

Optimasi Disaster Recovery Planning (DRP) pada Site Recovery Manager (SRM) Menggunakan Algoritma Genetika

Yoga Putra Setiawan¹, Setiyo Budiyo²

Program Studi Magister Teknik Elektro,
Fakultas Teknik,
Universitas Mercubuana, Jakarta
¹yogaputrasetiawan326@gmail.com, ²sbudiyanto@mercubuana.ac.id,

Ringkasan

Bencana teknologi informasi (TI) yang tak terduga dapat mengganggu operasional bisnis, terutama jika tidak ada persiapan yang matang. Dampak dari gangguan ini bisa sangat besar, termasuk hilangnya data dan informasi penting yang mengancam kelangsungan operasional. Untuk menghadapinya, analisis potensi bencana dan perencanaan *Disaster Recovery Planning* (DRP) yang baik sangat diperlukan. Salah satu tantangan utama dalam DRP adalah menentukan prioritas pemulihan mesin virtual dengan efisien serta mengelola *Recovery Point Objective* (RPO) secara optimal, mengingat keterbatasan waktu yang ada dalam proses pemulihan. Penelitian ini mengusulkan penerapan algoritma genetika untuk mengoptimalkan DRP pada *VMware Site Recovery Manager* (SRM). Algoritma ini digunakan untuk menentukan prioritas pemulihan VM berdasarkan *Service Level Agreement* (SLA), melalui tahapan pembentukan kromosom, inialisasi populasi, evaluasi *fitness*, seleksi, *crossover*, dan mutasi, untuk mencapai solusi pemulihan yang optimal. Evaluasi *fitness* difokuskan pada pengurangan *downtime* dan minimisasi gangguan operasional. Hasil penelitian menunjukkan bahwa algoritma genetika dapat mengurangi *downtime* IT secara signifikan, dengan peningkatan efisiensi pemulihan hingga 50%. Model ini terbukti efektif dalam mengoptimalkan SRM dan dapat menjadi acuan bagi strategi pemulihan bencana TI di masa depan.

Kata kunci: Optimasi *downtime*, *Recovery Time Objective*, DRP, Algoritma Genetika

1. Pendahuluan

Dalam era digital yang terus berkembang, teknologi informasi (TI) menjadi elemen krusial dalam menjaga kelangsungan bisnis. Infrastruktur TI yang kuat diperlukan untuk memastikan kelancaran operasional, terutama dengan pusat data sebagai komponen utama dalam penyimpanan dan pengelolaan data perusahaan [1], [2], [3], [4]. Namun, berbagai ancaman terhadap integritas dan ketersediaan data semakin kompleks, mulai dari bencana alam, serangan siber, hingga kegagalan perangkat keras [2], [5], [6]. Oleh karena itu, perusahaan harus memiliki strategi pemulihan bencana (*Disaster Recovery Planning/DRP*) yang komprehensif untuk menjaga kesinambungan operasional ketika gangguan terjadi [7], [8].

Salah satu pendekatan yang umum diterapkan dalam DRP adalah *VMware Site Recovery Manager* (SRM), yang memungkinkan replikasi otomatis antara situs utama dan cadangan, serta mempercepat pemulihan sistem setelah gangguan terjadi. Dengan implementasi

SRM, perusahaan dapat mengurangi *downtime* yang berpotensi menyebabkan kerugian finansial maupun reputasi. Studi IBM pada 2020 menyebutkan bahwa 40% gangguan layanan dapat mengakibatkan kerugian hingga \$1 juta, sehingga penerapan strategi DRP yang optimal menjadi hal yang sangat penting [5]. Meskipun SRM telah terbukti efektif, masih terdapat tantangan dalam optimalisasi waktu pemulihan dan keandalan sistem ketika menghadapi skenario bencana yang lebih kompleks. Penelitian sebelumnya lebih banyak membahas penerapan dasar DRP dengan SRM, namun masih minim eksplorasi terhadap optimalisasi SRM untuk mencapai pemulihan yang lebih cepat dan andal.

Oleh karena itu, diperlukan pendekatan inovatif dalam meningkatkan efisiensi SRM, salah satunya melalui algoritma genetika yang mampu mengoptimalkan pemilihan jalur pemulihan serta parameter kritis lainnya, seperti *Recovery Time Objective* (RTO) dan *Recovery Point Objective* (RPO) [6], [9], [10]. Algoritma genetika, yang terinspirasi dari teori evolusi, menawarkan solusi dalam mengatasi tantangan optimasi DRP dengan menentukan strategi pemulihan terbaik berdasarkan evaluasi sejumlah skenario dan variabel [11], [12], [13]. Dengan pendekatan ini, diharapkan sistem dapat menghasilkan solusi pemulihan yang lebih efisien, mempercepat proses pemulihan, serta mempertahankan kualitas dan keandalan sistem [14].

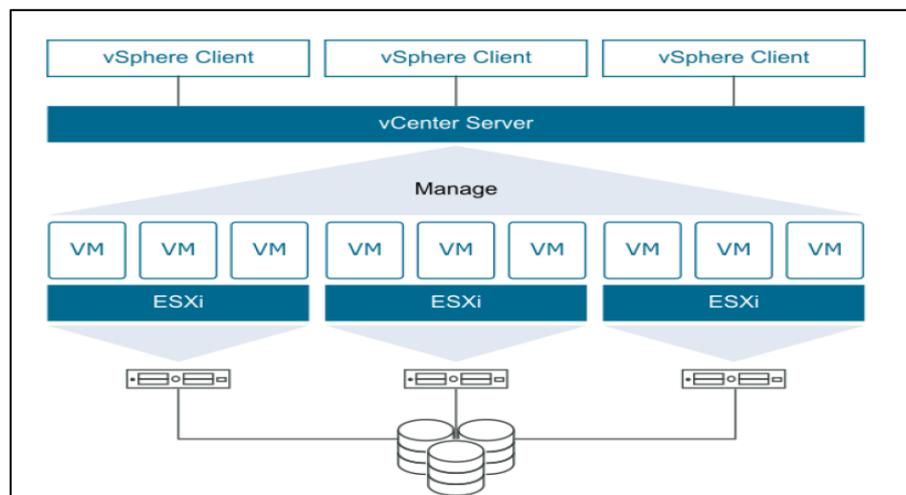
Optimalisasi DRP berbasis algoritma genetika memiliki aplikasi praktis yang luas. Perusahaan yang menerapkan model ini dapat merancang strategi pemulihan yang lebih terarah dan disesuaikan dengan kebutuhan operasional serta risiko yang dihadapi. Implementasi strategi ini tidak hanya mengurangi *downtime*, tetapi juga meminimalkan kerugian finansial dan dampak negatif terhadap reputasi perusahaan [15]. Penelitian ini bertujuan untuk mengoptimalkan DRP melalui penerapan algoritma genetika pada SRM untuk meningkatkan efisiensi dan keandalan pemulihan bencana TI. Selain mengisi celah dalam literatur terkait, model ini diharapkan dapat memberikan solusi praktis bagi organisasi dalam menyusun strategi pemulihan bencana yang lebih efektif dan hemat biaya.

2. Tinjauan Pustaka

2.1. VMware vCenter Server

VMware vCenter Server adalah perangkat lunak manajemen sumber daya dan pemantauan terpusat yang digunakan untuk infrastruktur virtual VMware vSphere. VMware vCenter Server memungkinkan administrator vSphere untuk mengelola semua sumber daya virtual dari satu tempat, termasuk server ESXi, VM, penyimpanan, dan jaringan. VMware vCenter Server juga memiliki banyak fitur yang membantu administrator memantau dan mengelola kinerja infrastruktur virtual mereka, serta mengotomatiskan sejumlah tugas [16].

Gambar 1 merupakan arsitektur vCenter Server yang terdiri atas tiga komponen utama: vSphere Web Client, vCenter Server Database, dan vCenter Single Sign-On. Aplikasi web vSphere berfungsi sebagai antarmuka pengguna untuk vCenter Server, memungkinkan administrator untuk mengelola instalasi, menangani objek inventaris dalam penyebaran vSphere, dan memberikan akses konsol ke VM. Database vCenter Server menyimpan dan mengelola data yang berkaitan dengan vCenter Server.



Gambar 1. Topologi VMware vCenter Server

2.2. Disaster Recovery

Pemulihan bencana (*Disaster Recovery*, DR) merujuk pada strategi dan teknik yang digunakan oleh organisasi untuk mengembalikan akses dan fungsi *infrastruktur* teknologi informasi setelah terjadinya bencana, termasuk bencana alam atau serangan siber. Pemulihan bencana melibatkan berbagai metode yang biasanya terdokumentasikan dalam rencana pemulihan bencana, yang merupakan komponen penting dari upaya kelangsungan bisnis. Di dalam konteks ini, penggandaan data (*data replication*) dan penyimpanan data serta pemrosesan komputer di tempat yang aman menjadi sangat krusial untuk memastikan pemulihan yang efektif [17].

2.2.1. Recovery Point Objective

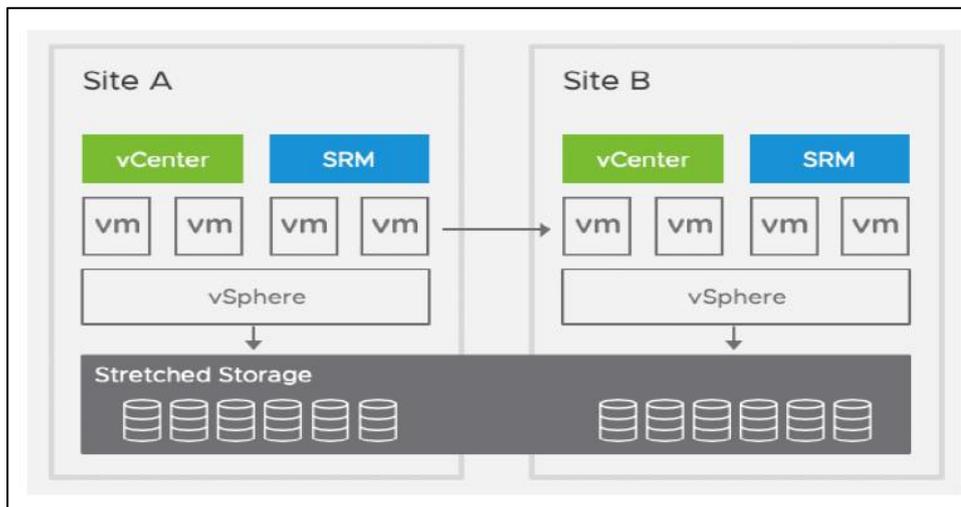
Recovery Point Objective (RPO) merupakan banyaknya data yang dapat hilang atau diizinkan setelah terjadinya sebuah *disaster* TI. Contoh kasus jika terjadi *disaster* TI pada jam 12.00, maka sistem harus dapat *recovery* data pada sistem sebelumnya, yaitu pada pukul 11.00 [3].

2.2.2. Recovery Time Objective

Recovery Time Objective (RTO) merupakan waktu yang dibutuhkan oleh sebuah aplikasi atau sistem untuk dapat aktif kembali menjadi sebuah layanan yang dapat diakses setelah terjadinya sebuah *disaster* TI. Nilai RTO bergantung kepada perintah yang diatur untuk mengembalikan pengaturan pada *backup* data yang sudah dilakukan sebelumnya. Contoh kasus jika sebuah sistem memiliki RTO 5 Jam dan *disaster* terjadi pada pukul 15.00, maka sebuah sistem harus dapat aktif kembali menjadi sebuah layanan pada pukul 20.00 [3].

2.3. VMware Site Recovery Manager

Sebagai bagian dari *VMware vCenter*, *VMware Site Recovery Manager* (SRM) membantu merencanakan, menguji, dan menjalankan pemulihan mesin *virtual* antara situs server *vCenter* yang dilindungi dan situs server *vCenter* pemulihan. SRM juga menyediakan migrasi situs dan pemulihan bencana untuk memastikan kontinuitas operasi [5].



Gambar 2. VMware Site Recovery Manager Topologi

Gambar 2 menjelaskan topologi SRM yang dapat digunakan untuk menerapkan berbagai jenis pemulihan dari situs yang dilindungi ke situs pemulihan dengan menggunakan mekanisme replikasi untuk mengurangi kehilangan data dan waktu tunggu sistem [17].

2.4. Quality Of Service (QOS)

Quality Of Service (QOS) merupakan sebuah metode pengukuran pada sebuah perusahaan dalam memenuhi harapan pelanggan dan menjawab kebutuhan mereka. Nilai dari QOS ini digunakan untuk mengukur seberapa maksimal hasil optimasi dari sebuah jaringan yang dilakukan [18]

Pengukuran nilai *High Availability* dilakukan dengan persamaan berikut:

$$HA = \left(1 - \frac{RTO}{Total\ Waktu}\right) \times 100\% \quad (1)$$

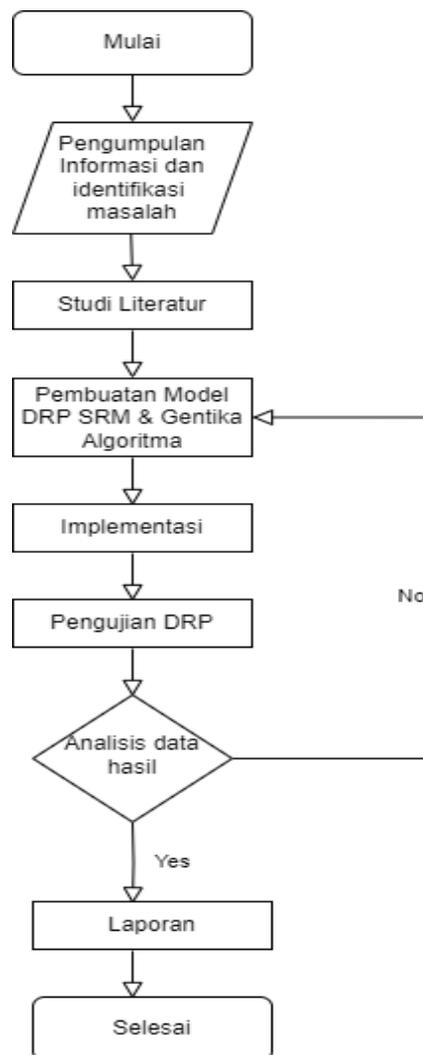
2.5. Algoritma Genetika

Algoritma genetika merupakan sebuah teknik pencarian dalam ilmu komputer yang dapat menemukan penyelesaian perkiraan untuk perbaikan atau optimalisasi sebuah masalah. Algoritma genetika merupakan sebuah perkembangan dari algoritma evolusioner dengan menggunakan teknik yang terinspirasi dari evolusi biologi seperti warisan, mutasi, seleksi alam, dan rekombinasi [19].

Algoritma genetika dikembangkan pertama kali oleh seorang ahli yang bernama John Holland pada tahun 1970-an di Amerika Serikat yang berhasil melahirkan buku berjudul "*Adaption in Natural and Artificial system*" di tahun 1975 [19].

3. Metode Penelitian

Diagram kerja penelitian ditunjukkan pada Gambar 3. Proses penelitian dilakukan melalui beberapa tahapan utama untuk mengembangkan dan mengimplementasikan model *Disaster Recovery Planning* (DRP) dengan memanfaatkan *VMWare Site Recovery Manager* (SRM) yang dioptimalkan menggunakan algoritma genetika.



Gambar 3. Diagram Kerja Penelitian

Penelitian ini dimulai dengan pengumpulan informasi dan identifikasi masalah terkait infrastruktur teknologi informasi, khususnya tantangan dalam penerapan *Disaster Recovery Planning* (DRP) yang efektif. Langkah berikutnya adalah studi literatur untuk memahami konsep DRP, *VMware Site Recovery Manager* (SRM), dan algoritma genetika. Berdasarkan hasil studi tersebut, dirancang model DRP yang mengintegrasikan SRM dengan algoritma genetika untuk mengoptimalkan parameter, seperti *Recovery Point Objective* (RPO) dan *Recovery Time Objective* (RTO).

Model yang dikembangkan kemudian diimplementasikan dan diuji dengan simulasi. Pengujian ini bertujuan untuk mengevaluasi performa model dalam mengurangi *downtime* dan meningkatkan efisiensi pemulihan. Hasil pengujian dianalisis untuk memastikan model memenuhi kriteria keberhasilan yang telah ditentukan. Jika diperlukan, revisi dilakukan hingga model optimal tercapai. Penelitian ini diakhiri dengan penyusunan laporan yang merangkum proses, hasil, dan rekomendasi sebagai kontribusi bagi pengembangan strategi DRP yang lebih andal dan efisien.

Untuk memperkuat pemahaman terhadap konsep yang digunakan dalam metode penelitian ini, berikut adalah penjelasan beberapa istilah dan terminologi utama yang

menjadi dasar dalam proses pengembangan dan implementasi strategi DRP berbasis algoritma genetika:

- a. *Service Level Agreement (SLA)* adalah perjanjian formal antara penyedia layanan TI dan pengguna, yang menetapkan tingkat layanan minimum yang harus dipenuhi. Dalam penelitian ini, SLA digunakan sebagai dasar pengelompokan VM berdasarkan tingkat kepentingan operasionalnya. VM yang memiliki SLA tinggi akan diprioritaskan dalam proses pemulihan karena menjalankan layanan yang kritikal bagi bisnis.
- b. *Recovery Time Objective (RTO)* menggambarkan waktu maksimum yang dapat ditoleransi untuk memulihkan sistem atau layanan setelah terjadi gangguan. Semakin kecil nilai RTO, semakin cepat layanan harus dipulihkan agar tidak mengganggu operasional.
- c. *Recovery Point Objective (RPO)* adalah batas maksimum kehilangan data yang dapat diterima akibat bencana, dihitung dari titik terakhir *backup* berhasil dilakukan. Nilai RPO yang rendah menunjukkan kemampuan sistem dalam menjaga integritas data yang tinggi.
- d. *VM Critical vs Non-Critical*: Dalam konteks ini, VM diklasifikasikan menjadi dua kelompok. *VM critical* adalah mesin *virtual* yang menjalankan aplikasi inti dan vital bagi organisasi, sehingga memiliki prioritas tinggi dalam proses *recovery* sedangkan *VM non-critical* adalah mesin *virtual* yang mendukung operasi sekunder dan memiliki toleransi lebih tinggi terhadap keterlambatan pemulihan.
- e. Iterasi 50 Kali pada Algoritma Genetika: Jumlah iterasi ini dipilih berdasarkan hasil uji awal serta referensi yang relevan, yang menunjukkan bahwa nilai *fitness* dalam algoritma genetika cenderung mencapai kestabilan setelah 50 generasi. Dengan batas ini, proses optimasi dianggap cukup untuk menghasilkan solusi mendekati optimal, tanpa membebani proses komputasi secara signifikan.

3.1. Implementasi Algoritma Genetika

Untuk melakukan perancangan algoritma genetika dilakukan persiapan awal, yaitu membuat diagram kerja algoritma genetika untuk dilakukan proses dalam uji penerapan, perancangan, dan analisis *disaster recovery planning VMware SRM*, seperti yang ditunjukkan pada Gambar 4. Implementasi model algoritma genetika yang telah dikembangkan diimplementasikan pada lingkungan simulasi untuk menguji keefektifannya dalam mengurangi *downtime* dan meningkatkan efisiensi pemulihan. Pengolahan data dilakukan melalui tahapan berikut:

1. Inisialisasi Populasi: Populasi awal dibentuk dari sejumlah solusi acak yang direpresentasikan dalam bentuk kromosom. Kromosom ini merepresentasikan urutan prioritas VM berdasarkan pengelompokan *critical* dan *non-critical* berdasarkan SLA. Masing-masing kromosom adalah kombinasi dari identitas VM seperti berikut:

$$K = \text{Shuffle (C)} + \text{Shuffle (N)} \quad (2)$$

Dengan $C = \{1,2,3,4,5\}$ adalah VM *critical* dan $N = \{6,7,8\}$ adalah VM *non-critical*.

2. Evaluasi Fitness: Setiap individu dalam populasi dievaluasi menggunakan fungsi kebugaran (*fitness function*) yang mempertimbangkan alokasi sumber daya (CPU, Memori, dan *disk*) dan posisi dalam kromosom sebagai indikator prioritas. Fungsi ini dirumuskan sebagai:

$$F(C) = \sum_{i=0}^4 (5 - i)(2vi + 1,5mi + 1,2di) + \sum_{i=5}^7 (8 - i)(1vi + 0,75mi + 0,6di) \quad (3)$$

dimana:

v_i = Jumlah CPU pada VM ke- i

m_i = Jumlah Memori

d_i = Kapasitas *Disk*

Bobot $5 - i$ (untuk *VM Critical*) dan $8 - i$ (untuk *VM Non Critical*) yang menunjukkan prioritas posisi kromosom.

3. Seleksi: Seleksi dilakukan menggunakan metode *elitism* dan *roulette wheel*, dimana individu dengan nilai *fitness* tertinggi akan memiliki probabilitas lebih besar untuk dipilih. Probabilitas seleksi dihitung dengan:

$$P(x_i) = \frac{f(x_i)}{\sum_{j=1}^n f(x_j)} \quad (4)$$

Dua individu akan diteruskan tanpa perubahan, sementara individu lain akan digunakan untuk membentuk generasi baru.

4. *Crossover*: Dua kromosom (*parent*) yang dipilih akan disilangkan pada suatu titik secara acak untuk menghasilkan keturunan baru (*offspring*). Proses ini dapat dirumuskan dengan sebagai :

$$\text{Offspring} = [x_1, x_2, \dots, x_k] + [y_{k+1}, \dots, y_n] \quad (5)$$

dimana x dan y adalah *parent* 1 dan 2, dan k adalah titik potong yang dipilih secara acak.

5. Mutasi: Untuk menjaga keragaman populasi, dilakukan mutasi acak dengan menukar dua gen dalam satu kromosom. Secara matematis dapat ditulis:

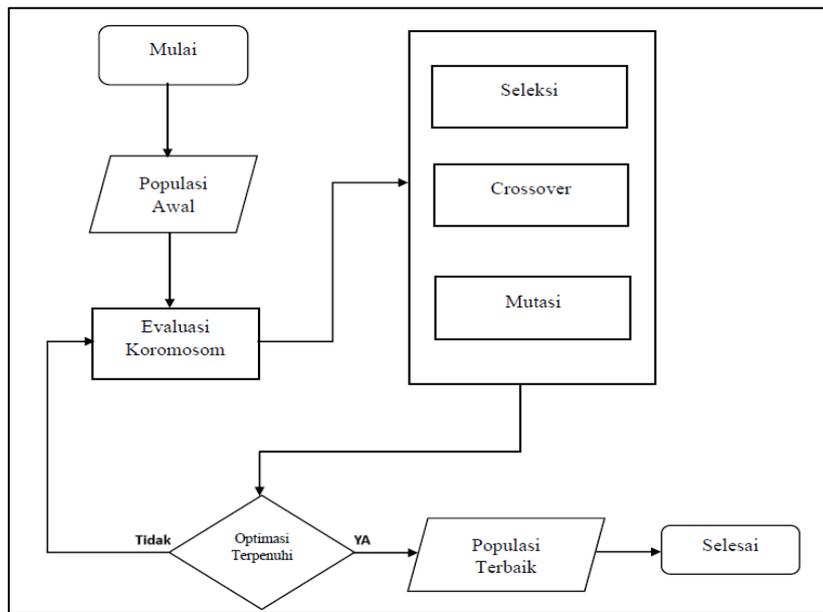
$$X_i \leftrightarrow X_j \quad (6)$$

Mutasi dilakukan dengan probabilitas tertentu untuk menghindari konvergensi pada solusi local.

6. Iterasi: Mengulangi proses seleksi, *crossover*, dan mutasi hingga populasi diperbarui sebanyak 50 kali. Proses ini akan berhenti ketika nilai kebugaran (*fitness value*) mencapai tingkat optimal atau ketika jumlah iterasi maksimum tercapai.

Pengujian dengan metode ini juga dapat digunakan untuk melihat pendekatan baru genetika algoritma dengan menilai bagaimana keunggulan dari sistem ini dan kekurangannya. Hal ini dapat dilihat dalam proses uji coba yang dilakukan setelah implemementasi, apakah sistem ini mampu memberikan efisiensi dan efektifitas yang baik untuk sebuah organisasi. Proses ini juga dapat melihat seberapa jauh peluang dan ancaman yang muncul dalam sistem ini sehingga hal tersebut dapat dijadikan antisipasi oleh sebuah perusahaan dan organisasi dalam menekan *cost* perusahaan untuk ke depannya.

Dalam fase pengujian dan penerapan *VMware SRM*, penulis akan mengidentifikasi aspek keunggulan, seperti waktu pulih sebuah sistem dan biaya yang dibutuhkan dari sistem tersebut, yang mencakup kemampuannya dalam menyelenggarakan pemulihan sistem secara cepat dan efisien, serta adaptasinya terhadap berbagai kondisi darurat.



Gambar 4. Diagram Kerja Algoritma Genetika

4. Hasil

Pada penelitian ini, eksperimen pertama dilakukan dengan mencari prioritas pada masing-masing *server*, kemudian setelah mendapatkan prioritas *server* tersebut, dilakukan optimasi dengan mencari nilai RPO terbaik berdasarkan SLA yang berlaku. Tabel 1 menunjukkan spesifikasi SLA, prioritas *server*, dan hasil RPO yang didapatkan dari optimasi algoritma genetika.

Tabel 1. Spesifikasi *Server* hasil Optimasi Algoritma Genetika

VMName	Spesifikasi				Prioritas	RPO (menit)
	vCpu	Memori	Disk	Severity		
VMTESTING-1	4	6GB	150GB	Critical	1	5
VMTESTING-2	2	2GB	100GB	Critical	3	45
VMTESTING-3	2	2GB	120GB	Critical	4	60
VMTESTING-4	4	4GB	200GB	Critical	2	10
VMTESTING-5	2	4GB	100GB	Critical	5	640
VMTESTING-6	4	4GB	250GB	Non Critical	8	720
VMTESTING-7	2	4GB	80GB	Non Critical	6	360
VMTESTING-8	4	6GB	180GB	Non Critical	7	720

Setelah diperoleh nilai pada Tabel 1, masing-masing *server* akan dikelompokkan berdasarkan tingkat prioritasnya, yaitu kategori kritis dan tidak kritis. *Server* yang telah dikelompokkan ini akan dimasukkan kedalam *protection group* pada *Site Recovery Manager* sebelum dilakukan pengukuran, seperti yang ditunjukkan pada Gambar 5.

Grup pada masing-masing juga dikelompokkan ke dalam prioritas nilai untuk membuktikan *recovery* berjalan sesuai dengan hasil dari algoritma genetika, dapat dilihat pada Gambar 6.

Sebelum dilakukan simulasi *disaster*, masing-masing *server* juga diatur nilai RPO-nya berdasarkan hasil optimasi dari algoritma genetika yang sudah didapatkan, seperti terlihat pada Gambar 7.

Optimasi Disaster Recovery Planning (DRP) pada Site Recovery Manager (SRM)
Menggunakan Algoritma Genetika
Yoga Putra Setiawan, Setiyo Budiyo

Non Critical VM						
Virtual Machine	Protection Status	Recovery Resource Pool	Recovery Host	Recovery Folder	Recovery Network	VMotion Eligible
VMTESTING-6	OK	DC Cluster	mbmjktex2.m...	Discovered virtual m	VM Network	N/A
VMTESTING-7	OK	DC Cluster	mbmjktex2.m...	Discovered virtual m	VM Network	N/A
VMTESTING-8	OK	DC Cluster	mbmjktex2.m...	Discovered virtual m	VM Network	N/A

Critical VM						
Virtual Machine	Protection Status	Recovery Resource Pool	Recovery Host	Recovery Folder	Recovery Network	VMotion Eligible
VMTESTING-1	OK	DC Cluster	mbmjktex1.mb...	Discovered virtual m	VM Network	N/A
VMTESTING-2	OK	DC Cluster	mbmjktex3.m...	Discovered virtual m	VM Network	N/A
VMTESTING-3	OK	DC Cluster	mbmjktex3.m...	Discovered virtual m	VM Network	N/A
VMTESTING-4	OK	DC Cluster	mbmjktex3.m...	Discovered virtual m	VM Network	N/A
VMTESTING-5	OK	DC Cluster	mbmjktex1.mb...	Discovered virtual m	VM Network	N/A

Gambar 5. Pengelompokan Server berdasarkan SLA

6. Power on priority 1 VMs	6.1. VMTESTING-1
7. Power on priority 2 VMs	7.1. VMTESTING-4
8. Power on priority 3 VMs	8.1. VMTESTING-2
9. Power on priority 4 VMs	9.1. VMTESTING-3
10. Power on priority 5 VMs	10.1. VMTESTING-5

6. Power on priority 1 VMs	6.1. VMTESTING-7
7. Power on priority 2 VMs	7.1. VMTESTING-8
8. Power on priority 3 VMs	8.1. VMTESTING-6

Gambar 6. Pengelompokan Server berdasarkan Prioritas

Virtual Machine	Status	RPO	Target	Replication Server	Protection Group
VMTESTING-1	OK	5 minutes	MBMJKTVCDCR01.mbma.c...	MBMJKTVRDR02	Critical VM
VMTESTING-2	OK	45 minutes	MBMJKTVCDCR01.mbma.c...	MBMJKTVRDR02	Critical VM
VMTESTING-3	OK	1 hour	MBMJKTVCDCR01.mbma.c...	MBMJKTVRDR02	Critical VM
VMTESTING-4	OK	10 minutes	MBMJKTVCDCR01.mbma.c...	MBMJKTVRDR02	Critical VM
VMTESTING-5	OK	8 hours	MBMJKTVCDCR01.mbma.c...	MBMJKTVRDR02	Critical VM
VMTESTING-6	OK	12 hours	MBMJKTVCDCR01.mbma.c...	MBMJKTVRDR02	Non Critical VM
VMTESTING-7	OK	6 hours	MBMJKTVCDCR01.mbma.c...	MBMJKTVRDR02	Non Critical VM
VMTESTING-8	OK	12 hours	MBMJKTVCDCR01.mbma.c...	MBMJKTVRDR02	Non Critical VM

Gambar 7. Nilai RPO pada masing-masing Server

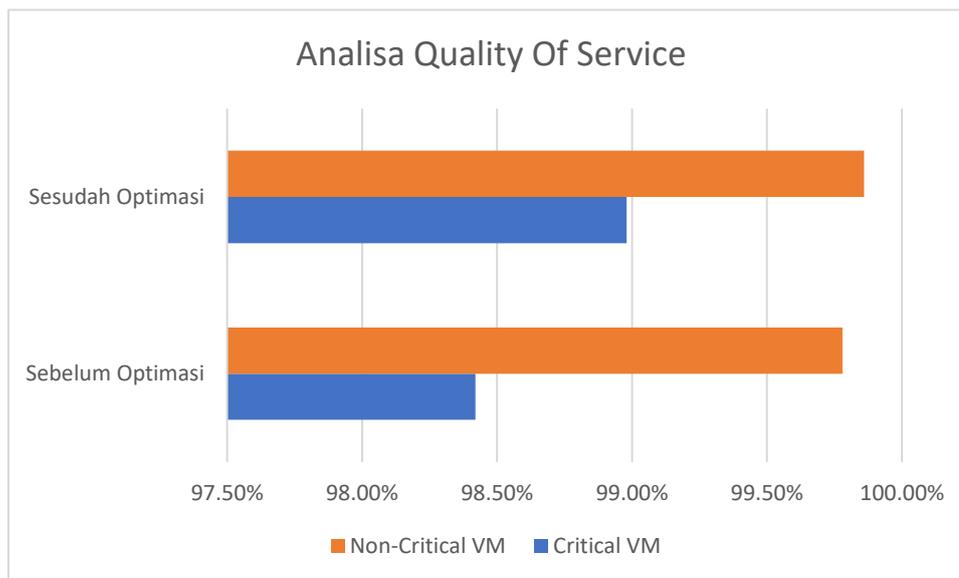
Setelah dilakukan pengelompokan pada masing-masing server berdasarkan hasil implementasi pada algoritma genetika, dilakukan pengukuran dengan menggunakan skenario *downtime* apabila terjadi *disaster* pada suatu sistem TI. Hasil penelitian ini menunjukkan bahwa optimasi *Disaster Recovery Planning* (DRP) dengan algoritma genetika pada *VMware Site Recovery Manager* (SRM) mampu meningkatkan efisiensi pemulihan, baik untuk *virtual machine* (VM) kritis maupun non-kritis. Pengukuran dilakukan dengan asumsi terjadinya bencana menyeluruh pada organisasi, dimana VM kritis menjadi prioritas utama dalam proses pemulihan. Hasilnya, semua server dalam grup kritis berhasil dipulihkan dengan status "OK" dan siap digunakan kembali. Proses pemulihan untuk grup kritis berlangsung selama 22 menit 45 detik, dimulai pada pukul 20:34:16,

menunjukkan peningkatan efisiensi yang signifikan dibandingkan dengan skenario sebelum optimasi. Untuk VM non-kritis, proses pemulihan dilakukan setelah VM kritis berhasil dipulihkan. *Server* dalam grup ini juga menunjukkan status "OK" sebelum pemulihan dan berhasil menjalani skenario *failover* dengan baik. Pengukuran menunjukkan bahwa proses pemulihan VM non-kritis berlangsung selama 3 menit 8 detik, dimulai pada pukul 21:20:42. Meskipun waktu pemulihan lebih singkat dibandingkan VM kritis, efektivitasnya tetap terukur melalui perhitungan *recovery point objective* (RPO) dan *recovery time objective* (RTO), yang dioptimalkan melalui algoritma genetika

Hasil perbandingan menunjukkan bahwa penerapan algoritma genetika memberikan dampak positif pada efisiensi pemulihan, terutama dalam mengurangi *downtime* untuk VM kritis hingga 36% dan VM non-kritis hingga 37%. Dengan pendekatan ini, sistem dapat mengalokasikan sumber daya secara lebih optimal berdasarkan tingkat *criticality*, memastikan bahwa VM yang paling berpengaruh terhadap operasional organisasi dapat dipulihkan terlebih dahulu. Penelitian ini membuktikan bahwa metode optimasi berbasis algoritma genetika tidak hanya meningkatkan kecepatan pemulihan tetapi juga memberikan solusi yang lebih adaptif dalam skenario DRP yang kompleks. Tabel 2 berikut menjelaskan secara rinci hasil perbandingan sebelum dan sesudah dilakukan optimasi DRP menggunakan algoritma genetika pada *Site Recovery Manager*.

Tabel 2. Perbandingan Hasil Sebelum dan Sesudah Optimasi

VMName	SLA	Sebelum Optimasi		Sesudah Optimasi	
		Durasi Recovery	RTO (detik)	Durasi Recovery	RTO (detik)
VMTESTING-1	Critical	22 menit 45 detik	366	14 Menit 34 detik	88
VMTESTING-2			254		163
VMTESTING-3			497		209
VMTESTING-4			383		383
VMTESTING-5			244		224
VMTESTING-6	Non Critical	3 menit 8 detik	58	1 menit 58 detik	33
VMTESTING-7			70		28
VMTESTING-8			61		33



Gambar 8. Quality of Service

Hasil tersebut membuktikan bahwa optimasi dengan menggunakan algoritma genetika dapat mempengaruhi nilai waktu *downtime* yang tentunya dapat secara efektif digunakan pada sebuah organisasi perusahaan untuk menyusun rencana pemulihan bencana yang efektif. Hasil tersebut dapat dijadikan sebagai acuan mengingat banyaknya proses *service* aplikasi lain yang terganggu ketika *disaster recovery* terjadi sehingga membuat waktu *downtime* juga semakin lama. Penggunaan algoritma genetika dapat dijadikan salah satu opsi efektifitas dalam mengurangi *disaster recovery* tersebut. Dari hasil optimasi juga dilakukan pengukuran *Quality of Service* untuk melihat seberapa jauh hasil optimasi dapat mempengaruhi nilai. Gambar 6 menunjukkan peningkatan *service* layanan pada *disaster recovery planning* menggunakan algoritma genetika.

4. Kesimpulan

Penelitian ini berhasil mengembangkan model algoritma genetika untuk optimasi *Disaster Recovery Planning* (DRP) pada *VMware Site Recovery Manager* (SRM), yang secara signifikan meningkatkan efisiensi dan keandalan pemulihan bencana. Model ini mampu mengurangi downtime VM kritis dari 22 menit 45 detik menjadi 14 menit 34 detik (36%) dan VM non-kritis dari 3 menit 8 detik menjadi 1 menit 58 detik (37%), membuktikan efektifitasnya dalam menganalisis serta mengoptimasi parameter DRP. Dengan mengintegrasikan tingkat prioritas *server* sebagai faktor utama, pendekatan ini memberikan solusi sistematis dalam memilih opsi pemulihan yang cepat dan efisien.

Hasil penelitian menunjukkan bahwa pendekatan ini berpotensi diterapkan dalam berbagai skenario TI yang kompleks, khususnya di industri yang membutuhkan pemulihan data cepat dan andal, seperti sektor keuangan dan kesehatan. Model ini juga membuka peluang penelitian lebih lanjut, termasuk eksplorasi penerapan pada *platform* virtualisasi lain *server* integrasi dengan teknologi lebih modern, seperti *hybrid cloud* dan *machine learning* untuk meningkatkan fleksibilitas dan prediktibilitas dalam perencanaan DRP.

Daftar Pustaka

- [1] M. Yenugula, S. K. Sahoo, and S. S. Goswami, "Cloud computing for sustainable development: An analysis of environmental, economic and social benefits," *J. Futur. Sustain.*, vol. 4, no. 1, pp. 59–66, 2024, doi: 10.5267/j.jfs.2024.1.005.
- [2] Derick Musundi Kesa, "Ensuring resilience: Integrating IT disaster recovery planning and business continuity for sustainable information technology operations," *World J. Adv. Res. Rev.*, vol. 18, no. 3, pp. 970–992, 2023, doi: 10.30574/wjarr.2023.18.3.1166.
- [3] Aureliu ZGUREANU, "the Role of Rpo and Rto in Disaster Recovery Planning Rolul Rto Şi Rpo În Planificarea Recuper Ārii În Caz De," *Conf. Int. Sci. Of, Years Reformsthe, Econ. Of, Repub. Progress, Econ. Innov. V I A*, no. august 1930, pp. 221–232, 2022.
- [4] A. Friyanto and S. I. Lestaringati, "Design and Implementation of Virtual Laboratory Infrastructure for Indonesia Skill Competition using Cloud Computing," in *2023 International Conference on Informatics Engineering, Science & Technology (INCITEST)*, 2023, pp. 1–4. doi: 10.1109/INCITEST59455.2023.10397061.
- [5] K. Elgdamsi and M. Embarak, "Implementing a Disaster Recovery Solution for Datacenters Using VMware Site Recovery Manager," vol. 04, no. 01, pp. 1–22, 2023.
- [6] A. Chaudhuri, R. K. Behera, and P. K. Bala, "Factors impacting cybersecurity transformation: An Industry 5.0 perspective," *Comput. Secur.*, vol. 150, p. 104267, 2025, doi: <https://doi.org/10.1016/j.cose.2024.104267>.

- [7] W. Sardjono, W. G. Perdana, and G. R. Putra, "Disaster recovery plan implementation evaluation model at the corporation," *Procedia Comput. Sci.*, vol. 234, pp. 1658–1663, 2024, doi: <https://doi.org/10.1016/j.procs.2024.03.170>.
- [8] R. Vallabhaneni and V. Veeramachaneni, "Developing a Disaster Recovery Plan to Enhance Corporate Resilience and Ensure Business Continuity," no. October, 2024, doi: 10.47191/etj/v9i10.11.
- [9] B. Guerrero Granados, C. G. Quintero M, and C. V. Núñez, "Improved genetic algorithm approach for coordinating decision-making in technological disaster management," *Neural Comput. Appl.*, vol. 36, no. 9, pp. 4503–4521, 2024, doi: 10.1007/s00521-023-09218-0.
- [10] D. A. Shafiq, N. Z. Jhanjhi, A. Abdullah, and M. A. Alzain, "A Load Balancing Algorithm for the Data Centres to Optimize Cloud Computing Applications," *IEEE Access*, vol. 9, pp. 41731–41744, 2021, doi: 10.1109/ACCESS.2021.3065308.
- [11] M. Shameem, M. Nadeem, and A. T. Zamani, "Genetic algorithm based probabilistic model for agile project success in global software development," *Appl. Soft Comput.*, vol. 135, p. 109998, 2023, doi: <https://doi.org/10.1016/j.asoc.2023.109998>.
- [12] Q. Wang, "Cloud Data Backup and Recovery Method Based on the DELTA Compression Algorithm," in *2021 IEEE International Conference on Industrial Application of Artificial Intelligence (IAAI)*, 2021, pp. 183–188. doi: 10.1109/IAAI54625.2021.9699943.
- [13] S. Liu and N. Wang, "Collaborative Optimization Scheduling of Cloud Service Resources Based on Improved Genetic Algorithm," *IEEE Access*, vol. 8, pp. 150878–150890, 2020, doi: 10.1109/ACCESS.2020.3016762.
- [14] P. Anand, M. Sharma, and A. Saroliya, "Comparative analysis of RTO and RPO in Amazon Web Service and Google Cloud Platform and DRaaS," in *2023 10th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON)*, 2023, pp. 1903–1907. doi: 10.1109/UPCON59197.2023.10434639.
- [15] Mehul Pawar and A. S. Phapale, "Enhancing Data Backup and Recovery in Cloud Computing with Secure Database Monitoring," *Int. J. Appl. Adv. Multidiscip. Res.*, vol. 1, no. 4, pp. 319–326, 2023, doi: 10.59890/ijaamr.v1i4.594.
- [16] I. VMware, "vCenter Server and Host Management," pp. 11–15, 2021.
- [17] B. A. Mulyani, "Analisa Infrastruktur Data Center Virtualisasi Dan Disaster Recovery Berbasis Site Recovery Manager Dalam Pemenuhan Service Level Agreement Pada Pt Xyz," *J. UMT*, pp. 8–22, 2018, [Online]. Available: <http://jurnal.umt.ac.id/index.php/jte/article/download/636/435>
- [18] E. M. Sinaga, "The influence of price on purchase decision with quality of service as intervening variable," *JPPi (Jurnal Penelit. Pendidik. Indones.)*, vol. 9, no. 1, p. 125, 2023, doi: 10.29210/020221734.
- [19] S. Saleh, N. Umar, and M. A. Nur, "Implementasi Algoritma Genetika Untuk Penjadwalan Ujian Pada Universitas Handayani Makassar," *Pros. SISFOTEK*, pp. 154–156, 2023, [Online]. Available: <http://www.seminar.iaii.or.id/index.php/SISFOTEK/article/view/402%0Ahttp://www.seminar.iaii.or.id/index.php/SISFOTEK/article/download/402/334>