



Exception Handling

Pengertian Exception Handling

- Exception handling adalah mekanisme untuk menangani kesalahan saat program berjalan (runtime).
- Digunakan agar program tidak berhenti secara tiba-tiba.
- Membantu menjaga stabilitas dan keamanan aplikasi.

Pengertian Exception

- Exception adalah objek yang merepresentasikan kesalahan.
- Terjadi ketika program mengalami kondisi tidak normal.
- Jika tidak ditangani, program dapat crash.

Tujuan Exception Handling

- Menghindari program crash.
- Mempermudah debugging.
- Menjaga keamanan data.
- Meningkatkan pengalaman pengguna.
- Membuat program lebih stabil.

Jenis-Jenis Error

- Syntax Error → kesalahan penulisan kode.
- Logical Error → kesalahan logika program.
- Runtime Error → kesalahan saat program berjalan.

Syntax Error

- Terjadi karena kesalahan penulisan sintaks.
- Contoh: kurang tanda titik koma atau salah keyword.
- Program tidak dapat dikompilasi.

Logical Error

- Program berjalan tetapi hasil salah.
- Terjadi karena kesalahan algoritma atau rumus.
- Sulit dideteksi compiler.

Runtime Error

- Terjadi saat program sedang berjalan.
- Contoh: pembagian dengan nol.
- Ditangani menggunakan exception handling.

Konsep Dasar Exception Handling

- Mendeteksi kesalahan.
- Melempar exception.
- Menangkap exception.
- Menangani exception.

Struktur Exception Handling

- `try` → mencoba menjalankan kode.
- `catch` → menangkap exception.
- `finally` → proses penutup.
- `throw` → melempar exception.
- `throws` → deklarasi exception.

Blok Try

- Berisi kode yang berpotensi menghasilkan error.
- Jika terjadi exception, program berpindah ke catch.

Blok Catch

- Digunakan untuk menangani exception.
- Dapat menampilkan pesan error.
- Mencegah program berhenti mendadak.

Blok Finally

- Selalu dijalankan baik ada error maupun tidak.
- Digunakan untuk menutup file atau koneksi database.
- Membantu cleanup resource.

Throw

- Digunakan untuk melempar exception secara manual.
- Membantu membuat validasi sendiri.

Throws

- Digunakan untuk mendeklarasikan exception.
- Memberi tahu bahwa method dapat menghasilkan exception.

Hirarki Exception

- Throwable
- → Error
- → Exception

Error

- Kesalahan serius pada sistem.
- Sulit ditangani programmer.
- Contoh: OutOfMemoryError.

Exception

- Kesalahan yang masih dapat ditangani.
- Biasanya terjadi pada aplikasi.
- Contoh: IOException.

Checked Exception

- Diperiksa compiler saat compile time.
- Wajib ditangani programmer.
- Contoh: IOException dan SQLException.

Unchecked Exception

- Terjadi saat runtime.
- Tidak wajib ditangani.
- Biasanya akibat kesalahan logika program.

Contoh Exception Umum

- ArithmeticException.
- NullPointerException.
- ArrayIndexOutOfBoundsException.
- NumberFormatException.
- FileNotFoundException.

Multiple Catch

- Satu try dapat memiliki beberapa catch.
- Digunakan untuk menangani jenis error berbeda.

Nested Try

- Try dapat berada di dalam try lainnya.
- Digunakan untuk penanganan bertingkat.
- Penggunaan berlebihan membuat kode sulit dibaca.

Keuntungan Exception Handling

- Program lebih stabil.
- Kode lebih terstruktur.
- Mempermudah maintenance.
- Meningkatkan keamanan sistem.

Kekurangan Exception Handling

- Kode menjadi lebih panjang.
- Menambah kompleksitas program.
- Dapat menurunkan performa.

Best Practice Exception Handling

- Tangani exception secara spesifik.
- Jangan mengabaikan exception.
- Gunakan finally untuk cleanup.
- Gunakan pesan error yang jelas.

Perbedaan Error dan Exception

- Error → kesalahan serius sistem.
- Exception → kesalahan yang dapat ditangani.
- Error sulit diperbaiki program.
- Exception dapat ditangani programmer.

Penerapan Dunia Nyata

- Sistem Login.
- ATM.
- Sistem Akademik.
- E-Commerce.

Exception Handling pada Sistem Login

- Menangani username kosong.
- Menangani password salah.
- Mencegah akses tidak valid.

Exception Handling pada ATM

- Menangani saldo tidak cukup.
- Menangani koneksi gagal.
- Menjaga keamanan transaksi.

Exception Handling pada Sistem Akademik

- Menangani input nilai salah.
- Menangani data mahasiswa tidak ditemukan.

Kesimpulan

- Exception handling digunakan untuk menangani runtime error.
- Membuat program lebih aman dan stabil.
- Meningkatkan kualitas aplikasi.
- Merupakan bagian penting dalam pengembangan software.



TERIMA KASIH