

PENERAPAN FLOW-BASED CHATBOT MENGENAI SKPM DI LINGKUNGAN UPN “VETERAN” JAWA TIMUR

Ivan Faiz Daniar^{1*}, Amalia Anjani Arifiyanti², Anindo Saka Fitri³

^{1,2,3} Universitas Pembangunan Nasional "Veteran" Jawa Timur, Indonesia

E-mail: 19082010054@student.upnjatim.ac.id

Abstract: The purpose of this study is to develop a WhatsApp-based Chatbot system that can answer questions related to SKPM, utilizing a platform that is familiar to many students. The methodology used in developing this system is a waterfall, which consists of the stages of Communication, Planning, Modeling, Construction, and Deployment. Starting with interviews and observations around SKPM. Then identifying the needs of the Chatbot system to be created, then creating a design for the Architecture, Flowchart, Usecase, Activity, Sequence, and Interface of the Chatbot system. System testing is carried out using the Blackbox testing method, while the measurement of system usability uses the Chatbot Usability Questionnaire (CUQ). The results of the CUQ test show that the Chatbot system successfully manages various interaction scenarios with users well, with a CUQ score of 74%. This finding confirms that the application of Chatbot as a means of communication in an academic context can increase efficiency and accelerate the dissemination of relevant information to students.

Keywords: Chatbot, Flow-Based, WhatsApp, Student Credit Points, UPNVJT

Abstrak: Tujuan penelitian ini mengembangkan sebuah sistem Chatbot berbasis WhatsApp yang dapat menjawab pertanyaan terkait SKPM, memanfaatkan platform yang sudah familiar bagi banyak mahasiswa. Metodologi yang digunakan dalam pengembangan sistem ini adalah waterfall, yang terdiri dari tahapan *Communication, Planning, Modelling, Construction, dan Deployment*. Diawali dengan wawancara dan observasi seputar SKPM. Kemudian mengidentifikasi kebutuhan sistem Chatbot yang akan dibuat, lalu membuat rancangan Arsitektur, Flowchart, Usecase, Activity, Sequence dan Antarmuka sistem Chatbot. Pengujian sistem dilakukan melalui metode *Blackbox testing*, sementara pengukuran *usability* sistem menggunakan *Chatbot Usability Questionnaire* (CUQ). Hasil pengujian CUQ menunjukkan bahwa sistem Chatbot berhasil mengelola berbagai skenario interaksi dengan pengguna dengan baik, dengan skor CUQ yang diperoleh adalah 74%. Temuan ini menegaskan bahwa penerapan Chatbot sebagai sarana komunikasi dalam konteks akademis dapat meningkatkan efisiensi dan mempercepat penyebaran informasi yang relevan kepada mahasiswa.

Kata kunci : Chatbot, Flow-Based, WhatsApp, Kredit Poin Mahasiswa, UPNVJT

Copyright (c) 2024 The Authors. This is an open-access article under the CC BY-SA 4.0 license (<https://creativecommons.org/licenses/by-sa/4.0/>)

PENDAHULUAN

Dalam dunia pendidikan, perguruan tinggi merupakan organisasi yang memberikan pelayanan kepada masyarakat umum (Fauzan Wakila, 2021; Sukmawati et al., 2023). Layanan yang ditawarkan di kampus meliputi layanan pendidikan, skripsi dan pengabdian masyarakat. UPN “Veteran” Jawa Timur atau UPNVJT merupakan salah satu organisasi yang menyediakan layanan-layanan tersebut. Dalam layanan

pendidikan, UPNVJT memiliki banyak program untuk membantu meningkatkan kualitas *hardskill* maupun *softskill* dari mahasiswanya, salah satu program tersebut adalah Satuan Kredit Poin Mahasiswa atau dikenal dengan SKPM.

Belum semua mahasiswa memahami pengertian dan tujuan dari adanya program SPKM. Banyak mahasiswa yang memilih untuk bertemu secara langsung dengan dosen wali dan menghadapi kendala akibat sulit mencari waktu bertemu. Hal ini menyebabkan timbulnya ketidakefektifan penyampaian informasi atau bahkan informasi tersebut belum tentu dipahami oleh mahasiswa dengan baik. Oleh karenanya, dibutuhkan penyediaan informasi mengenai program SKPM tersebut agar mahasiswa mendapatkan informasi yang mereka butuhkan serta informasi bisa tersampaikan dengan jelas dan efektif.

Hasil penelitian dari Cucus et al. (2019) diketahui bahwa permasalahan mengenai konsultasi akademik di lingkungan perguruan tinggi dapat menggunakan aplikasi Chatbot sebagai solusinya. Hal ini serupa dikemukakan Ramdani (2018), dimana pengembangan Chatbot mampu memberikan pelayanan kepada mahasiswa terkait dengan permintaan pengurusan surat. Sedangkan diketahui sebuah penelitian yang dihasilkan oleh Meltwater (2023) menyatakan bahwa dapat diketahui bahwa *platform* chat yang paling banyak digunakan oleh pengguna berusia 16 hingga 64 tahun merupakan WhatsApp. Persentase penggunaan WhatsApp berjumlah sebesar 92.1% dan dapat terbilang *platform* yang paling populer di Indonesia. Berbeda dengan hasil penelitian lainnya yang dinyatakan bahwa Mangain (2023) menyatakan bahwa WhatsApp API juga mudah untuk mengintegrasikan integrasi pihak ketiga. WhatsApp API dapat mengintegrasikan percakapan WhatsApp dan mengelola semua komunikasi pelanggan, permintaan support, dan tiket dari satu tempat. Sedangkan hasil kajian penelitian lainnya menyatakan bahwa WhatsApp API menyederhanakan tugas memberikan dukungan pengguna dan mengurangi friksi dalam menyampaikan informasi di seluruh saluran (Anjasmara et al., 2024; Facebook, 2020; Rizki et al., 2022). Penelitian ini berbeda dengan penelitian yang pernah dilakukan lainnya, pada penelitian ini memiliki tujuan penelitian ini untuk mengembangkan *Flow-Based* Chatbot sistem yang menjawab secara otomatis berbasis *WhatsApp* Chatbot mengenai SKPM di lingkungan UPNVJT.

METODE

Metode pengembangan sistem yang digunakan adalah metode *waterfall* dengan menggunakan *Object Oriented Programming* (OOP) dan teknik pemodelan *Unified Modeling Language* (UML) (Abdillah, 2018). Model *waterfall* sering disebut juga dengan “*classic life cycle*” (Wahid, 2020). Model ini melakukan pendekatan secara sistematis dan berurutan. Disebut dengan *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan. Menurut Redaksi Jagoan Hosting (2023), OOP mempermudah pengembangan perangkat lunak dengan menjadikan kode kompleks menjadi lebih mudah di gunakan, di baca, dan secara umum lebih baik. Pada bagian *communication* dalam pembuatan Chatbot ini dilakukan wawancara dan studi literatur sebagai modal awal pembuatan Chatbot ini. Admin membutuhkan admin *dashboard* untuk manajemen chatbot yang memungkinkan CRUD (*Create, Read, Update, Delete*) pertanyaan dan jawaban tanpa membuka *source code*, menggunakan *jsTree*. Perangkat keras yang digunakan adalah laptop dengan spesifikasi prosesor Intel(R) Core i7 Gen 9, RAM 16 GB, SSD 512 GB, Harddisk 1 TB, serta smartphone Android versi 13 dengan aplikasi *WhatsApp*. Perangkat lunak yang digunakan meliputi web browser, editor script PHP, Web Server Apache, bahasa pemrograman PHP, bahasa pemrograman *Java Script, Bootstrap*, dan *jsTree*.

HASIL DAN PEMBAHASAN

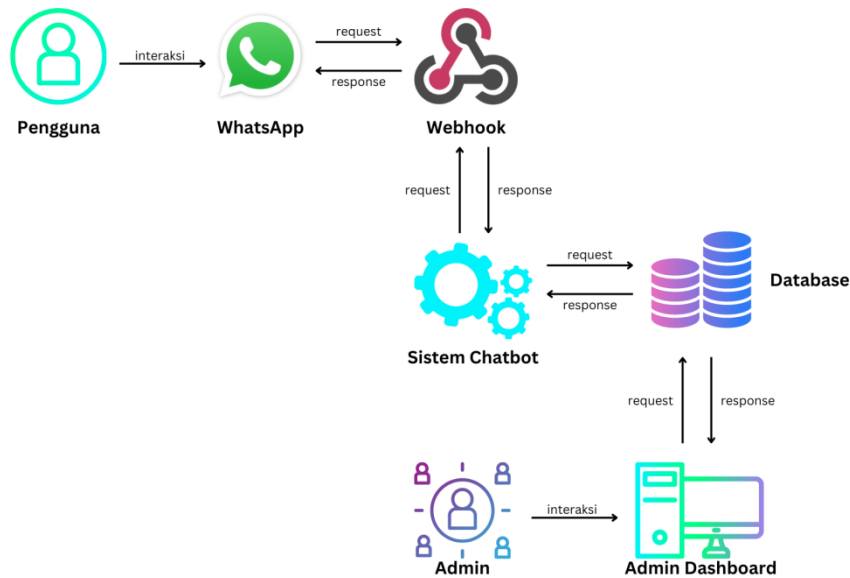
Tahap *communication* merupakan tahap yang berisi langkah-langkah untuk mendapatkan informasi atau pengumpulan data sebagai modal awal dari pembuatan suatu aplikasi (Chang et al., 2023; Ma & Ho, 2018; Villanueva & Palaoag, 2020). Metode pengumpulan data yang dilakukan yakni berupa wawancara, studi literatur dan observasi. *Hardware* yang digunakan dalam pengembangan Chatbot ini adalah laptop dengan spesifikasi prosesor Intel(R) Core i7 Gen 9, RAM 16 GB, SSD 512 GB, *harddisk* 1 TB dan *smartphone* Android versi 13 yang terpasang aplikasi *WhatsApp*. Sedangkan, *software* yang akan digunakan meliputi *web browser, text editor, Meta Developer, MySQL*, dan bahasa pemrograman php dan Javascript.

Planning merupakan tahap penting dalam metodologi *Waterfall* yang telah diadaptasi dari model yang dikembangkan oleh Pressman. Tahap ini berfokus pada pengorganisasian dan detail pengembangan sistem Chatbot berbasis aturan dan admin *dashboard* (Corrêa et al., 2023; Vinarti et al., 2024). Fokus utama dari perencanaan

adalah memastikan bahwa semua komponen sistem dirancang untuk memenuhi kebutuhan pengguna dengan efektif (Beilharz et al., 2021; Mamgain, 2023).

Melalui perencanaan yang terstruktur ini, proyek Chatbot diharapkan dapat menghasilkan alat yang tidak hanya efektif dan efisien dalam mengelola interaksi dengan pengguna, tetapi juga mudah dikelola oleh administrator sistem. Tahap ini menetapkan dasar yang kuat untuk langkah-langkah berikutnya dalam proses pengembangan, yaitu *modelling*, *construction*, dan *deployment*.

Modelling diperlukan agar pembuatan sistem dapat memenuhi tujuan dibuatnya sistem tersebut. *Modelling* dilakukan dengan meliputi perancangan alur interaksi Chatbot dan perancangan admin *dashboard* (Karim et al., 2023; Ma & Ho, 2018).



Gambar 1. Arsitektur Sistem Chatbot

Berdasarkan arsitektur sistem pada gambar 1, terdapat hubungan antar komponen yang dibutuhkan dalam pembuatan Chatbot ini. Komponen-komponen dalam arsitektur sistem tersebut adalah sistem Chatbot yang akan dibuat, *Webhook*, *Database*, *platform messenger* yaitu *WhatsApp*, dan *Admin Dashboard*. Data-data yang diperlukan dalam Chatbot akan disimpan dan diolah pada database yang dapat diakses secara *online*. *Webhook* merupakan layanan yang sudah tersedia di *WhatsApp Business API* yang dirancang sebagai layanan *callback*. Berdasar skema diatas, terlihat bahwa *Webhook* menjadi penghubung antara *WhatsApp Business API* dan sistem Chatbot. Pengelolaan admin dapat dilakukan melalui admin *dashboard*.

Pengguna akan melakukan interaksi dengan *WhatsApp*, sedangkan pesan pengguna akan diolah pada sistem Chatbot. Interaksi antara *WhatsApp Business API* dan sistem Chatbot akan ditangani oleh *Webhook*. Pesan yang terkirim akan diidentifikasi oleh sistem Chatbot dan sistem Chatbot akan mengirim respon ke *database* untuk mendapatkan jawaban dari pesan yang telah teridentifikasi.

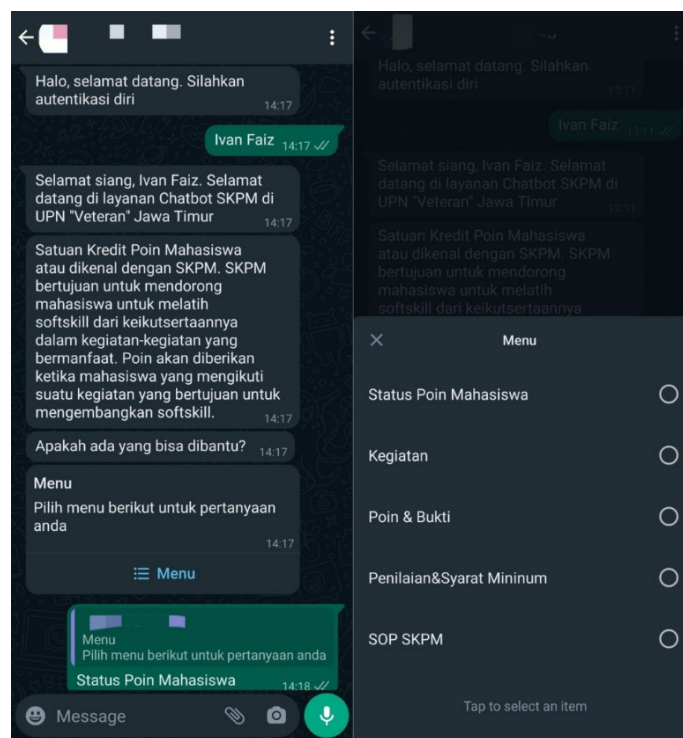
mahasiswa			data			user		
id_mahasiswa	<pi>	Integer <M>	id_data	<pi>	Integer <M>	ID_USER	<pi>	Integer <M>
nama		Variable characters (50)	parent		Integer	USERNAME		Variable characters (100)
npm		Variable characters (20)	type		Variable characters (20)	PASSWORD		Variable characters (100)
skpm		Integer	header		Variable characters (50)	FULLNAME		Variable characters (100)
Identifier_1	<pi>		body		Text	Identifier_1	<pi>	

Gambar 2. Rancangan Basis Data

Gambar 2 memperlihatkan rancangan basis data yang akan digunakan untuk menyimpan data-data yang dibutuhkan untuk menjalankan Chatbot dan Admin *Dashboard*.

Antarmuka

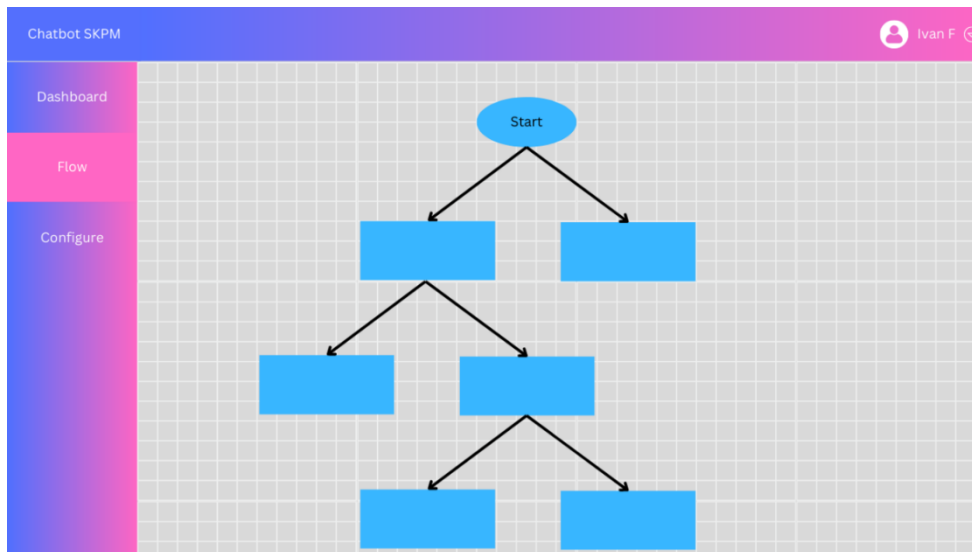
Antarmuka Pengguna



Gambar 3. *Mock-up* Antarmuka Pengguna

Pada rancangan antarmuka pengguna pada gambar 3, pengguna akan mengawali percakapan yang akan dilanjutkan dengan Chatbot meminta pengguna untuk melakukan autentikasi. Setelah autentikasi berhasil, Chatbot menampilkan salam pembuka yang dilanjutkan dengan menu yang tersedia.

Antarmuka Admin *Dashboard*



Gambar 4. *Mock-up* Antarmuka Admin Dashboard

Pada antarmuka admin *dashboard*, terdapat tabel dimana admin dapat melihat, menambahkan, menghapus, dan menyesuaikan pertanyaan atau jawaban Chatbot (CRUD).

Construction

Selanjutnya akan dilakukan *construction* sistem Chatbot WhatsApp yang dilakukan menggunakan Javascript, sedangkan admin *dashboard* akan dibuat menggunakan php dan jsTree untuk penggunaan CRUD yang lebih nyaman dan *database* yang digunakan adalah MySQL.

Coding

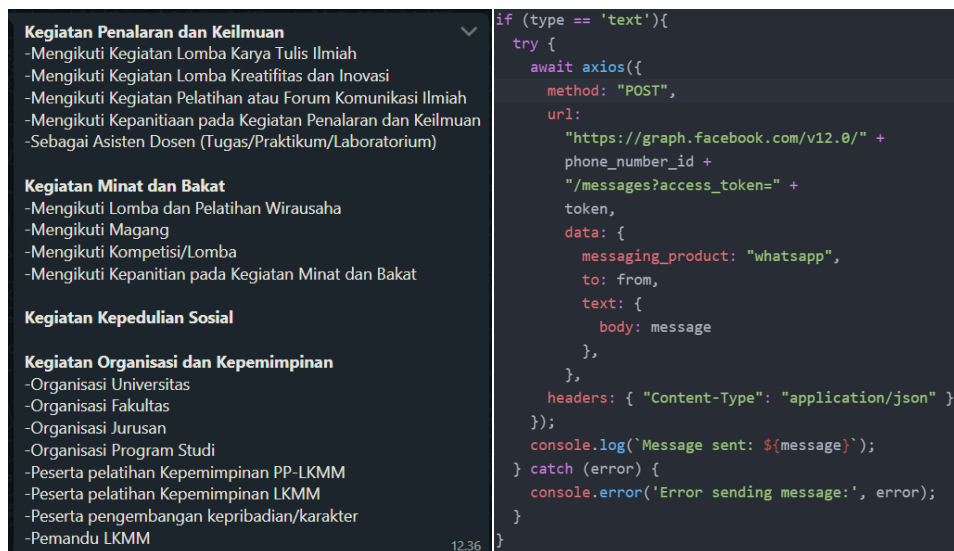
Chatbot

Kodingan utama di dalam Chatbot penelitian dibagi menjadi 4 untuk mengirim pertanyaan maupun jawaban kepada pengguna yakni, *Text*, *Interactive*, *Row* dan *Interactive_back*. Kodingan ini nantinya akan dibungkus di dalam permintaan *Axios* dan mengirimkan pesan atau jawaban ke *WhatsApp* melalui *WhatsApp Business API* Facebook.

```
axios({
  method: "POST",
  url:
    "https://graph.facebook.com/v12.0/" +
    phone_number_id +
    "/messages?access_token=" +
    token,
  data: {
    messaging_product: "whatsapp",
    to: from,
    text: {
      body: message
    },
  },
  headers: { "Content-Type": "application/json" },
});
```

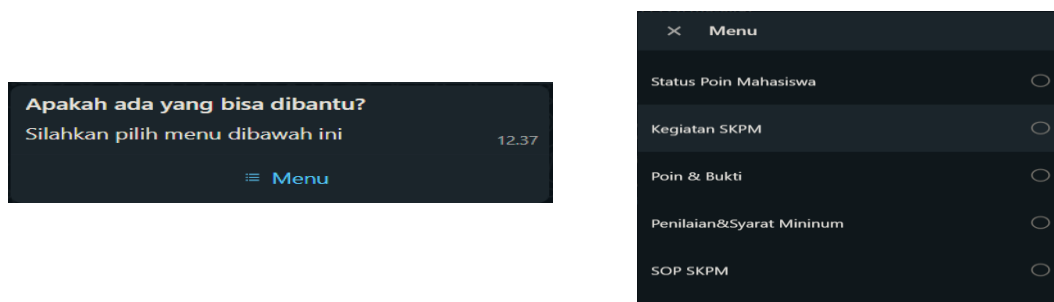
Gambar 5. Kodingan Axios

Dimulai dengan *Text*, *Text* hanya digunakan untuk menjawab pertanyaan karena hanya bisa mengandung teks saja. Berikut adalah contoh teks dan kodingannya



Gambar 6. Contoh *Text* dalam WhatsApp dan Kodingannya

Kemudian untuk *Interactive* dan *Row*, kedua kodingan membentuk menu pertanyaan. Berikut adalah contoh gambar dimana *Interactive* dan *Row* berkerja.



Gambar 7. Gambar *Interactive* dan *Row*

Gambar 7 dari kiri menunjukkan bahwa *Interactive* adalah sebagai tombol yang dapat dipencet untuk memunculkan pilihan *Row* atau menu pertanyaan. Berikut adalah koding dari *Interactive* dan *Row*.

```
else if (type == 'interactive') {
  async function sendFacebookMessage() {
    try {
      const formattedData = []; // Definisikan formattedData

      // Ambil data
      const row_results = await new Promise((resolve, reject) => {
        connection.query('SELECT id, parent, body FROM data WHERE type = "row" AND parent = ?', [id_title], (error, result) => {
          if (error) {
            console.error('Error executing the query:', error);
            reject(error);
          } else {
            resolve(result);
          }
        });
      });

      // Dimasukkan ke array
      let formattedRows = row_results.map(row => ({
        id: row.id,
        title: row.body
      }));
    }
  }
}
```

Gambar 8. Kodingan *Interactive* dan *Row* (1)

```
// Cek and proses tambahan data jika dibutuhkan
if (row_results.length > 0 && row_results[0].parent !== 1) {
  const additionalResult = await new Promise((resolve, reject) => {
    connection.query('SELECT parent FROM data WHERE id = ?', [id_title], (err, results) => {
      if (err) {
        console.error('Error executing the query:', err);
        reject(err);
      } else {
        resolve(results[0].parent);
      }
    });
  });

  if (additionalResult === 5 || additionalResult === 1) {
    // formattedRows.push({ id: additionalResult, title: 'Menu Sebelumnya' });
  } else {
    formattedRows.push({ id: additionalResult, title: 'Menu Sebelumnya' });
    formattedRows.push({ id: 5, title: 'Menu Utama' });
  }
}

formattedData.push(...formattedRows); // Masukkan semua rows ke formattedData
// Lognya formatted data
console.log("All rows fetched: ", formattedData);
```

Gambar 9. Kodingan *Interactive* dan *Row* (2)

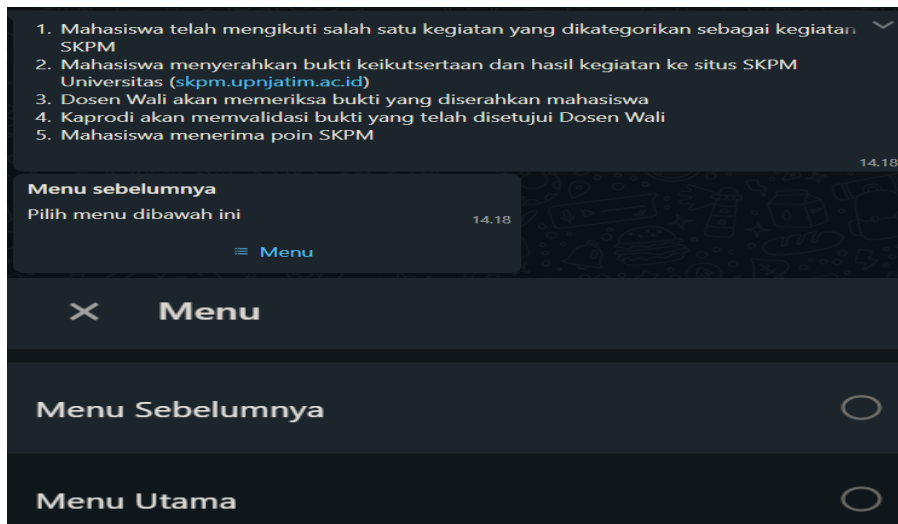

```
// Pisahkan formattedData menjadi batch 10 rows dan kirimkan setiap batch
const batchSize = 10;
for (let i = 0; i < formattedData.length; i += batchSize) {
  const batch = formattedData.slice(i, i + batchSize);

  // Make the Axios POST request for each batch
  await axios({
    method: "POST",
    url: "https://graph.facebook.com/v12.0/" + phone_number_id + "/messages?access_token=" + token,
    data: {
      messaging_product: "whatsapp",
      to: from,
      type: "interactive",
      interactive: {
        type: "list",
        header: {
          type: "text",
          text: header
        },
        body: {
          type: "text",
          text: message
        },
        action: {
          button: "Menu",
          sections: [{
            title: "Pilih menu dibawah ini",
            rows: batch // Use the current batch of rows here
          }]
        }
      }
    }
  });
}
```

Gambar 10. Kodingan *Interactive* dan *Row* (3)

Dalam gambar 8-10 dapat dilihat bahwa jika *Interactive* terpilih maka akan mengambil *Row* dan menggabungkannya menjadi satu pesan yang dikirim dengan *Axios* ke WhatsApp.

Kemudian yang terakhir ada *Interactive_back*, *Interactive_back* hanya dipakai pada saat pesan terakhir adalah *Text* untuk kembali ke menu sebelumnya atau kembali ke menu utama. Berikut contoh gambar *Interactive_back*.



Gambar 11. Contoh tampilan *Interactive_back*

Seperti pada gambar 11, *Interactive_back* mendampingi *Text* untuk kembali ke menu sebelumnya. Berikut adalah kode untuk *Interactive_back*.

```
else if (type === 'interactive_back') {
  async function sendFacebookMessage() {
    try {

      const additionalResult = await new Promise((resolve, reject) => {
        connection.query('SELECT parent FROM data WHERE id = ?', [id_title], (err, results) => {
          if (err) {
            console.error('Error executing the query:', err);
            reject(err);
          } else {
            resolve(results[0]?.parent || null);
          }
        });
      });

      const formattedData = [];

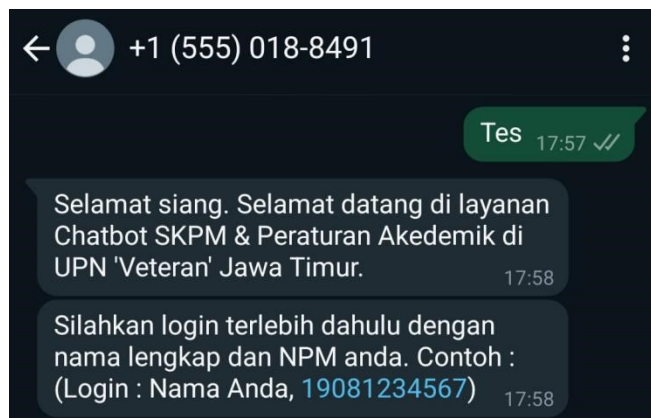
      if (additionalResult === 5 || additionalResult === 1) {
        formattedData.push({ id: additionalResult, title: 'Menu Sebelumnya' });
      } else if (additionalResult !== null) {
        formattedData.push({ id: additionalResult, title: 'Menu Sebelumnya' });
        formattedData.push({ id: 5, title: 'Menu Utama' });
      }

      console.log("All rows fetched: ", formattedData);
    }

    await axios({
      method: "POST",
      url: "https://graph.facebook.com/v12.0/" + phone_number_id + "/messages?access_token=" + token,
      data: {
        messaging_product: "whatsapp",
        to: from,
        type: "interactive",
        "interactive": {
          "type": "list",
          "header": {
            "type": "text",
            "text": header
          },
          "body": {
            "text": message
          },
          "action": {
            "button": "Menu",
            "sections": [{
              "title": "Pilih menu dibawah ini",
              rows: formattedData
            }]
          }
        }
      },
      headers: { "Content-Type": "application/json" },
    });
  }
}
```

Gambar 12. Kode *Interactive_back*

Salam Pengguna dalam Chatbot



Gambar 13. Salam Pengguna Chatbot

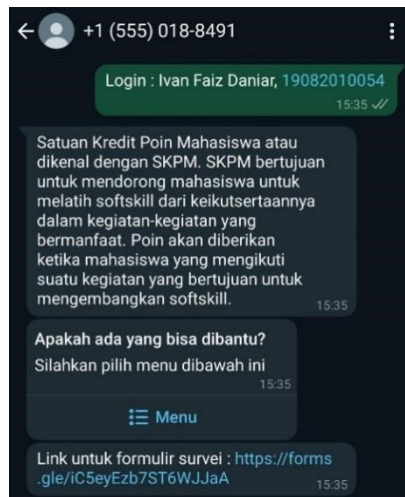
Gambar 13 memperlihatkan tampilan *WhatsApp* dimana sistem Chatbot akan berfungsi. Pengguna mengawali percakapan yang akan dilanjutkan dengan Chatbot

meminta pengguna untuk melakukan autentikasi. Pada gambar 13 diperlihatkan kode untuk salam pengguna Chatbot

```
default_pembukaan()
.then() => {
  console.log('Query results:', queryResult);
  if (Array.isArray(queryResult) && queryResult.length > 0) {
    for (const result of queryResult) {
      if (result && result.body) {
        const body = result.body;
        console.log('Query body result:', body);
      } else {
        console.log('Invalid query result format:', result);
      }
    }
  }
  async function sendMessages() {
    for (const messageObj of queryResult) {
      const message = messageObj.body;
      try {
        await axios({
          method: "POST",
          url:
            "https://graph.facebook.com/v12.0/" +
            phone_number_id +
            "/messages?access_token=" +
            token,
          data: {
            messaging_product: "whatsapp",
            to: from,
            text: {
              body: message
            },
          },
          headers: { "Content-Type": "application/json" },
        });
        console.log("Message sent: ${message}");
      } catch (error) {
        console.error("Error sending message:", error);
      }
    }
  }
  sendMessages();
} else {
  console.log('No query results or invalid result format.');
```

Gambar 14 Kode Salam Pengguna Chatbot

Autentikasi pengguna dalam Chatbot



Gambar 15 Autentikasi Pengguna

Gambar 15 memperlihatkan tampilan *WhatsApp* dimana sistem Chatbot yang berfungsi. Setelah autentikasi berhasil, Chatbot menampilkan salam pembuka yang dilanjutkan dengan menu yang tersedia. Berikut pada gambar 4.27 adalah kode untuk memperlihatkan tampilan setelah autentikasi berhasil.

```
//Login Menu
const default_login = () => {
  return new Promise((resolve, reject) => {
    connection.query(query_login, (err, results) => {
      if (err) {
        console.error('Error executing the query: ' + err);
        reject(err);
      } else {
        resolve(results);
      }
    });
  });
};

default_login()
  .then(() => {
    // 'queryResult' is now defined with the query results
    console.log('Query results:', queryResult);

    // Ensure queryResult is an array and has data
    if (Array.isArray(queryResult) && queryResult.length > 0) {
      for (const result of queryResult) {
        if (result && result.body) {
          const body = result.body;
          // console.log('Query body result:', body);
        } else {
          console.log('Invalid query result format:', result);
        }
      }
    }

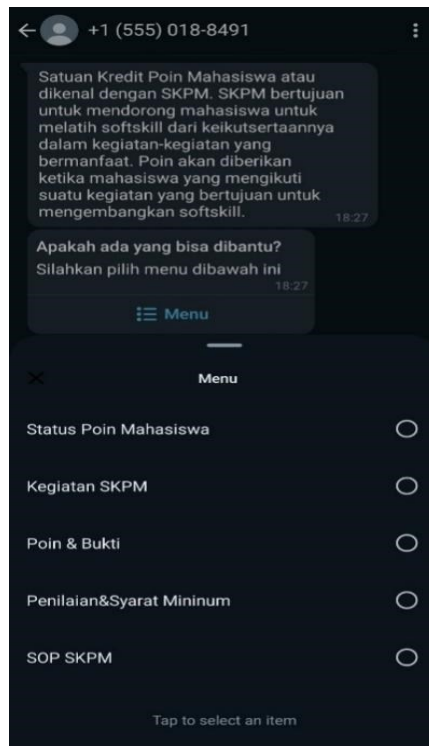
    async function sendMessages() {
      for (const messageObj of queryResult) {
        const id = messageObj.id;
        const parent = messageObj.parent;
        const type = messageObj.type;
        const header = messageObj.header;
        const message = messageObj.body;

        if (type == 'text'){=}
        else if (type == 'interactive') {=}
        else {
          console.log('Theres no determinable message type in this cell');
        }
      }
    }

    sendMessages();
  });
};
```

Gambar 16. Kode Autentikasi Pengguna

Pemilihan Menu Pertanyaan



Gambar 17. Tampilan Menu Pertanyaan

Gambar 17 memperlihatkan tampilan *WhatsApp* dimana sistem Chatbot menampilkan menu yang tersedia. Pada gambar 19 adalah kode untuk membuat menu *Interactive* dan *Row* untuk membuat seperti tampilan pada gambar 18

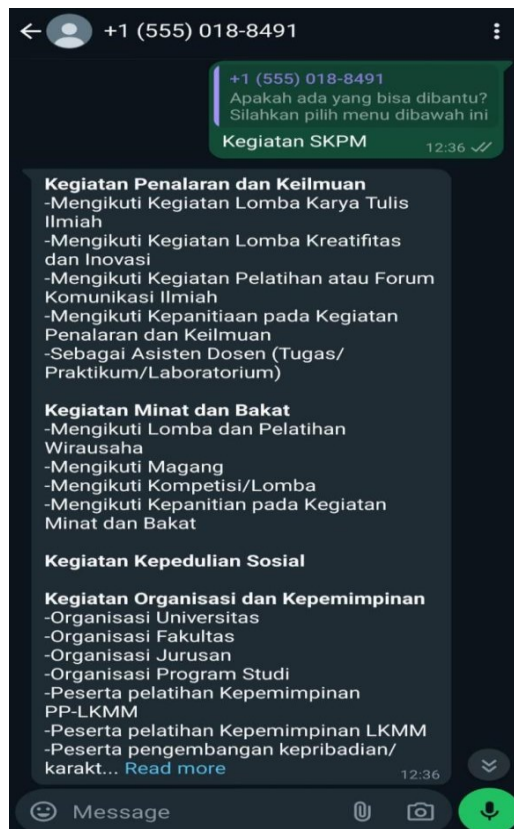
```
else if (type == 'interactive') {
  async function sendFacebookMessage() {
    try {
      const formattedData = []; // Definisikan formattedData

      // Ambil data
      const row_results = await new Promise((resolve, reject) => {
        connection.query('SELECT id, parent, body FROM data WHERE type = "row" AND parent = ?', [id_title], (error, result) => {
          if (error) {
            console.error('Error executing the query:', error);
            reject(error);
          } else {
            resolve(result);
          }
        });
      });

      // Dimasukkan ke array
      let formattedRows = row_results.map(row => ({
        id: row.id,
        title: row.body
      }));
    }
  }
}
```

Gambar 18. Kode Menu Pertanyaan

Contoh Pemilihan Pertanyaan



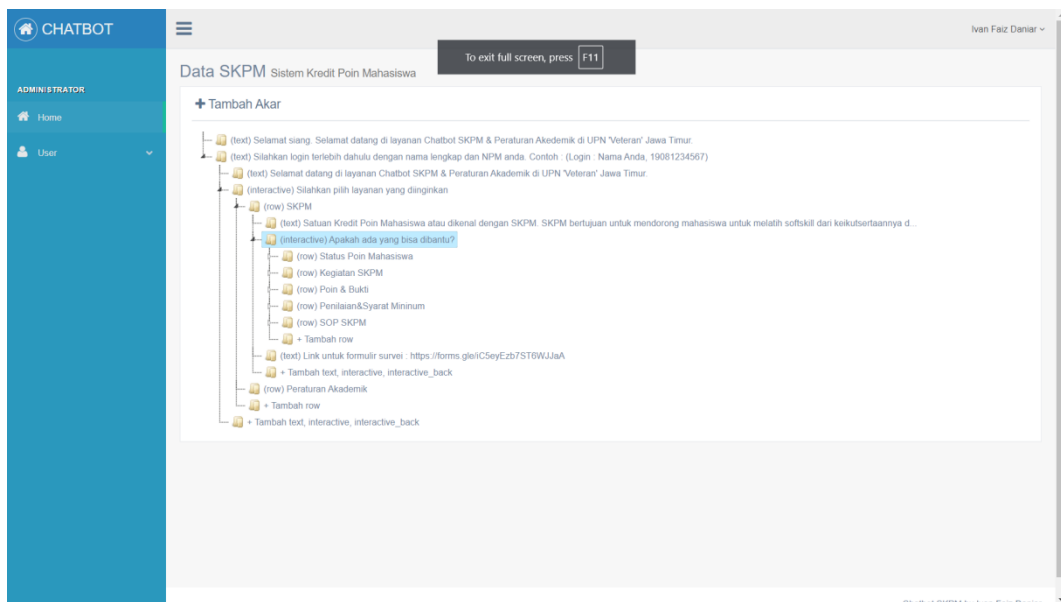
Gambar 19. Tampilan Jawaban Ketika Setelah Memilih Pertanyaan

Gambar 19 memperlihatkan tampilan *WhatsApp* dimana sistem Chatbot menampilkan jawaban pertanyaan jika kita memilih menu pertanyaan “Kegiatan SKPM”. Gambar 20 menampilkan kode untuk menampilkan jawaban *Text* dari Chatbot.

```
if (type == 'text'){
  try {
    await axios({
      method: "POST",
      url:
        "https://graph.facebook.com/v12.0/" +
        phone_number_id +
        "/messages?access_token=" +
        token,
      data: {
        messaging_product: "whatsapp",
        to: from,
        text: {
          body: message
        },
      },
      headers: { "Content-Type": "application/json" },
    });
    console.log(`Message sent: ${message}`);
  } catch (error) {
    console.error('Error sending message:', error);
  }
}
```

Gambar 20. Kode Memilih Pertanyaan

Tampilan Admin Dashboard



Gambar 21. Tampilan utama Admin *Dashboard*

Gambar 21 memperlihatkan tampilan utama dari admin *dashboard* dimana admin dapat melihat semua pertanyaan dan jawaban yang ada dalam *database* Chatbot. Gambar 22 menampilkan kode utama untuk menunjukkan pembuatan pohon interaktif menggunakan jsTree.

```
$query = "SELECT * FROM data ORDER BY parent ASC, id ASC";
$result = $conn->query($query);

$nodes = array();
$interactiveParents = array();
$parentChildren = array();

while ($row = $result->fetch_assoc()) {
    $parent = ($row['parent'] == NULL) ? '#' : $row['parent'];
    if ($row['type'] == 'interactive') {
        $interactiveParents[$parent] = $row['id'];
    }

    if ($row['type'] == 'row' && array_key_exists($parent, $interactiveParents)) {
        $parent = $interactiveParents[$parent];
    }

    $text = "(" . $row['type'] . ") ";
    $text .= (!empty($row['header'])) ? $row['header'] : $row['body'];
    $text = (strlen($text) > 150) ? substr($text, 0, 150) . "...": $text;

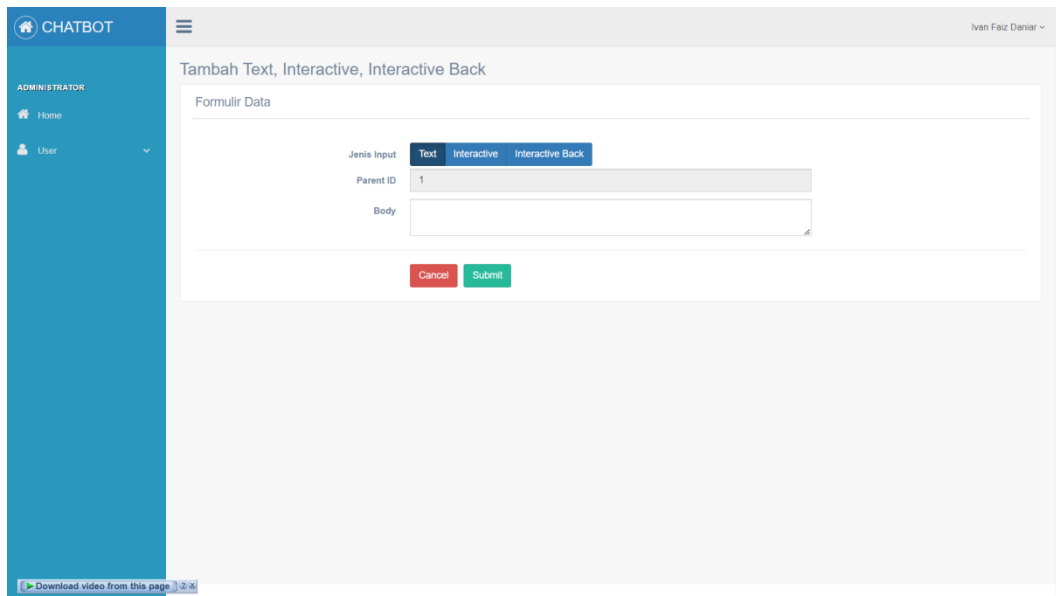
    // Determine the correct update Link based on the node type
    $updateLink = "";
    switch ($row['type']) {
        case "text":
            $updateLink = "update_text.php?id=" . $row['id'];
            break;
        case "interactive":
            $updateLink = "update_interactive.php?id=" . $row['id'];
            break;
        case "row":
            $updateLink = "update_row.php?id=" . $row['id'];
            break;
        case "interactive_back":
            $updateLink = "update_interactive_back.php?id=" . $row['id'];
            break;
    }

    $nodes[] = array(
        "id" => $row['id'],
        "parent" => $parent,
        "text" => $text,
        "a_attr" => array("href" => $updateLink)
    );

    $parentChildren[$parent] = isset($parentChildren[$parent]) ? $parentChildren[$parent] + 1 : 1;
}
```

Gambar 22. Kode Utama Admin *Dashboard*

Membuat Pertanyaan atau Jawaban



Gambar 23. Tampilan Menu Penambahan Pertanyaan Atau Jawaban

Gambar 23 memperlihatkan tampilan untuk membuat pertanyaan atau jawaban untuk kemudian disimpan kedalam *database* Chatbot. Gambar 24 menampilkan

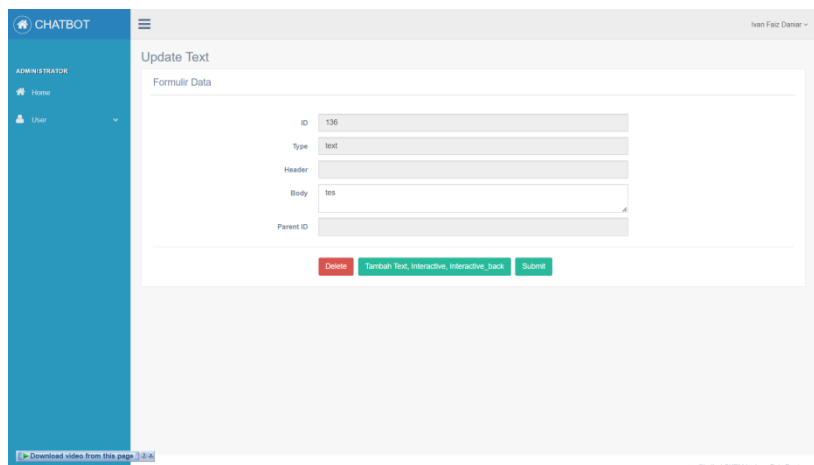
kodingan utama untuk menyimpan pembuatan pertanyaan atau jawaban ke dalam *database*.

```
<?php
include "../koneksi.php"; // Make sure this path correctly points to your database connection script

// Check if the form has been submitted using POST method
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Extract data from POST
    $parent = isset($_POST['parent']) && $_POST['parent'] != '' ? $_POST['parent'] : NULL;
    $type = $_POST['type'] ?? '';
    $header = isset($_POST['header']) && $_POST['header'] != '' ? $_POST['header'] : NULL;
    $body = $_POST['body'] ?? '';

    // SQL query to insert data into the database
    $query = "INSERT INTO `data` (`parent`, `type`, `header`, `body`) VALUES (?, ?, ?, ?)";
    if ($stmt = mysqli_prepare($conn, $query)) {
        mysqli_stmt_bind_param($stmt, "isss", $parent, $type, $header, $body);
        if (mysqli_stmt_execute($stmt)) {
            header("Location: dashboard.php"); // Redirect to dashboard on successful insert
            exit();
        } else {
            echo "Error: " . mysqli_stmt_error($stmt); // Error handling
        }
        mysqli_stmt_close($stmt);
    } else {
        echo "Error preparing the statement: " . mysqli_error($conn);
    }
    mysqli_close($conn);
}
?>
```

Gambar 24. Kode Utama Menu Penambahan Pertanyaan Atau Jawaban Mengubah dan Menghapus Pertanyaan atau Jawaban



Gambar 25. Tampilan Mengubah dan Menghapus Pertanyaan atau Jawaban.

Gambar 25 memperlihatkan tampilan untuk merubah pertanyaan atau jawaban untuk disimpan ke dalam *database* Chatbot. Gambar 26 memperlihatkan kodingan utama untuk menyimpan perubahan pertanyaan atau jawaban untuk disimpan ke dalam *database* Chatbot.


```

<?php
include "../koneksi.php"; // Ensure your database connection file is correctly referenced

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $id = $_POST['id'] ?? null;
    $type = $_POST['type'] ?? '';
    $header = $_POST['header'] ?? NULL; // Accept NULL if header is not provided
    $body = $_POST['body'] ?? '';
    $parentId = $_POST['parentId'] ?? null;

    // SQL query to update the data
    $query = "UPDATE `data` SET `parent` = ?, `type` = ?, `header` = ?, `body` = ? WHERE `id` = ?";
    if ($stmt = mysqli_prepare($conn, $query)) {
        mysqli_stmt_bind_param($stmt, "isssi", $parentId, $type, $header, $body, $id);
        if (mysqli_stmt_execute($stmt)) {
            header("Location: dashboard.php"); // Redirect to dashboard on successful update
            exit();
        } else {
            echo "Error: " . mysqli_stmt_error($stmt); // Display SQL execution error
        }
        mysqli_stmt_close($stmt);
    } else {
        echo "Error preparing the statement: " . mysqli_error($conn); // Display statement preparation error
    }
    mysqli_close($conn);
}
}
    
```

Gambar 26. Kode Utama Mengubah Pertanyaan Atau Jawaban.

Testing

Tahap ini merupakan tahap dilakukannya pengujian kesesuaian terhadap sistem Chatbot dan admin *dashboard* yang berhasil dibangun. Pengujian dilakukan dengan menggunakan metode pengujian *Blackbox* dan survei untuk mengukur *usability* Chatbot.

Blackbox

Setelah pengembangan Chatbot dan Admin *Dashboard* selesai, langkah berikutnya adalah melakukan pengujian sistem dengan menggunakan teknik *Blackbox*. Berikut ini adalah hasil dari pengujian *Blackbox* yang telah dilaksanakan.

Tabel 1. Tabel Pengujian *Blackbox*

No.	Skenario	Hasil Yang Diharapkan	Hasil Sebenarnya	Status
1.	Pengguna menyapa Chatbot	Chatbot menyapa balik pengguna	Chatbot menyapa balik pengguna	OK
2.	Pengguna melakukan autentifikasi	Chatbot mengenali pengguna	Chatbot mengenali pengguna	OK
3.	Pengguna menayakan pertanyaan yang telah disediakan	Chatbot menjawab pertanyaan pengguna	Chatbot menjawab pertanyaan pengguna	OK
4.	Admin melihat semua pertanyaan dan jawaban chatbot	Menampilkan semua pertanyaan dan jawaban	Menampilkan semua pertanyaan dan jawaban	OK
5.	Admin mengubah suatu	Suatu pertanyaan atau	Suatu pertanyaan atau	OK

No.	Skenario	Hasil Yang Diharapkan	Hasil Sebenarnya	Status
	pertanyaan atau jawaban chatbot	jawaban chatbot dirubah	jawaban chatbot dirubah	
6.	Admin menambah suatu pertanyaan atau jawaban	Suatu pertanyaan atau jawaban ditambahkan	Suatu pertanyaan atau jawaban ditambahkan	OK
7.	Admin menghapus suatu pertanyaan atau jawaban	Suatu pertanyaan atau jawaban dihapus	Suatu pertanyaan atau jawaban dihapus	OK

CUQ (Chatbot Usability Questionnare)

Pengujian sistem menggunakan CUQ dilaksanakan setelah fase konstruksi selesai. Pengujian ini melibatkan partisipan yang merupakan 32 mahasiswa Prodi Sistem Informasi UPN "Veteran" Jawa Timur, yang memiliki perwakilan dari setiap tingkatan semester, yang menggunakan sistem Chatbot dalam skenario penggunaan nyata. Partisipan diminta untuk mencoba apakah Chatbot akan menghasilkan pertanyaan dan jawaban yang sesuai dan kemudian mengisi kuesioner CUQ yang mengukur berbagai aspek seperti kerpribadian, keramahan, dan kepuasan pengguna dalam menggunakan sistem.

Deployment

Deployment sistem merupakan tahap akhir dalam metodologi *Waterfall* yang telah digunakan dalam pengembangan Chatbot dan admin *dashboard*. Pada tahap ini, fokus utama adalah mengimplementasikan sistem yang telah dirancang dan diuji ke lingkungan nyata di UPA TIK UPN "Veteran" Jawa Timur. Tahap ini melibatkan serangkaian kegiatan yang dirancang untuk memastikan bahwa transisi dari sistem yang dikembangkan menjadi sistem yang beroperasi dapat berjalan lancar serta tanpa hambatan.

Proses *Deployment* dimulai dengan penyerahan secara formal dari sistem Chatbot dan admin *dashboard* yang telah selesai dikembangkan. Penyerahan ini melibatkan pemberian akses penuh ke seluruh fitur dan fungsionalitas dari kedua sistem kepada tim teknologi informasi di UPA TIK UPN "Veteran" Jawa Timur. Kegiatan ini diawali dengan sesi orientasi yang memberikan panduan mendalam tentang cara kerja sistem, termasuk cara melakukan pengelolaan konten melalui admin *dashboard*. Ini memastikan bahwa tim Puskom memiliki semua alat dan pengetahuan yang diperlukan untuk mengelola dan memelihara sistem secara efektif setelah penyerahan.

SIMPULAN

Penelitian ini berhasil mengembangkan sebuah sistem Chatbot berbasis *WhatsApp* yang dirancang untuk melayani pertanyaan mengenai Sistem Kredit Prestasi Mahasiswa (SKPM) di lingkungan Universitas Pembangunan Nasional "Veteran" Jawa Timur (UPNVJT). Sistem Chatbot dan admin *dashboard* yang dikembangkan telah berhasil melewati serangkaian pengujian yang ketat, termasuk pengujian *Blackbox* dan survei untuk mengukur *usability*. Hasil pengujian menunjukkan bahwa semua skenario yang dirancang berjalan dengan sukses pada setiap aspek yang diuji. Survei yang melibatkan 32 mahasiswa dari berbagai tingkatan semester menghasilkan skor *Chatbot Usability Questionnaire* (CUQ) sebesar 74%.

DAFTAR RUJUKAN

- Abdillah, W. (2018). Metode Penelitian Terpadu Sistem Informasi, Pemodelan Teoretis, Pengukuran, dan Pengujian Statis. In *Metode Penelitian Terpadu Sistem Informasi, Pemodelan Teoretis, Pengukuran, dan Pengujian Statis*.
- Anjasmara, D. B., Rosid, M. A., & Eviyanti, A. (2024). Implementasi Fitur Notifikasi Whatsapp API pada Sistem Manajemen Tugas Akhir. *Physical Sciences, Life Science and Engineering*, 1(2). <https://doi.org/10.47134/pslse.v1i2.197>
- Beilharz, F., Sukunesan, S., Rossell, S. L., Kulkarni, J., & Sharp, G. (2021). Development of a positive body image chatbot (KIT) with young people and parents/carers: Qualitative focus group study. *Journal of Medical Internet Research*, 23(6). <https://doi.org/10.2196/27807>
- Chang, J., Park, J., & Park, J. (2023). Using an Artificial Intelligence Chatbot in Scientific Inquiry: Focusing on a Guided-Inquiry Activity Using Inquirybot. *Asia-Pacific Science Education*, 2(2). <https://doi.org/10.1163/23641177-bja10062>
- Corrêa, J. S., Neto, A. P. de A., Pinto, G. R., Lima, L. D. B., & Teles, A. S. (2023). Lhia: A Smart Chatbot for Breastfeeding Education and Recruitment of Human Milk Donors. *Applied Sciences (Switzerland)*, 13(12). <https://doi.org/10.3390/app13126923>
- Cucus, A., Endra, R. Y., & Naralita, T. (2019). Chatter Bot Untuk Konsultasi Akademik Di Perguruan Tinggi. *Explore: Jurnal Sistem Informasi Dan Telematika*, 10(1). <https://doi.org/10.36448/jsit.v10i1.1214>
- Facebook. (2020). WhatsApp Business API. *Developers.Facebook.Com*.
- Fauzan Wakila, Y. (2021). Konsep dan Fungsi Manajemen Pendidikan. *Equivalent : Jurnal Ilmiah Sosial Teknologi*, 3(1). <https://doi.org/10.46799/jequi.v3i1.33>
- Karim, I. A., Sayarizki, P., & Prawita, F. N. (2023). Sistem Penjawab Otomatis Berbasis Chatbot Untuk Unit Pengelola Pembelajaran Daring Universitas Telkom. *E-Proceeding of Applied Science*, 9(1).
- Ma, S. P., & Ho, C. T. (2018). Modularized and flow-based approach to Chatbot design and deployment*. *Journal of Information Science and Engineering*, 34(5).

[https://doi.org/10.6688/JISE.201809_34\(5\).0005](https://doi.org/10.6688/JISE.201809_34(5).0005)

- Mamgain, D. (2023). *WhatsApp chatbot Vs Telegram Chatbot: Which is best for your business*.
- Meltwater. (2023). Digital 2023: Indonesia. *Wearesocial*.
- Ramdani, B. (2018). *Pembangunan Aplikasi Layanan Pengurusan Surat Berbasis Line Bot (Studi Kasus di Teknik Informatika Universitas Pasundan)*. November, 1–31.
- Redaksi Jagoan Hosting. (2023). Pengertian OOP (ObjectOriented Programming) dan 4Prinsipnya. <https://www.jagoanhosting.com/blog/oop-adalah/>, 19.
- Rizki, F., Irawan, D., & Hidayat, A. T. (2022). Penerapan Api Whatsapp Dalam Pelayanan Uji Plagiasi Universitas Bina Insan Berbasis Web Mobile. *JURIKOM (Jurnal Riset Komputer)*, 9(3). <https://doi.org/10.30865/jurikom.v9i3.4054>
- Sukmawati, E., Imanah, N. D. N., & Rantauni, D. A. (2023). Implementation and challenges of project-based learning of STEAM in the university during the pandemic: A systematic literature review. *JINoP (Jurnal Inovasi Pembelajaran)*, 9(1). <https://doi.org/10.22219/jinop.v9i1.25177>
- Villanueva, G. R., & Palaoag, T. (2020). Design architecture of FAQ chatbot for higher education institution. *Journal of Advanced Research in Dynamical and Control Systems*, 12(1 Special Issue). <https://doi.org/10.5373/JARDCS/V12SP1/20201062>
- Vinarti, R. A., Sani, N. A., Amalia, R., Sulistyani, E., Adistha Eka Noveyani, Edward Suryaputra, & Muhammad Azhar Arwan. (2024). KIA-CHAT: A QnA Chatbot for Postnatal and Newborn Care. *Jurnal Promkes*, 12(SI 1). <https://doi.org/10.20473/jpk.v12.isi1.2024.97-101>
- Wahid, A. A. (2020). Analisis Metode Waterfall Untuk Pengembangan Sistem Informasi. *Jurnal Ilmu-Ilmu Informatika Dan Manajemen STMIK*, November, 1–5.