Sistem Terdistribusi (ST)

Pertemuan 8 : Failure Model dan Recovery

Alauddin Maulana Hirzan

Fakultas Teknologi Informasi dan Komunikasi
Universitas Semarang

Outline

Konsep Failure

Definisi

Failure merupakan kondisi ketika sistem tidak mampu memberikan layanan sesuai dengan spesifikasi yang telah ditentukan (functional maupun non-functional requirements). Dalam konteks sistem terdistribusi, failure tidak selalu berarti sistem berhenti total, tetapi dapat berupa:

1. Respon yang salah
2. Respon yang tidak lengkap
3. Respon yang tidak tepat waktu

Failure bersifat eksternal, artinya dapat diamati oleh:

1. Pengguna akhir
2. Sistem lain (client atau node lain)

Konsep Failure

Istilah Dalam Failure

- Fault $\rightarrow$ penyebab
- Error $\rightarrow$ kondisi salah
- Failure $\rightarrow$ output salah

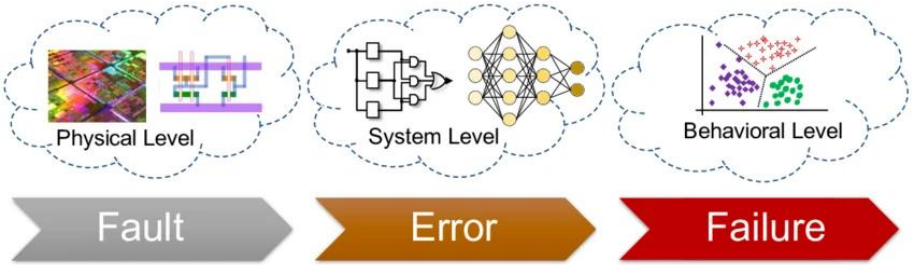
Konsep Failure

Penjelasan:

Ketiga konsep ini membentuk rantai sebab-akibat (causal chain) dalam dependability system:

1. Fault (kesalahan awal / defect)
   - Sumber masalah
   - Contoh: bug pada kode, kerusakan hardware, gangguan jaringan
2. Error (state yang salah)
   - Kondisi internal sistem yang sudah tidak benar
   - Belum tentu terlihat oleh pengguna
   - Contoh: variabel berisi nilai salah akibat bug
3. Failure (kegagalan layanan)
   - Terjadi ketika error mencapai boundary sistem
   - Terlihat oleh user atau sistem lain
   - Contoh: API mengembalikan data yang salah

Konsep Failure



Pentingnya Failure Model

Failure model adalah abstraksi yang digunakan untuk mendeskripsikan bagaimana sistem dapat gagal. Hal ini penting karena:

1. Membantu desain sistem tahan gangguan (robust system)
   - Sistem dapat dirancang untuk mengantisipasi jenis kegagalan tertentu
2. Dasar fault tolerance
   - Teknik seperti replication, retry, consensus sangat bergantung pada model failure
3. Menyederhanakan kompleksitas
   - Developer tidak perlu menangani semua kemungkinan, cukup sesuai model

Konsep Failure

Klasifikasi Failure

1. Crash
2. Omission
3. Timing
4. Byzantine
5. Partition

Penjelasan:

Klasifikasi ini umum digunakan dalam literatur sistem terdistribusi untuk memahami perilaku kegagalan node atau jaringan.

Konsep Failure

Crash Failure

Node berhenti total

Penjelasan:

Crash failure terjadi ketika node:

1. Berhenti bekerja secara tiba-tiba
2. Tidak mengirim atau menerima pesan
3. Tidak memberikan respon sama sekali

Konsep Failure

Contoh Crash

1. Server mati
2. Container berhenti

Penjelasan:

Contoh nyata:

1. Server fisik mengalami power failure
2. Container Docker crash akibat out-of-memory
3. VM dihentikan secara tiba-tiba

Konsep Failure

Dampak Crash

- Service unavailable
- Data inconsistency

Penjelasan:

Dampak utama:

1. Service Unavailability
2. Layanan tidak dapat diakses
3. Data Inconsistency
4. Jika crash terjadi saat proses write atau transaksi

Konsep Failure

Omission Failure

Pesan hilang

Penjelasan:

Omission failure terjadi ketika Pesan yang seharusnya dikirim atau diterima tidak terjadi. Tidak seperti crash, node masih aktif tetapi komunikasi gagal.

Konsep Failure

Jenis Omission Failure

1. Send omission : Node gagal mengirim pesan
2. Receive omission : Node tidak menerima pesan yang dikirim
3. Channel omission : Pesan hilang di jaringan

Konsep Failure

Contoh Omission

- Packet loss
- Network drop

Penjelasan:

Contoh:

- Packet hilang akibat congestion
- Wireless interference menyebabkan data tidak sampai
- Router membuang packet (buffer overflow)

Konsep Failure

iming Failure

Respon terlambat

Penjelasan:

- Timing failure terjadi ketika respon sistem tidak sesuai dengan batas waktu (deadline)
- Umumnya terjadi pada:
  - Real-time system
  - Distributed system dengan SLA ketat

Konsep Failure

Dampak Timing Failure

- Deadline miss
- QoS menurun

Penjelasan:

1. Deadline miss
   - Sistem real-time gagal memenuhi waktu respon
2. Quality of Service (QoS) menurun
   - Latensi tinggi
   - Throughput menurun

Konsep Failure

Byzantine Failure

Node bertindak tidak konsisten

Penjelasan:

Byzantine failure adalah jenis kegagalan paling kompleks:

- Node dapat memberikan informasi berbeda ke node lain
- Tidak dapat dipercaya

Konsep Failure

Contoh Byzantine

- Data korup
- Node diserang

Penjelasan:

Contoh:

- Node mengirim data berbeda ke dua node berbeda
- Malware memanipulasi hasil komputasi
- Sistem blockchain diserang oleh malicious node

Konsep Failure

Kompleksitas Byzantine

- Sulit dideteksi
- Butuh algoritma khusus

Penjelasan:

Mengapa sulit:

- Tidak ada pola tetap
- Node terlihat “hidup” tetapi memberikan data salah

Solusi:

- Byzantine Fault Tolerance (BFT)
- Practical BFT (PBFT)

Konsep Failure

Partition Failure

Jaringan terpisah

Penjelasan:

Partition failure terjadi ketika:

- Jaringan terpecah menjadi beberapa bagian
- Node tidak dapat saling berkomunikasi

Konsep Failure

Contoh Partition

Cluster split-brain

Penjelasan:

Split-brain:

- Dua bagian cluster berjalan independen
- Keduanya menganggap dirinya sebagai “primary”

Akibat:

- Konflik data
- Inkonsistensi

Konsep Failure

Failure Detection

Menentukan node gagal

Failure detection adalah mekanisme untuk:

- Mengidentifikasi node yang gagal
- Menjadi dasar recovery (failover, re-election)

Konsep Failure

Heartbeat

Sinyal periodik antar node

Heartbeat:

- Node mengirim sinyal secara berkala
- Jika sinyal tidak diterima $\rightarrow$ indikasi failure

Contoh: Leader dalam cluster mengirim heartbeat ke follower

Konsep Failure

Timeout Detection

Tidak ada respon $\rightarrow$ gagal

Penjelasan:

- Jika dalam waktu tertentu: Tidak ada respon $\rightarrow$ node dianggap gagal
- Masalah: Delay jaringan bisa menyebabkan kesalahan deteksi

Konsep Failure

Gossip Protocol

- Node berbagi informasi secara acak ke node lain
- Informasi menyebar seperti epidemi

Keunggulan:

- Scalability tinggi
- Toleran terhadap failure

Digunakan pada:

- Cassandra
- Kubernetes cluster state

Konsep Recovery

Definisi

Recovery adalah proses untuk mengembalikan sistem setelah terjadi failure agar kembali ke kondisi operasional yang benar (correct state). Tujuan utama recovery:

1. Restorasi layanan (service restoration)
   - Sistem kembali melayani request
2. Menjaga konsistensi data (data integrity)
   - Tidak terjadi corrupt atau partial update
3. Meminimalkan downtime (availability)
   - Recovery harus cepat dan efisien

Konsep Recovery

Retry Mechanism

Retry adalah mekanisme sederhana namun sangat efektif untuk menangani failure sementara (transient failure), seperti:

1. Network glitch
2. Temporary overload
3. Timeout

Prinsip: Jika operasi gagal $\rightarrow$ ulangi kembali

Konsep Recovery

Replication

Replication adalah teknik menyimpan salinan data di beberapa node untuk:

1. Fault tolerance
2. High availability
3. Load balancing

Jika satu node gagal: Node lain dapat menggantikan (failover)

Konsep Recovery

Checkpointing

Checkpointing adalah proses menyimpan snapshot kondisi sistem pada waktu tertentu.

- Tujuan: Memudahkan recovery tanpa mengulang dari awal
- Contoh:
  - Snapshot memory aplikasi
  - Snapshot database

Konsep Recovery

Rollback Recovery

Rollback recovery mengembalikan sistem ke checkpoint terakhir sebelum failure.

Proses:

- Failure terjadi
- Sistem kembali ke checkpoint
- Eksekusi ulang dari titik tersebut

Masalah utama:

- Cascading rollback
- Lost computation

Lastly

Terima Kasih