

BAB II

TINJAUAN PUSTAKA DAN TEORI

2.1 Tinjauan Pustaka

Pada penelitian ini ada beberapa referensi terkait Pengembangan Sistem Backup Backupflow Berbasis *Microservices* Dan *gRPC* Menggunakan *Cloud Computing*) yang digunakan sebagai acuan, antara lain sebagai berikut:

Johannes Fernandes Andry (2017), aplikasi *backup* yang bertujuan mencegah adanya *crash* atau *error* pada aplikasi, sistem ini menggunakan metode SDLC.

Kadek Surya Mahedy (2021), menggunakan metode *paradigma prototyping*. *Prototyping* merupakan sebuah proses yang memungkinkan pengembang untuk membuat model perangkat lunak yang akan direalisasikan. Hasil yang diperoleh adalah bertujuan agar penggunaan backup yang mudah dan juga efisien, dan jika terdapat error maka aplikasi akan me *restore file* dari server cadangan.

Andi Rosano dan Djadjat Sudaradjat (2020), penelitian ini meneliti sistem *backup* pada bank XYZ, metode penelitian ini menggunakan Model *Mirroring Sistem*. Dimana lokasi *Disaster Recovery Center* terletak di kota lain, untuk mengantisipasi hal hal seperti kebakaran dan sebagainya, hasil dari penelitian ini dinyatakan bahwa dalam 30 menit pertama nampak normal, namun setelah 30

menit selanjutnya terjadi *fatal* yang mengharuskan sistem kembali ke posisi semula.

Maksy Sendiang, Sonny Kasenda dan Jerry Purnama, mengenai *micro-service*, penelitian ini berfokus pada pengembangan *Machine Learning* menggunakan teknologi *micro-service* akan membentuk aplikasi yang memberikan dampak positif pada proses belajar mengajar.

Mohammad Harry Khomas Saputra dan Luthfi Muhammad Nabil, berfokus pada efek-efek yang didapatkan dari arsitektur *micro-service*. Arsitektur ini dapat disimpulkan tidak mengganggu system lain pada saat melakukan pengembangan dan perbaikan system karena sistem dibuat secara terpisah sehingga dapat memperkecil ruang lingkup dari pengembangan.

Untuk penelitian ini, sistem berfokus pada sistemisasi dan efisiensi aplikasi menggunakan *microservices* dan *gRPC*. Dan dengan fitur-fitur yang membuat hasil *backup* aman, cepat dan efisien.

Perbandingan pada penelitian tersebut dapat dilihat pada table 2.1

Tabel 2.1 Tabel Perbandingan

Penulis	Judul	Metode	Hasil
Johanes Fernandes Andry (2017)	Pengembangan Aplikasi <i>Backup</i> Dan <i>Restore</i> Secara Otomatisasi Menggunakan <i>Sdlc</i> Untuk Mencegah Bencana	SDLC yaitu <i>Sistem Development Life Cycle Model Waterfall</i> .	Mencegah <i>file</i> pada komputer terdapat <i>crash</i> atau <i>error</i> .

Kadek Surya Mahedy (2021)	Implementasi Sistem <i>Backup Data</i> Pada Sistem Informasi Perpustakaan Universitas Pendidikan Ganesha	paradigma <i>prototyping</i>	meminimalisasi terjadinya kehilangan data yang disebabkan oleh kerusakan perangkat atau kelalaian <i>user</i> , karena semua data akan diduplikasi dan disimpan pada server yang berbeda. Jika terjadi kehilangan data maka <i>user</i> dapat mengembalikan data dari server cadangan.
Andi Rosano, Djadjat Sudaradjat (2020)	Manajemen <i>Backup Data</i> untuk Penyelamatan Data Nasabah pada Sistem Informasi Perbankan	Model <i>Mirroring Sistem</i>	Setelah dilakukan <i>Switch over</i> tidak timbul permasalahan yang serius dan hasil pengecekan menyatakan normal, barulah sistem dilanjutkan selama 24 jam penuh. Namun apabila dalam 30 menit pertama setelah <i>Switch over</i> dilakukan terjadi masalah yang bersifat fatal, misalkan semua transaksi gagal, atau staf bank tidak bisa login ke sistem, dan sebagainya, maka proses <i>switch over</i> dianggap gagal. Sistem harus dikembalikan ke posisi semula.
Maksy Sendiang, Sonny Kasenda, Jerry Purnama (2018)	Implementasi Teknologi <i>Microservice</i> pada Pengembangan <i>Mobile Learning</i>	<i>Rational Unified Process (RUP)</i>	Arsitektur <i>microservice</i> memecah aplikasi menjadi entity – entity independent yang dipanggil melalui API Gateway
Mohammad Harry Khomas Saputra, Luthfi Muhammad Nabil (2021)	Penerapan Arsitektur <i>Microservice</i> Pada Sistem Tata Kelola Matakuliah	<i>Use Case Diagram</i>	Efek yang dihasilkan saat mengembangkan sistem saat arsitektur <i>microservice</i> lebih tidak mengganggu sistem lain saat melakukan

	Proyek Politeknik Pos Indonesia		pengembangan sehingga dapat memperkecil ruang lingkup dari pengembangan
Adnan Nuryono (2022)	Pengembangan Sistem <i>Backup Backupflow</i> Berbasis <i>Microservices</i> Dan <i>GRPC</i> Menggunakan	<i>Waterfall</i>	Dapat mengimplementasikan kinerja dan efisiensi penggunaan <i>microservices</i> dan <i>grpc</i> dalam mekanisme <i>backup</i> .

2.2 Dasar Teori

2.2.1 Backup

Backup adalah proses duplikasi atau menyalin data atau *file* dari satu perangkat atau media penyimpanan ke media sekunder, seperti *flashdisk*, *hardisk eksternal*, sistem *cloud* atau media lainnya, yang dilakukan secara *offline* maupun *online*. *Backup* memiliki dua tujuan. untuk mengembalikan data apabila data tersebut hilang, baik karena terhapus atau karena rusak (*corrupt*) dan untuk mengembalikan data ke titik tertentu pada masa lalu (Nagitec, 2022).

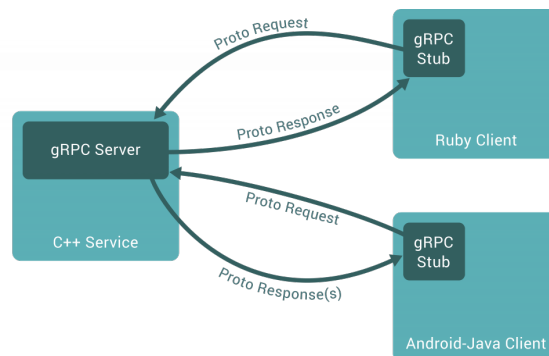
Menurut Nagitec (2022),

2.2.2 Restore

Proses mengembalikan kembali sebuah data atau *file* ketempat semula. Jadi misalnya data Anda terhapus secara tidak sengaja, maka Anda masih dapat mencari *file* data tersebut dalam recycle bin komputer untuk kemudian dikembalikan ketempat lokasi semula *file* itu berada, di suatu folder tertentu (Reny Pebriasari, 2022).

2.2.3 gRPC

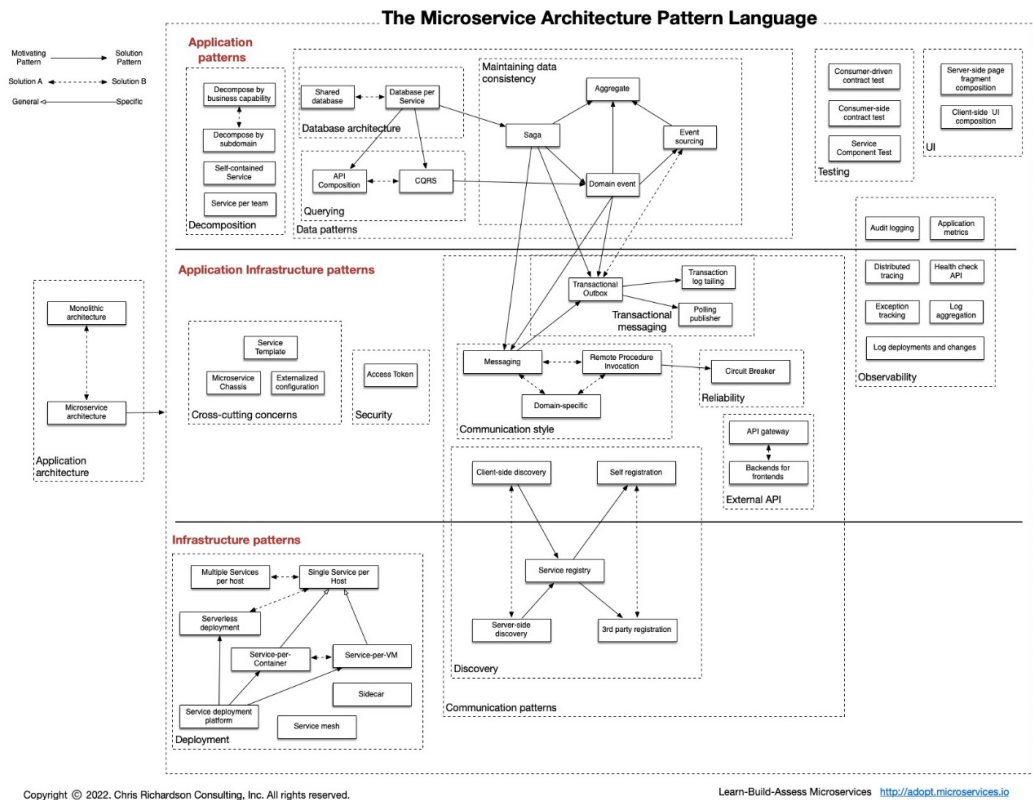
gRPC merupakan salah satu *framework RPC open source* yang dikembangkan oleh *Google*. Pada *gRPC* aplikasi klien dapat secara langsung memanggil metode pada aplikasi server pada mesin yang berbeda seolah-olah itu adalah objek lokal. Dan menjadi mudah membuat aplikasi dan layanan pun terdistribusi (Habibi Mustafa, 2019).



Gambar 2.1 Mekanisme *gRPC*

2.2.4 Microservice

Desain arsitektur untuk membuat sebuah aplikasi yang terdiri dari berbagai unit layanan tersendiri tapi tetap saling terhubung. Setiap unit layanan dalam aplikasi tersebut menjalankan fungsi berbeda, tapi tetap mendukung satu sama lain (Benefita, 2022).



Copyright © 2022, Chris Richardson Consulting, Inc. All rights reserved.

Learn-Build-Assess Microservices <http://adopt.microservices.io>

Gambar 2.2 Konsep *Microservices*

2.2.5 Golang

Golang adalah bahasa pemrograman yang diketik dan dikompilasi secara statis dan menghasilkan kode biner mesin. Menariknya, bahasa pemrograman yang satu ini bersifat *open source*. Golang dihimpun dan ditulis menggunakan bahasa pemrograman C maka, tak heran jika banyak orang menganggap Golang adalah bahasa pemrograman C di abad ke-21. Selain itu, tak heran pula jika banyak orang jadi tertarik untuk belajar Golang (Yalantis, 2022). Bahasa golang banyak digunakan lantaran bahasa golang adalah bahasa yang bisa dipakai untuk

kebutuhan apa saja, seperti web, aplikasi, dll. Serta kemudahan terhadap golang untuk *build* yang tidak membutuhkan *syntax* yang rumit.

2.2.6 Cloud Computing

Menurut indonesiancloud (2022), dinamakan *Cloud Computing* karena informasi yang diakses secara remote di “awan” atau ruangan virtual. Perusahaan-perusahaan yang menyediakan layanan cloud, memungkinkan para penggunanya menyimpan *file* dan aplikasi dari server jarak jauh. Mereka juga bisa mengaksesnya asalkan ada sambungan internet. Ini berarti, seorang pengguna tidak perlu berada di tempat tertentu untuk mendapatkan akses *file*-nya.

Cloud computing bisa bersifat *public* atau *private*. *Public cloud* menyediakan layanannya secara publik di internet. Sementara di lain sisi, *private cloud* hanya menyediakan layanannya ke orang-orang tertentu. Juga ada opsi *hybrid*, yang mengombinasikan baik *public cloud* maupun *private cloud*.

2.2.7 MySQL

MySQL adalah sebuah *database* management system (manajemen basis data) menggunakan perintah dasar SQL (Structured Query Language) yang merupakan suatu bahasa yang dipakai di dalam pengambilan data pada relational *database* atau *database* yang terstruktur. Jadi MySQL adalah *database* management system yang menggunakan bahasa SQL sebagai bahasa penghubung antara perangkat lunak aplikasi dengan *database* server (Yasin K, 2022).

2.2.8 GORM

Gorm adalah sebuah library dengan menggunakan bahasa pemrograman *golang* untuk mendukung penggunaan SQL (Structure Query Language). GORM dapat mendukung aplikasi *database* seperti MySQL, PostgreSQL dan SQLite (Alvin Khair, 2020).

2.2.9 Firebase

Firebase adalah platform pengembangan aplikasi mobile dan web yang disediakan oleh Google. Platform ini memberikan berbagai layanan dan alat untuk membantu pengembang membangun, mengelola, dan memperbarui aplikasi dengan lebih mudah. Firebase menawarkan berbagai fitur, termasuk penyimpanan data real-time, autentikasi pengguna, analitik, pembayaran, dan masih banyak lagi.

2.2.10 Redis

Redis adalah sebuah penyimpanan *in-app-memory* yang berarti penyimpanan sementara pada memory, tujuan dari adanya redis ini seperti untuk cache data, maupun untuk menyimpan data yang tidak butuh di simpan di database