BAB II
KAJIAN LITERATUR

2.1. Penjadwalan

Penjadwalan (*scheduling*) merupakan konsep yang sering dijumpai dalam kehidupan sehari-hari, terutama dalam bentuk jadwal kegiatan seperti jadwal transportasi, jadwal perkuliahan, maupun jadwal kerja. Jadwal pada dasarnya merupakan rencana yang menunjukkan kapan suatu aktivitas akan dilaksanakan atau diselesaikan. Dengan adanya jadwal, seseorang dapat mengetahui waktu pelaksanaan suatu kegiatan serta urutan aktivitas yang harus dilakukan untuk mencapai suatu tujuan tertentu. Secara umum, penjadwalan dapat dipahami sebagai proses penyusunan atau penentuan waktu pelaksanaan berbagai aktivitas dengan memanfaatkan sumber daya yang tersedia. Proses ini melibatkan penentuan urutan pekerjaan (*sequencing*) serta waktu mulai dan selesai setiap aktivitas. Dalam praktiknya, penjadwalan sering dilakukan melalui dua tahapan utama, yaitu menentukan urutan tugas yang harus dikerjakan terlebih dahulu dan menentukan waktu pelaksanaan dari setiap tugas tersebut [13].

Dalam berbagai organisasi, penjadwalan melibatkan dua komponen utama, yaitu tugas (*tasks*) dan sumber daya (*resources*). Tugas merupakan aktivitas yang harus diselesaikan, sementara sumber daya merupakan sarana yang digunakan untuk melaksanakan tugas tersebut, seperti tenaga kerja, mesin, atau fasilitas lainnya [14]. Permasalahan penjadwalan muncul ketika sejumlah tugas harus diselesaikan menggunakan sumber daya yang terbatas sehingga diperlukan pengaturan waktu dan urutan pekerjaan yang tepat agar seluruh aktivitas dapat berjalan secara efisien. Kondisi tersebut menjadikan penjadwalan sebagai bagian dari permasalahan optimasi yang bertujuan mencari solusi terbaik dalam pengalokasian sumber daya terhadap berbagai aktivitas dengan mempertimbangkan berbagai batasan seperti waktu pelaksanaan, kapasitas sumber daya, serta hubungan ketergantungan antar tugas [15].

Permasalahan serupa juga ditemukan dalam berbagai bidang lain, seperti penjadwalan kegiatan olahraga, khususnya pada pertandingan sepak bola, di mana setiap tim harus dijadwalkan untuk bertanding tanpa mengalami bentrok waktu dengan mempertimbangkan berbagai batasan seperti ketersediaan stadion, waktu pertandingan, serta keadilan jumlah pertandingan. Penjadwalan pertandingan sepak bola termasuk dalam kategori permasalahan optimasi yang kompleks karena melibatkan banyak variabel dan

constraint yang harus dipenuhi secara bersamaan [16]. Permasalahan ini juga dikenal sebagai bagian dari *sports scheduling problem* yang banyak diteliti dalam bidang optimasi.

Dalam konteks organisasi pelayanan seperti gereja, konsep penjadwalan ini juga diterapkan dalam pengaturan jadwal pelayanan ibadah. Penjadwalan dilakukan dengan mengalokasikan petugas ke dalam berbagai divisi pelayanan seperti musik, *worship leader* (WL), *singer*, dan *choir*, sesuai dengan kebutuhan setiap sesi ibadah. Setiap kegiatan ibadah memiliki waktu dan kebutuhan sumber daya yang berbeda, sehingga penyusunan jadwal harus mempertimbangkan ketersediaan pelayan, peran atau kemampuan yang dimiliki, serta pemerataan pembagian tugas agar pelayanan dapat berjalan dengan baik.

Pada GBI Pelita Medan, proses penjadwalan pelayanan dilakukan secara berkala untuk mengatur petugas dalam beberapa sesi ibadah setiap minggunya. Contoh jadwal pelayanan tersebut dapat dilihat pada Gambar 2.1, yang menunjukkan pembagian tim pelayanan ke dalam beberapa kategori seperti musik, *Worship Leader* (WL), *Singer*, dan *Choir* pada berbagai waktu ibadah. Melalui gambar tersebut, terlihat bahwa setiap sesi ibadah memiliki kombinasi petugas yang berbeda-beda, sehingga penyusunan jadwal harus mempertimbangkan banyak aspek seperti ketersediaan pelayan, pembagian tugas yang merata, serta penghindaran bentrok jadwal. Kompleksitas ini menjadi semakin tinggi ketika jumlah pelayan yang terlibat semakin banyak dan jadwal ibadah semakin beragam, sehingga penyusunan jadwal secara manual menjadi kurang efisien dan berpotensi menimbulkan konflik.

Tanggal	IBRA 1-2 Jam 09.00 & 11.00					IBRA 3 (LICC) - Ibra 4 JAM 14.00 & 16.00				IBRA 5-6 - 18.00 & 20.00				NextGen	
	Musik		WL	Singer	Choir	WL	Singer	Choir		Musik	WL	Singer	Choir	Vokal	Musik
01-Mar-26	Piano	Erickson	Meli	Helmin		Desi	Ester			Kristhoven	Elis	Grace Moko		Yolanda	Darmin
	Filler	Mey		Theresya		Mey	Marietta			Desi		Cecilia			
	Bass	Misael		Desi		Adi P	Yogi			Adriel		Clay		Team NG	Hendri
	Drum	Jamie		Imel		Perlita	Aby			Yoas		Adi H			Meyland
	Gitar/Saxo	Iban													
	Soundman:					Soundman:						Soundman:			
	EXCELLENT WORSHIPER GBI PELITA MEDAN														
08-Mar-26	Piano	Desi	Zivo	Hesta		Desi	Clay			Kristhoven	Meyland	Debora		Grace Moko	Dandi
	Filler	Mey		Valdy		Darman	Agustina			Dea		Hari			
	Bass	Ian		Evie		Misael	Yogi			Dewald		Cecilia		Team NG	Adriel
	Drum	Ninik		Serta		Steven	Aby			Yoas		Mery			Justin
	Gitar/Saxo														
	Soundman: Dandi					Soundman : Putra / Daud				Soundman : Putra					
	EXCELLENT WORSHIPER GBI PELITA MEDAN														
15-Mar-26	Piano	Desi	Meli	Adi H		Erickson	Zivo			Kristhoven	Odde	Ester		Marketta	Adi P
	Filler	Mey		Serta		Mey	Hesta			Desi		Marietta			
	Bass	Adi P		Theresya		Jhon	Cecilia			Adi P		Clay		Team NG	Hendri
	Drum	Imanuel		Mery		Dean	Imel			Eman		Yogi			Dean
	Gitar/Saxo					Hendri				Dandi					
	Soundman: Putra					Soundman: Dandi				Soundman : Daud / Putra					
	EXCELLENT WORSHIPER GBI PELITA MEDAN														
22-Mar-26	Piano	Desi	Odde	Serta		Desi	Marietta			Erickson	Meli	Imel		Yolanda	Darmin
	Filler	Mey		Roria		Mey	Herlin			Kristhoven		Hafi			
	Bass	Ian		Cecilia		Oswald	Delfia			Adi P		Yogi		Team NG	Dandi
	Drum	Ninik		Merdiani		Steven	Agustina			Yoas		Clay			Immanuel
	Gitar/Saxo														
	Soundman: Putra					Soundman: Dandi				Soundman: Dandi					
	Gitar/Saxo														
29-Mar-26	Piano	Erickson	Meyland	Valdy		Desi	Yogi			Desi	Odde	Zivo		Merdiani	Desi
	Filler	Mey		Evie		Mey	Herlin			Darman		Hesta			
	Bass	Adi P		Dewi		Jhon	Agustina			Dewald		Grace Moko		Team NG	Misael
	Drum	Perlita		Delfia		Steven	Adi H			Yoas		Roria			Justin
	Gitar/Saxo					Hendri				Dandi					
	Soundman: Dandi					Soundman: Putra / Daud				Soundman: Daud / Putra					

Gambar 2. 1 Jadwal Pelayan GBI Pelita Medan

Berdasarkan kompleksitas tersebut, permasalahan penjadwalan dapat dikategorikan sebagai permasalahan optimasi kombinatorial yang cukup rumit karena melibatkan banyak

variabel dan batasan yang harus dipenuhi secara bersamaan. Oleh karena itu, diperlukan suatu metode yang mampu membantu menghasilkan jadwal secara optimal dengan mempertimbangkan berbagai batasan yang ada. Salah satu metode yang dapat digunakan untuk menyelesaikan permasalahan tersebut adalah Algoritma Genetika, yang merupakan metode optimasi berbasis evolusi yang mampu mencari solusi terbaik melalui proses seleksi, dan mutasi berdasarkan nilai *fitness* dari setiap solusi yang dihasilkan.

2.2. Algoritma Genetika

Algoritma Genetika (*Genetic Algorithm,* GA) merupakan salah satu metode optimasi yang terinspirasi oleh mekanisme evolusi biologis yang diperkenalkan oleh John Holland pada tahun 1975. Algoritma ini digunakan untuk menemukan solusi optimal dalam masalah pencarian dan optimasi. Algoritma Genetika bekerja dengan cara menciptakan populasi solusi yang disebut individu, yang direpresentasikan dalam bentuk kromosom [17]. Setiap individu dievaluasi berdasarkan fungsi kecocokan (*fitness function*) untuk menentukan seberapa baik solusi tersebut dalam memenuhi kriteria yang ditetapkan. Algoritma Genetika menjadi salah satu metode yang paling populer karena kemampuannya dalam menjelajahi ruang solusi yang luas dan menghasilkan jadwal dengan tingkat konflik rendah [18]. Algoritma Genetika memiliki kemampuan eksplorasi ruang solusi yang baik sehingga mampu mencari solusi optimal pada ruang pencarian yang luas serta mengurangi kemungkinan terjebak pada solusi optimum lokal [19]. Selain itu, operasi genetika seperti mutasi membuat proses pencarian solusi menjadi lebih adaptif terhadap perubahan kondisi maupun batasan (*constraints*) yang ada.

2.2.1. Istilah dan Komponen Kunci

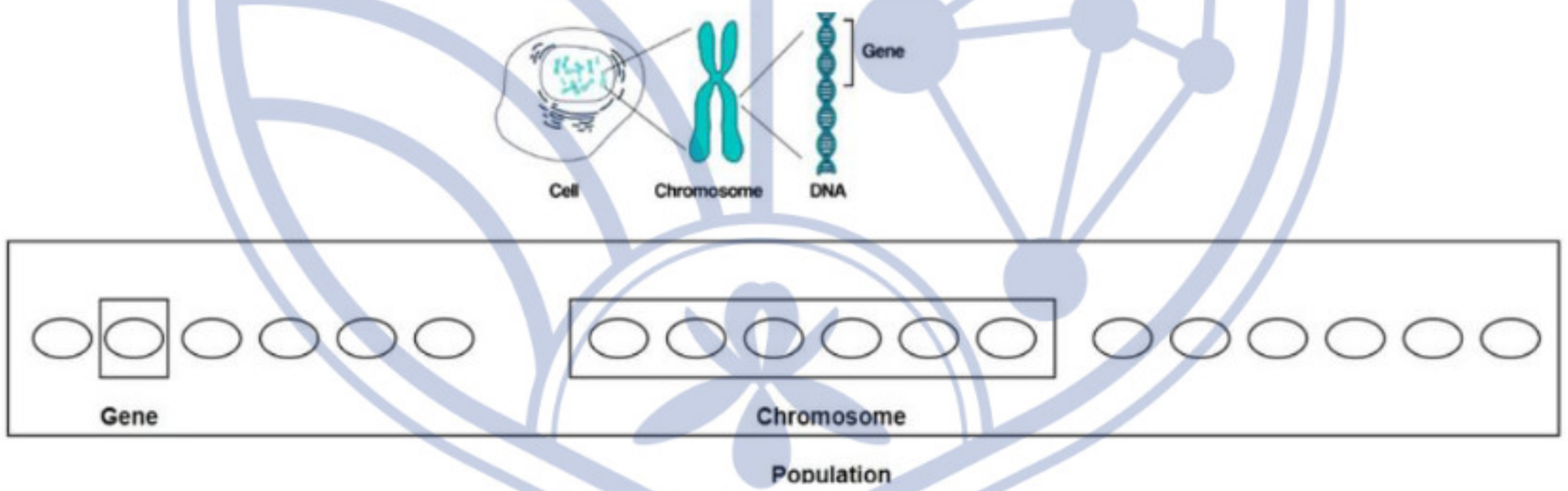
Dalam Algoritma Genetika terdapat beberapa istilah dan komponen utama yang perlu dipahami, yaitu gen, kromosom, populasi, generasi, fungsi *fitness,* serta operator genetika seperti seleksi, dan mutasi. Gen merupakan unit informasi terkecil yang merepresentasikan suatu parameter atau variabel dalam solusi. Dalam konteks sistem penjadwalan pelayanan gereja, gen dapat merepresentasikan satu elemen jadwal, seperti penugasan seorang *Volunteer* pada waktu dan divisi tertentu. Kromosom adalah representasi lengkap dari satu solusi kandidat yang terdiri dari kumpulan gen. Satu kromosom dapat menggambarkan satu susunan jadwal pelayanan secara utuh dalam satu periode tertentu. Struktur kromosom inilah akan dievaluasi untuk menentukan kualitas solusi [20]. Populasi

adalah sekumpulan kromosom yang dibangkitkan pada satu waktu tertentu. Algoritma Genetika tidak bekerja pada satu solusi saja, melainkan mengevaluasi banyak solusi sekaligus dalam bentuk populasi untuk meningkatkan peluang menemukan solusi terbaik. Generasi merupakan satu siklus proses evolusi yang terdiri dari evaluasi *fitness,* seleksi, *crossover*, dan mutasi. Setiap generasi menghasilkan populasi baru yang diharapkan memiliki kualitas solusi lebih baik dibandingkan generasi sebelumnya [21].

Fungsi *fitness* digunakan untuk mengukur tingkat kecocokan atau kualitas suatu kromosom terhadap tujuan optimasi yang ditetapkan. Dalam penelitian ini, fungsi *fitness* dapat dirancang untuk meminimalkan konflik jadwal, menghindari benturan tugas, serta memastikan pemerataan pelayanan. Operator genetika terdiri dari beberapa mekanisme utama, yaitu:

1. Seleksi, yaitu proses pemilihan kromosom terbaik berdasarkan nilai *fitness* untuk dijadikan induk.
2. *Crossover*, yaitu proses pertukaran sebagian gen antar dua kromosom induk untuk menghasilkan kromosom baru.
3. Mutasi, yaitu perubahan acak pada gen tertentu untuk menjaga keberagaman populasi dan mencegah solusi terjebak pada kondisi optimum lokal.

Hubungan antara gen, kromosom, dan populasi dalam Algoritma Genetika dapat dilihat pada Gambar 2.2.



Gambar 2.2 *Representasi, Gen* dan *Chromosome* pada Algoritma Genetika[22]

Gambar 2.2 menunjukkan bahwa gen merupakan bagian penyusun kromosom, dan sekumpulan kromosom membentuk populasi yang akan mengalami proses evolusi secara iteratif hingga diperoleh solusi terbaik.

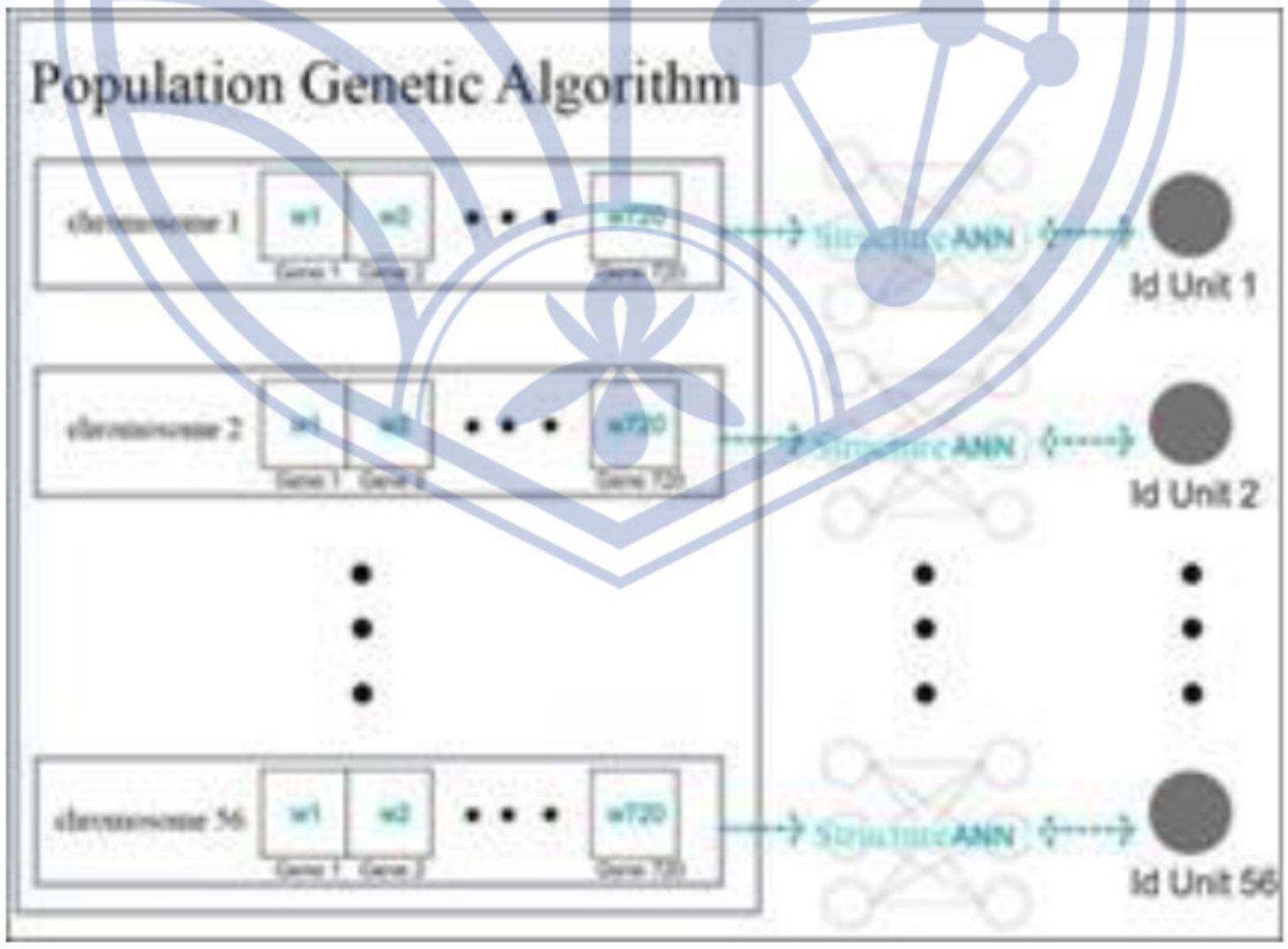
2.2.2. Representasi Gen dan Kromosom

Dalam Algoritma Genetika, gen merupakan unit informasi terkecil yang merepresentasikan karakteristik atau nilai tertentu dalam sebuah solusi. Sementara itu,

kromosom merupakan representasi dari solusi potensial untuk masalah yang sedang dipecahkan. Setiap kromosom terdiri dari gen-gen yang menyimpan informasi penting mengenai solusi tersebut [19]. Dalam konteks penjadwalan, setiap gen menyimpan informasi terkait elemen penugasan, seperti *Volunteer,* divisi pelayanan, dan waktu pelayanan, sementara kromosom merepresentasikan satu susunan jadwal pelayanan secara keseluruhan dalam satu periode tertentu.

Pada penelitian ini, *gen* merepresentasikan satu elemen penugasan pelayanan yang mencakup nama *Volunteer,* jenis pelayanan, serta waktu pelayanan. Kombinasi gen dalam satu kromosom membentuk berbagai kemungkinan susunan jadwal pelayanan, sehingga memungkinkan Algoritma Genetika untuk mengeksplorasi berbagai alternatif solusi penjadwalan. Oleh karena itu, perancangan struktur gen dan kromosom yang tepat menjadi faktor penting dalam menghasilkan solusi jadwal yang optimal serta meminimalkan konflik penjadwalan.

Sebagai ilustrasi, representasi gen dan kromosom biasanya digambarkan dalam bentuk deretan gen yang membentuk sebuah kromosom, di mana setiap gen mewakili variabel keputusan dalam sistem penjadwalan. Melalui struktur tersebut, proses seleksi, *crossover*, dan mutasi dapat dilakukan untuk menghasilkan generasi solusi yang lebih baik. Ilustrasi ini dapat dilihat pada Gambar 2.3 yang menunjukkan hubungan antara gen sebagai unit data dan kromosom sebagai representasi solusi secara keseluruhan.



Gambar 2.3 Representasi Kromosom terhadap Unit Data[23]

2.2.3. Populasi dan Generasi

Populasi dalam Algoritma Genetika merupakan sekumpulan kromosom yang merepresentasikan beberapa kandidat solusi dalam suatu proses optimasi. Setiap kromosom terdiri dari sekumpulan gen yang memyimpan informasi mengenai solusi yang dihasilkan [24]. Berbeda dengan metode optimasi yang hanya mengevaluasi satu solusi, Algoritma Genetika melakukan pencarian terhadap beberapa solusi secara bersamaan melalui suatu populasi sehingga mampu mengeksplorasi ruang solusi yang luas. Setiap individu dalam populasi akan melalui proses evaluasi menggunakan fungsi *fitness* untuk mengetahui tingkat kualitas solusi yang di hasilkan. Nilai *fitness* digunakan sebagai indikator dalam menentukan seberapa baik satu kromosom dalam memenuhi tujuan optimasi dan batasan (*constrain*) yang telah ditentukan. Kromosom dengan nilai *fitness* yang lebih baik memiliki kualitas solusi yang lebih tinggi dibangingkan kromosom lainnya [24].

Pada Algoritma Genetika konvensional, populasi awal umumnya dibentuk dengan membangkitkan sejumlah kromosom secara acak. Namun, pada penelitian ini proses pembentukan populasi awal menggunakan pendekatan *hybrid initization*. Pendekatan ini diterapkan karena permasalahan penjadwalan pelayanan memiliki banyak batasan yang harus dipenuhi, sehingga pembangkitan solusi secara acak dapat menghasilkan banyak kandidat solusi yang tidak memenuhi *constrain*. Tahapan inisialisasi populasi pada sistem ini dimulai dengan membentuk satu kromosom awal menggunakan metode *Deterministic Scheduler*. Metode tersebut menghasilkan solusi awal berdasarkan aturan penjadwalan yang telah ditentukan, seperti ketersediaan *Volunteer* (*Availabiliti*), kesesuaian kemampuan pelayanan (*skill)*, serta batas jumlah pelayanan setiap *Volunteer*. Kromosom awal yang dihasilkan digunakan sebagai *baseline solution* dalam proses pembentukan populasi.

Generasi merupakan siklus proses evolusi dalam Algoritma Genetika yang dilakukan secara berulang untuk meningkatkan kualitas solusi. Pada setiap generasi, kromosom dalam populasi akan di evaluasi berdasarkan nilai *fitness*, kemudian dilakukan proses seleksi terhadap individu terbaik untuk membentuk populasi generasi berikutnya [24]. Pada penelitian ini, proses seleksi menggunakan metode seleksi individu terbaik (*elite selection*). Setelah seluruh individu dalam populasi dievaluasi, kromosom akan diurutkan berdasarkan nilai fitness secara menurun (descending). Sebanyak 10 individu dengan nilai fitness tertinggi dari total populasi akan dipilih sebagai individu elit yang digunakan untuk menghasilkan populasi generasi berikutnya.

Berbeda dengan Algoritma Genetika konvensional yang menggunakan *crossover* untuk menghasilkan kombinasi solusi baru, penelitian ini tidak menerapkan proses *crossover*. Hal tersebut dilakukan karena perubahan struktur gen melalui *crossover* berpotensi merusak susunan jadwal yang sebelumnya telah memenuhi *constraint*. Oleh karena itu, proses pembentukan solusi baru dilakukan menggunakan mekanisme mutasi terarah (*Directed Mutation*) terhadap individu terpilih.

Mutasi terarah dilakukan dengan melakukan perubahan pada bagian tertentu dari kromosom berdasarkan kandidat volunteer yang memenuhi kriteria penjadwalan, seperti ketersediaan waktu, kesesuaian skill, dan jumlah pelayanan yang telah diterima. Pendekatan ini memungkinkan proses pencarian solusi tetap memiliki variasi, namun tetap mempertahankan kualitas solusi yang telah diperoleh sebelumnya. Proses evolusi dilakukan secara berulang hingga mencapai kondisi terminasi yang telah ditentukan oleh sistem. Melalui proses tersebut, Algoritma Genetika mampu menghasilkan solusi jadwal pelayanan dengan nilai fitness yang lebih baik serta meminimalkan pelanggaran terhadap batasan penjadwalan yang telah ditetapkan.

2.2.4. Fungsi *fitness*

Fungsi *fitness* adalah mekanisme untuk mengukur kualitas suatu kromosom dalam Algoritma Genetika, di mana semakin tinggi nilainya, semakin baik solusi yang dihasilkan. Fungsi ini menjadi dasar seleksi, karena kromosom dengan nilai *fitness* lebih tinggi berpeluang lebih besar untuk berkembang pada generasi berikutnya [26], [27]. Pada penelitian ini, fungsi *fitness* digunakan untuk menilai susunan penjadwalan pelayan berdasarkan kriteria seperti tidak adanya bentrok jadwal, penugasan sesuai ketersediaan waktu, terpenuhinya jumlah petugas tiap divisi, dan pembagian tugas yang merata. Semakin sedikit pelanggaran terhadap kriteria tersebut, semakin tinggi nilai *fitness,* sehingga solusi yang dihasilkan memiliki konflik minimal dan distribusi tugas yang lebih adil.

Dalam penerapannya, fungsi *fitness* sering dirancang berdasarkan kombinasi beberapa kriteria atau *constraint* yang harus dipenuhi dalam sistem penjadwalan. Setiap pelanggaran terhadap *constraint* biasanya akan diberikan nilai penalti, sehingga nilai *fitness* akan menurun jika solusi tidak memenuhi kriteria yang ditentukan. Pendekatan ini memungkinkan Algoritma Genetika untuk secara bertahap memperbaiki kualitas solusi melalui proses evolusi hingga diperoleh jadwal yang optimal [28].

Secara umum, fungsi *fitness* dalam penelitian ini adalah sebagai berikut [29]:

$$Fitness = \sum_{i=1}^{n} Rewar\, d_i - \sum_{i=1}^{n} Penalt\, y_i - Penalty_{fairness} \quad \text{.....(1)}$$

Dimana:

$Fitness$ = Nilai kualitas suatu kromosom yang menunjukkan tingkat kelayakan solusi

Σ Reward$_i$ = Total nilai tambahan dari kondisi solusi yang sesuai dengan kriteria ideal.

Σ Penalty$_i$ = Total nilai pengurangan akibat pelanggaran constraint pada solusi.

Penalty$_{(fairness)}$ = Nilai pengurangan akibat ketidakseimbangan pembagian tugas antar volunteer.

Pada sistem ini, nilai *reward* diberikan berdasarkan tingkat kesesuaian *volunteer* terhadap kebutuhan pelayanan. *Volunteer* yang ditempatkan sesuai dengan kemampuan utama (*primary skill*) memperoleh nilai reward lebih tinggi dibandingkan penempatan pada kemampuan pendukung (*secondary skill*).

Sementara itu, penalty diberikan terhadap kondisi yang menyebabkan kualitas solusi menurun. Beberapa bentuk penalty yang diterapkan yaitu:

1. *Penalty* kekosongan slot pelayanan

   Diberikan ketika terdapat posisi pelayanan yang belum memiliki *volunteer* sehingga kebutuhan pelayanan belum terpenuhi.

2. *Penalty* batas jumlah pelayanan

   Diberikan apabila seorang volunteer mendapatkan jumlah pelayanan melebihi batas yang telah ditentukan dalam periode tertentu.

3. *Penalty* pelanggaran *availability*

   Diberikan apabila *volunteer* dijadwalkan pada waktu yang tidak tersedia berdasarkan data ketersediaan *volunteer*.

4. *Penalty* bentrok jadwal (*double assignment*)

   Diberikan apabila seorang *volunteer* mendapatkan lebih dari satu tugas pada sesi pelayanan yang sama.

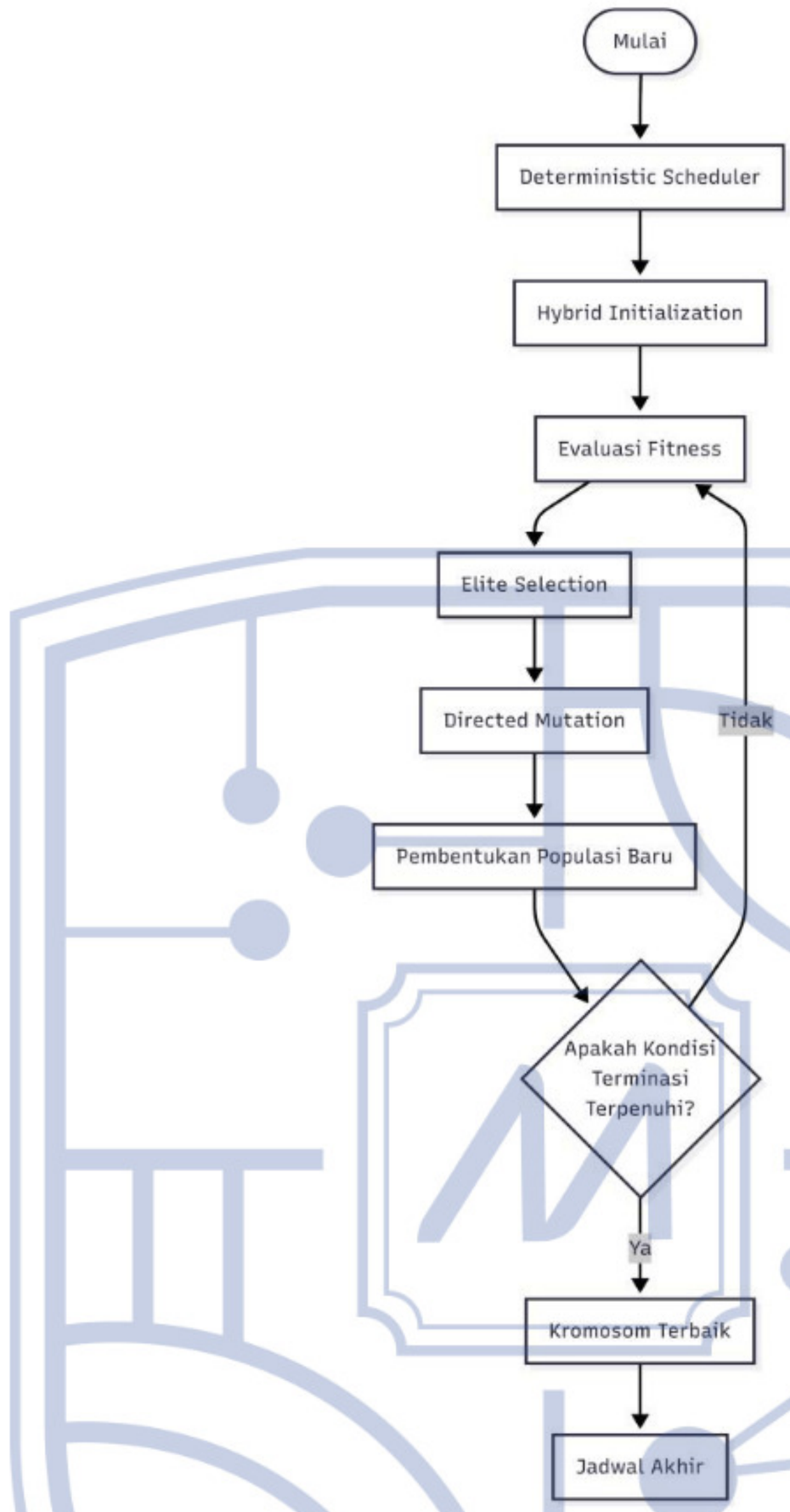
Pada pelanggaran yang bersifat fatal, seperti pelanggaran *availability* dan *double assignment*, sistem memberikan penalti mutlak sebesar -9.999.999. Nilai penalti yang sangat besar tersebut digunakan agar kromosom yang melanggar constraint utama memiliki nilai *fitness* yang sangat rendah sehingga tidak terpilih dalam proses seleksi. Selain mempertimbangkan pemenuhan constraint, sistem juga memperhatikan pemerataan

pembagian tugas melalui *fairness penalty*. Komponen ini digunakan untuk mengurangi kemungkinan terjadinya ketimpangan jumlah pelayanan antar *volunteer*, sehingga hasil jadwal yang diperoleh tidak hanya memenuhi aturan, tetapi juga menghasilkan pembagian pelayanan yang lebih seimbang. Dengan pendekatan fungsi *fitness* berbasis *reward* dan *penalty* tersebut, proses evaluasi tidak hanya berfokus pada jumlah pelanggaran, tetapi juga mempertimbangkan kualitas penempatan *volunteer* dan pemerataan pelayanan. Kromosom dengan nilai *fitness* tertinggi kemudian menjadi kandidat solusi terbaik yang akan dipertahankan pada proses evolusi berikutnya [19].

2.2.5. Siklus Algoritma Genetika

Siklus Algoritma Genetika merupakan proses evolusi yang dilakukan secara berulang (*iterative process*) untuk memperoleh solusi dengan kualitas terbaik berdasarkan nilai *fitness* yang dihasilkan [30]. Secara umum, Algoritma Genetika terdiri atas beberapa tahapan utama, yaitu inisialisasi populasi, evaluasi *fitness*, seleksi, reproduksi melalui operator *crossover* dan mutasi, pembentukan populasi baru, serta terminasi [31]. Namun, pada penelitian ini operator crossover tidak digunakan dan digantikan dengan *Directed Mutation* agar solusi tetap memenuhi constraint penjadwalan. Proses tersebut diulang hingga diperoleh solusi yang memenuhi kriteria yang telah ditentukan.

Pada penelitian ini, siklus Algoritma Genetika dimodifikasi untuk menyesuaikan dengan karakteristik permasalahan penjadwalan pelayanan yang memiliki banyak batasan (*constraint*). Modifikasi dilakukan dengan menggunakan *Hybrid Initialization* melalui *Deterministic Scheduler* sebagai pembentuk populasi awal, menerapkan *Elite Selection* sebagai metode seleksi, serta mengganti operator *crossover* dengan *Directed Mutation*. Modifikasi ini bertujuan menghasilkan solusi yang lebih sesuai dengan kebutuhan penjadwalan sekaligus mengurangi kemungkinan pelanggaran terhadap *constraint*. Alur Algoritma Genetika yang digunakan pada penelitian ini ditunjukkan pada Gambar 2.4.



Gambar 2. 4 Siklus Algoritma Genetika yang digunakan pada penelitian ini (diadaptasi dari [34] dan dimodifikasi oleh penulis)

Tahap pertama dimulai dengan *Deterministic Scheduler*, yaitu proses penyusunan solusi awal berdasarkan aturan dan *constraint* yang telah ditentukan. Solusi awal yang dihasilkan kemudian digunakan pada tahap *Hybrid Initialization* untuk membentuk populasi awal. Berbeda dengan Algoritma Genetika konvensional yang membangkitkan populasi secara acak, pendekatan ini menghasilkan populasi awal yang telah memenuhi *constraint* dasar sehingga proses evolusi menjadi lebih terarah. Variasi antar kromosom selanjutnya diperoleh melalui proses duplikasi dan mutasi sehingga dihasilkan sejumlah alternatif solusi. Setelah populasi awal terbentuk, setiap kromosom dievaluasi menggunakan fungsi *fitness* berbasis *reward*. Nilai *fitness* dihitung berdasarkan beberapa kriteria, yaitu kesesuaian *skill* volunteer dengan kebutuhan pelayanan, ketersediaan volunteer, tidak adanya bentrok

jadwal, serta pemerataan pembagian tugas. Semakin tinggi nilai *fitness*, semakin baik kualitas solusi penjadwalan yang dihasilkan.

Tahap berikutnya adalah *Elite Selection*, yaitu proses memilih individu terbaik berdasarkan nilai *fitness*. Seluruh kromosom diurutkan secara menurun (*descending*) berdasarkan nilai *fitness*, kemudian dipilih sebanyak 10 kromosom dengan nilai *fitness* tertinggi sebagai individu elit. Individu-individu tersebut dipertahankan untuk membentuk generasi berikutnya karena dianggap memiliki kualitas solusi terbaik. Selanjutnya dilakukan *Directed Mutation*, yaitu proses mutasi yang dilakukan secara terarah tanpa menggunakan operator *crossover*. Mutasi dilakukan dengan mempertimbangkan ketersediaan volunteer, kesesuaian *skill*, serta pemerataan jumlah pelayanan yang telah diterima oleh setiap volunteer. Pendekatan ini menghasilkan variasi solusi baru tanpa menghilangkan karakteristik solusi yang telah memiliki nilai *fitness* tinggi, sehingga tetap memenuhi *constraint* utama dalam penjadwalan.

Hasil dari proses mutasi digunakan untuk membentuk populasi baru yang kemudian dievaluasi kembali melalui perhitungan nilai *fitness*. Siklus evaluasi *fitness*, *Elite Selection*, *Directed Mutation*, dan pembentukan populasi baru dilakukan secara berulang hingga kondisi terminasi terpenuhi, misalnya ketika jumlah generasi maksimum telah tercapai atau tidak terjadi peningkatan nilai *fitness* dalam beberapa generasi berturut-turut. Setelah kondisi terminasi tercapai, kromosom dengan nilai *fitness* tertinggi dipilih sebagai solusi akhir berupa jadwal pelayanan yang memiliki tingkat pelanggaran *constraint* paling rendah.

2.3. Aplikasi Berbasis Web

Aplikasi berbasis Web merupakan perangkat lunak yang dapat diakses melalui jaringan internet menggunakan peramban (Web *browser*) tanpa perlu melakukan instalasi secara langsung pada perangkat pengguna. Aplikasi ini berjalan pada Web *server* dan memungkinkan pengguna untuk mengakses sistem dari berbagai perangkat seperti komputer, laptop, maupun smartphone selama terhubung dengan jaringan internet. Penggunaan aplikasi berbasis Web memungkinkan pengelolaan data dilakukan secara terpusat sehingga memudahkan proses penyimpanan, pengolahan, serta distribusi informasi kepada pengguna secara lebih efektif [35].

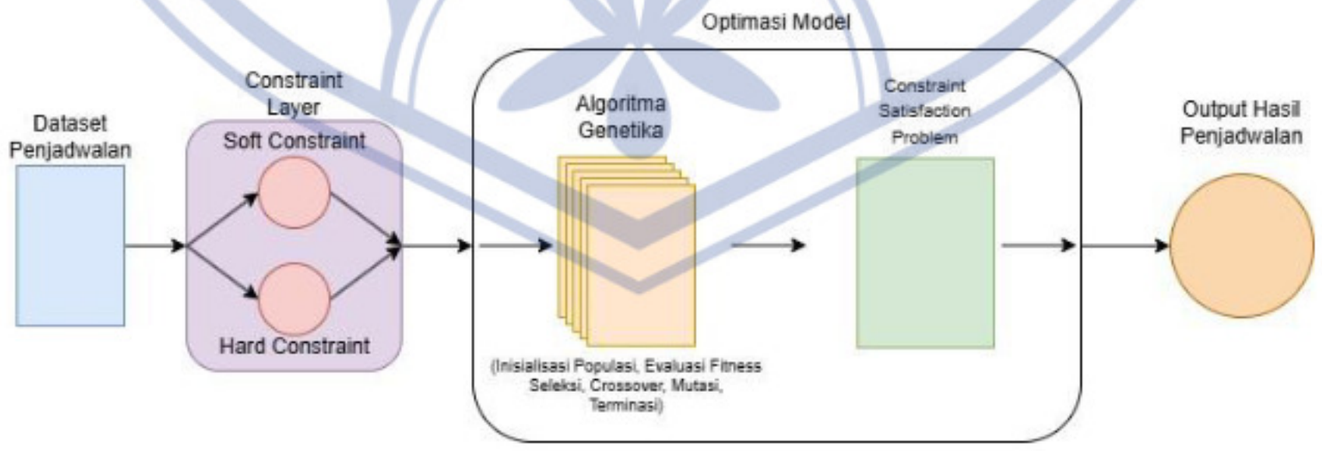
Salah satu keunggulan utama aplikasi berbasis Web adalah kemampuannya dalam mendukung akses *multi-user* secara bersamaan serta kemudahan dalam proses pemeliharaan sistem. Pembaruan sistem dapat dilakukan langsung pada *server* sehingga pengguna tidak

perlu melakukan instalasi atau pembaruan aplikasi pada perangkat masing-masing. Selain itu, aplikasi berbasis Web juga memungkinkan penyajian informasi secara *real-time* sehingga pengguna dapat memperoleh data terbaru dengan lebih cepat dan akurat. Hal ini menjadikan aplikasi berbasis Web banyak digunakan dalam berbagai sistem informasi organisasi, termasuk dalam pengelolaan data dan sistem penjadwalan [36].

Berdasarkan sifat dan cara kerjanya, *website* dapat dibedakan menjadi tiga jenis, yaitu *website* statis, *website* dinamis, dan *website* interaktif. *Website* statis merupakan jenis *website* yang memiliki konten tetap dan jarang mengalami perubahan, sehingga informasi yang disajikan biasanya bersifat sederhana dan tidak memerlukan pengolahan data secara kompleks. *Website* dinamis merupakan *website* yang kontennya dapat berubah secara otomatis berdasarkan data yang tersimpan dalam basis data, sehingga memungkinkan pengelolaan informasi yang lebih fleksibel [36]. Sementara itu, *website* interaktif merupakan *website* yang memungkinkan terjadinya interaksi antara pengguna dan sistem, seperti pengisian formulir, pengolahan data, serta berbagai aktivitas yang melibatkan respon langsung dari sistem terhadap input pengguna [37].

Dalam konteks sistem penjadwalan pelayanan gereja, penggunaan aplikasi berbasis Web dapat membantu pengelolaan jadwal secara lebih efektif dan terstruktur. Sistem berbasis Web memungkinkan pengurus gereja untuk mengelola data *Volunteer,* menyusun jadwal pelayanan, serta membagikan informasi jadwal kepada seluruh pelayan secara lebih cepat dan transparan. Dengan demikian, proses koordinasi antar divisi pelayanan dapat dilakukan dengan lebih efisien serta meminimalkan potensi terjadinya konflik jadwal.

Proses pengolahan data hingga menghasilkan jadwal pelayanan yang optimal dapat digambarkan melalui alur sistem optimasi yang ditunjukkan pada Gambar 2.5.



Gambar 2. 5 Alur Optimasi Penjadwalan Menggunakan Algoritma Genetika[38]

2.4. Unified Modeling Language (UML)

Unified Modeling Language (UML) merupakan bahasa pemodelan standar yang digunakan untuk memvisualisasikan, merancang, dan mendokumentasikan sistem perangkat lunak. UML digunakan untuk membantu pengembang dalam menggambarkan struktur sistem, kebutuhan fungsional, serta alur proses yang terjadi dalam suatu sistem sebelum tahap implementasi dilakukan. Dengan menggunakan UML, rancangan sistem dapat disajikan dalam bentuk model visual yang terstruktur sehingga memudahkan proses analisis dan komunikasi antara pengembang dengan pemangku kepentingan sistem [39]. UML banyak digunakan dalam pengembangan sistem berbasis objek karena mampu memodelkan berbagai aspek sistem seperti interaksi pengguna dengan sistem, alur proses, serta struktur data dalam sistem. Diagram UML dapat membantu pengembang dalam memahami kebutuhan sistem secara lebih jelas sehingga dapat meminimalkan kesalahan pada tahap implementasi sistem [40].

Dalam penelitian ini, UML digunakan untuk memodelkan sistem penjadwalan otomatis multi-divisi pelayanan gereja berbasis Web. Penggunaan UML membantu dalam menggambarkan kebutuhan sistem secara terstruktur sehingga mempermudah proses analisis dan perancangan sistem. Diagram yang digunakan dalam penelitian ini meliputi *Use Case Diagram*, *Activity Diagram*, dan *Entity-Relationship Diagram* (ERD).

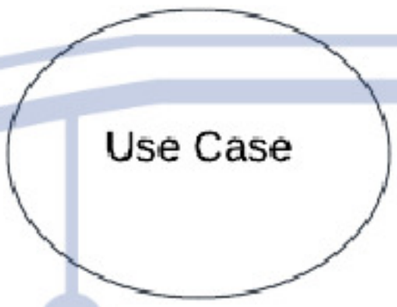
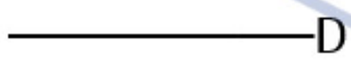
2.4.1. *Use Case* Diagram

Use Case Diagram digunakan untuk menggambarkan interaksi antara aktor dan sistem serta fungsi-fungsi utama yang tersedia dalam sistem. Diagram ini menunjukkan batas sistem dan peran masing-masing aktor, yaitu *Admin*, Kepala Departemen, Ketua Divisi, dan *Volunteer*. *Use Case Diagram* membantu dalam mengidentifikasi kebutuhan fungsional sistem agar seluruh proses penjadwalan dapat terakomodasi dengan baik. *Use Case Diagram* menampilkan batas sistem (*system boundary*) yang membedakan bagian dalam sistem dengan entitas eksternal. Aktor merupakan pihak luar yang berinteraksi dengan sistem, baik berupa manusia maupun sistem lain. Setiap aktor memiliki peran tertentu dan berinteraksi dengan *Use Case* sesuai hak akses atau tanggung jawabnya. Dalam sistem penjadwalan otomatis multi-divisi pelayanan gereja berbasis Web, aktor yang terlibat meliputi: *Admin*, Kepala Departemen, Ketua Divisi, *Volunteer*. Masing-masing aktor memiliki hak akses dan fungsi yang berbeda sesuai dengan kebutuhan sistem. *Use Case Diagram* pada penelitian ini digunakan untuk memastikan bahwa seluruh proses

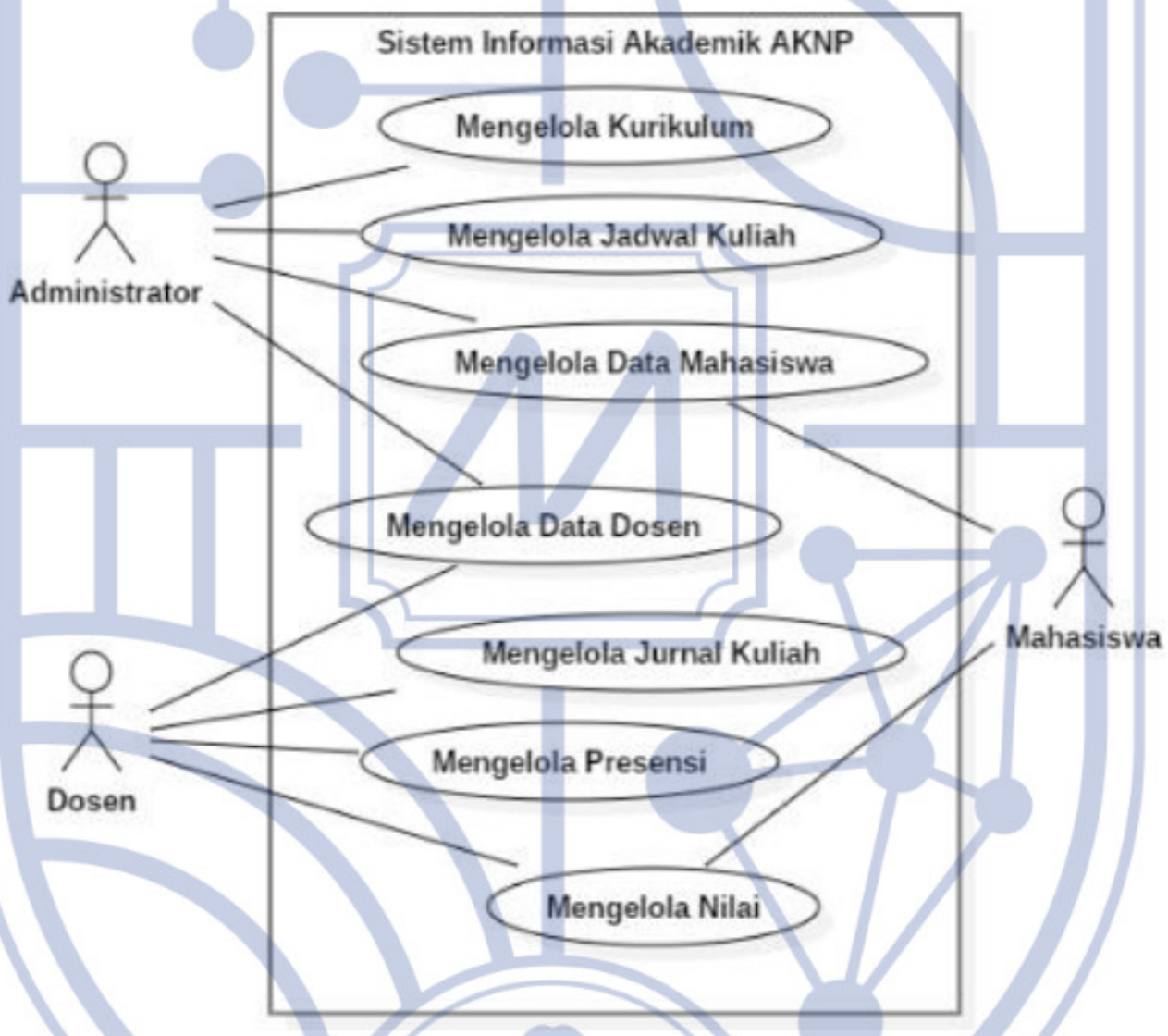
penjadwalan, mulai dari pengelolaan data hingga publikasi jadwal, telah teridentifikasi dan terakomodasi secara lengkap dalam sistem.

Beberapa simbol yang umum digunakan dalam *Use Case Diagram* dapat dilihat pada Tabel 2.1 berikut :

Tabel 2.1 Simbol *Use Case* Diagram[40]

Nama	Simbol	Keterangan
Use Case		Digunakan untuk menunjukkan fungsi atau layanan yang disediakan oleh sistem
Aktor (*Actor*)		Merepresentasikan entitas eksternal yang berinteraksi dengan sistem
Asosiasi (*Association*)		Menghubungkan aktor dengan *Use Case* yang dapat diaksesnya
Extend		Menunjukkan bahwa suatu *Use Case* dapat memperluas perilaku *Use Case* lain dalam kondisi tertentu.
Include		Menunjukkan bahwa suatu *Use Case* selalu menyertakan *Use Case* lain sebagai bagian dari prosesnya.
Generalization		Menunjukkan hubungan antara dua aktor atau dua *use case*, di mana salah satunya mewarisi fungsi dari yang lain dan dapat menambahkan atau mengubah fungsinya.

Sebagai contoh penerapan, *Use Case* Diagram digunakan pada Sistem Informasi Akademik AKNP untuk menggambarkan interaksi antara aktor dan sistem. Aktor yang terlibat meliputi *admin*istrator, dosen, dan mahasiswa. *Admin*istrator bertugas mengelola kurikulum, jadwal kuliah, data mahasiswa, dan data dosen. Dosen dapat mengelola jurnal kuliah, presensi, serta nilai mahasiswa. Sementara itu, mahasiswa dapat mengakses informasi akademik yang berkaitan dengan perkuliahan dan nilai. *Use Case* Diagram membantu menggambarkan kebutuhan fungsional sistem secara jelas sehingga memudahkan proses analisis dan perancangan sistem. Contoh *Use Case* Diagram Sistem Informasi Akademik AKNP dapat dilihat pada Gambar 2.6.



Gambar 2. 6 Contoh *Use Case Diagram* Sistem Informasi Akademik AKNP [41]

Pada Gambar 2.6 terlihat bahwa setiap aktor memiliki fungsi dan hak akses yang berbeda sesuai dengan perannya masing-masing. *Admin*istrator memiliki akses penuh dalam pengelolaan data akademik, dosen berperan dalam proses pengelolaan kegiatan perkuliahan seperti presensi dan nilai, sedangkan mahasiswa berinteraksi dengan sistem untuk memperoleh informasi akademik. Setiap aktor terhubung dengan *Use Case* melalui relasi asosiasi yang menunjukkan keterlibatan pengguna terhadap fungsi sistem. Diagram ini menunjukkan pembagian fungsi yang jelas sehingga sistem dapat berjalan lebih terstruktur dan mengurangi terjadinya tumpang tindih hak akses pengguna.

2.4.2. *Activity Diagram*

Activity Diagram digunakan untuk menggambarkan alur aktivitas atau proses kerja dalam sistem secara berurutan. Diagram ini menunjukkan langkah-langkah yang dilakukan oleh pengguna maupun sistem, mulai dari awal hingga akhir proses. *Activity Diagram* memodelkan bagaimana suatu aktivitas dijalankan, bagaimana percabangan keputusan terjadi berdasarkan kondisi tertentu, serta bagaimana proses dapat berjalan secara berurutan maupun paralel. Diagram ini tidak hanya menggambarkan aktivitas yang dilakukan oleh sistem, tetapi juga dapat menunjukkan interaksi antara pengguna dan sistem, terutama jika menggunakan *swimlane* untuk memisahkan tanggung jawab masing-masing pihak.

Beberapa simbol yang umum digunakan dalam *Activity Diagram* dapat dilihat pada Tabel 2.2 berikut.

Tabel 2.2 Simbol *Activity* Diagram [42]

Nama	Simbol	Keterangan
Status Awal (*Initial Node*)		Menunjukkan awal dari suatu proses atau aktivitas. Setiap *Activity Diagram* memiliki satu initial node.
Aktivitas (*Activity*)		Menunjukkan aktivitas atau proses yang dilakukan dalam sistem. Biasanya diawali dengan kata kerja.
Alur Kontrol (*Control Flow*)	→	Arah aliran proses dari satu aktivitas ke aktivitas berikutnya.
Percabangan (*Decvision Node*)		Menunjukkan titik pengambilan keputusan berdasarkan kondisi tertentu.
Penggabungan (*Merge Node*)		Menggabungkan kembali beberapa alur hasil percabangan menjadi satu alur.
Penggabungan (*Join*)		Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Status Akhir (*Final Node*)		Menunjukkan akhir dari suatu proses dalam *Activity Diagram*.

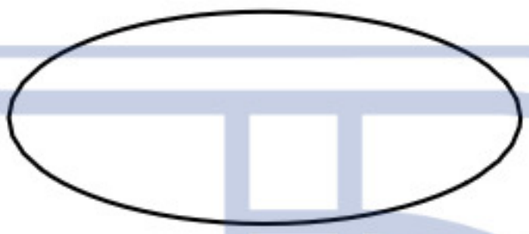
2.4.3. *Entity-Relationship Diagram* (ERD)

Entity-Relationship Diagram (ERD) merupakan salah satu model konseptual yang digunakan untuk merancang struktur basis data dalam suatu sistem informasi. ERD berfungsi untuk menggambarkan entitas yang terlibat dalam sistem, atribut yang dimiliki oleh masing-masing entitas, serta hubungan (*relationship*) antar entitas tersebut. Model ini pertama kali diperkenalkan oleh Peter Chen dan banyak digunakan dalam tahap perancangan

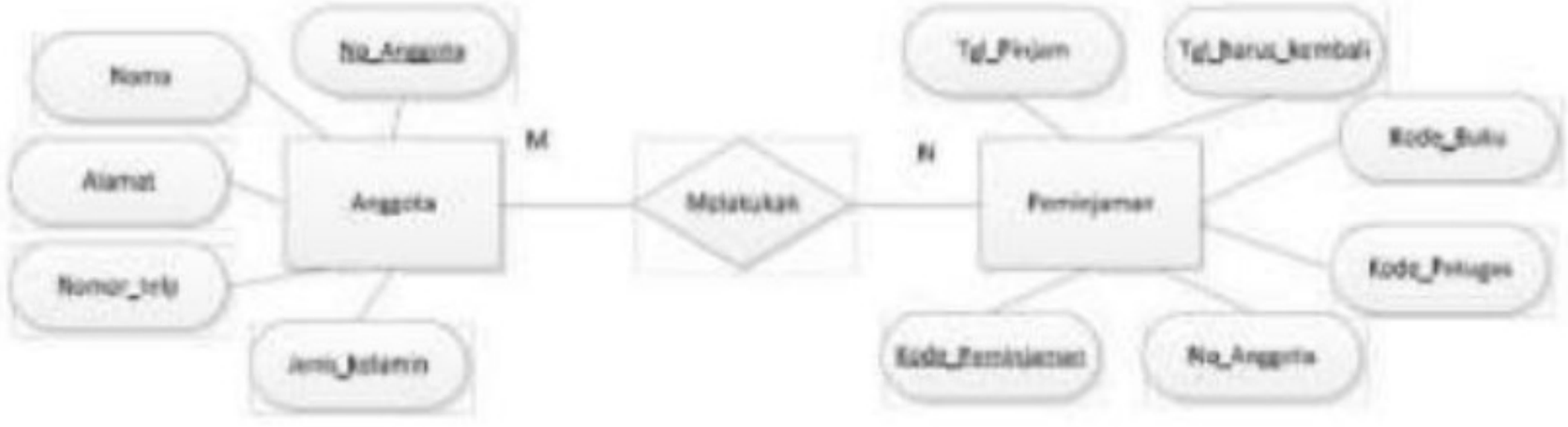
basis data karena mampu merepresentasikan struktur data secara sistematis dan mudah dipahami [43]. ERD terdiri dari beberapa komponen utama yang digunakan dalam pemodelan data, yaitu entitas, atribut, relasi, dan konektor. Masing-masing komponen tersebut memiliki simbol dan fungsi yang berbeda dalam merepresentasikan struktur data. Untuk mempermudah pemahaman, komponen ERD disajikan pada tabel berikut:

Entity-Relationship Diagram (ERD) merupakan salah satu model konseptual yang digunakan untuk merancang struktur basis data dalam suatu sistem informasi. ERD berfungsi untuk menggambarkan entitas yang terlibat dalam sistem, atribut yang dimiliki oleh masing-masing entitas, serta hubungan (*relationship*) antar entitas tersebut. Model ini pertama kali diperkenalkan oleh Peter Chen dan banyak digunakan dalam tahap perancangan basis data karena mampu merepresentasikan struktur data secara sistematis dan mudah dipahami[43]. ERD terdiri dari beberapa komponen utama yang digunakan dalam pemodelan data, yaitu entitas, atribut, relasi, dan konektor. Masing-masing komponen tersebut memiliki simbol dan fungsi yang berbeda dalam merepresentasikan struktur data seperti yang dijelaskan pada Tabel 2.3. Untuk mempermudah pemahaman, komponen ERD disajikan pada tabel berikut:

Tabel 2.3 Simbol *Entity-Relationship Diagram* (ERD)[44]

Nama	Simbol	Keterangan
Entitas (*Entity*)		Kumpulan objek yang dapat diidentifikasikan secara unik.
Atribut (*Attribute*)		Mengaitkan karakteristik atau properti dari satu entitas atau relasi yang memberikan informasi terperinci.
Relasi (Relationship)		Keterkaitan antara satu atau lebih entitas. Jenis-jenisnya satu-ke-satu, satu-ke-banyak, banyak-ke-banyak.
Konektor (*Connector*)		Menghubungkan entitas dengan atributnya atau atribut dengan relasinya

Dalam pengembangan sistem informasi, ERD digunakan sebagai dasar dalam perancangan basis data relasional. Dengan adanya ERD, struktur data dapat dirancang secara terorganisir, meminimalkan redundansi, serta menjaga konsistensi dan integritas data. Selain itu, ERD juga membantu dalam mengidentifikasi kebutuhan data sebelum tahap implementasi sistem dilakukan. Sebagai contoh penerapan, dalam sistem perpustakaan terdapat entitas Anggota dan Buku. Entitas Anggota dapat memiliki atribut seperti No_Anggota, Nama, dan Alamat, sedangkan entitas Buku memiliki atribut seperti Kode_Buku, Judul, dan Pengarang. Kedua entitas tersebut memiliki relasi "meminjam", di mana seorang anggota dapat meminjam banyak buku (relasi satu ke banyak). Dalam ERD, relasi tersebut digambarkan dengan simbol belah ketupat yang menghubungkan entitas Anggota dan Buku melalui konektor garis. Gambar 2.7 menunjukkan visualisasi notasi hubungan antara entitas dan relasi tersebut, di mana setiap komponen direpresentasikan menggunakan simbol standar ERD sehingga hubungan antar data dapat dipahami secara sistematis. Dengan demikian, ERD mampu menggambarkan hubungan antar data secara jelas dan terstruktur.

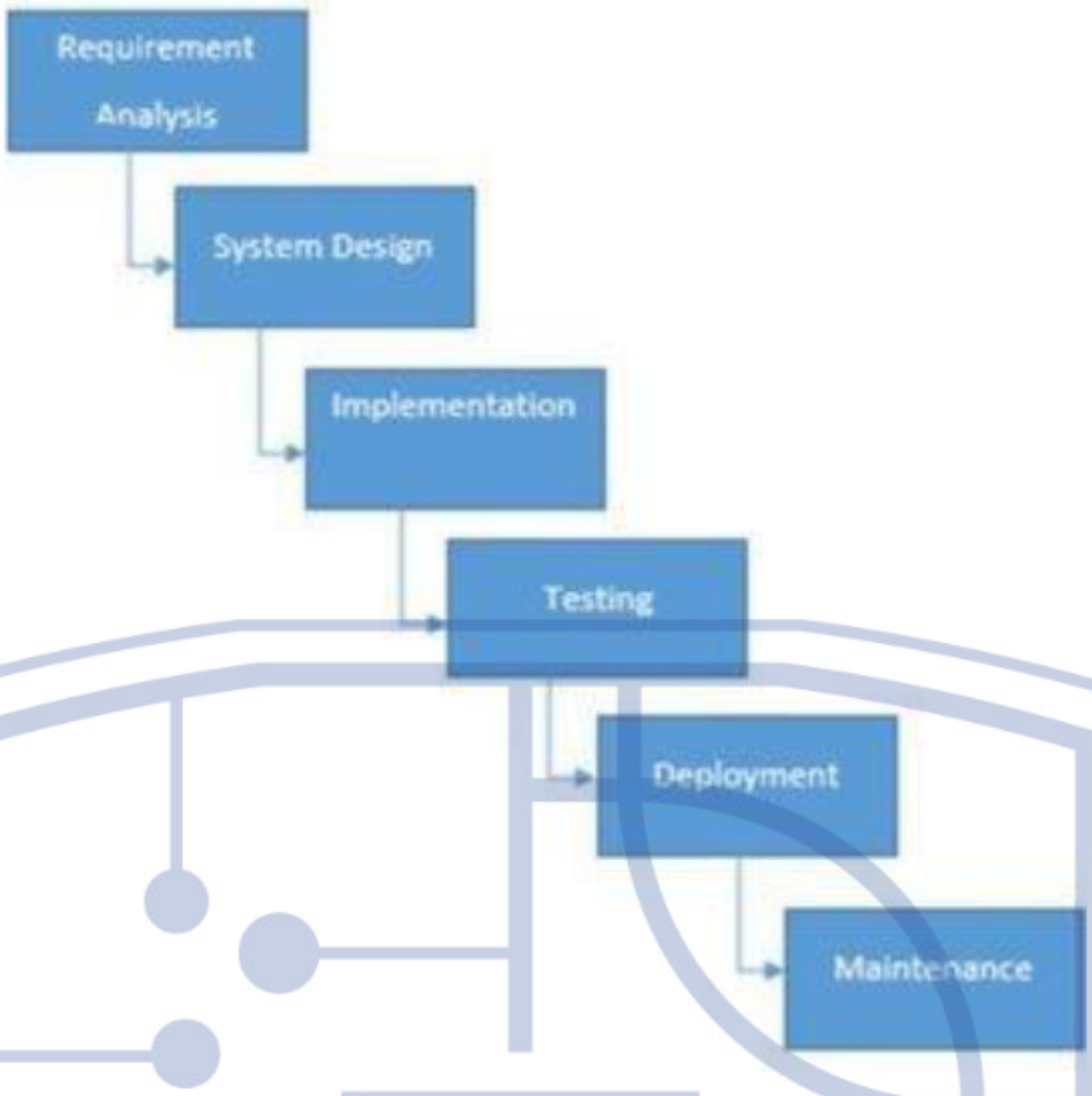


Gambar 2. 7 Notasi hubungan entitas dan relasi[43]

2.5. Waterfall

Metode *Waterfall* merupakan salah satu model pengembangan perangkat lunak yang bersifat sistematis dan berurutan. Model ini menggambarkan proses pengembangan perangkat lunak yang dilakukan melalui beberapa tahapan yang berjalan secara bertahap mulai dari analisis kebutuhan, perancangan sistem, implementasi, pengujian, hingga pemeliharaan sistem [45]. Setiap tahap dalam metode ini dilakukan secara berurutan sehingga hasil dari satu tahap akan menjadi dasar bagi tahap berikutnya. Pendekatan ini membantu pengembang dalam memahami kebutuhan sistem secara lebih terstruktur sebelum proses pengembangan dimulai. Selain itu, metode *Waterfall* menekankan pentingnya dokumentasi pada setiap tahap pengembangan sistem. Dengan dokumentasi yang baik, proses pengembangan perangkat lunak dapat lebih mudah dipantau dan dikontrol oleh tim pengembang maupun pihak terkait lainnya.

Pada model *Waterfall*, setiap tahap harus diselesaikan terlebih dahulu sebelum melanjutkan ke tahap berikutnya. Tahapan pertama biasanya dimulai dengan analisis kebutuhan sistem yang bertujuan untuk mengidentifikasi kebutuhan pengguna serta spesifikasi sistem yang akan dikembangkan. Setelah tahap analisis selesai, proses dilanjutkan dengan tahap perancangan sistem yang meliputi perancangan arsitektur sistem, basis data, serta antarmuka pengguna. Tahap berikutnya adalah implementasi atau proses pengkodean sistem menggunakan bahasa pemrograman yang telah ditentukan. Setelah implementasi selesai, sistem akan melalui tahap pengujian untuk memastikan bahwa sistem dapat berjalan sesuai dengan kebutuhan yang telah ditentukan. Tahap terakhir dalam model *Waterfall* adalah pemeliharaan sistem yang dilakukan untuk memperbaiki kesalahan serta melakukan pembaruan sistem apabila diperlukan.



Gambar 2. 8 Model Pengembangan Perangkat Lunak Metode *Waterfall*[46]

Berdasarkan Gambar 2.8, metode *Waterfall* terdiri dari beberapa tahapan utama dalam pengembangan perangkat lunak, yaitu:

3. *Requirement Analysis*. Tahap ini merupakan tahap awal dalam pengembangan sistem yang bertujuan untuk mengidentifikasi kebutuhan pengguna serta menentukan spesifikasi sistem yang akan dikembangkan. Pada tahap ini dilakukan proses pengumpulan data melalui observasi, wawancara, maupun studi literatur untuk memahami kebutuhan sistem secara menyeluruh.
4. *System Design*. Tahap perancangan sistem dilakukan setelah kebutuhan sistem telah ditentukan. Pada tahap ini pengembang merancang arsitektur sistem, struktur basis data, serta desain antarmuka pengguna yang akan digunakan dalam sistem. Perancangan ini bertujuan untuk memberikan gambaran teknis mengenai sistem sebelum tahap implementasi dilakukan.
5. *Implementation*. Tahap implementasi merupakan proses penerjemahan desain sistem ke dalam bentuk kode program menggunakan bahasa pemrograman tertentu. Pada tahap ini pengembang mulai membangun sistem sesuai dengan rancangan yang telah dibuat sebelumnya.
6. *Testing*. Setelah sistem selesai dikembangkan, dilakukan tahap pengujian untuk memastikan bahwa sistem dapat berjalan dengan baik dan sesuai dengan kebutuhan

pengguna. Pengujian dilakukan untuk menemukan kesalahan atau bug dalam sistem sehingga dapat diperbaiki sebelum sistem digunakan.

7. *Deployment*. Tahap deployment merupakan proses penerapan sistem agar dapat digunakan oleh pengguna. Pada tahap ini sistem yang telah selesai diuji akan dipasang pada *server* atau lingkungan operasional sehingga dapat diakses oleh pengguna.
8. *Maintenance*. Tahap maintenance merupakan tahap pemeliharaan sistem setelah sistem digunakan. Pada tahap ini dilakukan perbaikan kesalahan, peningkatan kinerja sistem, serta penyesuaian sistem apabila terdapat perubahan kebutuhan pengguna.

Metode *Waterfall* banyak digunakan dalam pengembangan sistem informasi karena memiliki alur kerja yang jelas dan terstruktur. Model ini cocok digunakan pada proyek yang memiliki kebutuhan sistem yang relatif stabil serta tidak mengalami perubahan yang signifikan selama proses pengembangan. Dengan adanya tahapan yang sistematis, metode *Waterfall* dapat membantu pengembang dalam mengelola proses pengembangan sistem secara lebih terorganisir [46].

Berdasarkan tahapan metode *Waterfall* yang telah dijelaskan, pada penelitian ini proses pengembangan sistem difokuskan hingga tahap pengujian (*testing*). Tahapan yang dilakukan meliputi analisis kebutuhan, perancangan sistem, implementasi, dan pengujian. Pembatasan ini dilakukan karena penelitian berfokus pada perancangan dan pembangunan sistem serta pengujian fungsionalitas untuk memastikan sistem berjalan sesuai dengan kebutuhan yang telah ditentukan. Sementara itu, tahap *deployment* dan *maintenance* tidak dibahas secara mendalam karena memerlukan implementasi pada lingkungan operasional serta pemantauan jangka panjang yang berada di luar ruang lingkup penelitian.

2.6. Penelitian Terkait

Penelitian terkait digunakan untuk mengetahui perkembangan penelitian sebelumnya yang berkaitan dengan sistem penjadwalan dan penerapan Algoritma Genetika. Beberapa penelitian terdahulu menunjukkan bahwa metode Algoritma Genetika banyak digunakan untuk menyelesaikan permasalahan penjadwalan karena mampu menangani kombinasi variabel yang kompleks serta berbagai batasan yang harus dipenuhi secara bersamaan. Penelitian yang dilakukan oleh Mutasar dan Hasdyna mengembangkan sistem informasi penjadwalan mata kuliah berbasis Web dengan menggunakan Algoritma Genetika. Penelitian ini bertujuan untuk mengatasi permasalahan bentrok jadwal, ketersediaan ruang, serta distribusi beban mengajar dosen. Hasil penelitian menunjukkan bahwa Algoritma

Genetika mampu menghasilkan kombinasi jadwal yang optimal melalui proses seleksi, *crossover*, dan mutasi sehingga konflik jadwal dapat diminimalkan secara signifikan[8].

Penelitian lain dilakukan oleh Rahellea Joiby Onibala, Sondy, Efraim. yang mengembangkan aplikasi penjadwalan petugas pemadam kebakaran berbasis Web dengan menggunakan Algoritma Genetika. Permasalahan utama yang dihadapi adalah proses penjadwalan manual yang menyebabkan distribusi tugas tidak merata dan sulit diperbarui ketika terjadi perubahan personel. Hasil penelitian menunjukkan bahwa sistem penjadwalan berbasis Web dengan Algoritma Genetika mampu meningkatkan pemerataan distribusi tugas serta mempermudah proses pengelolaan jadwal secara otomatis [9].

Penelitian yang dilakukan oleh Maulana dkk. mengembangkan sistem penjadwalan mata pelajaran menggunakan Algoritma Genetika untuk mengatasi kompleksitas penyusunan jadwal yang melibatkan banyak batasan seperti ketersediaan guru, ruang kelas, dan waktu pembelajaran. Dalam penelitian tersebut, jadwal direpresentasikan dalam bentuk kromosom yang berisi kombinasi data mata pelajaran, waktu, dan guru. Hasil penelitian menunjukkan bahwa Algoritma Genetika efektif dalam menghasilkan jadwal dengan konflik minimal serta mampu menangani permasalahan penjadwalan yang bersifat kompleks[10].

Penelitian lain yang dilakukan oleh Fajarlestari dan Hardiyanti mengkaji penerapan Algoritma Genetika dalam sistem penjadwalan karyawan. Penelitian ini bertujuan untuk mengoptimalkan pembagian jadwal kerja karyawan dengan mempertimbangkan berbagai faktor seperti jumlah shift, kebutuhan tenaga kerja, serta permintaan jadwal dari karyawan. Hasil penelitian menunjukkan bahwa Algoritma Genetika mampu menghasilkan jadwal kerja yang lebih optimal dan efisien dibandingkan metode penjadwalan manual[11].

Berdasarkan beberapa penelitian tersebut dapat disimpulkan bahwa Algoritma Genetika merupakan metode yang efektif untuk menyelesaikan permasalahan penjadwalan yang kompleks karena mampu menghasilkan solusi optimal dari berbagai kemungkinan kombinasi. Namun, sebagian besar penelitian sebelumnya masih berfokus pada penjadwalan akademik atau penjadwalan tenaga kerja. Oleh karena itu, penelitian ini mengembangkan sistem penjadwalan otomatis multi-divisi pelayanan gereja berbasis Web dengan menggunakan Algoritma Genetika guna membantu proses penyusunan jadwal pelayanan yang lebih optimal, adil, dan terstruktur.

2.7. Constraint Validation Test (CVT)

Constraint Validation Test (CVT) merupakan metode pengujian perangkat lunak yang berfokus pada verifikasi tingkat kepatuhan sistem terhadap batasan-batasan (*constraints*) yang telah ditetapkan. Dalam sistem berbasis data, CVT memastikan integritas input sesuai aturan bisnis, sedangkan dalam optimasi, CVT menjadi filter untuk menentukan kelayakan (*feasibility*) sebuah solusi. Dalam konteks sistem basis data, *constraint* berfungsi sebagai aturan yang membatasi data yang dapat dimasukkan ke dalam sistem agar tetap sesuai dengan struktur dan kebutuhan yang telah ditentukan. *Constraint* dapat diterapkan baik pada tingkat aplikasi maupun basis data, dan berperan penting dalam menjaga integritas serta konsistensi data. Dengan adanya *constraint*, sistem akan menolak data yang tidak sesuai dengan aturan yang telah ditetapkan, sehingga kesalahan input dapat diminimalkan [47].

Selain itu, proses pengujian terhadap *constraint* dilakukan untuk memastikan bahwa setiap aturan yang telah dirancang benar-benar berjalan dengan baik pada sistem. Pengujian ini dilakukan dengan memasukkan data yang sesuai maupun tidak sesuai dengan aturan, sehingga dapat diketahui apakah sistem mampu memvalidasi data secara tepat. Hasil pengujian menunjukkan bahwa penerapan *constraint* dapat berfungsi sebagai mekanisme validasi yang efektif dalam menyaring data serta menjamin kualitas dan kebenaran data yang dihasilkan oleh sistem [48]. Dengan demikian, CVT tidak hanya berfungsi sebagai metode pengujian, tetapi juga sebagai mekanisme kontrol kualitas dalam sistem, khususnya dalam memastikan bahwa setiap proses pengolahan data maupun hasil optimasi telah memenuhi seluruh batasan yang ditentukan.

2.8. Black Box Testing

Black Box Testing dikenal sebagai *behavior testing*, *closed-box testing* atau *specification-based testing*, metode ini merupakan teknik pengujian fungsionalitas perangkat lunak di mana penguji mengevaluasi kebenaran sistem tanpa perlu mengetahui atau memperhatikan logika internal dari kode program [49]. Pengujian *Black Box* ada beragam metode seperti *Equivalence Partitioning*, *Boundary Value Analysis*, *Fuzzing*, *Cause-Effect Graph*, *Orthogonal Array*, *All Pair*, dan *State Transition* yang digunakan untuk memvalidasi kualitas perangkat lunak melalui pengolahan input, hubungan sebab-akibat, dan skenario uji yang terstruktur [50]. Salah satu teknik yang diterapkan adalah Equivalence Partitioning, yaitu metode pembagian data masukan ke dalam beberapa kelompok

berdasarkan fungsinya. Prinsip utamanya adalah mengklasifikasikan data untuk membedakan antara masukan yang benar (*valid*) dan salah (tidak valid). Dengan menguji perwakilan dari setiap kelompok tersebut, teknik ini secara efektif dapat menemukan kesalahan sekaligus menghemat waktu karena jumlah skenario pengujian yang dikembangkan lebih sedikit, namun tetap menjamin kualitas pemeriksaan pada seluruh fungsi sistem [50]. Penerapan teknik ini bertujuan untuk mendeteksi berbagai kesalahan sistem yang mencakup ketidaksesuaian fungsionalitas, kendala antarmuka, gangguan pada struktur data atau akses database, penurunan performa, hingga kegagalan pada proses inisialisasi dan penghentian aplikasi.

Pada penelitian ini, metode *Black Box Testing* digunakan untuk menguji fungsionalitas sistem penjadwalan pelayanan gereja yang dikembangkan. Pengujian difokuskan pada setiap fitur utama seperti pengelolaan data *Volunteer,* proses penyusunan jadwal, serta validasi data input. Teknik *Equivalence Partitioning* digunakan untuk mengelompokkan data uji ke dalam kategori valid dan tidak valid sehingga pengujian dapat dilakukan secara efektif dan efisien tanpa harus menguji seluruh kemungkinan data yang ada.

2.9. Kuesioner

Kuesioner merupakan salah satu teknik pengumpulan data yang digunakan untuk memperoleh informasi dari responden melalui serangkaian pertanyaan tertulis. Metode ini banyak digunakan dalam penelitian karena mampu mengumpulkan data secara sistematis serta menjangkau responden dalam jumlah yang relatif besar. Selain itu, penggunaan kuesioner juga memudahkan proses pengolahan dan analisis data yang diperoleh dari responden [51]. Dalam penelitian ini, kuesioner digunakan sebagai instrumen untuk mengevaluasi sistem yang telah dikembangkan, khususnya dalam menilai tingkat kemudahan penggunaan (*usability*), tampilan antarmuka, serta fungsionalitas sistem penjadwalan pelayanan gereja. Responden yang terlibat merupakan pengguna sistem, seperti pengurus dan pelayan gereja, yang telah menggunakan aplikasi yang dibangun. Penggunaan kuesioner dalam evaluasi sistem bertujuan untuk melengkapi hasil pengujian dengan memperoleh penilaian langsung dari pengguna terhadap sistem yang digunakan [52].

Penyusunan kuesioner dilakukan dengan menggunakan skala *Likert*, yaitu skala pengukuran yang digunakan untuk mengetahui tingkat persetujuan responden terhadap suatu pernyataan. Skala *Likert* terdiri dari beberapa kategori jawaban seperti sangat setuju, setuju,

netral, tidak setuju, dan sangat tidak setuju yang masing-masing memiliki bobot nilai tertentu sehingga memudahkan dalam proses analisis data [53]. Selain itu, penggunaan skala *Likert* juga memungkinkan peneliti untuk mengukur sikap atau persepsi responden secara kuantitatif sehingga hasil yang diperoleh dapat diolah secara statistik.

Dalam penyusunan butir pertanyaan kuesioner, setiap pertanyaan dirancang agar jelas, mudah dipahami, serta sesuai dengan tujuan penelitian. Pertanyaan yang disusun mencakup aspek kemudahan penggunaan sistem, kejelasan tampilan antarmuka, serta kinerja sistem dalam membantu proses penjadwalan. Dengan demikian, data yang diperoleh diharapkan mampu merepresentasikan pengalaman pengguna secara menyeluruh.

Hasil dari kuesioner kemudian dianalisis untuk mengetahui tingkat keberhasilan sistem yang dikembangkan berdasarkan respon pengguna. Data yang diperoleh akan diolah dalam bentuk persentase atau nilai rata-rata yang dihitung dengan Rumus 2 sehingga dapat memberikan gambaran mengenai kualitas sistem dari sudut pandang pengguna. Dengan adanya evaluasi ini, pengembang dapat mengetahui kelebihan dan kekurangan sistem serta melakukan perbaikan yang diperlukan di masa mendatang.

$$\boldsymbol{Persentase = \frac{Total\ Skor}{Skor\ Maksimal}\ x\ 100\%} \quad \text{.....(2)}$$

Dimana:

$Persentase$ = nilai persentase hasil kuesioner
$Total\ Skor$ = jumlah seluruh skor jawaban responden
$Skor\ Maksimal$ = skor tertinggi yang mungkin diperoleh
100% = konstanta untuk mengubah hasil menjadi bentuk persentase

2.10. Gereja Bethel Indonesia (GBI) Pelita IV Medan

Gereja Bethel Indonesia (GBI) Pelita IV Medan merupakan gereja lokal yang lahir dari kerinduan pelayanan keluarga Pdm. dr. Suheri P. Gultom, M.A., M.Kes untuk menjadi berkat bagi kota dan bangsa serta melahirkan generasi yang berdampak. Pelayanan ini pada awalnya banyak dimulai dari kalangan pemuda yang memiliki kerinduan untuk melayani Tuhan dan menjadi saluran berkat bagi banyak orang. Dari kerinduan tersebut, pada Januari 2001 dimulai perintisan Pos Pekabaran Injil (Pos PI) yang berlokasi di Klinik Denai, Jalan Denai. Perintisan ini kemudian berkembang dan pada 19 Juni 2001 didewasakan menjadi gereja lokal di Aula RSU Imelda Medan dengan nama GKB Kasih Karunia. Pada masa awal pelayanan, kegiatan ibadah juga diadakan di Klinik Perjuangan di Jalan Tempuling dengan jemaat yang terdiri dari keluarga gembala serta staf perawat dan dokter di klinik tersebut.

Selanjutnya jemaat lokal mulai beribadah di rumah keluarga gembala di Jalan Pelita IV Gang Pertama No. 3 dengan jumlah jemaat awal sekitar 45 orang.

Seiring dengan perkembangan pelayanan, gereja juga melakukan berbagai perintisan pelayanan di beberapa daerah. Salah satu perintisan pelayanan dilakukan di Pulau Nias pada tahun 2004, tepatnya sekitar dua minggu sebelum terjadinya gempa bumi dan tsunami yang melanda wilayah tersebut. Pelayanan ini dimulai dengan keberangkatan Bapak dan Ibu Gembala bersama beberapa rekan pelayanan internasional seperti Ps. Kim dari Korea Selatan, Ps. Bryan Lee, dan Ps. Nick Burt dari Amerika Serikat untuk mengadakan Kebaktian Kebangunan Rohani (KKR). Setelah terjadinya bencana gempa dan tsunami, tim pelayanan kembali ke Nias untuk memberikan bantuan kemanusiaan serta melakukan pelayanan misi. Pelayanan tersebut berkembang dengan berdirinya beberapa gereja lokal seperti GBI Baithani Teluk Dalam, Hili Sondreka, Bawomaenamolo, Hilizalo'otano, dan GBI Bawolahusa. Hingga saat ini gereja-gereja tersebut telah memiliki gedung ibadah permanen dan terus melayani jemaat di wilayah tersebut.

Selain pelayanan misi di Nias, GBI Pelita IV Medan juga terus mengembangkan pelayanan di wilayah Kota Medan dan sekitarnya. Pada tahun 2006, gereja merintis pelayanan ibadah di lingkungan Akbid Sehati yang kemudian berkembang menjadi salah satu pusat pelayanan jemaat. Dalam perkembangannya, pelayanan gereja juga melahirkan beberapa cabang gereja seperti GBI Rs. Imelda, GBI Stikes Sehati, dan GBI Pelita Cabang Simpang Pemda. Perintisan cabang gereja ini dilakukan untuk memperluas jangkauan pelayanan dan menjangkau lebih banyak jemaat. Hingga tahun 2026, pelayanan GBI Pelita IV Medan terus mengalami perkembangan baik dari segi kegiatan pelayanan maupun jumlah jemaat. Saat ini jumlah jemaat yang hadir dalam ibadah mingguan mencapai sekitar kurang lebih 3000 orang setiap minggunya sehingga pengelolaan pelayanan dan penjadwalan *Volunteer* menjadi semakin penting[54]. Bangunan dapat dilihat dari Gambar 2.9.

Gambar 2. 9 Gedung Gereja Bethel Indonesia (GBI) Pelita IV Medan