_faskes	Varchar	100	NOT NULL
Health_facility_address	varchar	255	NOT NULL
Phone number	varchar	50	NULL
Photo_faskes	varchar	10	NOT NULL
Desk_faskes	text		NOT NULL

Table 13 Structure table Nakes

Fields	Type	Size	Information
Id_nakes	Varchar	4	NOT NULL, Primary Key
Nama_nakes	Varchar	50	NOT NULL
Profesi	Varchar	255	NOT NULL
User_nakes	Varchar	50	NOT NULL
Pass_nakes	Varchar	32	NOT NULL
Foto	Varchar	10	NOT NULL

Table 14 Structure table Konsultasi

Fields	Type	Size	Information
Id_consult	Int	11	NOT NULL, Primary Key
Id_telusur	Int	11	NOT NULL, Foreign Key tabel telusur
Id_user	Int	11	NOT NULL, Foreign Key tabel user
Id_nakes	Varchar	4	NOT NULL, Foreign Key tabel nakes
Hasil_telusur	Varchar	255	NOT NULL
Pertanyaan	Text		NOT NULL
Tgl_input	Datetime		
dibaca	enum	('Y','T')	

Table 15 Structure table Balas_Konsultasi

Fields	Type	Size	Information
Id_balas	Int	11	NOT NULL, Primary Key
Id_consult	Int	11	NOT NULL, Foreign Key tabel konsultasi
Dari	Enum	('nakes','user')	NOT NULL
tgl_balas	datetime		NOT NULL
jawaban	text		NOT NULL
dibaca	enum('Y','T')		NOT NULL

The table structure is then implemented through the MySQL Data Definition Language (DDL) command. The DDL command is applied in the child behavioral disorder expert system. The DDL commands are shown in table 16:

No	DDL Commands

1	`CREATE database perilaku_anak;`
	`CREATE TABLE 'admin' (`
	`'id_admin' varchar(4) NOT NULL,`
	`'user_admin' varchar(50) NOT NULL,`
	`'pass_admin' varchar(32) NOT NULL`
	`) ENGINE=InnoDB DEFAULT CHARSET=latin1;`

2	`CREATE TABLE 'admin' (`
	`'id_admin' varchar(4) NOT NULL,`
	`'user_admin' varchar(50) NOT NULL,`
	`'pass_admin' varchar(32) NOT NULL`
	`) ENGINE=InnoDB DEFAULT CHARSET=latin1;`

3	`CREATE TABLE 'balas_consult' (`
	`'id_balas' int(11) NOT NULL,`
	`'id_consult' int(11) NOT NULL,`
	`'dari' enum('nakes','user') NOT NULL DEFAULT 'nakes',`
	`'tgl_balas' datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,`
	`'jawaban' text NOT NULL,`
	`'dibaca' enum('Y','T') NOT NULL DEFAULT 'T'`
	`) ENGINE=InnoDB DEFAULT CHARSET=latin1;`

4	`CREATE TABLE 'basis_detail' (`
	`'id_basis' varchar(4) NOT NULL,`
	`'id_gejala' varchar(4) NOT NULL,`
	`'MB' double NOT NULL,`
	`'MD' double NOT NULL`
	`) ENGINE=InnoDB DEFAULT CHARSET=latin1;`

5	`CREATE TABLE 'basis_pengetahuan' (`
	`'id_basis' varchar(4) NOT NULL,`
	`'id_penyakit' varchar(4) NOT NULL`
	`) ENGINE=InnoDB DEFAULT CHARSET=latin1;`

6	`CREATE TABLE 'detail_telusur' (`
	`'id_telusur' int(11) NOT NULL,`
	`'id_gejala' varchar(4) NOT NULL`
	`) ENGINE=InnoDB DEFAULT CHARSET=latin1;`

No	DDL Commands
7	CREATE TABLE 'faskes' ('id_faskes' varchar(4) NOT NULL, 'nama_faskes' varchar(255) NOT NULL, 'jenis_faskes' varchar(50) NOT NULL, 'provinsi_faskes' varchar(50) NOT NULL, 'kota_kab_faskes' varchar(100) NOT NULL, 'alamat_faskes' varchar(255) NOT NULL, 'no_telp' varchar(50) NOT NULL, 'foto_faskes' varchar(10) NOT NULL, 'desk_faskes' text NOT NULL) ENGINE=InnoDB DEFAULT CHARSET=latin1;
8	CREATE TABLE 'gejala' ('id_gejala' varchar(4) NOT NULL, 'nama_gejala' varchar(255) NOT NULL) ENGINE=InnoDB DEFAULT CHARSET=latin1;
9	CREATE TABLE 'info_perilaku' ('id_info' int(11) NOT NULL, 'tgl_info' datetime NOT NULL DEFAULT CURRENT_TIMESTAMP, 'judul_info' varchar(255) NOT NULL, 'info' text NOT NULL, 'gambar' varchar(10) DEFAULT NULL) ENGINE=InnoDB DEFAULT CHARSET=latin1;
10	CREATE TABLE 'nakes' ('id_nakes' varchar(4) NOT NULL, 'nama_nakes' varchar(50) NOT NULL, 'profesi' varchar(255) NOT NULL, 'user_nakes' varchar(50) NOT NULL, 'pass_nakes' varchar(32) NOT NULL, 'foto' varchar(10) NOT NULL) ENGINE=InnoDB DEFAULT CHARSET=latin1;
11	CREATE TABLE 'penyakit' ('id_penyakit' varchar(4) NOT NULL, 'nama_penyakit' varchar(255) NOT NULL, 'penyebab' text NOT NULL, 'tata_laksana' text NOT NULL) ENGINE=InnoDB DEFAULT CHARSET=latin1;

No	DDL Commands
12	CREATE TABLE 'konsultasi' ('id_consult' int(11) NOT NULL, 'id_telusur' int(11) NOT NULL, 'id_user' int(11) NOT NULL, 'id_nakes' varchar(4) NOT NULL, 'hasil_telusur' varchar(255) NOT NULL DEFAULT 'tidak ditemukan penyakit yang cocok dari gejala yang dimasukkan', 'pertanyaan' text NOT NULL, 'tgl_input' datetime NOT NULL DEFAULT CURRENT_TIMESTAMP, 'dibaca' enum('Y','T') NOT NULL DEFAULT 'T') ENGINE=InnoDB DEFAULT CHARSET=latin1;
13	CREATE TABLE 'telusur_gejala' ('id_telusur' int(11) NOT NULL, 'id_user' int(11) NOT NULL, 'tgl_telusur' datetime NOT NULL DEFAULT CURRENT_TIMESTAMP, 'id_penyakit' varchar(4) DEFAULT NULL, 'nilai' double DEFAULT NULL) ENGINE=InnoDB DEFAULT CHARSET=latin1;
14	CREATE TABLE 'user' ('id_user' int(11) NOT NULL, 'nama_user' varchar(50) NOT NULL, 'tgl_lahir' date DEFAULT NULL, 'umur' int(2) DEFAULT NULL, 'jk' enum('L','P') DEFAULT NULL, 'notelp' varchar(15) DEFAULT NULL, 'alamat' varchar(255) DEFAULT NULL, 'username' varchar(50) NOT NULL, 'pass_user' varchar(32) NOT NULL

CONCLUSION

The database design in this study applies the database life cycle, capable of producing a good and systematic database design. Each stage in the database life cycle ensures that the database can meet the needs of child behavioral disorder expert system users, function efficiently, and be easy to manage. Conceptual database design can produce entities that represent the characteristics of the system being built. Logical database design can produce efficient database relation modeling. Physical database design can produce effective table structures applied to DBMS software.

REFERENCES

Anwar, M. R., & Purnama, S. (2022). Boarding House Search Information System Database Design. *International Journal of Cyber and IT Service Management*, *2*(1),

70–81. https://doi.org/10.34306/ijcitsm.v2i1.89

Bahry, F. D. S., Amran, N., Putri, T. E., & Ramli, M. I. (2022). Database design of the Malaysia public figures web archive repository: a social and cultural heritage web collections. *Collection and Curation*, *41*(4), 133-.

Imbulpitiya, A., Whalley, J., & Senapathi, M. (2021). Examining the Exams: Bloom and Database Modelling and Design. *ACE 21*, 21–29.

Samidi, S., & Hidayat, R. (2023). Desain Model Database Mutasi Siswa Dengan Menerapkan Metode Database Life Cycle. *Technomedia Journal*, *8*(2SP), 221–235. https://doi.org/10.33050/tmj.v8i2sp.2063

Septiawati, E., & Sauda, S. (2020). Implementasi Basis Data Terpusat Untuk Penagihan Tunggakan Listrik Pada Cv. Cahaya Abadi. *Jurnal Nasional Ilmu Komputer*, *1*(3), 119–126. https://doi.org/10.47747/jurnalnik.v1i3.158

Setiyadi, D. (2021). Database System Development Life Cycle (DSDLC) on System Libraries for Data Manipulation Language (DML) Using SQL Server 2008. *Jurnal Mantik*, *5*(2), 1065–1066.

Triandini, E. (2022). Pengembangan Database E-commerce De Janggelan Menggunakan Metode Database Life Cycle. *Jurnal Sistem Dan Informatika (JSI)*, *16*(2), 1–11. https://djanggelan.pagora.id.

Wijaya, T. A., Menteng, C., Surya, A., Julianto, A., & Utami, E. (2021). Perancangan Desain Basis Data Sistem Informasi Geografis Tanah Penduduk Dengan Menerapkan Model Data Relasional (Studi Kasus : Desa Tumbang Mantuhe Kabupaten Gunung Mas Provinsi Kalimantan Tengah). *Jurnal Teknologi Informasi: Jurnal Keilmuan Dan Aplikasi Bidang Teknik Informatika*, *15*(1), 72–81. https://doi.org/10.47111/jti.v15i1.1867