

## PENERAPAN CHATGPT DAN DRAW.IO UNTUK OTOMATISASI FLOWCHART MENGGUNAKAN MERMAID CODE

<sup>1</sup>Sitti Wetenriajeng Sidehabi, <sup>2</sup>Hamdan Gani, <sup>3</sup>Lutfi  
<sup>1,2,3</sup>Politeknik ATI Makassar  
<sup>1</sup>[teni@atim.ac.id](mailto:teni@atim.ac.id), <sup>2</sup>[hamdangani@atim.ac.id](mailto:hamdangani@atim.ac.id), <sup>3</sup>[lutfi@atim.ac.id](mailto:lutfi@atim.ac.id)

### ABSTRACT

*This research examines the potential of automating flowchart creation by combining ChatGPT, an advanced language model, and Draw.io, an intuitive diagramming tool. The process begins with inputting a workflow description into ChatGPT, which generates Mermaid code to be converted into a visual flowchart in Draw.io. This approach was tested in discrete manufacturing systems courses, where students often struggle to design flowcharts for complex processes. The study is categorized as development research and involves needs analysis, system design, implementation, testing, and evaluation. Results indicate that this method significantly reduces the time and effort needed to create flowcharts, particularly for students without an IT background. Although manual adjustments are still required to meet certain standards, the offered automation provides a solid foundation for further development. The integration of ChatGPT and Draw.io has the potential to enhance understanding of complex systems across various fields, allowing students to focus on analysis and problem-solving rather than time-consuming diagram creation.*

**Keywords:** Automation, flowchart, ChatGPT, Draw.io, business process.

### ABSTRAK

Penelitian ini mengkaji potensi otomatisasi pembuatan flowchart dengan menggabungkan ChatGPT, model bahasa canggih, dan Draw.io, alat diagram yang intuitif. Proses dimulai dengan memasukkan deskripsi alur kerja ke ChatGPT, yang menghasilkan kode Mermaid untuk dikonversi menjadi flowchart visual di Draw.io. Pendekatan ini diuji dalam mata kuliah sistem manufaktur diskrit, di mana mahasiswa sering kesulitan merancang flowchart untuk proses yang kompleks. Penelitian ini merupakan penelitian pengembangan dan melibatkan analisis kebutuhan, perancangan sistem, implementasi, pengujian, dan evaluasi. Hasilnya menunjukkan bahwa metode ini secara signifikan mengurangi waktu dan usaha dalam membuat flowchart, terutama bagi mahasiswa tanpa latar belakang teknologi informasi. Meskipun masih diperlukan penyesuaian manual untuk memenuhi standar tertentu, otomatisasi yang ditawarkan memberikan dasar yang kuat untuk pengembangan lebih lanjut. Integrasi ChatGPT dan Draw.io berpotensi meningkatkan pemahaman sistem kompleks di berbagai bidang, memungkinkan mahasiswa fokus pada analisis dan pemecahan masalah daripada pembuatan diagram yang memakan waktu.

**Kata Kunci:** Otomatisasi, flowchart, ChatGPT, Draw.io, bisnis proses

## **LATAR BELAKANG**

Visualisasi bisnis proses melalui flowchart telah menjadi hal yang krusial dalam era revolusi industri 4.0, terutama dalam rekayasa otomasi sistem manufaktur diskrit. Flowchart membantu dalam memahami alur kerja, mengidentifikasi masalah, dan melakukan perbaikan. Namun, proses pembuatan flowchart secara manual seringkali memakan waktu dan rentan kesalahan. Oleh karena itu, kebutuhan akan otomatisasi proses pembuatan flowchart semakin mendesak. (Cook, 2017)

Perkembangan teknologi telah memunculkan berbagai perangkat lunak dan bahasa pemrograman yang dapat digunakan untuk otomatisasi, termasuk Mermaid code, yang dirancang khusus untuk membuat diagram secara visual. Di samping itu, alat seperti ChatGPT, model bahasa alami dari OpenAI, dan Draw.io, aplikasi penggambaran diagram online, menawarkan pendekatan baru dalam pengembangan sistem ini. ChatGPT dapat digunakan untuk menghasilkan kode Mermaid berdasarkan deskripsi proses dari pengguna, sementara Draw.io berfungsi sebagai platform untuk mengonversi kode tersebut menjadi flowchart interaktif. (Rehwaldt, 2024) (Cook, 2017) (Mahendra, 2021) (Sveidqvist, 2024) (Sveidqvist, 2024)

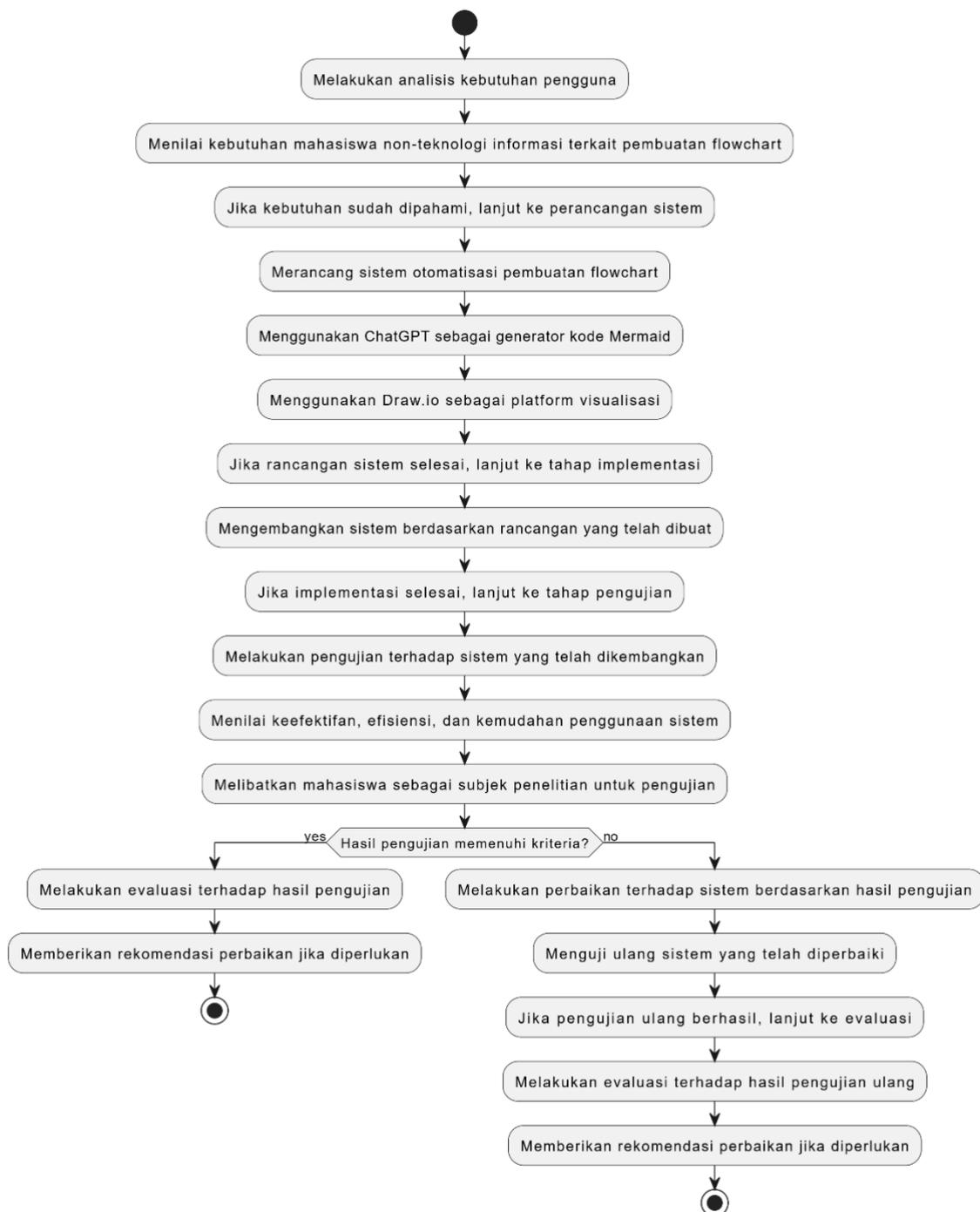
Penerapan metode ini tidak hanya diharapkan dapat meningkatkan produktivitas dalam pengembangan sistem manufaktur diskrit, tetapi juga mempermudah pemahaman terhadap proses yang kompleks. Selain itu, mahasiswa non-teknologi informasi sering menghadapi kesulitan dalam membuat flowchart meskipun telah menyelesaikan kode program. Hal ini menunjukkan perlunya solusi otomatisasi yang dapat meningkatkan efisiensi dan efektivitas pembuatan flowchart, sehingga memudahkan mahasiswa dalam menyajikan alur pemrograman mereka.

Dengan demikian, penelitian ini bertujuan untuk mengeksplorasi potensi kombinasi ChatGPT dan Draw.io dalam mengotomatisasi pembuatan flowchart menggunakan Mermaid code. Diharapkan, penelitian ini dapat memberikan solusi yang lebih cepat dan akurat dalam pembuatan flowchart, khususnya bagi mahasiswa non-teknologi informasi yang sering mengalami kesulitan dalam membuat flowchart. Hasil penelitian ini juga diharapkan dapat menjadi referensi bagi peneliti lain serta diaplikasikan dalam dunia industri dan Pendidikan.

## **METODE PENELITIAN**

Penelitian yang dilakukan adalah penelitian pengembangan (development research) atau penelitian terapan (applied research) yang bertujuan untuk menemukan solusi praktis terhadap masalah pembuatan flowchart secara manual. Proses otomatisasi pembuatan flowchart direncanakan untuk meningkatkan efisiensi dan efektivitas, dengan melibatkan teknologi terkini seperti ChatGPT, Draw.io, dan kode Mermaid.

Tahapan yang dilaksanakan meliputi: (1) Analisis kebutuhan: Melakukan analisis terhadap kebutuhan pengguna, terutama mahasiswa non-teknologi informasi, terkait dengan pembuatan flowchart. (2) Perancangan sistem: Merancang sistem otomatisasi pembuatan flowchart dengan menggunakan ChatGPT sebagai generator kode Mermaid, dan Draw.io sebagai platform visualisasi. (3) Implementasi: Mengembangkan sistem berdasarkan rancangan yang telah dibuat. (4) Pengujian: Melakukan pengujian terhadap sistem yang telah dikembangkan untuk mengukur keefektifan, efisiensi, dan kemudahan penggunaan. Pengujian akan melibatkan sejumlah mahasiswa sebagai subjek penelitian. (5) Evaluasi: Mengevaluasi hasil pengujian dan memberikan rekomendasi perbaikan jika diperlukan.



Gambar 1. Diagram Alir Penelitian

Metode pengumpulan data yang digunakan adalah diskusi dan wawancara dengan mahasiswa dan dosen untuk menggali kesulitan dalam pembuatan flowchart serta pengamatan langsung terhadap penggunaan sistem, yang kemudian dianalisis secara kuantitatif dan kualitatif.

## HASIL DAN PEMBAHASAN

Langkah pertama dalam penggunaan ChatGPT adalah mengunjungi platform atau aplikasi yang menyediakan akses ke layanan tersebut. Setelah itu, pertanyaan atau

pernyataan yang ingin didiskusikan dapat dimasukkan ke dalam kolom yang tersedia. Konteks tambahan atau spesifikasi mengenai jenis jawaban yang diinginkan juga dapat diberikan. Setelah pengiriman pesan, respons dari model akan diterima dalam waktu singkat. Jika diperlukan, percakapan dapat dilanjutkan dengan memberikan tanggapan atau pertanyaan lanjutan untuk memperdalam diskusi.

Flowchart bisnis proses untuk sistem tertentu dapat dibuat dengan format pertanyaan berikut: “Tolong buat saya contoh *use case* yang terdefinisi dengan baik dan terstruktur dalam bentuk paragraf untuk sistem apa pun.” Dalam kasus ini, sebuah flowchart tentang sistem monitoring suhu dan kelembaban menggunakan ESP32 akan dibuat dengan mengetikkan pertanyaan “**Tolong buat saya contoh use case yang terdefinisi dengan baik dan terstruktur dalam bentuk paragraf untuk sistem monitoring suhu dan kelembaban menggunakan esp32**” sehingga sebuah use case yang terdiri dari aktor utama, deskripsi, prasyarat, alur utama, dan post-kondisi akan muncul. Setelah itu, pertanyaan kedua dapat diberikan, seperti “**Tolong buat kode program esp32 untuk use case di atas**” yang akan menghasilkan kode program seperti pada gambar 2.

Pada kode program yang ditunjukkan dalam gambar 2, kode program dengan struktur yang terdiri dari file header, deklarasi variabel global, dan dua fungsi, yakni fungsi “setup” dan fungsi “loop,” dapat terlihat. Agar flowchart yang dihasilkan sesuai dengan logika program alur utama, pembuatan kode mermaid khusus pada fungsi “loop” perlu dilakukan. Pertanyaan yang dapat diketik pada ChatGPT adalah “tolong buat kode mermaid untuk flowchart di atas hanya untuk function loop” sehingga kode mermaid akan muncul seperti pada gambar 3.

Kode mermaid dapat disalin dengan menekan tulisan “copy code,” lalu perlu dimodifikasi dengan menghapus tanda kutip atau tanda kurung buka dan tutup di antara pesan agar tidak terjadi error saat ditempelkan ke aplikasi draw.io. Penggunaan draw.io dimulai dengan mengunjungi situs web draw.io dan memilih opsi untuk membuat diagram baru. Setelah itu, template yang diinginkan dapat dipilih atau diagram kosong dapat dibuka. Elemen-elemen diagram kemudian dapat diseret dari panel sebelah kiri dan ditempatkan di area kerja. Pengaturan dan modifikasi elemen dapat dilakukan dengan mengklik dan mengedit properti yang muncul. Setelah selesai, diagram dapat disimpan dalam format yang diinginkan atau diekspor ke berbagai format file. Terakhir, diagram dapat dibagikan atau diintegrasikan ke dalam dokumen lain sesuai kebutuhan penelitian.

```
#include <WiFi.h>
#include <DHT.h> // Menampilkan data ke Serial Monitor
#include <HTTPClient.h>
Serial.print("Suhu: ");
Serial.print(t);
// Ganti dengan kredensial Wi-Fi Anda
Serial.print(" °C, Kelembaban: ");
const char* ssid = "NAMA_WIFI";
Serial.print(h);
const char* password = "PASSWORD_WIFI";
Serial.println(" %");
// Ganti dengan URL server Anda
const char* serverUrl = "http://your-server- // Mengirim data ke server
url/api/sensor_data";
if (WiFi.status() == WL_CONNECTED) {
// Definisikan pin dan tipe sensor
HTTPClient http;
#define DHTPIN 4 // Pin yang digunakan untuk
DHT // Membuat URL endpoint
#define DHTTYPE DHT22
String url = String(serverUrl) + "?temperature=" +
DHT dht(DHTPIN, DHTTYPE); t + "&humidity=" + h;
```

```

void setup() {
  Serial.begin(115200);
  dht.begin();

  // Menghubungkan ke Wi-Fi
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Menghubungkan ke Wi-Fi...");
  }
  Serial.println("Terhubung ke Wi-Fi");
}

void loop() {
  float h = dht.readHumidity(); // Membaca suhu
  dan kelembaban
  float t = dht.readTemperature(); // Dalam Celsius

  // Memeriksa apakah pembacaan berhasil
  if (isnan(h) || isnan(t)) {
    Serial.println("Gagal membaca dari sensor
    DHT!");
    return;
  }

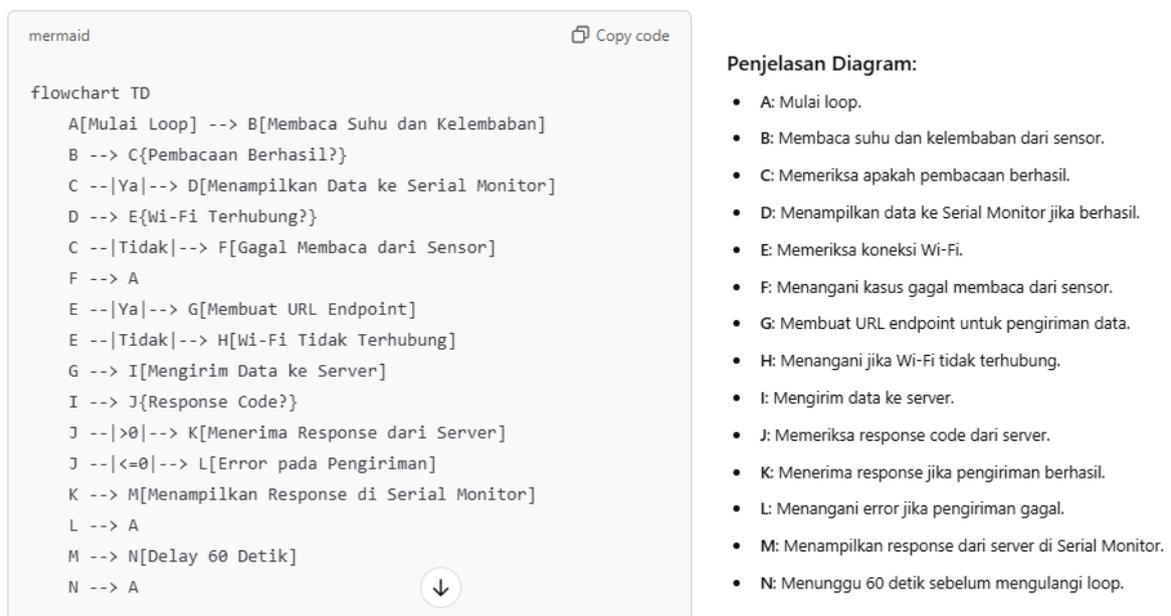
  http.begin(url);
  int httpResponseCode = http.GET();

  if (httpResponseCode > 0) {
    String response = http.getString();
    Serial.println(httpResponseCode);
    Serial.println(response);
  } else {
    Serial.print("Error pada pengiriman: ");
    Serial.println(httpResponseCode);
  }

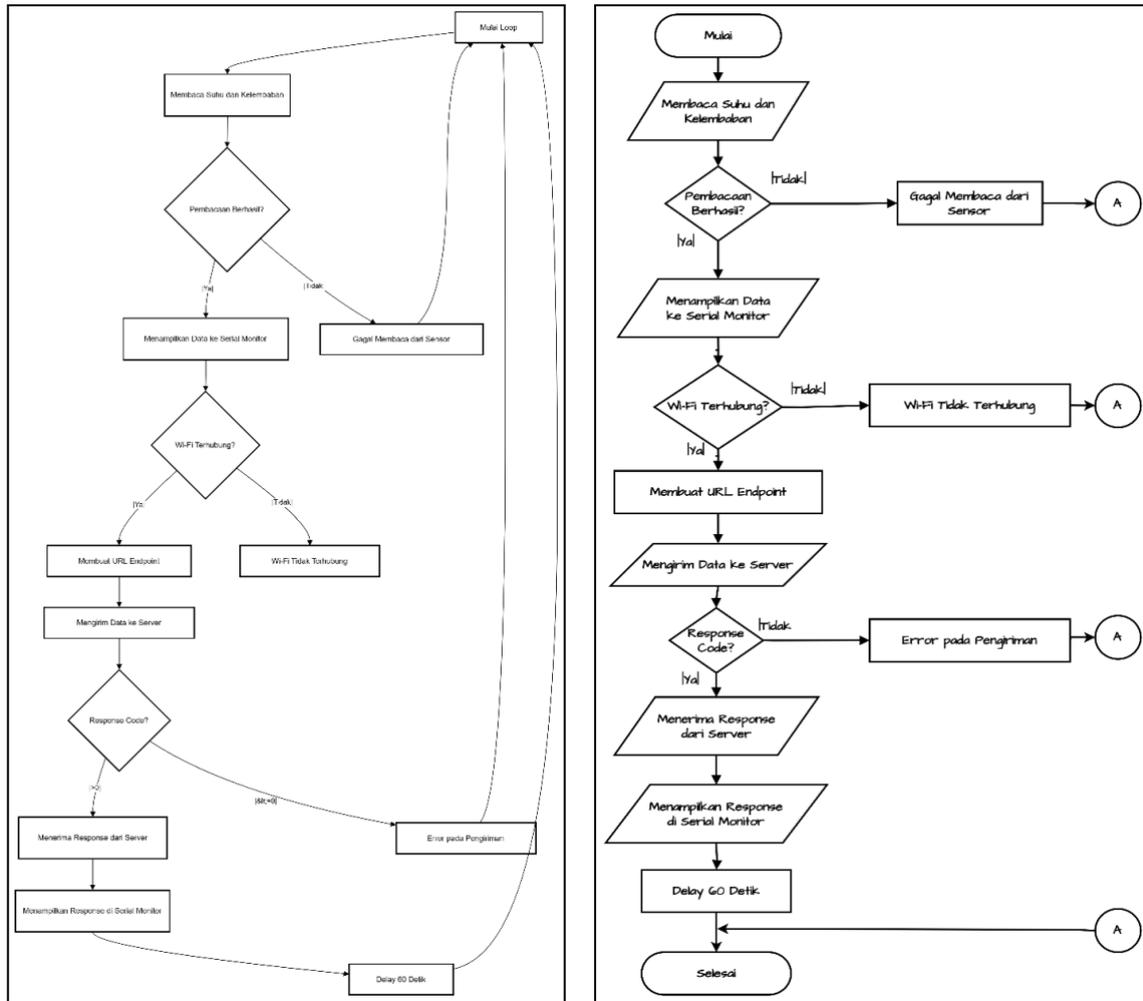
  http.end();
} else {
  Serial.println("Wi-Fi tidak terhubung");
}

// Delay sebelum membaca lagi
delay(60000); // Membaca setiap 60 detik
}
    
```

**Gambar 2.** Hasil yang dibuat chatGPT untuk kode program ESP32 pada sistem monitoring suhu dan kelembaban.



**Gambar 3.** Hasil yang dibuat chatGPT untuk kode mermaid dari kode program esp32 bagian function loop



Gambar 4. Hasil flowchart yang dibuat oleh draw.io, sebelum diedit (kiri) dan setelah diedit (kanan)

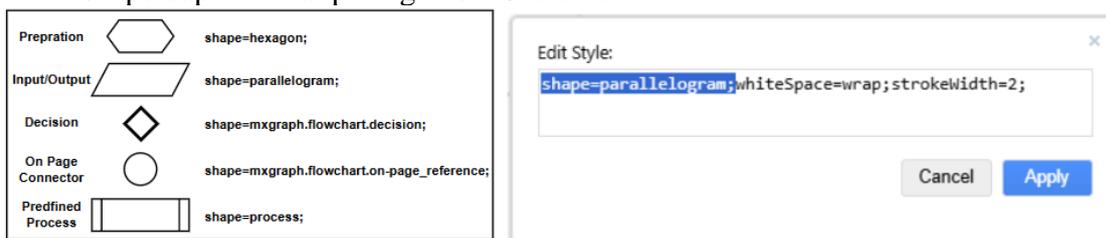
Pada gambar 4 di atas, dua buah flowchart dapat dilihat, yaitu yang dihasilkan dari konversi kode mermaid dan yang telah diedit secara manual agar sesuai dengan standar ANSI/ISO. Untuk mengubah kode mermaid menjadi flowchart, setelah draw.io dibuka di browser internet dan diagram baru dibuat, menu “Arrange - Insert - Advanced - Mermaid...” harus dipilih, sehingga sebuah kotak dialog untuk menempelkan kode mermaid yang sebelumnya sudah disalin dari ChatGPT akan muncul. Tekan insert flowchart, sehingga tampilan di layar diagram dapat ditampilkan seperti pada gambar 3 bagian kiri.

Beberapa pengaturan diperlukan agar flowchart pada gambar kiri dapat berubah menjadi flowchart pada gambar kanan. Semua objek di layar dapat dipilih dengan menekan Ctrl + A, setelah itu, waypoints diatur menjadi orthogonal. Posisi dan ukuran objek simbol flowchart diperbaiki agar terlihat lebih rapi, dan “Connection Points” disesuaikan sehingga posisi panah di antara objek simbol mengikuti aturan yang berlaku.

Garis-garis yang terlalu menumpuk atau tumpang tindih dapat diganti dengan objek lingkaran sebagai connector atau link. Umumnya, garis-garis yang saling tumpang tindih adalah garis yang menuju simbol terminal sebagai titik mulai atau “start.” Arah garis-garis tersebut diatur agar mengarah ke simbol lingkaran untuk menghubungkan ke titik akhir. Untuk simbol kotak yang tidak memiliki panah keluar, yang menunjukkan titik akhir aliran

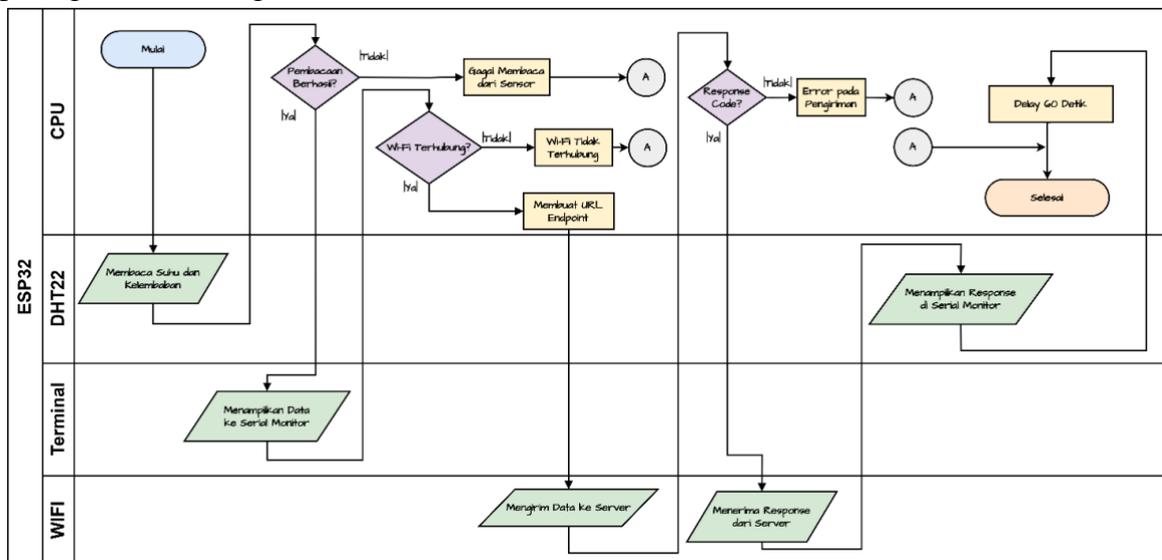
program, simbol lingkaran juga perlu dibuat untuk menunjukkan connector ke titik akhir atau “End.”

Simbol kotak yang bertuliskan “start” atau “mulai” dan “end” atau “selesai” perlu diubah menjadi simbol terminator. Proses ini dilakukan dengan memilih kotak tersebut dan mencentang opsi “rounded” di tab “style” pada jendela sebelah kanan, kemudian sudut kotaknya diseret agar berbentuk seperti kapsul. Meskipun simbol-simbol lain dapat diubah melalui kode mermaid, perubahan lebih mudah dilakukan setelah flowchart dibuat di draw.io. Dengan mengklik kanan pada objek yang ingin diubah dan memilih “edit style,” kotak dialog akan muncul berisi kode tentang properti dengan nilai. Pada awal kode, property shape ditempelkan dengan nilai yang sesuai dengan objek yang diinginkan, dan daftar shape dapat dilihat pada gambar 5 berikut:

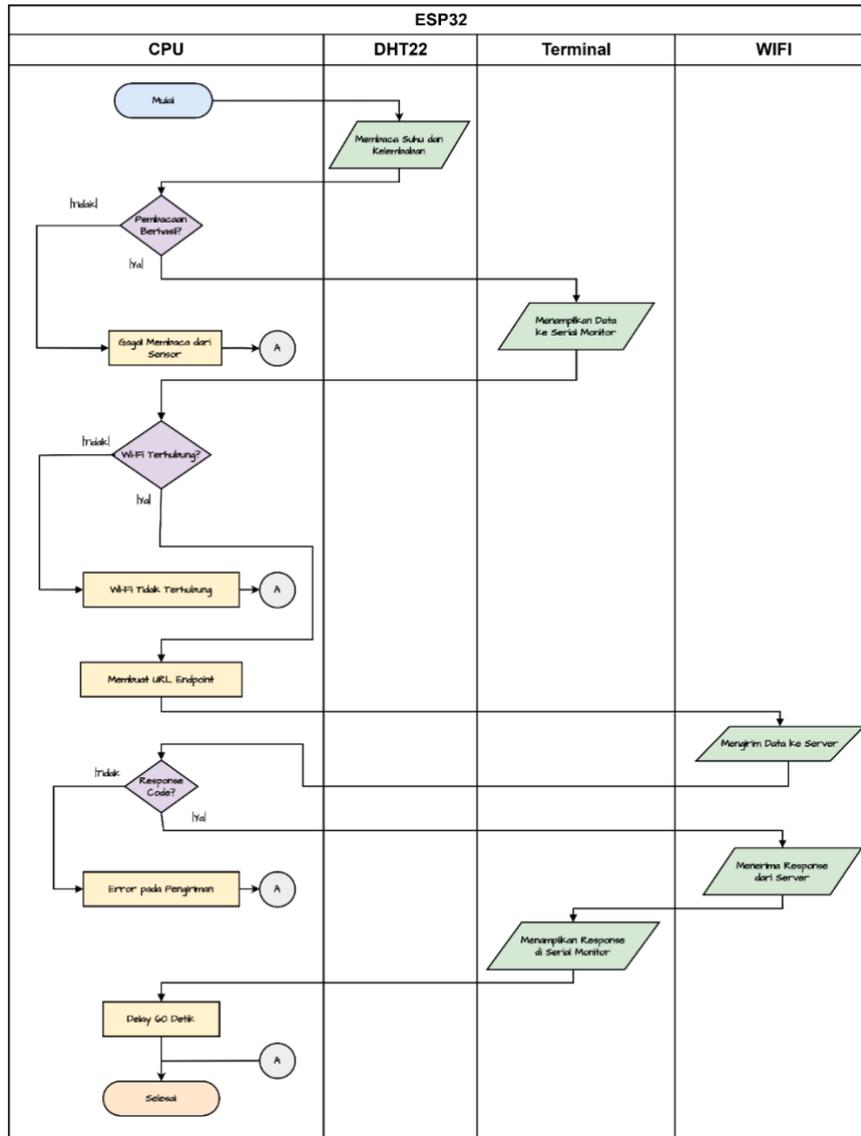


**Gambar 5. Daftar kode property dan value untuk mengubah bentuk objek kotak melalui edit style**

Sebagai opsi tambahan, flowchart dapat dirapikan dengan mendekati setiap objek sehingga panahnya menjadi lebih pendek dan ukuran objek diperkecil sesuai dengan ukuran tulisannya. Swimlane diagram dapat ditambahkan secara horizontal maupun vertikal, dan flowchart dimodifikasi sesuai dengan aktor utama masing-masing. Hasil akhir dapat dilihat pada gambar 6 dan gambar 7 berikut:



**Gambar 6. Tampilan akhir flowchart dengan swimlane diagram horizontal**



**Gambar 7. Tampilan akhir flowchart dengan swimlane diagram vertical**

Pada tampilan akhir, flowchart masih diedit secara manual menggunakan Draw.io, namun alur logikanya sudah terbentuk dan dapat digunakan meskipun belum sesuai dengan standar ANSI/ISO. Proses pengeditan dapat disesuaikan dengan kebutuhan dan estetika tampilan flowchart. Bentuk flowchart mencerminkan logika dari kode mermaid yang telah dihasilkan oleh chatGPT. Meskipun kode mermaid yang dihasilkan berasal dari pertanyaan dan kode program yang sama, hasilnya kadang berbeda-beda, demikian pula dengan kode program yang dihasilkan meskipun pertanyaannya serupa. Meskipun demikian, logika proses yang ada tidak jauh berbeda.

## KESIMPULAN DAN SARAN

Berdasarkan analisis paper di atas, dapat disimpulkan bahwa potensi kombinasi ChatGPT, Draw.io, dan Mermaid code dalam mengotomasi pembuatan flowchart telah berhasil dieksplorasi. Melalui penelitian ini, suatu sistem otomatisasi telah dikembangkan yang mampu menghasilkan flowchart berdasarkan deskripsi proses yang diberikan. Dengan demikian, kesulitan mahasiswa dalam membuat flowchart secara manual dapat diatasi, dan efisiensi dalam pengembangan sistem dapat ditingkatkan.

## **REFERENSI**

- Cook, D. (2017). (Sacramento state university) Retrieved from <http://flowgorithm.org/documentation/index.html>
- Mahendra, M. A. (2021). The Use of Draw Io As Digital Mind Map to Improve Students' Creativity and Students' Concept Mastery in Learning Human Influence on Ecosystem. Bandung: Program Studi International Program on Science Education (IPSE) Fakultas Pendidikan Matematika dan Ilmu Pengetahuan Alam Universitas Pendidikan Indonesia.
- Mujiyono, Simarmata, M. K., Mustofa, Agus, M., Heriyanto, L., Siregar, M. T., & Lutfi. (2021). TRANSFORMASI INDUSTRI 4.0 MANUFAKTUR DISKRIT. Jakarta Selatan: Pusat Pengembangan Pendidikan Vokasi Industri Badan Pengembangan Sumber Daya Manusia Industri Kementerian Perindustrian Republik Indonesia.
- Nishadha. (2024, Juni 5). (Creately) Retrieved from <https://creately.com/blog/diagrams/uml-diagram-types-examples/>
- Rehwaldt, N. (2024, April 1). Retrieved from <https://bpmn.io/blog/posts/2024-ebpmn>
- Sveidqvist, K. (2024). (mermaid-js) Retrieved from <https://github.com/mermaid-js/mermaid>
- Sveidqvist, K. (2024). (mermaid-js) Retrieved from <https://mermaid.js.org/syntax/flowchart.html>