BAB II
LANDASAN TEORI

2.1 Permasalahan Pengelompokan dengan Diversitas Maksimal

Maximally Diverse Grouping Problem (MDGP) adalah permasalahan tentang pengelompokan sekumpulan elemen M ke dalam kelompok G yang saling terpisah sedemikian rupa sehingga keragaman di antara elemen-elemen dalam setiap kelompok bernilai maksimal. Feo and Khellaf (1990) membuktikan bahwa Maximally Diverse Grouping Problem (MDGP) adalah NP-Hard Problem.

Masalah ini banyak terjadi di dunia nyata. Pertama kali ditekankan oleh Weitz dan Jelassi (1992) dalam pekerjaan mereka dalam membentuk kelompok kerja siswa, MDGP memiliki beragam aplikasi dalam konteks yang berbeda, seperti pendidikan, industri, lembaga atau organisasi pendanaan ilmiah. Di sekolah bisnis, semakin umum untuk membuat kelompok kerja siswa yang beragam atau tim pelatihan untuk memberikan siswa lingkungan yang beragam (Weitz dan Lakshminarayanan, 1998)

Setiap elemen M yang akan dikelompokan ke dalam sejumlah kelompok G, mempunyai atribut sebanyak K. Sehingga s_{ik} adalah nilai elemen atribut k, di mana k = 1, ..., K dan i = 1, ..., M. Kemudian, untuk menghitung diversitas atau jarak d_{ij} antara elemen i dan j dapat dengan ditentukan oleh perhitungan Euclidean:

$$d_{ij} = \sqrt{\sum_{k=1}^{K} (s_{ik} - s_{jk})^2}$$

Persamaan 2.1

Terdapat dua buah varian MDGP. Yang pertama (MDGP1) adalah yang semua kelompok untuk memiliki jumlah elemen yang sama, dengan $S = M / G$. Varian kedua (MDGP2) memungkinkan jumlah elemen S_g dari setiap grup g berada dalam rentang [a_g, b_g], di mana a_g <= $_{bg}$ untuk $g = 1,\ldots, G$. Pada studi kasus penelitian ini, varian MDGP yang sesuai adalah MDGP1 karena setiap kamar memiliki jumlah mahasiswa yang sama yaitu empat mahasiswa per kamar asrama.

Untuk memformulasikan MDGP, diperlukan sebuah variabel x_{ig} yang berisi nilai 1 jika elemen i berada dalam grup g dan berisi 0 jika sebaliknya. Sehingga, formulasi matematis MDGP adalah sebagai berikut:

$$Maximize \sum_{g=1}^{G} \sum_{i=1}^{M-1} \sum_{j>1}^{M} d_{ij} x_{ig} x_{jg}$$ Persamaan 2.2

Persamaan 2.3
$$subject\ to \sum_{g=1}^{G} x_{ig} = 1 \qquad i = 1, 2, \ldots, M$$

Persamaan 2.4
$$\sum_{i=1}^{M} x_{ig} = S \qquad g = 1, 2, \ldots, G$$

Persamaan 2.5
$$x_{ig} \in \{0,1\} \qquad i = 1, \ldots M\ g = 1, \ldots, G$$

Fungsi objektif yang ada pada persamaan pertama adalah menjumlahkan jarak semua pasangan elemen yang dimiliki oleh sebuah grup. Persamaan kedua memastikan setiap elemen telah masuk kedalam satu grup. Persamaan ketiga memastikan jumlah elemen pada semua kelompok adalah sama dengan S. Dan persamaan terakhir memastikan nilai variabel x dari setiap i pada masing-masing g adalah 0, atau 1.

2.2 Metode Lofti-Cerveny-Weitz

Metode Lofti-Cerveny-Weitz adalah versi modifikasi dari metode Lofti-Cerveny (LC), yang awalnya diterbitkan oleh Lofti dan Cerveny (1991) sebagai bagian dari metode komprehensif untuk penjadwalan ujian akhir dengan tujuan meminimalkan (bukan memaksimalkan) keragaman di masing-masing kelompok. Weitz dan Lakshminarayanan (1996) menemukan dan memperbaiki sejumlah kesalahan dalam metode Lofti dan Cerveny tersebut.

```
Konstruksi solusi awal
do {
    for (i = 1, …, M) {
        1. Cari elemen j pada semua grup yang mana perpindahan grup
           antara elemen i dan j menghasilkan perubahan terbesar pada
           fungsi objektif
        2. Jika perubahan pada fungsi objektif bernilai positif, lakukan
           perubahan
    }
}
```

Gambar 2.1 Metode LCW

Metode LCW adalah variasi dari LC di mana pencarian untuk elemen *j* tidak terbatas pada grup *g*. LCW mempertimbangkan semua grup saat mencari elemen *j* kecuali pada grup dimana elemen *i* berada. Metode LCW dirangkum dalam Gambar 2.1.

2.3 Algoritma Genetik Pencarian Lokal

Algoritma genetik telah berhasil digunakan untuk menemukan solusi untuk berbagai masalah optimasi (misalnya, Goldberg (1989) Gen dan Cheng (1997)), sejak diperkenalkan oleh Holland (1975). GA adalah teknik optimisasi yang cerdas berdasarkan simulasi evolusi biologis. ZP Fan (2011) menerangkan langkah-langkah heuristik berbasis GA untuk MDGP dilakukan sebagai berikut:

1.3.1 Pengkodean

MDGP pada dasarnya terdiri dari dua jenis informasi, elemen dan kelompok. Skema pengkodean Falkenauer (1998) terdiri dari informasi elemen dan informasi grup. Di bagian objek, setiap gen ditugaskan ke nomor grup, yang menunjukkan bahwa elemen yang sesuai dialokasikan ke grup itu. Di bagian grup, semua nomor grup tercantum dalam urutan acak. Notasi berikut digunakan untuk representasi kromosom:

$$q_1, q_2, ..., q_M \mid Q_1, Q_2, ..., Q_G$$

dimana $q_i \in \{1,2,...,G\}$ menunjukan elemen p_i dialokasikan ke grup q_i $(1 \leq i \leq M)$, dan Q_g adalah nomor grup $(1 \leq g \leq G)$.

1.3.2 Inisialisasi

Proses inisialisasi dilakukan dengan memilih secara acak anggota M untuk masing-masing grup G. Setiap grup diisi oleh S anggota. Jika M / G adalah bilangan bulat, semua grup memiliki ukuran grup yang sama yaitu M / G; jika tidak, grup sebanyak (G - 1) pertama memiliki ukuran grup yang sama yaitu M / G, dan kelompok yang terakhir memiliki ukuran grup (M – S[G-1])

1.3.3 Seleksi

Proses seleksi menggunakan strategi *roulette-wheel* berbasis peringkat untuk memilih *parent* untuk proses *crossover*. Setiap kromosom diberi bagian dari roda roulette imajiner, berdasarkan peringkat nilai *fitness*-nya. Nilai *fitness* dihitung dengan fungsi objektif. Sebelum pemilihan dua buah *parent*, persentase antara 0 dan 100 dihasilkan secara acak. Kromosom yang menempati bagian roda roulette yang dicakup oleh persentase yang dihasilkan secara acak dipilih sebagai *parent*. *Parent* kedua juga dipilih dengan cara yang sama.

1.3.4 Crossover

Proses *crossover* menghasilkan individu baru dengan menggabungkan dua buah *parent*. *Crossover* algoritma genetik standar menghasilkan keturunan dengan menukar sebagian kromosom induk. Seperti yang ditunjukkan oleh Falkenauer (1998), jenis operasi ini, ketika diterapkan pada masalah pengelompokan, dapat menghasilkan tumpang tindih anggota di antara kelompok. Ini berarti kromosom keturunan mungkin mengandung satu atau lebih kelompok yang berbagi beberapa elemen yang sama. Jenis keturunan ini tidak valid karena masalah pengelompokan mengharuskan setiap elemen hanya untuk sebuah grup. Jadi, operasi *crossover* khusus untuk masalah pengelompokan diusulkan oleh Falkenauer (1998).

Langkah *crossover* yaitu pertama, pilih titik tengah dari masing-masing parent. Kemudian masukan konten dari *parent* kedua ke titik tengah dari *parent* pertama. Berikutnya, hapus semua elemen yang ada dua setelah dilakukan injeksi. Setelah prosedur *crossover*, keturunan yang dihasilkan mungkin masih ilegal karena ukuran grup bisa tidak sesuai. Dalam situasi ini, algoritma penyesuaian ukuran grup dilakukan untuk menyesuaikan ukuran grup.

1.3.5 Mutasi

Operasi mutasi yaitu memilih secara acak dua kelompok dan kemudian memilih β elemen yang ada pada kelompok tersebut, di mana β adalah jumlah gen yang hendak dimutasi. Mutasi dilakukan dengan menukar β elemen ke kelompok lain yang telah dipilih.

1.3.6 Kriteria Berhenti

Kriteria berhenti paling mudah untuk GA adalah jumlah generasi (iterasi). Jika jumlah generasi rendah, probabilitas untuk menemukan hasil terbaik adalah jgua rendah. Dan jika jumlah generasi terlalu tinggi, maka akan menyebabkan waktu pemrosesan terlalu lama. Pada Algoritma Genetik Pencarian Lokal, terdapat g_LS yaitu jumlah iterasi LS dan g_GA yaitu jumlah iterasi GA. Fan (2011) bereksperimen dan mendapatkan bahwa nilai terbaik dapat diperoleh dengan g_LS = 1 dan g_GA = 30.

1.3.7 Pencarian Lokal

LS pada MDGP kami adalah prosedur berulang. Dalam setiap iterasi, prosedur memindai semua solusi tetangga dan memilih yang terbaik untuk menggantikan solusi saat ini. Solusi tetangga diperoleh dengan menukar dua elemen dari kelompok yang berbeda dari solusi saat ini. Prosedur ini juga digambarkan sebagai prosedur perbaikan oleh Baker dan Powell (2002). LS untuk MDGP dapat dilihat dalam Lampiran.