

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

HRP merupakan suatu proses perencanaan, identifikasi dan analisa segala kebutuhan sumber daya manusia untuk memenuhi tujuan organisasi (Aslam dkk., 2014). HRP yang sistematis dapat membantu memenuhi tujuan yang diinginkan dengan memaksimalkan seluruh potensi yang dimiliki dalam mewujudkan semua tujuan organisasi. Salah satu cara untuk memaksimalkan tujuan adalah dengan mengetahui karakteristik yang dimiliki masing-masing pegawai karena keterampilan dan pengalaman kerja yang dimiliki sangat berpengaruh pada prestasi kerja (Chen dkk., 2019). Konsep manajemen beban kerja pegawai merupakan penentu penting dari produktivitas dan pergantian pegawai (Inegbedion dkk., 2020). Manajemen beban pekerjaan meliputi pendistribusian tugas dan pemerataan beban kerja agar seluruh pekerjaan dapat dibagi secara adil dan proporsional. Manajemen beban kerja juga bertujuan untuk meminimalkan ketidakseimbangan beban kerja dalam organisasi dan meningkatkan kepuasan kerja pegawai dalam kaitannya dengan hasil pekerjaan. Pendistribusian bertujuan untuk menempatkan pegawai kedalam pos pekerjaan yang sesuai untuk meningkatkan kinerja organisasi secara menyeluruh.

Pendistribusian pekerjaan kepada pegawai merupakan masalah umum yang dihadapi dalam manajemen organisasi. Optimalisasi distribusi pekerjaan telah dilakukan pada sistem pencocokan perekrutan pegawai untuk memaksimalkan tingkat kesesuaian antara pencari kerja dan perusahaan (Lee dan Ahn, 2020). Penelitian mengenai pencocokan perekrutan pegawai dilakukan untuk mencari kesesuaian antara calon pegawai dan perusahaan, proses optimalisasi dilakukan menggunakan dasar preferensi keterampilan, wilayah dan MBTI (Lee dan Ahn, 2020). Optimalisasi pengelolaan sumber daya manusia perlu dilakukan agar dapat melakukan pengendalian rasio pencocokan pegawai dan pekerjaan (Lee dan Ahn, 2020). Kesesuaian antara pegawai dan pekerjaan merupakan penentu penting efektifitas kinerja dan mendukung perusahaan dalam

merekrut calon pegawai. Masalah optimalisasi beban pekerjaan pegawai dapat diselesaikan dengan *job matching*, preferensi dan penjadwalan. *Job matching* dapat digunakan untuk menentukan kesesuaian antara pencari kerja dan perusahaan yang sesuai (Mhamdi dkk., 2020). *Job matching* melakukan pencocokan data kemudian membuat rekomendasi pekerjaan berdasarkan peringkat (Mhamdi dkk., 2020). Penelitian mengenai *job matching* mempelajari masalah perencanaan proyek yang dibatasi sumber daya yang fleksibel dengan perbedaan kompetensi. Optimalisasi bertujuan untuk meminimalkan biaya total dan total *makespan* dengan kendala keterbatasan sumber daya manusia dan sumber daya material. Hasil penelitian *job matching* menunjukkan bahwa model yang diusulkan dengan mempertimbangkan perbedaan karakteristik dapat menyelesaikan masalah secara efektif.

Perencanaan keseimbangan beban pekerjaan dilakukan dengan membandingkan jumlah pegawai yang dibutuhkan dengan jumlah pekerjaan (Aslam dkk., 2014). Keahlian pegawai dapat diketahui berdasarkan latar belakang pendidikan, namun pada kenyataannya banyak pegawai yang melakukan pekerjaan tidak sesuai dengan latar belakang pendidikannya. Penelitian mengenai teknik pengelompokan menggunakan *K-means clustering* digunakan pada evaluasi kinerja pegawai, *K-means* digunakan untuk mengklasifikasikan pegawai berdasarkan kinerja kedalam *cluster* yang berbeda agar mempermudah proses pengambilan keputusan (Sarker dkk., 2018). Penelitian tersebut menggunakan *dataset* pegawai dengan indikator penilaian berdasarkan sembilan belas kategori. *Dataset* pegawai tersebut di kelompokkan kedalam empat *cluster* (sangat baik, baik, sedang dan kurang). Setelah proses pengelompokan, dihasilkan empat *cluster* berdasarkan kinerja pegawai selama satu tahun.

Proses pecocokan antara pekerjaan dan karakteristik pegawai merupakan salah satu upaya dalam efisiensi distribusi beban pekerjaan. Algoritma *Tabu Search* dikembangkan untuk memecahkan masalah penentuan waktu kursus pada universitas dalam menentukan jadwal mingguan (Chen dkk., 2020). Penelitian tersebut mengintegrasikan strategi pengacakan terkontrol dan mekanisme batas pada pencarian *Tabu*. Proses pertama dilakukan pengenalan dua

lingkungan untuk memastikan pencarian secara intensif dan mengurangi waktu pencarian, kemudian mekanisme ambang batas diadopsi dalam pencarian lingkungan. Strategi penerimaan acak diusulkan untuk menentukan hasil pencarian, apabila hasil tidak terlalu baik akan tetap diterima dengan probabilitas tertentu sehingga kualitas solusi dapat diterima. *Tabu Search* merupakan solusi dalam memenuhi kendala dengan mengintegrasikan strategi pengacakan terkontrol untuk menghasilkan kecocokan dengan jarak terdekat (Chen dkk., 2020). Kecocokan dapat terjadi apabila semua variabel yang dimiliki sesuai dengan pasangan yang didapat.

Berdasarkan beberapa studi kasus yang telah diuraikan, maka proses pengelompokan dilakukan menggunakan metode *K-Means Clustering* dan proses pencocokan dilakukan menggunakan algoritma *Tabu Search*. Dengan keunggulan yang dimiliki masing-masing metode tersebut, maka kombinasi metode diperlukan dalam merancang sistem *Human Resource Planning* untuk pemerataan beban kerja pegawai. Pengembangan penelitian terkait terdapat pada kombinasi performa dari masing-masing metode. Studi kasus terkait yang diperoleh belum ada yang mengombinasikan kedua metode tersebut, sehingga penelitian ini dilakukan untuk mengoptimalkan hasil pemerataan beban agar dapat meningkatkan kinerja organisasi.

2.2 Dasar Teori

2.2.1 Clustering

Clustering merupakan proses pengelompokan dan penempatan suatu objek yang memiliki keterkaitan satu sama lain dan memiliki karakteristik serupa untuk menyelidiki struktur yang mendasari data (Nas, 2020). *Clustering* merupakan proses partisi suatu objek kedalam himpunan bagian yang disebut sebagai *cluster*, partisi tersebut dilakukan melalui proses analisis menggunakan algoritma *clustering* (Aggarwal dan Chandan, 2014). Analisis *clustering* merupakan metode pengelompokan, *clustering* dibagi menjadi beberapa wilayah yang mana objek yang memiliki kesamaan diletakkan kedalam wilayah yang sama (Jain, 2010).

Dalam implikasinya, setiap proses membutuhkan analisis *clustering*. Analisis *clustering* atau pengelompokan data memiliki tujuan utama yakni (Jain, 2010), (i) Mencari struktur yang mendasari untuk mendapatkan detil data, menghasilkan hipotesis, mendeteksi anomali dan mengidentifikasi fitur yang menonjol; (ii) Melakukan klasifikasi untuk identifikasi kesamaan bentuk; (iii) Melakukan kompresi untuk mengatur data dan meringkasnya melalui proses *clustering*. Algoritma *clustering* secara garis besar dibagi menjadi dua kelompok yaitu hierarki dan partisi. Hierarki *clustering* dimulai dari setiap titik data dalam *clustering* sendiri yang kemudian digabungkan dengan pasangan *clustering* lain yang paling mirip secara berturut-turut untuk membentuk hierarki (Jain, 2010). Partisi *clustering* menemukan semua *cluster* secara bersamaan sebagai partisi data dan tidak memaksakan struktur hierarki (Jain, 2010). Berdasarkan perbedaan tersebut, maka teknik yang akan digunakan dalam penelitian ini adalah partisi *clustering* yaitu metode *K-means*.

2.2.2 *K-means Clustering*

K-means Clustering termasuk dalam partisi *clustering* dan merupakan salah satu algoritma pada model *distance-based algorithm* yang mana perwakilan partisi sesuai dengan rata-rata setiap *cluster*. *K-means* adalah salah satu jenis pembelajaran tanpa pengawasan, yang digunakan dalam data tak berlabel (Sarker dkk., 2018). *K-means* merupakan metode yang paling banyak digunakan karena kesederhanaan dan efisiensinya, namun parameter k harus ditentukan terlebih dahulu (Aggarwal dan Chandan, 2014). Pada *K-means*, k merupakan banyaknya parameter *input* yang membagi sekumpulan n objek kedalam k *cluster* untuk menemukan tingkat kemiripan yang tinggi dalam suatu *cluster* dan menentukan tingkat kemiripan rendah dengan anggota *cluster* lain. Kemiripan anggota terhadap kelompoknya berdasarkan kedekatan nilai rata-rata pada *cluster* atau sebagai *centroid cluster* (Nas, 2020). Proses pengelompokan data atau objek kedalam *cluster* dilakukan mengikuti tahapan metode *K-means*.

Adapun tahapan utama dalam algoritma *K-means* adalah sebagai berikut (Aggarwal dan Chandan, 2014):

1. Menentukan banyak *cluster* dengan nilai k adalah jumlah *cluster* dan menentukan k *cluster* sebagai *centroid* awal.
2. Menentukan nilai *centroid*.

Penentuan nilai *centroid* pada awal iterasi dilakukan secara acak, yang digunakan sebagai titik awal untuk setiap *cluster*. Nilai *centroid* saat proses iterasi berlangsung menggunakan persamaan (2. 1).

$$v_{ij} = \frac{\sum_{k=1}^{N_i} x_{kj}}{N_i} \quad (2. 1)$$

Persamaan 2.1 merupakan proses iterasi penentuan nilai *centroid*, v_{ij} merupakan nilai *centroid*/rata-rata *cluster* ke- i untuk variabel ke- j , N_i merupakan banyaknya anggota *cluster* ke- i , i, k merupakan indeks dari *cluster*, j merupakan indeks dari variabel dan x_{kj} merupakan nilai data ke- k yang ada didalam *cluster* tersebut untuk variabel ke- j .

3. Membuat partisi baru berdasarkan k *cluster* yang terdekat dengan *centroid*. Jarak setiap data terhadap masing-masing *centroid* dengan menggunakan persamaan (2. 2).

Sekolah Pascasarjana

$$d(P, Q) = \sqrt{\sum_{j=1}^p \{x_j(P) - x_j(Q)\}^2} \quad (2. 2)$$

Pada pembuatan partisi baru, d menyatakan jarak data ke *centroid*, x merupakan nilai data, P merupakan data pada *record* dan Q merupakan nilai *centroid*.

4. Pengalokasian masing-masing data ke *centroid* terdekat dan kemudian hitung *centroid cluster* dari k *cluster* dengan menghitung rata-rata anggota *cluster*. Penentuan alokasi data ke *cluster* terdekat atau minimal menggunakan persamaan (2.3) dan penentuan *centroid* baru dapat menggunakan persamaan (2.4).

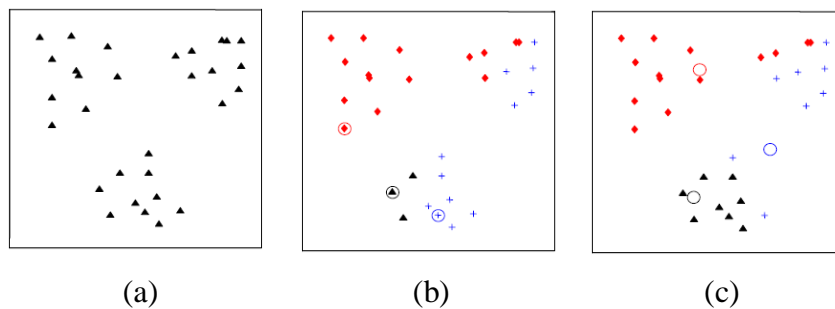
$$d = \min\{D(P_i, C_1)\}, a_{ij} \begin{cases} 1 \\ 0 \end{cases} \quad (2.3)$$

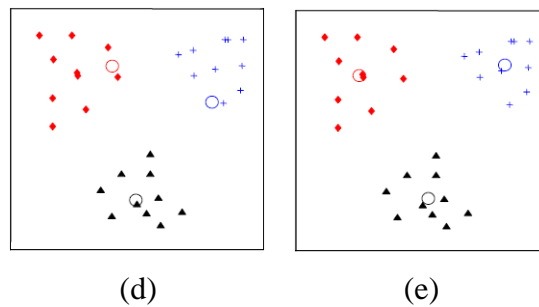
$$C(i) = \frac{x_1+x_2+\dots+x_n}{\sum x} \quad (2.4)$$

Pada persamaan 2.3, a_{ij} merupakan nilai keanggotaan titik P_i ke *centroid* C_1 , C_1 adalah nilai *centroid* ke-1 dan d adalah jarak terpendek dari data P_i ke k kelompok setelah dibandingkan. Pada persamaan 2.4, C merupakan *centroid* baru yang diperoleh, $x_1 \dots x_n$ merupakan nilai data *record* 1 sampai n dan $\sum x$ adalah jumlah data *record*.

5. Ulangi langkah 3 dan 4 hingga tidak ada perubahan dalam fungsi kriteria setelah iterasi atau kriteria konvergensi ditemukan.
6. Proses akan berhenti dan telah selesai dalam optimalisasi *cluster* apabila *centroid* telah stabil atau tidak ada perubahan nilai dan jumlah iterasi yang ditemukan telah tercapai.

Gambar 2.1 merupakan contoh ilustrasi algoritma *K-means* dengan tiga *cluster* (Jain, 2010).





Gambar 2. 1 Ilustrasi proses algoritma *K-means*

Ilustrasi pada gambar 2.1 menjelaskan bahwa:

- Proses *input* data dengan tiga *cluster*;
- Tiga titik terbaik dipilih sebagai *centroid cluster* dan merupakan penugasan awal dari titik data pada *cluster*;
- Proses iterasi setelah tahap sebelumnya;
- Iterasi perantara yang dapat memperbarui label setiap *cluster* dengan *centroid*;
- Proses akhir dari *clustering* berdasarkan algoritma *K-means* pada konvergensi.

2.2.3 Algoritma *Tabu Search*

Tabu Search merupakan metode pencarian lokal metaheuristik yang digunakan untuk optimasi matematika (Glover, 1990). Salah satu komponen utama pada *Tabu Search* adalah penggunaan memori adaptif yang membuat prinsip pencarian menjadi lebih fleksibel. *Tabu Search* didasarkan pada premis bahwa pemecahan masalah bertujuan untuk mencari nilai optimal pada memori adaptif dan eksplorasi responsif (Glover, 1990). Fitur memori adaptif memungkinkan pencarian ruang solusi efektif dan efisien karena pilihan lokal berdasarkan informasi yang dikumpulkan selama pencarian. Apabila pencarian yang pernah dilakukan (*tabu*) tidak lebih baik, maka algoritma ini akan mengabaikan langkah pencarian tersebut dan tetap menggunakan solusi yang terbaik (Glover dan Laguna, 2013). Algoritma *Tabu Search* dapat melakukan eksplorasi dalam pencarian ruang yang lebih luas (Chen dkk., 2020).

Prinsip utama algoritma *Tabu Search* adalah penggunaan struktur memori berbasis atribut yang fleksibel, sehingga kriteria evaluasi dan pencarian dilakukan

secara menyeluruh (Glover, 1990). Mekanisme *Tabu Search* dilakukan dengan mengevaluasi gerakan berdasarkan kriteria yang diberikan dalam penelusuran *tabu*. Penggunaan fungsi memori berdasarkan rentang waktu berbeda (jangka pendek hingga panjang) dalam menerapkan strategi agar fokus pada area tertentu yang mendorong pencarian wilayah baru.

2.2.4.1 Strategi

Tabu Search memiliki tiga strategi utama dalam melakukan proses implementasi yakni *forbidding*, *freeing* dan *short-term*. *Forbidding strategy* digunakan untuk melakukan kontrol masukan pada *tabu list*, *freeing strategy* untuk mengontrol informasi yang keluar dari *Tabu Search* dan kapan saja waktunya, kemudian *short-term strategy* untuk mengatur pengaruh *forbidding strategy* dan *freeing strategy* dalam pemilihan solusi yang optimal (Glover dkk., 1997). Dalam implementasinya, *Tabu Search* merupakan metode optimasi yang menggunakan komponen *short term memory* untuk menjaga agar proses pencarian agar tidak terjebak pada nilai optimal lokal. Struktur *short-term memory* dalam *Tabu Search* disebut sebagai *tabu list*, yang mana setiap langkah yang dilakukan pada iterasi sebelumnya disimpan pada *Tabu list* untuk menyimpan solusi akhir sementara dan menghindari siklus jangka pendek (Glover dkk., 1997). Prinsip utama *Tabu Search* adalah melakukan pemeriksaan tetangga secara berurutan dari suatu solusi ke solusi berikutnya yang paling optimum tanpa terjebak pada solusi awal (Graiet dkk., 2020).

Metode optimasi tersebut menggunakan *tabu list* dalam menyimpan solusi baru yang telah di evaluasi. Pada proses optimasi, setiap iterasi yang menghasilkan solusi yang akan dicocokkan dengan isi *tabu list* untuk melihat apakah solusi tersebut sudah ada pada *tabu list*. Apabila solusi tersebut sudah ada maka tidak akan di evaluasi lagi pada iterasi berikutnya dan tidak ada lagi solusi yang akan menjadi anggota *tabu list* (Glover dan Laguna, 2013). Selanjutnya, didapatkan nilai terbaik yang merupakan kecocokan dan solusi terbaik. Strategi yang dimiliki *Tabu Search* adalah dapat menemukan solusi optimum local sehingga optimasi dapat dilakukan secara maksimal.

2.2.4. 2 Tahap Pengembangan Algoritma *Tabu Search*

Implementasi dan pengembangan algoritma *Tabu Search* dilakukan dengan tiga tahap, meliputi:

1. Tahap Pencarian Awal

Perencanaan awal dilakukan dengan beberapa tahap sebagai berikut:

a. Tahap 0: Inisialisasi

Tahap inisialisasi diawali dengan memasukkan *input* berupa kendala presedensi, waktu siklus dan waktu proses masing-masing operasi. Kemudian menentukan nilai parameter yang digunakan antara lain (i) *Tabu list size* (T_s) yang mengatur waktu dan jumlah *move* pada kondisi tabu; (ii) Iterasi maksimal (N_{maks}) yang mengatur jumlah iterasi maksimum yang diinginkan. Tahap terakhir adalah menentukan solusi awal.

b. Tahap 1: Pembentukan *Candidate move*

Pembentukan *candidate move* dilakukan dengan metode *neighborhood search*. Pencarian *candidate move* dilakukan hingga ditemukan jumlah yang diinginkan (J_{lmaks}). Proses ini dihasilkan melalui proses *swap* dan *insert* dengan masing-masing memiliki probabilitas yang sama untuk menghasilkan solusi. Apabila tidak menghasilkan solusi yang diinginkan, maka proses akan dilanjutkan menggunakan metode lainnya (*swap* dan *insert*).

c. Tahap 2: Pemilihan *Move* Terbaik

Tahap ini akan dilakukan pemilihan *move* terbaik dari beberapa kandidat yang telah diperoleh, namun sebelumnya perlu di periksa status tabu pada setiap kandidat. Apabila *candidate move* berstatus tabu, maka akan dipilih berdasarkan kriteria aspirasi. Apabila *candidate move* tidak berstatus *tabu* maka dipilih yang terbaik berdasarkan hasil evaluasi pada tahap 1.

d. Tahap 3: Kriteria Aspirasi

Kriteria ini mengesampingkan pembatasan tabu atau mengijinkan pergerakan sebuah *move* yang berstatus tabu.

2. Tahap Intensifikasi

Tahap intensifikasi merupakan proses pencarian yang difokuskan pada solusi terbaik yang telah ditemukan. Proses ini dilakukan berulang hingga mencapai nilai iterasi intensifikasi maksimum (*nimaks*). Langkah dalam proses intensifikasi adalah sebagai berikut:

- a. Langkah 0 : inisiasi $S = S^*$, kosongkan *tabu list* dan *set* variabel menjadi 0.
- b. Langkah 1 : pembentukan *candidate move*.
- c. Langkah 2 : pemilihan *move* terbaik.
- d. Langkah 3 : kriteria aspirasi.
- e. Langkah 4 : proses evaluasi *candidate move* terpilih.
- f. Langkah 5 : kriteria pemberhentian.

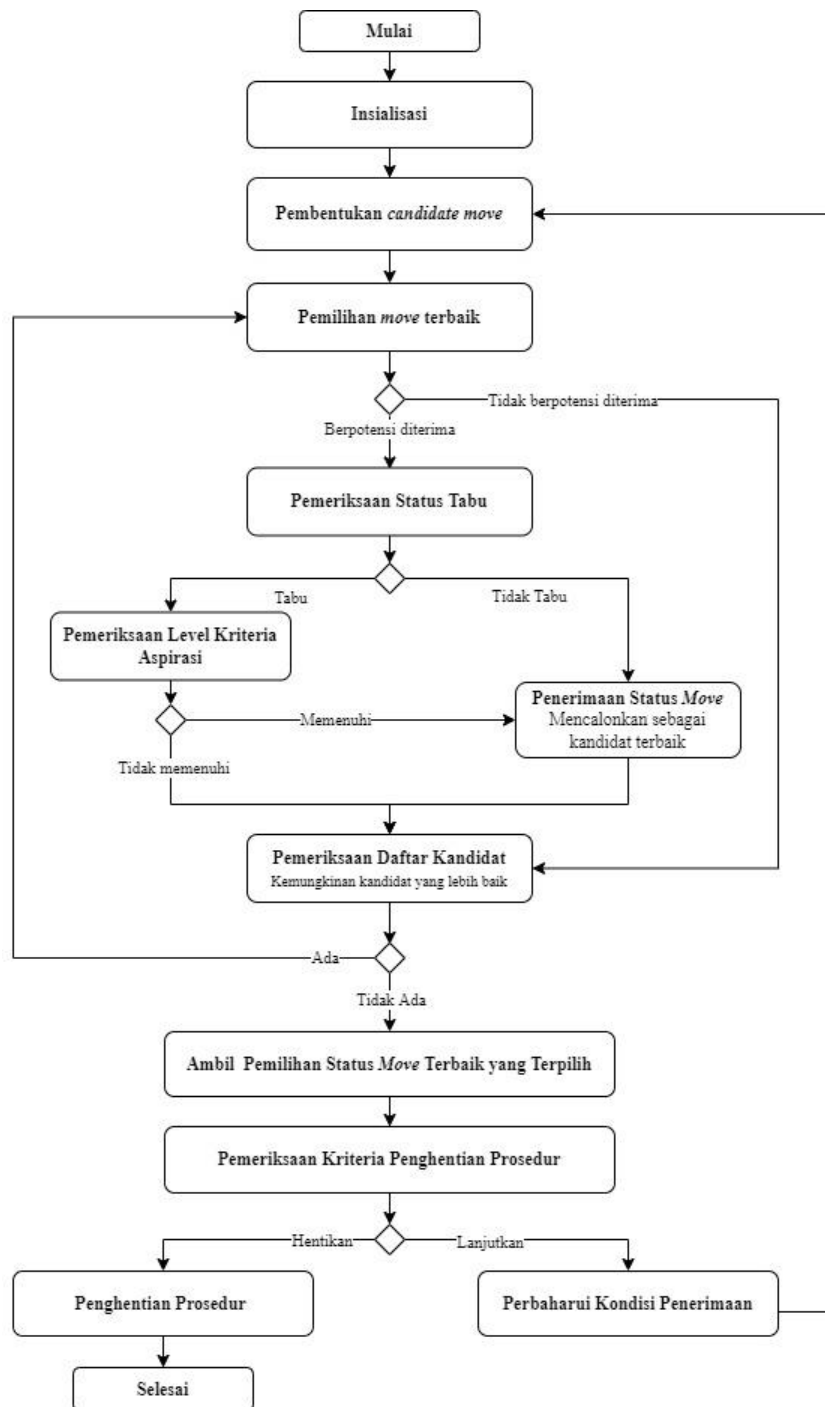
3. Tahap Diversifikasi

Tahap diversifikasi merupakan proses pencarian yang lebih menyebar untuk membuka daerah baru yang belum dikunjungi. Proses diversifikasi akan dilakukan terus hingga mencapai nilai iterasi diversifikasi maksimum (*ndmaks*). Langkah dalam proses diversifikasi adalah sebagai berikut:

- a. Langkah 0 : inisialisasi.
- b. Langkah 1 : pembentukan *candidate move*.
- c. Langkah 2 : pemilihan *move* terbaik.
- d. Langkah 3 : kriteria aspirasi.
- e. Langkah 4 : proses evaluasi *candidate move* terpilih.
- f. Langkah 5 : kriteria pemberhentian.

2.2.4.3 Algoritma *Tabu Search* untuk Menyelesaikan Pencocokan

Flowchart diagram umum dari algoritma *Tabu Search* dapat dilihat pada gambar 2.3.



Gambar 2. 2 *Flowchart* diagram algoritma *Tabu Search*

(Glover dkk., 1997)

Berdasarkan *flowchart* pada gambar 2.2, penerapan *Tabu Search* dilakukan dengan tahapan sebagai berikut:

1. Inisialisasi merupakan permulaan proses dengan sebuah solusi awal saat ini. Solusi saat ini diperoleh dari sebuah tahap insialisasi atau dari sebuah komponen memori jangka pendek. Proses ini juga bertujuan untuk menentukan solusi awal dan menetapkannya sebagai solusi optimum berdasarkan nilai fungsi tujuan.
2. Pembentukan kandidat *move* yaitu menentukan solusi alternatif dengan melakukan penukaran dua titik dalam solusi. Pada tahap ini juga dibuat sebuah daftar kandidat perpindahan solusi, jika daftar tersebut sudah dipakai maka setiap perpindahan akan menghasilkan sebuah solusi baru dari solusi saat ini.
3. Evaluasi setiap kandidat solusinya dan pemilihan *move*/kandidat terbaik. Proses evaluasi kandidat dilakukan untuk mengetahui apakah kandidat solusi memberikan nilai tujuan yang lebih baik dibandingkan kandidat lain yang telah dibuat sejauh ini. Penerimaan kandidat ini juga akan didasarkan pada *tabu restriction* dan *aspiration criteria*.
4. Pemeriksaan status tabu bertujuan untuk mengetahui apakah kandidat pada proses pemilihan *move* tersebut berstatus tabu atau tidak tabu.
 - a. Apabila kandidat bersifat tidak tabu maka kandidat tersebut dicalonkan sebagai kandidat yang diterima terbaik dan proses akan berlanjut pada penerimaan status *move*.
 - b. Apabila kandidat bersifat tabu maka akan diperiksa pada *level* aspirasi berdasarkan calon kandidat tabu.
5. Penerimaan status *move* dilakukan dengan mencalonkan sebagai kandidat terbaik yang diterima. Apabila status *move* diterima, penerimaan akan dicatat dan dilanjutkan pada proses pemeriksaan daftar kandidat.
6. Pemeriksaan *level* aspirasi.

Pemeriksaan *level* aspirasi bertujuan untuk memeriksa kandidat yang tabu berdasarkan kriteria aspirasi. *Aspiration criteria* (kriteria aspirasi) merupakan suatu penanganan khusus terhadap *move* (perpindahan) yang dapat

menghasilkan solusi yang baik namun *move* tersebut berstatus tabu. Apabila *move* tersebut memenuhi kriteria aspirasi yang telah ditetapkan sebelumnya, maka *move* tersebut dapat digunakan untuk membentuk solusi terbaik berikutnya (status tabu dibatalkan) dan proses akan dilanjutkan pada penerimaan status *move*.

- a. Apabila memenuhi kriteria aspirasi maka perpindahan (status tabu ke status tidak tabu) tersebut diterima atau dicalonkan sebagai kandidat diterima yang terbaik (status tidak tabu).
 - b. Apabila tidak memenuhi kriteria aspirasi maka proses akan berlanjut pada pemeriksaan daftar kandidat tanpa pencatatan penerimaan untuk diperiksa kembali apakah masih ada kemungkinan yang lebih baik dari perpindahan yang tersisa. Jika memang masih ada kemungkinan yang lebih baik maka dilakukan pengulangan proses pemeriksaan status tabu.
7. Pemeriksaan daftar kandidat apakah masih terdapat sebuah “kemungkinan yang lebih baik” dari perpindahan yang tersisa (belum diperiksa). Jika tidak maka proses akan berlanjut pada pengambilan perpindahan terbaik.
 8. Ambil perpindahan diterima terbaik yang terpilih.
 9. Kriteria pemberhentian prosedur.
 - a. Apabila kondisi pemberhentian terpenuhi, maka hentikan seluruh prosedur dan diperoleh solusi terbaiknya.
 - b. Apabila masih ada akan dilanjutkan dengan membuat kembali daftar perpindahan dari perpindahan berdasarkan solusi terbaik terakhir.

2.2.4 *Sillhouette Method*

Sillhouette Method merupakan metode yang digunakan pada *K-Means Clustering* untuk menemukan jumlah *cluster* yang optimal serta melakukan validasi konsistensi dalam *clustering* data. Metode ini digunakan untuk mengukur tingkat kemiripan suatu titik dengan *cluster*-nya sendiri dan dibandingkan dengan *cluster* lainnya (Werdiningsih dkk., 2019). Koefisien siluet atau titik ke-*i* menggunakan persamaan (2.5) dibawah ini (Werdiningsih dkk., 2019).

$$s(i) = \frac{b(i)-a(i)}{\max(b(i), a(i))} \quad (2.5)$$

Pada persamaan 2.5, $s(i)$ merupakan koefisien siluet, $a(i)$ merupakan jarak rata-rata titik tersebut dengan semua titik lain dalam kelompok yang sama dan $b(i)$ merupakan jarak rata-rata dari titik tersebut dengan semua titik dalam *cluster* terdekat dengan *cluster*-nya sendiri.

Perhitungan koefisien *silhouette* memiliki rentang nilai antara [-1,1], nilai 1 menunjukkan arti terbaik bahwa titik i sangat dekat dengan *cluster* nya sendiri dan jauh dari *cluster* lainnya. Nilai terburuk adalah -1 karena nilai yang mendekati 0 menunjukkan *cluster* yang tumpang tindih. Setelah menghitung koefisien *silhouette* pada setiap titik pada *dataset*, dilakukan analisis siluet untuk mengukur kinerja *clustering* data yang telah dilakukan. Analisis siluet menggunakan plot untuk mendapatkan representasi visual tentang seberapa baik *dataset* tersebut dikelompokkan menjadi k *cluster*. Plot tersebut menampilkan seberapa dekat setiap titik dalam satu *cluster* dengan *cluster* tetangga.

2.2.5 Standar Deviasi

Standar deviasi merupakan variasi data yang digunakan sebagai simpangan baku yang merupakan akar kuadrat dari varian. Penghitungan standar deviasi digunakan untuk mengetahui keragaman kelompok data (Suryanto, 2017). Pada standar deviasi, apabila ukuran sampel memiliki nilai yang kecil menunjukkan data pada himpunan tersebut mendekati atau berada pada rata-rata nilai himpunan. Standar deviasi digunakan untuk melakukan validasi dalam memastikan hasil komputasi. Validasi dilakukan dengan membandingkan data hasil keluaran model dengan perhitungan matematis untuk mengetahui hasil sebaran data. Standar deviasi dituangkan dalam persamaan (2.6) (Suryanto, 2017).

$$s = \sqrt{\frac{n \sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2}{n(n-1)}} \quad (2.6)$$

Pada persamaan 2.6, s merupakan standar deviasi, x_i merupakan nilai x ke i , \bar{x} merupakan nilai rata-rata x dan n adalah ukuran sampel datanya. Nilai standar deviasi rendah menunjukkan titik data cenderung dekat dengan *means* himpunan, sedangkan standar deviasi tinggi menunjukkan titik data lebih besar dari rentang nilai dan dapat dikatakan data sebaran kurang baik. Nilai rentang tersebut menunjukkan kualitas sebaran data yang diperoleh dari hasil penelitian.



Sekolah Pascasarjana