BAB II
TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1. Tinjauan Pustaka

Penelitian yang dilakukan tidak terlepas dari dukungan hasil-hasil penelitian yang telah ada sebelumnya yang berikaitan dengan penelitian ini.

Pada tahun 2013, Irfrans Kusmarna melakukan penelitian tentang rancang bangun aplikasi penjadwalan mata kuliah menggunakan algoritma *Particle Swarm Optimization* dengan studi kasus Jurusan Teknik Informatika UIN Suska Riau. Hasil dari penelitian ini adalah algoritma PSO berhasil dalam merancang jadwal perkuliahan di Jurusan Teknik Informatika UIN Suska Riau walaupun masih terdapat beberapa *constraint* yang tidak terpenuhi. Hasil dari penjadwalan kuliah juga belum berhasil memenuhi dari segi kualitas yaitu jam perkuliahan yang difokuskan pada jam-jam yang efektif. Kekurangan dari aplikasi penjadwalan mata kuliah menggunakan PSO adalah *constraint* yang tidak bisa berubah sesuai kondisi.

Pada tahun 2010. Yanti Nitra Sianipar melakukan penelitian tentang optimasi penjadwalan mata kuliah di Perguruan Tinggi dengan menggunakan algoritma Memetika. Hasil penelitian yang didapat bahwa algoritma Memetika mampu dalam membuat penjadwalan mata kuliah yang layak. Hasil yang didapat dengan memenuhi semua kendala *hard* adalah dengan menggunakan ukuran populasi 16, probabilitas mutasi sebesar 0,5 dan 1500 iterasi dengan waktu *running* kurang dari 3 menit. Jadwal yang dhasilkan oleh algoritma ini mendekati optimal dengan memenuhi tujuan dari penelitian. Walaupun semua kendala *hard* telah terpenuhi dan menghasilkan jadwal yang layak digunakan, namun tidak semua kendala *soft* dapat terpenuhi

Pada tahun 2017, Ricky Nugraha Tendi melakukan penelitian terkait tentang penyelesaian *travelling salesman problem with time windows* untuk meminimasi *cost* dengan Algoritma *Grey Wolf Optimization.* Pada penelitian ini GWO diimplementasikan ke dalam 9 kasus TSPTW yang terdiri dari 5 sub-kasus pada setiap kasusnya. Parameter yang diuji pada penelitian ini adalah jumlah serigala (n), jumlah iterasi (T), dan nilai awal parameter a (awal). Hasil pengujian dari pengaruh parameter membuktikan bahwa adanya pengaruh dari masing-masing

parameter pada kasus TSPTW. Hasil penelitian ini menunjukkan bahwa GWO menunjukkan performansi yang sangat baik dalam permasalahan kasus 20 kota, tetapi algortima GWO belum dapat menghasilkan solusi yang lebih baik pada sebagian kasus 40 kota dan seluruh kasus 60 kota.

Pada tahun 2019, Riqqah Izzah Adibah juga melakukan penelitian tentang penyelesaian *Traveling Salesman Problem* pada perusahaan Batu Bhumi Suryatama dengan menggunakan Algoritma *Grey Wolf Optimization*. Penelitian ini bertujuan untuk meminimasi jarak pada TSP. Metode GWO akan dibandingkan dengan metode yang digunakan oleh perusahaan. Hasil dari penelitian ini membuktikan bahwa metode algoritma GWO menghasilkan jarak paling minimun dibandingkan dengan metode perusahaan dengan selisih total jarang 46,76 Km dan efisiensi sebesar 55,03%.

Pada tahun 2017, Milica Petrovic dan Zoran Miljkovic melakukan penelitian yang membahas tentang sistem penjadwalan robot seluler tunggal dengan menggunakan Algoritma *Grey Wolf Optimization*. Pada penelitian ini, algoritma GWO diusulkan untuk menemukan solusi optimal dari masalah penjadwalan robot seluler tunggal. Tujuan dari penelitian ini adalah untuk meminimalkan waktu total transportasi robot seluler saat melakukan pengangkutan internal bahan baku, barang, dan alat-alat dalam sistem manufaktur. Penjadwalan didapatkan dari Matlab dan diuji dengan sistem robot seluler Khepera II dalam model laboratorium. Hasil dari penelitian ini menunjukkan bahwa penerapan dan keefektifan pendekatan cerdas yang dilakukan mendekati kondisi di dunia nyata.

2.2. Penjadwalan

Penjadwalan adalah masalah yang berkaitan dengan pengoptimalan sumber daya yang terbatas terhadap beberapa aktivitas pada waktu yang tepat. Masalah ini adalah sebuah proses pengambilan keputusan yang memiliki tujuan untuk mengoptimalkan satu atau beberapa tujuan [5]. Masalah penjadwalan biasanya diselesaikan dengan menggunakan sumber daya manusia di suatu lembaga pendidikan dan selama proses penyelesaian penjadwalan ini, banyak aspek yang harus pertimbangkan [13]. Hasil yang didapatkan pun sering kali tidak memuaskan

yaitu seperti tidak memenuhi prasyarat dan ketika tiba-tiba prasyarat berubah yang membuat seluruh pekerjaan tidak dapat digunakan dan harus dimulai dari awal.

Masalah penjadwalan pada umumnya adalah alokasi sejumlah kegiatan pada suatu periode atau waktu tertentu sehingga semua kendala dapat terpenuhi [14]. Masalah penjadwalan adalah masalah yang cukup rumit bahkan akan menjadi lebih rumit jika harus mempertimbangkan banyak syarat yang harus terpenuhi seperti dosen, mahasiswa, mata kuliah, dan ruangan yang harus ditentukan secara optimal. Hal-hal yang harus dipertimbangkan sebagai faktor penyusunan jadwal yaitu adanya batasan waktu pengerjaan (*completion time)*, aliran waktu (*flow time*) yang merupakan urutan pengerjaan aktivitas termasuk waktu tunggu (*waiting time*), dan waktu (*processing time*)[5]. Penjadwalan yang baik adalah penjadwalan yang memenuhi semua prasyarat yang telah ditentukan.

2.2.1. Penjadwalan Kuliah Internasional

Masalah penjadwalan mata kuliah ini juga sering disebut dengan *University Course Timetabling Problem* (UCTP). Masalah penjadwalan kuliah ini berawal dari sebuah kompetisi membuat jadwal perkuliahan di 20 instansi di Universitas Napier Edinburg, Skotlandia [15]. Model permasalahan sudah ditetapkan, namun peserta dapat menggunakan berbagai metode dalam penyelesaiannya. Tujuan dari UCTP adalah untuk menemukan suatu metode yang mengalokasikan sumber daya pada waktu dan ruang yang telah ditentukan sebelumnya, dimana semua kendala harus terpenuhi [16]. Sumber daya ini meliputi mahasiswa, dosen, dan mata kuliah, dan ruangan.

Terdapat beberapa definisi masalah yang terdapat pada UCTP, yaitu *events*, *timeslots*, *resource*, *constraints*, *people*, dan *conflicts* [16]. *Events* adalah sebuah aktivitas yang telah dijadwalkan, yang didalamnya terdapat dosen, mahasiswa, dan mata kuliah. *Timeslots* adalah selang waktu dimana setiap *events* dijadwalkan, seperti slot waktu mingguan hari selasa dan slot waktu harian jam 9 sampai 10 pagi. *Resource* adalah sumber daya yang digunakan oleh *events*, seperti ruangan, waktu, dan sebagainya. *Constraint* adalah sebuah kendala dalam penjadwalan *events*. Pelanggaran ini dibagi menjadi 2 kategori kendala yaitu kendala lunak (*soft constraints*) dan kendala keras (*hard constraints*), contohnya seperti kapasitas

ruanga, waktu pelaksanaan, dan lain-lain. *People* adalah orang yang termasuk di dalam *events*, seperti dosen dan mahasiswa. *Conflict* adalah permasalahan dari dua *events* yang saling bertentangan seperti, dosen mengajar 2 mata kuliah pada waktu yang sama.

Pelanggaran yang terdapat pada UCTP dibagi menjadi 2 macam, yaitu kendala keras (*hard constraints)* dan kendala lunak (*soft constraints)*. Kendala keras (*hard contraints)* mempunyai prioritas lebih tinggi daripada kendala lunak (*soft contraints)* karena semua kendala yang ada pada kendala keras harus terpenuhi, sehingga terciptanya jadwal yang layak digunakan [17]. Kendala lunak merupakan fungsi objektif, yang berarti memaksimalkan semua kendala lunak terpenuhi. Tidak seperti kendala keras, kendala lunak tidak harus semuanya terpenuhi. Namun, ketika kendala lunak semakin banyak terpenuhi maka solusi yang didapatkan akan lebih maksimal.

Kendala keras yang timbul akibat interaksi antar entitas adalah sebagai berikut [15]:

1. Dua atau lebih kelas tidak dapat diberikan pada suatu waktu pada ruangan yang sama
2. Setiap kuliah harus diselenggarakan pada ruangan yang memenuhi fasilitas dan kapasitas untuk kuliah tersebut
3. Tidak ada mahasiswa yang mendapat dua atau lebih kelas pada waktu yang bersamaan

Kendala soft yang bisa meningkatkan hasil dari penjadwalan, yaitu [15] :

1. Penjadwalan mata kuliah menghindari selang waktu pada sore hari.
2. Mahasiswa tidak mendapat kuliah lebih dari 6 jam berturut.
3. Mahasiswa mengikuti lebih dari 1 mata kuliah dalam 1 hari.

2.3. Algoritma *Grey Wolf Optimization*

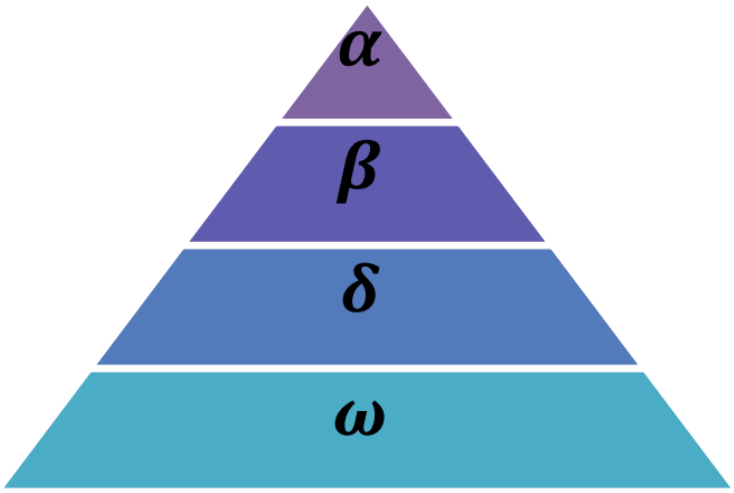
Grey Wolf Optimization adalah algoritma yang terinspirasi dari serigala abu-abu yang termasuk ke dalam keluarga Canidae. Algoritma *Grey Wolf Optimization* termasuk algoritma metaheuristik yang baru. Algoritma metaheuristik adalah algoritma yang terinspirasi dari cara kerja alam yang dirancang untuk memberikan solusi yang mendekati optimal untuk suatu masalah optimisasi [17]. Algoritma

GWO ini termasuk ke dalam kategori *Swarm Intelligence* (SI) karena terinspirasi dari hirarki sosial dan proses berburu Serigala abu-abu. Karakteristik dari *Swarm Intelligence* adalah mudah untuk dilakukan, yang berguna untuk menutupi kekurangan algoritma dan meningkatkan kinerja dari algoritma tersebut [18]. Serigala abu-abu (Canis Lupus) termasuk kepada keluarga Canidae. Serigala abu-abu berada di atas rantai makanan, yang membuat mereka disebut sebagai predator puncak. Serigala abu-abu ini biasanya hidup dalam sebuah grup. Pada satu kelompok serigala abu-abu rata-rata terdiri dari 5-12 ekor serigala abu-abu. Algortima ini terinspirasi dari tingkah laku dan mekanisme pemburuan pada serigala abu-abu. Dalam satu grup serigala abu-abu ini biasanya terdiri dari beberapa dominan hirarki yang berbeda [11], yaitu :

1. Alpha (α) yang berperan untuk memimpin suatu grup serigala abu-abu. Pemimpin dalam suatu grup serigala abu-abu yaitu 1 ekor serigala laki-laki dan 1 ekor serigala perempuan. Alpha mempunyai tanggung jawab untuk membuat keputusan tentang berburu, tempat istirahat, waktu bangun dan lain-lain. Keputusan yang diambil oleh alpha, harus diikut oleh semua anggota dalam grup. Namun, alpha bukanlah anggota terkuat dari grup tetapi yang terbaik dalam memanajemen grup. Ini membuktikan bahwa memanajemen dan mendisiplinkan suatu grup lebih penting daripada kekuatan.
2. Beta (β) yang berperan sebagai bawahan alpha. Beta membantu alpha dalam pengambilan keputusan dan kegiatan lainnya. Beta menempati posisi kedua paling atas pada suatu grup serigala abu-abu. Beta bisa saja serigala laki-laki atau perempuan. Beta bisa menjadi kandidat untuk menggantikan alpha, jika pada suatu saat alpha mati atau sudah sangat tua. Beta mempunyai tugas seperti penasihat untuk alpha dan meneruskan perintah dari alpha ke seluruh anggota serigala abu-abu.
3. Serigala abu-abu yang tidak termasuk pada alpha, beta, dan omega adalah delta (δ). Delta harus tunduk kepada alpha dan beta, namun tetap mendominasi omega. Serigala abu-abu delta mempunyai beberapa peran yaitu, pengintai, *sentinel*, sesepuh, pemburu dan pengasuh. Pengintai bertugas untuk mengawasi batas wilayah dan memperingatkan jika terdapat

bahaya. *Sentinel* bertugas untuk menjamin dan melindungi keamanan grup. Sesepuh adalah serigala abu-abu tua yang telah berpengalaman sebelumnya menjadi alpha atau beta. Pemburu bertugas untuk membantu alpha dan beta saat memburu suatu mangsa dan menyediakan makanan untuk grup. Pengasuh bertugas untuk merawat serigala abu-abu yang sedang lemah, sakit, dan terluka pada grup.

4. Omega (ω) yang berperan sebagai kambing hitam pada suatu grup serigala abu-abu. Omega menempati posisi paling rendah pada suatu grup serigala abu-abu. Omega harus selalu mengikuti semua perintah dari semua serigala abu-abu yang dominan. Omega adalah serigala abu-abu terakhir yang diizinkan untuk makan. Omega mungkin tidak terlihat tidak terlalu penting, namun setelah diamati ternyata seluruh anggota grup akan menghadapi pertengkaran dan masalah internal jika kehilangan omega. Omega membantu dalam memuaskan seluruh anggota grup dan menahan struktur dominasi.



Gambar 2.1 Urutan Dominan Hierarki Serigala Abu-Abu

(Sumber : S. Mirjalili, S. M. Mirjalili, and A. Lewis, “Grey Wolf Optimizer”)

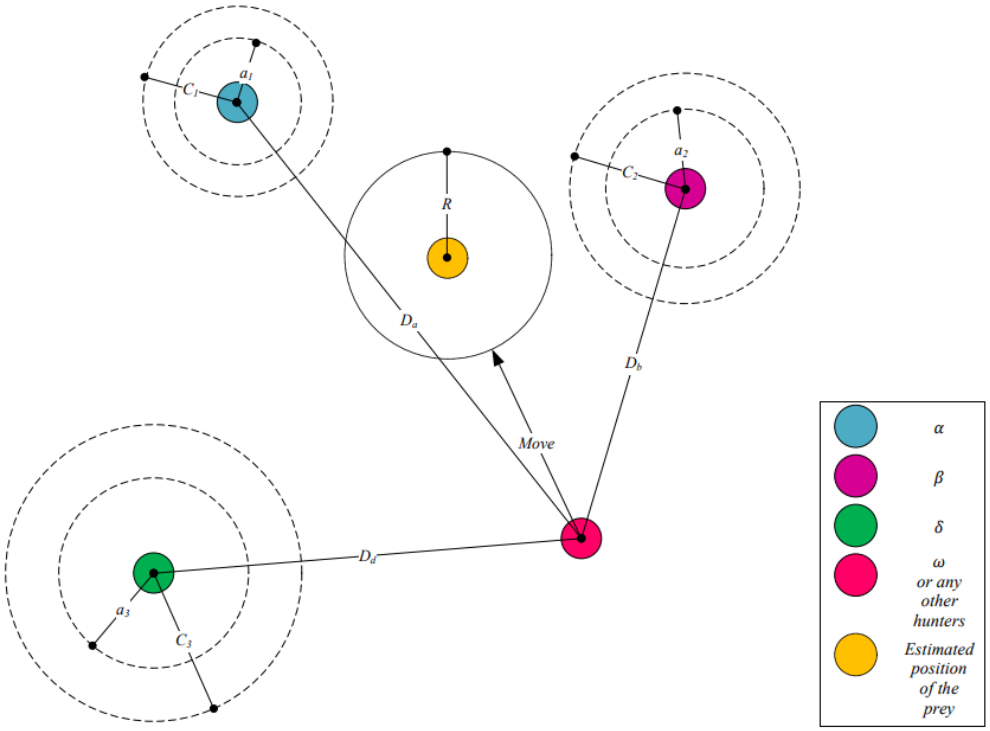
Algoritma GWO telah diuji dengan fungsi standar yang menunjukkan bahwa ia memiliki karakteristik eksplorasi dan eksploitasi yang lebih baik dibandingkan dengan teknik kecerdasaran berkoloni (*swarm intelligent techniques)* yang lain. Selain itu, sebagian teknik kecerdasan berkoloni biasa yang biasa digunakan untuk menyelesaikan permasalahan optimasi tidak mempunyai pemimpin untuk

mengendalikan keseluruhan periode. Kekurangan inilah yang diperbaiki dalam algortima GWO, dimana serigala abu-abu secara natural memiliki mekanisme kepemimpinan. Algoritma ini hanya memiliki beberapa parameter dan cukup mudah diimplementaskan sehingga membuatnya lebih unggul daripada sebelumnya. Karena kemudahannya inilah, algoritma GWO telah diimplementasikan untuk menyelesaikan beberapa permasalahan optimasi.

2.4. Skema Algoritma *Grey Wolf Optimization*

Algoritma *Grey Wolf Optimization* ini tidak hanya meniru hirarki sosial yang ada disuatu kelompok serigala abu-abu saja. Proses algoritma GWO ini juga meniru proses dari perburuan yang dilakukan oleh suatu kelompok serigala abu-abu. Langkah-langkah perburuan yang dilakukan oleh suatu serigala abu-abu yaitu melacak dan mendekati mangsa, mengejar dan mengitari mangsa hingga berhenti bergerak, dan menyerang mangsa. Dimana pada algoritma GWO proses tersebut dimodel matematiskan sebagai berikut :

2.4.1. Serigala Abu-Abu Mengelilingi Mangsa



Gambar 2.2 Serigala Abu-Abu Mengelilingi Mangsa

(Sumber : S. Mirjalili, S. M. Mirjalili, and A. Lewis, “Grey Wolf Optimizer”)

Pada Gambar 2.2 dapat dilihat proses setiap parameter memodelkan perilaku serigala abu-abu dalam mengelilingi mangsa. hal ini dilakukan untuk mempersempit pergerakan dari mangsa, sehingga mangsa akan kesulitan untuk melarikan diri dari perburuan. Algoritma GWO menirukan proses dari memburu mangsa pada suatu kelompok serigala abu-abu. Pada proses serigala abu-abu mengelilingi mangsanya, algoritma GWO memodelkan perilaku serigala abu-abu secara matematis sebagai berikut :

$$\vec{D} = \left|\vec{C} \,.\, \overrightarrow{X_P}(t) - \vec{X}\,(t)\right| \quad \text{.....(2.1)}$$

$$\vec{X}(t+1) = \overrightarrow{X_p}(t) - \vec{A} \,.\, \vec{D} \quad \text{.....(2.2)}$$

Keterangan :

$\vec{D}$ = posisi jarak ω terhadap α, β, dan δ

$\overrightarrow{X_p}$ = posisi mangsa

$\vec{X}$ = posisi serigala

t = jumlah iterasi

$\vec{A}$ = koefisien vektor

$\vec{C}$ = koefisien vektor

Perhitungan nilai koefisien vektor $\vec{A}$ dan $\vec{C}$ didapatkan dengan menggunakan rumus berikut :

$$\vec{A} = 2\vec{a} \,.\, \overrightarrow{r_1} - \vec{a} \quad \text{.....(2.3)}$$

$$\vec{C} = 2 \,.\, \vec{r}_2 \quad \text{.....(2.4)}$$

Keterangan :

$\vec{a}$ = nilai yang diturunkan secara linier dari 2 ke 0 sebanyak iterasi

$\overrightarrow{r_1}$ = nilai vektor acak dengan rentang [0,1]

$\overrightarrow{r_2}$ = nilai vektor acak dengan rentang [0,1]

2.4.2. Serigala Abu-Abu Memburu Mangsa

Gambar 2.3 Serigala Abu-Abu Berburu Mangsa

(Sumber : S. Mirjalili, S. M. Mirjalili, and A. Lewis, “Grey Wolf Optimizer”)

Terlihat pada Gambar 2.3, serigala abu-abu sedang memburu mangsa dengan cara mendekati mangsa secara mengelilinginya. Serigala abu-abu memiliki kemampuan untuk mengetahui lokasi mangsa. Dalam melakukan perburuan mangsa, serigala abu-abu dipimpin oleh alpha dan dibantu oleh beta dan delta. Pada proses awal perburuan, serigala abu-abu belum mengetahui posisi dari mangsa. Oleh karena itu, serigala alpha, beta, dan delta dianggap sebagai serigala abu-abu yang paling mengetahui posisi dari mangsa yang akan diburu. Serigala abu-abu yang lain akan mengikuti pergerakan serigala alpha, beta, dan delta dalam mendekati mangsa. Oleh sebab itu, perlu untuk menyimpan 3 solusi terbaik yaitu alpha, beta, dan delta. Setelah itu, memerintahkan serigala abu-abu yang lain untuk memperbarui posisinya berdasarkan posisi dari alpha, beta dan delta. Untuk mendeskripsikan proses tersebut dalam model matematis, pertama yang perlu dilakukan ada menghitung posisi jarak serigala dan mangsa dengan menggunakan rumus berikut :

$$\overrightarrow{D_\alpha} = \left|\overrightarrow{C_1} . \overrightarrow{X_\alpha}(t) - \vec{X}(t)\right| \quad \text{.....(2.5)}$$

$$\overrightarrow{D_\beta} = \left|\overrightarrow{C_2} . \overrightarrow{X_\beta}(t) - \vec{X}(t)\right| \quad \text{.....(2.6)}$$

$$\overrightarrow{D_\delta} = \left|\overrightarrow{C_3} . \overrightarrow{X_\delta}(t) - \vec{X}(t)\right| \quad \text{.....(2.7)}$$

Kemudian menghitung nilai $\vec{X}_1 . \vec{X}_2$ dan $\vec{X}_3$ dengan menggunakan rumus berikut :

$$\overrightarrow{X_1} = \overrightarrow{X_\alpha} - \overrightarrow{A_1} . \left(\overrightarrow{D_\alpha}\right) \quad \text{.....(2.8)}$$

$$\overrightarrow{X_2} = \overrightarrow{X_\beta} - \overrightarrow{A_2} . \left(\overrightarrow{D_\beta}\right) \quad \text{.....(2.9)}$$

$$\overrightarrow{X_3} = \overrightarrow{X_\delta} - \overrightarrow{A_3} . \left(\overrightarrow{D_\delta}\right) \quad \text{.....(2.10)}$$

Kemudian perbarui posisi setiap individu dengan menggunakan rumus berikut :

$$\vec{X}(t+1) = \frac{\overrightarrow{X_1} + \overrightarrow{X_2} + \overrightarrow{X_3}}{3} \quad \text{.....(2.11)}$$

Keterangan :

$\vec{X}(t+1)$ = hasil perubahan posisi serigala

$\overrightarrow{X_1}$ = perubahan posisi serigala berdasarkan alpha

$\overrightarrow{X_2}$ = perubahan posisi serigala berdasarkan beta

$\overrightarrow{X_3}$ = perubahan posisi serigala berdasarkan delta

2.4.3. Serigala Abu-Abu Menyerang Mangsa (*exploitation)*

Gambar 2. 4 Serigala Abu-Abu Menyerang Mangsa

(Sumber : S. Mirjalili, S. M. Mirjalili, and A. Lewis, “Grey Wolf Optimizer”)

Pada Gambar 2.4, ketika mangsa tidak dapat bergerak, serigala abu-abu melakukan penyerangan dengan cara mengigit mangsat tersebut. Serigala abu-abu akan menyelesaikan proses perburuan dengan menyerang mangsa saat ia berhenti bergerak. Proses serigala abu-abu dalam mendekati posisi mangsa dimodel matematiskan dengan cara mengurangi nilai $\vec{a}$. Dengan operator ini, algoritma GWO memungkinkan *search agents* dalam memperbarui posisi berdasarkan lokasi

alpha, beta, dan delta, sekaligus menyerang mangsa. Namun, algoritma GWO cenderung stagnasi dalam solusi lokal.

2.4.4. Serigala Abu-Abu Mencari Mangsa (*Exploration*)

Gambar 2.5 Serigala Mencari Mangsa

(Sumber : S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer")

Gambar 2.5 memperlihatkan proses serigala abu-abu dalam mencari mangsa. Serigala abu-abu akan mencari mangsa berdasarkan posisi alpha, beta, dan delta. Mereka akan berpencar untuk mencari mangsa dan akan berkumpul kearah mangsa untuk menyelesaikan perburuan. Proses pencarian dimulai dengan membuat posisi acak dari setiap serigala abu-abu. Selama proses perulangan, serigala alpha, beta, dan delta memperkirakan posisi dari mangsa. Vektor $\vec{C}$ berguna untuk menyimulasikan keadaan dimana, serigala akan menemui rintangan dalam mencari mangsa [18]. Adapun rintangan tersebut adalah pada saat mangsa mencoba untuk menghindari serigala abu-abu secara cepat dan aman. Vektor $\vec{C}$ secara acak akan membuat mangsa menjadi sulit untuk ditangkap atau sebaliknya. Setelah itu, setiap serigala abu-abu akan memperbarui jaraknya untuk mendekati mangsa. Parameter $\vec{a}$ yang akan menurun dari 2 ke 0 berguna untuk menekan proses ekplorasi dan eksploitasi, sehingga serigala abu-abu akan dipaksa untuk menyerang mangsa ketika sudah mendekati iterasi maksimal. Oleh karena itu, serigala abu-abu akan berpencar mencari mangsa pada saat $|\vec{A}| > 1$ dan serigala abu-abu akan mulai berkumpul dan menyerang mangsa pada saat $|\vec{A}| < 1$ [17]. Pada akhirnya, algoritma GWO akan berhenti ketika kriteria yang diinginkan terpenuhi atau sudah mencapai maksimal iterasi.