The Indonesian Journal of Computer Science
www.ijcs.net
Volume 14, Issue 2, April 2025
https://doi.org/10.33022/ijcs.v14i2.4748

Development of a Backend Architecture for an Online Mental Counselling Platform to Enhance Performance and Security

Misna Asqia[1], Muhammad Al Fajri[2], Yahya Zulkarnain[3], Davied Wahyu Wismanindra[4], Khoirul Umam[5]
misna@nurulfikri.ac.id[1], muha21101ti@student.nurulfikri.ac.id[2], yahya_zulkarnain@staff.gunadarma.ac.id[3], david@nurulfikri.ac.id[4], khoirul.umam@nurulfikri.ac.id[5]
[1] Department of Information System, Sekolah Tinggi Teknologi Terpadu Nurul Fikri, Depok
[2,5] Department of Informatics Engineering, Sekolah Tinggi Teknologi Terpadu Nurul Fikri, Depok
[3] Department of Industrial Engineering, Gunadarma University, Depok
[4] Department of Digital Business, Sekolah Tinggi Teknologi Terpadu Nurul Fikri, Depok

Article Information

Received : 20 Feb 2025
Revised : 7 Mar 2025
Accepted : 15 Apr 2025

Keywords

Mental, Counselling, Online Platform, Backend, VS Code

Abstract

The increasing demand for mental health services during the COVID-19 pandemic emphasizes the importance of developing a web-based psychological counselling platform. This study aims to create a robust backend architecture to support the functional and non-functional requirements of an online mental counselling platform. The backend plays a crucial role in managing business logic, data management, user authentication, counselling session settings, and integration with third-party services. The Research and Development (R&D) methodology was implemented through the steps of requirements analysis, architecture design, module creation, and performance and security testing. The backend was built with a layered architecture approach, ensuring optimal load management, information security, and scalability. The application of authentication features with JSON Web Token (JWT) provides an extra layer of protection for user data. System tests were conducted using Postman through three main scenarios. The initial trial showed a "User does not exist" error (code 400) when user data could not be found. The second test resulted in an "Incorrect password" (code 400) when the password entered was incorrect. The third trial showed that the login was successful with code 200 OK, issuing an access token to the user. These results demonstrate the stability and accuracy of the backend implementation in managing user validation. Research findings include the design of an Entity Relationship Diagram (ERD) for data management as well as the development of backend modules that support CRUD functions. This backend platform improves service efficiency, protects user privacy, and enables wider access, including to remote areas. This study makes a meaningful contribution to the innovation of technology-enabled mental health services, creating opportunities for further development in support of inclusive and sustainable digital psychology services.

A. Introduction

Mental health is a vital component of individual well-being, influencing a person's ability to cope with life stressors, realize their potential, learn effectively, work efficiently, and actively contribute to their community [1]. The global rise in mental health disorders during the COVID-19 pandemic has underscored the urgent need for improved access to mental health services. This situation has led to a surge in demand for mental health counselling, particularly digital-based services, which offer flexibility and accessibility without geographical constraints. Online mental health counselling platforms hold significant potential to address barriers such as time limitations, distance, stigma, and cost, making them an innovative solution for individuals in need of professional support.

Online counselling is a form of interpersonal communication conducted digitally between a counsellor and a client for therapeutic purposes, either in real-time (synchronous) or delayed (asynchronous). This service involves professional interactions where counsellors and clients communicate through electronic devices, such as computers or smartphones, despite being in different locations [2]. Another definition describes online counselling as a remote meeting process between a professional counsellor and a client, utilizing electronic devices, either in real-time (synchronous) or with time delays (asynchronous) [3].

Online counselling offers numerous advantages, including easy access anytime and anywhere, providing comfort for users concerned about social stigma, enabling flexibility in time and location, offering space for deeper reflection, and ensuring anonymity to protect user identities. However, online counselling also has several limitations, such as challenges in maintaining confidentiality, lack of qualifications and understanding of ethical codes among counsellors, and technical issues in video conferencing applications. Additionally, the absence of non-verbal cues may hinder the development of an effective therapeutic relationship [4].

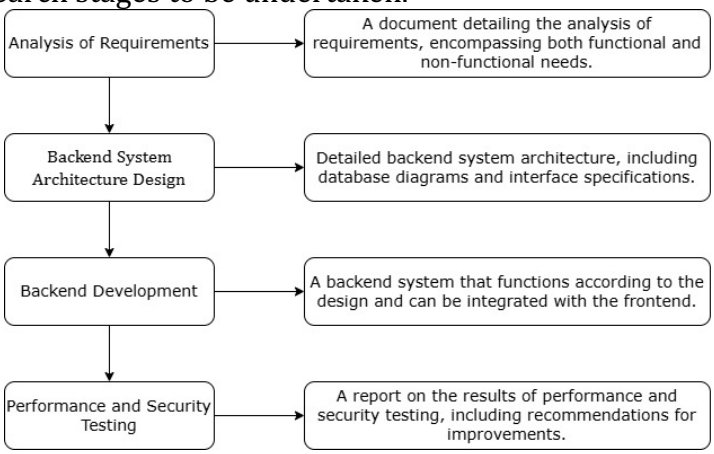
In previous research, web-based mental health counselling platforms have begun with a focus on frontend development [5]. These platforms are designed to provide an intuitive interface and high accessibility, enabling users to interact efficiently with available services [6], [7]. However, the success of such platforms is not solely determined by the user interface but also by the reliability and security of the backend system that supports it. An effective backend is essential to ensure user data is managed securely, counselling processes run smoothly, and services are accessible quickly and stably.

In this context, the development of the platform's backend represents a critical next step. The backend serves as the core that manages business logic, database management, user authentication, counselling session scheduling, and integration with external services such as payment or notification systems. A robust backend not only enhances platform performance and efficiency but also ensures the protection of user data in compliance with security standards and regulations [8]. This study aims to develop a web-based online mental health counselling platform backend capable of supporting various functional and non-functional requirements, such as scalability, access speed, and data security. By developing this backend, the study seeks to provide an improved user experience, support the sustainability of services, and enhance user trust in online mental health counselling platforms.

Thus, the proposed backend development focuses on improving the platform's technical performance and addressing broader impacts on mental health service accessibility, particularly in remote areas underserved by conventional mental health services. This research is expected to significantly contribute to advancing technology-based mental health services by integrating modern technology and innovative approaches.

B. Research Method

This study employs a Research and Development (R&D) approach to develop a web-based online mental health counselling platform backend, focusing on fulfilling functional and non-functional requirements such as scalability, access speed, and data security. The R&D method was selected because this research centres on the process of technological development, with the expected outcome being a practical product or system. The research methodology consists of several stages designed to achieve the research objectives systematically. Figure 1 illustrates the research stages to be undertaken.



Source: Data Processing (2025)

Figure 1. Research Stages

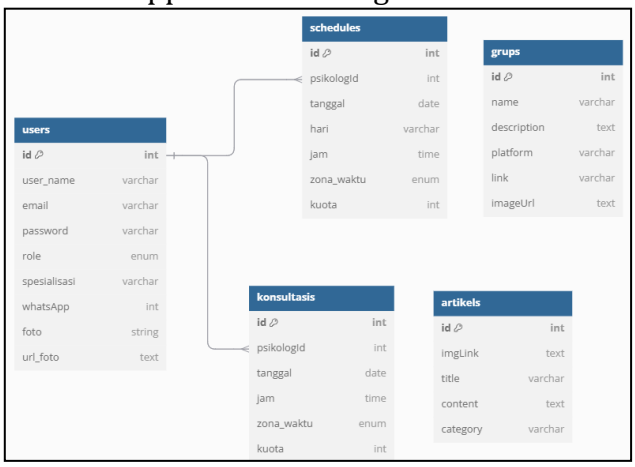
1. Requirement Analysis
   This stage involves the collection and analysis of backend system requirements, a process that is deeply rooted in a user-centric approach. The input from users, psychologists, and other stakeholders is crucial in understanding the functional and non-functional requirements. These requirements will be documented as the foundation for design and development, ensuring that the system is tailored to the needs of its users. This activity also includes the identification of data security requirements and compliance with privacy regulations, further emphasizing our commitment to user safety and privacy [9].
2. Backend System Architecture Design
   Based on the needs analysis results, the backend architecture design will be developed to ensure a scalable, responsive, and secure system. This design includes selecting appropriate technologies and tools, designing the database, and defining API interfaces for integration with the front end. Emphasis is placed on solutions that support horizontal scalability, efficient load management, and user data protection [10].

3. Backend Development
   Backend development is carried out at this stage by implementing the designed features. The development includes creating modules for user management, authentication, session management, and integration with external services. This development focuses on ensuring optimal performance, access speed, and the system's ability to handle high user loads. Good coding practices and continuous unit testing are implemented to ensure code quality [11].
4. Performance and Security Testing
   Performance testing is conducted to evaluate system responsiveness, access time, and the ability to handle a large number of users. Security testing includes penetration testing, vulnerability scanning, and verification of compliance with data security standards. These tests aim to identify and address potential issues before final implementation.

C. Result and Discussion

The backend development process involves retrieving, processing, and delivering data managed by the system to the front end for display to the end user. The backend is also responsible for processing the system's data requirements, including CRUD operations, which encompass creating new data, reading existing data, updating data, and deleting data as needed.

An Entity Relationship Diagram (ERD) depicts the characteristics and connections between entities to illustrate the relationships between tables. The ERD helps visualize how data is organized and how tables are interconnected. In backend development, the CRUD (Create, Read, Update, Delete) method serves as the core functionality implemented to support data management.



Source: Data Processing (2025)

Figure 2. Entity Relationship Diagram (ERD)

In Figure 2, the Entity Relationship Diagram (ERD) illustrates the relationships between the users table and the schedules, consultations, groups, and articles tables. All tables are connected by the same primary key, which is the user ID. The user's table includes a key attribute called role, which determines the user's role within the application [12]. Three types of roles can utilize the application: admin, psychologist, and patient, each with specific access rights and responsibilities.

1. Admin
   The admin has full access rights to the application. Their role includes managing the Groups and Articles tables and overseeing other users, such as patients and psychologists. The admin can create, update, or delete data in the relevant tables to ensure the application's smooth operation.
2. Psychologist
   The psychologist has access rights to manage meeting schedules entered into the Schedules table. These schedules allow patients to request consultations based on the availability determined by the psychologist. Once approved, the meeting schedule is linked to the Consultation table, which records the details of the consultation process.
3. Patient
   The patient can request consultations with psychologists whose schedules have been previously arranged. Information regarding these requests is recorded in the Consultation table, which includes data such as the meeting time, the designated psychologist, and the consultation status.

After the Entity Relationship Diagram (ERD) is established, the next step is to create the backend framework for the web-based mental health consultation platform application. Figure 4 illustrates the backend framework, which outlines the structure and components of the application to be developed. The backend is developed using Visual Studio Code (VS Code), a code editor that facilitates the design of applications, including backend systems. VS Code is developed by Microsoft [13].

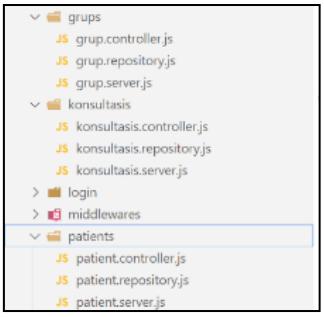
```
generator client {
  provider = "prisma-client-js"
}

datasource db {
  provider = "mysql"
  url      = env("DATABASE_URL")
}

enum Gender {
  male
  female
}

enum Role {
  psychologists
  patients
  admin
}

enum Availability {
  available
  unavailable
}

enum ZonaWaktu {
  WIT
  WITA
  WIB
}

model users {
  id                      Int          @id @default(autoincrement())
  userName                String
  email                   String       @unique
  password                String
  role                    Role         @default(patients)
  spesialisasi            String?
  whatsapp                String?
  foto                    String?
  urlFoto                 String?
  availability            Availability @default(unavailable)
  refresh_token           String?      @db.Text
  konsultasiSebagaiPasien   konsultasis[] @relation("PasienKonsultasis")
  konsultasiSebagaiPsikolog konsultasis[] @relation("PsikologKonsultasis")
  schedules               schedules[]  @relation("UserSchedules")
}
```

```
model schedules {
  id          Int          @id @default(autoincrement())
  user        users        @relation("UserSchedules", fields: [psikologId], references: [id])
  psikologId  Int
  tanggal     DateTime
  hari        String
  jam         DateTime
  zonaWaktu   ZonaWaktu    @default(WIB)
  kuota       Int
  konsultasis konsultasis[] @relation("JadwalKonsultasi") // Relasi ke konsultasi
}

model konsultasis {
  id           Int           @id @default(autoincrement())
  pasien       users         @relation("PasienKonsultasis", fields: [pasienId], references: [id])
  psikolog     users         @relation("PsikologKonsultasis", fields: [psikologId], references: [id])
  pasienId     Int
  psikologId   Int
  keluhan      String        @db.Text
  status       String        @default("Menunggu")
  conversation Conversation[] @relation("KonsultasiConversations")
  jadwal       schedules     @relation("JadwalKonsultasi", fields: [jadwalId], references: [id]) // Relasi ke jadwal
  jadwalId     Int // Foreign key ke jadwal
}

model Conversation {
  id           Int         @id @default(autoincrement())
  konsultasiId Int
  senderId     Int // ID pengirim pesan (bisa psikolog atau pasien)
  message      String // Isi pesan
  timestamp    DateTime    @default(now()) // Waktu pengiriman pesan
  konsultasi   konsultasis @relation("KonsultasiConversations", fields: [konsultasiId], references: [id])
}

model grups {
  id          Int    @id @default(autoincrement())
  name        String
  description String
  platform    String
  link        String
  imageUrl    String
}

model artikels {
  id       Int    @id @default(autoincrement())
  imgLink  String @db.Text
  title    String
  content  String @db.Text
  category String
}
```

Source: Processing with VS Code (2025)

Figure 3. Database Schema

Figure 3 displays the code used to create the database schema, which serves as the foundational framework for data management in the online consultation platform application. This schema is designed to ensure that the database structure is well-organized, enabling it to support various necessary operations, such as data storage, retrieval, updating, and deletion.



Source: Processing with VS Code (2025)

Figure 4. Layered Architecture Backend

Figure 4 illustrates the implementation of a layered architecture in the backend development of the Online Consultation Platform application. The layered architecture employed in this backend development is designed to facilitate easier

development for programmers. Layers are created for groups, consultations, and patients. Each layer consists of controller.js, repository.js, and server.js, with each component serving distinct roles and functions. Controller.js is responsible for delivering the API to be used in index.js. It also handles request and response management, as well as body validation. Repository.js is tasked with database communication. Moreover, Server.js manages the backend's business logic.

```
// register
router.post("/register", async (req, res) => {
  const { userName, email, password } = req.body;

  // Validasi input
  if (!userName || !email || !password) {
    return res.status(400).json({ msg: "All fields are required" });
  }

  // Validasi email
  const isValidEmail = /^[^\s@]+@[^\s@]+\.[^\s@]+$/.test(email);
  if (!isValidEmail) {
    return res.status(400).json({ msg: "Invalid email format" });
  }

  try {
    // Cek apakah email sudah terdaftar
    const existingUser = await prisma.users.findUnique({
      where: { email },
    });
    if (existingUser) {
      return res.status(400).json({ msg: "Email already registered" });
    }

    // Hash password
    // const salt = await bcrypt.genSalt(10); // Tambahkan rounds jika perlu
    const hashPassword = await bcrypt.hash(password, 10);

    // Buat user baru
    await prisma.users.create({
      data: {
        userName,
        email,
        password: hashPassword,
      },
    });

    res.status(201).json({ msg: "Register Success" });
  } catch (error) {
    console.error(error);
    res.status(500).json({ msg: "Something went wrong during registration" });
  }
});

// login
router.post("/login", async (req, res) => {
  try {
    const { email, password } = req.body;

    // Validasi input
    if (!email || !password) {
      return res.status(400).json({ msg: "Email and password are required" });
    }

    // Validasi format email
    const validateEmail = (email) => /^[^\s@]+@[^\s@]+\.[^\s@]+$/.test(email);
    if (!validateEmail(email)) {
      return res.status(400).json({ msg: "Invalid email format" });
    }

    // Cari user berdasarkan email
    const user = await prisma.users.findUnique({
```

```
      where: { email },
    });
    if (!user) {
      return res.status(404).json({ msg: "User does not exist" });
    }

    // Verifikasi password
    const isPasswordValid = await bcrypt.compare(password, user.password);
    if (!isPasswordValid) {
      console.log("Wrong Password");
      return res.status(401).json({ msg: "Wrong Password" });
    }

    // Generate token
    const userId = user.id;
    const userName = user.userName;
    const role = user.role;

    // Access token
    const accessToken = jwt.sign(
      { userId, userName, email, role },
      process.env.ACCESS_TOKEN_SECRET,
      { expiresIn: "2h" }
    );

    // Refresh token
    const refreshToken = jwt.sign(
      { userId, userName, email, role },
      process.env.REFRESH_TOKEN_SECRET,
      { expiresIn: "1d" }
    );

    // Update refresh token di database
    await prisma.users.update({
      where: { id: userId },
      data: { refresh_token: refreshToken },
    });

    // Set cookie untuk refresh token
    res.cookie("refreshToken", refreshToken, {
      httpOnly: true,
      secure: process.env.NODE_ENV === "production",
      maxAge: 24 * 60 * 60 * 1000, // 1 day
    });

    // Kirim response
    res.status(200).json({ accessToken });
  } catch (error) {
    console.error("Error during login:", error);
    res.status(500).json({ msg: "An error occurred during login" });
  }
});
```

Source: Processing with VS Code (2025)

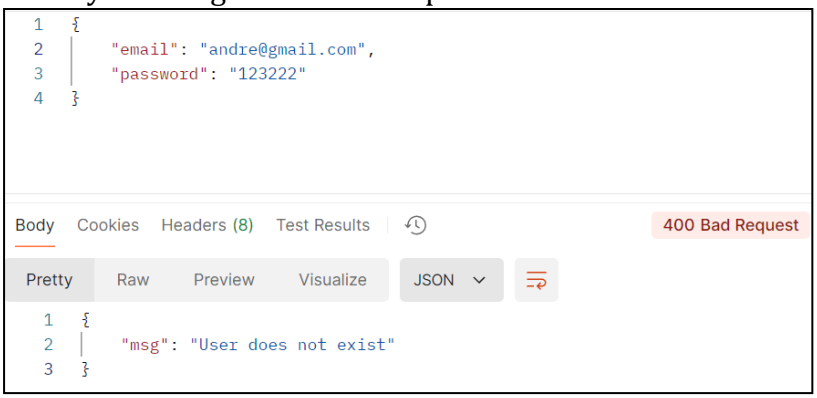
Figure 5. Autentification of Login JSON Web Token (JWT)

Figure 5 illustrates the authentication login process using JSON Web Token (JWT) in the backend of the online mental health consultation platform application. This process involves retrieving several key attributes from the Users table: ID, Username, Email, and Role. These attributes are used to generate a token provided to users upon successful login.

Code Testing Using Postman Tool

Testing using the Postman tool is conducted to evaluate whether the API implementation within the system functions as expected. The testing involves checking whether the user attempting to log in already has an account or not. The

output of this testing is in the form of JSON data along with a status code [14]. The codes used in this testing are standardized, including 200 OK, indicating that the test was successful and 400 Bad Request, indicating an error in the testing process. The testing is performed by entering the email and password credentials.



Source: Processing with VS Code (2025)

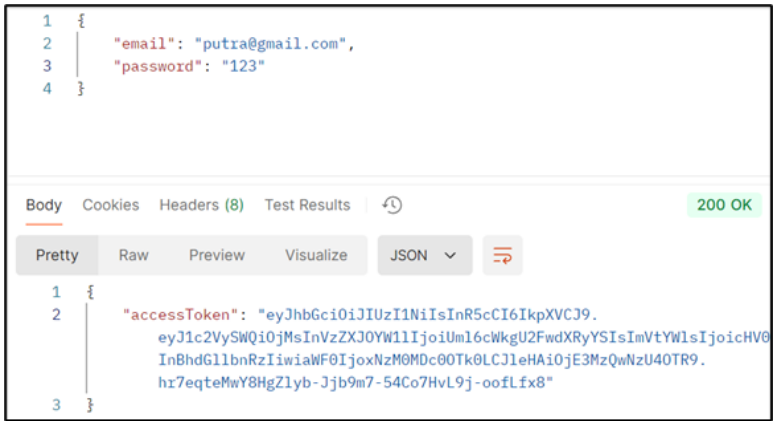
Figure 6. First Postman Testing

In Figure 6, the first test is conducted by entering the email andre@gmail.com and password 123222. After running the test, the result displays the message "User does not exist" with a 400 Bad Request status code. This code indicates that the entered data is not in the database, requiring the user to register first. Subsequently, a second test is performed by entering a different email.

Source: Processing with VS Code (2025)

Figure 7. Second Postman Testing

In Figure 7, the second test is conducted by entering the email putra@gmail.com and password 123222. After running the test, the result displays the "wrong password" message with a 400 Bad Request status code. This code indicates that the entered email exists in the database, but the password input is incorrect or does not match the password stored in the database. Subsequently, a third test is performed by entering a different password.



Source: Processing with VS Code (2025)

Figure 8. Third Postman Testing

In Figure 8, the third test is conducted by entering the email putra@gmail.com and password 123. After running the test, the result displays a 200 OK status code. This code indicates that the entered data exists in the database and that the input password matches the stored password. In the final section, the result includes an access token used to access the web.

D. Conclusion

Based on the discussion that has been conducted regarding the development of a web-based online mental health consultation platform backend using the CRUD (Create, Read, Update, Delete) approach. First, the relationships between tables were established using an ERD consisting of the User, Schedules, Konsultasis, Grups, and Artikels tables. This ERD generated three roles that can use the application based on the created backend, namely the admin, psychologist, and patient roles. After that, the backend development framework was formed using the VS Code application. The database schema and table grouping based on layer architecture were successfully created from VS Code. Once the backend framework was formed, testing was then conducted using the Postman tool. The testing results showed that users cannot access the application if they enter an incorrect email and password. However, if the user can enter the correct email and password, they can access the application. This is in accordance with the security protocol implemented in the backend, with a 400 code if the user enters an incorrect email and/or password and a 200 code if the user can enter the correct email and password into the application.

E. Acknowledgement

Appreciation is extended to all individuals and institutions who contributed to this research. Special thanks are given for the invaluable support and collaboration, which have greatly facilitated the successful completion of this study in accordance with its intended objectives.

F. References

[1] F. Rahmawaty, R. P. Silalahiy, Berthiana, and B. Mansyah, “Faktor-faktor yang Mempengaruhi Kesehatan Mental pada Remaja,” *Jurnal Surya Medika (JSM)*, no. 8, pp. 276–281, 2022.

[2] S. G. Zeren, S. M. Erus, Y. Amanvermez, A. B. Genc, M. B. Yilmaz, and B. Duy, "The effectiveness of online counseling for university students in Turkey: A non-randomized controlled trial," *European Journal of Educational Research*, vol. 9, no. 2, pp. 825–834, Apr. 2020, doi: 10.12973/eu-jer.9.2.825.

[3] P. M. Amos, P. K. A. Bedu-Addo, and T. Antwi, "Experiences of Online Counseling Among Undergraduates in Some Ghanaian Universities," *Sage Open*, vol. 10, no. 3, Jul. 2020, doi: 10.1177/2158244020941844.

[4] N. M. Harahap, "Konseling Online sebagai Solusi di Masa Pandemi Covid 19," *Jurnal Bimbingan Konseling Islam*, vol. 3, no. 1, pp. 51–64, May 2021, [Online]. Available: http://jurnal.iain-padangsidimpuan.ac.id/index.php/Irsyad

[5] M. N. R. Saputra, "Rancang Bangun Aplikasi Konsultasi Mental Online Berbasis Website dengan Implementasi Frontend Menggunakan ReactJS," Depok, Aug. 2024.

[6] Nabilla and A. Ichwani, "Sistem Informasi Layanan e-Konseling Psikologi Untuk Mahasiswa Berbasis Website dengan Menggunakan Prototype," *Jurnal MNEMONIC*, vol. 5, no. 2, pp. 191–198, Sep. 2022.

[7] I. Nawali and B. R. Suteja, "Pembuatan Sistem Aplikasi Berbasis Website Konsiltasi Orang Tua dengan Psikolog untuk Kesehatan Mental Anak," *Jurnal Strategi*, pp. 110–9, May 2023.

[8] B. R. Faturizky and Ri. Komalasari, "Sistem Informasi Layanan Konsultasi Kesehatan Mental Berbasis Website," *Jurnal Teknologi Sistem Informasi (JTSI)*, 2024.

[9] A. A. Iriarte, G. L. Erle, and M. M. Etxabe, "Evaluating User Experience With Physiological Monitoring: A Systematic Literature Review," *DYNA NEW TECHNOLOGIES*, vol. 8, no. 1, pp. 1–18, 2021, doi: 10.6036/nt10072.

[10] W. S. Lestari and M. Ulina, "Pengembangan sistem informasi berbasis web pada pesantren Tahfizh Daarul Mafaza," *SELAPARANG: Jurnal Pengabdian Masyarakat Berkemajuan*, vol. 8, no. 2, pp. 1052–1061, 2024.

[11] M. Satya Nugraha, K. Candra Brata, and A. Hendra Brata, "Pembangunan Aplikasi Perangkat Bergerak Konseling Online pada Anxiety Disorder berbasis Android menggunakan Metode Personal Extreme Programming," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 5, no. 4, pp. 1370–1379, 2021, [Online]. Available: http://j-ptiik.ub.ac.id

[12] S. Nugraha, A. B. Prasetijo, and D. Eridani, "Perancangan Back-End Aplikasi Reservasi Talanoa Kopi and Space Menggunakan Framework Express.js Back End Design of the Talanoa Kopi and Space Reservation Application Using Express.js Framework," *Jurnal Teknik Komputer*, vol. 1, no. 3, pp. 126–131, 2022, doi: 10.14710/jtk.v1i3.36901.

[13] D. H. P. Harahap and N. Nasir, "Backend CRUD Aplikasi Website E-Commerce," *Jurnal Ilmiah Teknik Informatika (TEKINFO)*, vol. 24, no. 2, pp. 21–28, 2023.

[14] H. Setiawan, D. Yusuf, and G. H. Wibowo, "Penerapan Extreme Programming pada Pengembangan Backend Sistem Informasi e-YM," *Journal Zetroem*, vol. 6, no. 2, pp. 45–50, 2024.