**PENGEMBANGAN WEB (TEORI)**

**LAPORAN EKSPERIMEN PERBANDINGAN METODE ERROR HANDLING TRY CATCH DAN ERROR BOUNDARIES**

*Laporan ini disusun untuk memenuhi tugas 1 mata kuliah Pengembangan Web (Teori)*



Disusun oleh kelompok B4:

Asri Husnul Rosadi 221524035

Faris Abulkhoir 221524040

Mahardika Pratama 221524044

Muhamad Fahri Yuwan 221524047

Najib Alimudin Fajri 221524053

Sarah 221524059

Septyana Agustina 221524060

Dosen Pengampu:

Joe Lian Min, M.Eng.

**JURUSAN TEKNIK KOMPUTER DAN INFORMATIKA**

**PROGRAM STUDI D4 TEKNIK INFORMATIKA**

**POLITEKNIK NEGERI BANDUNG**

**2024**

# **DAFTAR ISI**

[**DAFTAR ISI** i](#_Toc176232797)

[**A.** **IDENTIFIKASI PROBLEM** 1](#_Toc176232798)

[**B.** **DESKRIPSI PROBLEM** 1](#_Toc176232799)

[**C.** **METODOLOGI EKSPERIMEN** 1](#_Toc176232800)

[**D.** **PELAKSANAAN EKSPERIMEN** 2](#_Toc176232801)

[**E.** **ANALISIS HASIL EKSPERIMEN** 2](#_Toc176232802)

# **IDENTIFIKASI PROBLEM**

Dalam pengembangan aplikasi React, penanganan kesalahan merupakan hal yang penting untuk memastikan aplikasi berjalan dengan baik dan memberikan pengalaman pengguna yang baik. Dua pendekatan utama untuk menangani kesalahan di React adalah menggunakan try-catch dan Error Boundaries. Eksperimen ini bertujuan untuk membandingkan kedua metode ini dalam hal efektivitas dan penerapan mereka dalam menangani kesalahan.

# **DESKRIPSI PROBLEM**

Kedua metode penanganan kesalahan memiliki kegunaan dan konteks yang berbeda. try-catch adalah teknik umum yang digunakan untuk menangani kesalahan dalam blok kode tertentu, seperti fungsi atau event handler. Sementara itu, Error Boundaries adalah fitur spesifik React yang dirancang untuk menangani kesalahan yang terjadi selama proses rendering dan lifecycle methods.

Eksperimen ini dilakukan untuk mengidentifikasi:

* Bagaimana masing-masing metode menangani kesalahan yang sama.
* Keefektifan masing-masing metode dalam memberikan umpan balik kesalahan kepada pengguna.
* Kemudahan implementasi dan pemeliharaan masing-masing metode.

# **METODOLOGI EKSPERIMEN**

Eksperimen dilakukan dengan dua proyek React terpisah, masing-masing menggunakan satu dari dua metode penanganan kesalahan yang diuji. Berikut adalah metodologi yang diterapkan:

1. Proyek 1: try-catch

* Komponen ErrorProneComponent yang menyebabkan kesalahan saat event handler dipicu.
* Komponen TryCatchWrapper yang menangani kesalahan menggunakan try-catch di dalam event handler.

1. Proyek 2: Error Boundaries

* Komponen ErrorProneComponent yang menyebabkan kesalahan selama rendering.
* Komponen ErrorBoundary yang menangkap kesalahan yang terjadi di dalam komponen anak selama rendering dan lifecycle methods.

Langkah-langkah yang diambil:

1. Membangun dua proyek React dengan struktur yang sesuai.
2. Implementasi komponen ErrorProneComponent yang menyebabkan kesalahan untuk kedua proyek.
3. Implementasi metode penanganan kesalahan untuk setiap proyek.
4. Menjalankan aplikasi dan mengamati bagaimana kesalahan ditangani dan ditampilkan kepada pengguna.

# **PELAKSANAAN EKSPERIMEN**

1. Proyek 1: try-catch

* ErrorProneComponent berisi kode yang melempar kesalahan saat tombol diklik.
* TryCatchWrapper menangani kesalahan dengan try-catch dan menampilkan pesan kesalahan di UI jika terjadi.

1. Proyek 2: Error Boundaries

* ErrorProneComponent menyebabkan kesalahan saat rendering.
* ErrorBoundary menangkap kesalahan dan menampilkan UI fallback jika terjadi kesalahan.

# **ANALISIS HASIL EKSPERIMEN**

1. Proyek 1: try-catch

* Hasil:

Kesalahan ditangani hanya ketika tombol diklik. Pesan kesalahan ditampilkan secara langsung di UI menggunakan state lokal.

* Pros:
  + Fleksibel dan dapat digunakan untuk menangani kesalahan di dalam fungsi atau event handler tertentu.
  + Mudah diterapkan dalam kasus di mana kesalahan dapat diprediksi dan ditangani di dalam event handler.
* Cons:
  + Tidak efektif untuk menangani kesalahan yang terjadi di luar event handler, seperti kesalahan rendering atau lifecycle methods.
  + Tidak memberikan mekanisme pemulihan otomatis; pengguna mungkin tetap melihat UI yang rusak jika kesalahan tidak ditangani dengan benar.

1. Proyek 2: Error Boundaries

* Hasil:

Kesalahan ditangani secara otomatis selama rendering dan lifecycle methods. UI fallback ditampilkan jika terjadi kesalahan.

* Pros:
  + Menangani kesalahan di tingkat komponen dan subkomponen, termasuk kesalahan yang tidak dapat ditangani oleh try-catch.
  + Menyediakan mekanisme pemulihan otomatis dengan menampilkan fallback UI, yang membantu menjaga aplikasi tetap fungsional meskipun ada kesalahan.
* Cons:
  + Tidak dapat menangani kesalahan di luar rendering atau lifecycle methods, seperti kesalahan dalam event handlers.
  + Memerlukan perancangan yang lebih hati-hati untuk memastikan bahwa semua komponen penting dilindungi oleh Error Boundaries

1. Kesimpulan

* Try-catch efektif untuk menangani kesalahan dalam event handler dan operasi asinkron tetapi memiliki keterbatasan dalam menangani kesalahan di luar konteksnya.
* Error Boundaries adalah alat yang sangat berguna untuk menangani kesalahan rendering dan lifecycle methods, dan membantu menjaga aplikasi tetap stabil dengan menampilkan UI fallback.