



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB II

TINJAUAN PUSTAKA

2.1 Penjadwalan Sidang Magang Prodi Informatika UMN

Proses sidang magang di Prodi Informatika UMN dilakukan dengan tujuan untuk menyampaikan laporan mahasiswa terkait praktik kerja lapangan yang dilakukan, serta menjadi bukti bahwa mahasiswa tersebut telah melaksanakan kerja magang. Komponen utama dari sidang magang yaitu, Mahasiswa sebagai pihak yang akan diuji, Dosen pembimbing sebagai pihak yang mengawasi dan membantu mahasiswa selama pembuatan laporan dan sidang magang, dan Dosen penguji sebagai pihak yang menguji dan memberikan nilai kepada mahasiswa yang telah melaksanakan magang (Widyanto & Wicaksana, 2019).

Sidang magang saat ini penjadwalannya masih dilakukan secara manual oleh Sekertaris Prodi Informatika UMN (Widyanto & Wicaksana, 2019). Jadwal yang dihasilkan diharapkan akan memenuhi sejumlah kriteria seperti setiap dosen memiliki jumlah kuota yang sama untuk menguji sidang magang, baik sebagai pembimbing, maupun sebagai penguji, dan tidak ada bentrok dengan waktu mengajar dosen. Jadwal sidang magang dapat dikatakan optimal bila memenuhi semua kriteria yang ada (Widyanto & Wicaksana, 2019).

2.2 Penjadwalan

Salah satu pokok persoalan terpenting di dalam proses perencanaan dan operasi di sistem manufaktur yaitu proses penjadwalan (Zhang dkk, 2008). Penjadwalan (scheduling) adalah proses pengalokasian sumber daya atau mesin yang ada untuk menjalankan sekumpulan pekerjaan dalam jangka waktu tertentu

(Baker, 1974). Menurut Morton dan Pentico (1993) sistem penjadwalan secara dinamis adalah proses pengambilan keputusan tentang penyesuaian aktivitas dan sumber daya untuk menyelesaikan pekerjaan atau proyek secara tepat waktu dan sekaligus memaksimalkan hasil yang bermutu tinggi dan meminimalkan biaya operasional langsung. Jenis keputusan penjadwalan dasar yang dibuat meliputi (Morton & Pentico, 1993):

1. Pengurutan pekerjaan (*sequencing*)
2. Pengaturan waktu (*timing*)
3. Perutean atau pengurutan operasi untuk suatu pekerjaan (*routing*)

Sedangkan pembahasan penjadwalan menurut Tomas Muller dan Roman Bartak (2001) adalah.

1. Aktivitas yang dilakukan

Maksudnya adalah bahwa penentuan dari permasalahan yang dibahas.

2. Sumber-sumber yang dipakai

Sumber-sumber yang dipakai berarti hal-hal yang dapat digunakan untuk menyelesaikan permasalahan penjadwalan (aktivitas) yang telah ditentukan atau bisa juga dikatakan sebagai data yang digunakan.

3. Syarat-syarat yang diperlukan

Syarat disini adalah hal-hal yang perlu diperhatikan ketika menyusun suatu penjadwalan.

4. Hubungan timbal balik

Yang dimaksud dengan hubungan timbal balik adalah bagaimana jadwal yang telah dibuat tersebut dapat sesuai dengan yang diinginkan oleh *pengguna*.

2.3 Job-Shop Scheduling Problem

Job-Shop Scheduling Problem (JSSP) merupakan permasalahan pengalokasian beberapa mesin untuk menyelesaikan beberapa pekerjaan yang terdiri dari beberapa operasi yang telah diketahui durasi penyelesaiannya. Terdapat suatu himpunan mesin $M = \{M1, M2, \dots, Mm\}$ dan himpunan *job* $J = \{J1, J2, \dots, Jn\}$. Setiap *job* J harus melalui mesin untuk menyelesaikan pekerjaannya. Setiap *job* terdiri dari satu himpunan operasi $O = \{O1, O2, \dots, Om\}$, dan urutan operasi untuk mesin sudah ditentukan sebelumnya. Setiap operasi menggunakan salah satu mesin untuk menyelesaikan satu pekerjaan untuk interval waktu yang tetap. Permasalahan yang muncul adalah bagaimana menyusun penjadwalan operasi yang akan dikerjakan mesin dengan memenuhi kendala yang ada untuk menghasilkan kualitas solusi jadwal yang optimal (Gonçalves, dkk, 2005).

Tiap jadwal dapat dievaluasi dengan berbagai macam pengukuran, tingkat keberhasilan pengukuran berhubungan dengan waktu yang dihabiskan satu pekerjaan dalam jadwal atau pengoptimalan waktu mesin atau ke duanya. Beberapa tingkat keberhasilan pengukurannya yaitu (Taillard, 1989):

1. *Flowtimes* adalah jumlah waktu yang dihabiskan tiap pekerjaan dalam jadwal, ini hampir mirip dengan pekerjaan dalam proses inventori.
2. *Makespan* adalah total waktu untuk seluruh pekerjaan dalam prosesnya.
3. *Lateness* and *earliness* adalah pengukuran nilai deviasi dari pertama dan ke dua.
4. *Machine and labor utilization* adalah pengoptimalan penggunaan mesin dan pekerja.

Google (2019) menyatakan bahwa terdapat beberapa batasan yang harus ditaati dan tidak boleh dilanggar dalam permasalahan Job Shop, yaitu:

1. Pekerjaan baru bisa dikerjakan oleh suatu sumber daya apabila pekerjaan sebelumnya telah diselesaikan
2. Sumber daya yang ada hanya dapat mengerjakan satu pekerjaan saja dalam waktu tertentu. Atau dalam kata lain, sumber daya tersebut tidak dapat multitasking.
3. Ketika suatu pekerjaan sudah dimulai, maka sumber daya tersebut berkewajiban untuk menyelesaikan pekerjaan tersebut.

2.4 Knapsack Problem

Knapsack problem atau *rucksack problem* merupakan masalah optimasi kombinasi yang bertujuan untuk mencari solusi terbaik dari berbagai solusi lainnya. Diumpamakan sebuah wadah yang bervolume V memiliki kapasitas yang bernilai bilangan bulat positif, wadah ini berpotensi untuk diisi oleh N barang yang memiliki nilai yang berbeda dengan tujuan untuk mencari nilai keuntungan yang paling tinggi (Hristakeva, 2004).

Dalam literatur Kellerer (2004) terdapat beberapa variasi *Knapsack Problem* yang telah diteliti, diantaranya adalah *Single Knapsack Problem*, *Fractional Knapsack Problem*, *Bounded Knapsack Problem*, dan *Unbounded Knapsack Problem*. *Knapsack Problem* banyak diaplikasikan dalam kasus-kasus di dunia nyata seperti pada bidang industri, transportasi, logistik, dan ilmu komputer (Hembecker, dkk., 2007).

2.5 Algoritma Genetika

Algoritma Genetika adalah algoritma pencarian yang didasarkan pada mekanisme seleksi alamiah dan Genetika alamiah. Algoritma Genetika mengambil sekumpulan individu dari sebuah populasi dan akan terus mencari individu terbaik dengan menghasilkan generasi baru secara iterative (Haldurai, 2016).

AG adalah optimasi dan teknik pencarian berdasarkan prinsip Genetika dan seleksi alam. Algoritma Genetika memungkikan populasi yang terdiri dari banyak individu untuk dapat berkembang berdasarkan aturan seleksi tertentu ke keadaan fitness maksimal (fungsi biaya minimal). Metode ini dikembangkan oleh John Holland (1975) selama tahun 1960-an dan 1970-an dan akhirnya dipopulerkan oleh salah seorang muridnya, David Goldberg (Haupt, 2004).

Beberapa definisi penting yang perlu diperhatikan untuk membangun penyelesaian permasalahan dengan algoritma Genetika adalah sebagai berikut (Stuart, 2010).

1. *Genotype* (Gen), sebuah nilai yang menyatakan satuan dasar yang membentuk suatu arti tertentu dalam satu kesatuan gen yang dinamakan kromosom. Dalam algoritma Genetika, gen ini bisa berupa nilai biner, *float*, *integer* maupun karakter, atau kombinatorial.
2. *Allele*, nilai dari gen.
3. Kromosom, gabungan gen-gen yang membentuk nilai tertentu.
4. Individu, menyatakan satu nilai keadaan yang menyatakan salah satu solusi yang mungkin dari permasalahan yang diangkat.
5. Populasi, merupakan sekumpulan individu yang akan diproses bersama dalam satu siklus proses evolusi.

6. Generasi, menyatakan satu siklus proses evolusi atau satu iterasi di dalam algoritma Genetika.

Algoritma Genetika memiliki 5 komponen dasar yang disimpulkan sebagai berikut (Gen, 2000).

1. Algoritma Genetika merepresentasikan solusi untuk setiap masalah.
2. Suatu Teknik untuk inisialisasi populasi.
3. Evaluasi rating solusi dengan nilai *fitness*.
4. Operator-operator genetik yang mengubah komposisi genetik orang tua menjadi anak.
5. Nilai-nilai parameter dari algoritma Genetika.

Sebelum memanfaatkan algoritma Genetika, hal yang harus dilakukan adalah menyandikan solusi dari masalah yang diberikan ke dalam kromosom pada algoritma Genetika dan membandingkan nilai *fitness*nya. Sebuah representasi algoritma Genetika yang efektif dan nilai *fitness* yang bermakna adalah kunci keberhasilan dalam aplikasi Genetika (Pardede, 2014).

2.5.1 Nilai Fitness

Suatu individu dievaluasi berdasarkan suatu fungsi tertentu sebagai ukuran nilai *fitness*-nya. Dalam proses evolusi, individu yang *survive* dari proses seleksi alam akan mempunyai kesempatan untuk memproduksi kembali. Dengan demikian kemampuan individu untuk dapat beradaptasi dan bertahan hidup untuk selamat sangat penting. Dalam terminologi algoritma Genetika, kemampuan suatu individu atau kromosom untuk bertahan dapat diukur berdasarkan nilai *fitness*-nya. Semakin baik nilai *fitness* suatu kromosom maka akan semakin baik peluang kromosom

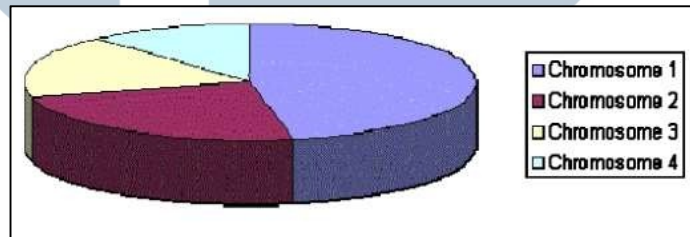
tersebut untuk bertahan hidup dan ikut dalam proses reproduksi. Nilai *fitness* suatu kromosom dapat dihitung dengan menggunakan fungsi objektif (Goldberg, 1989).

2.5.2 Seleksi Orang Tua

Terdapat beberapa jenis metode seleksi, berikut ini adalah beberapa metode seleksi yang sering digunakan.

1. Seleksi roda *roulette* (*roulette wheel selection*)

Pada metode seleksi ini, *parent* dipilih berdasarkan nilai *fitness*-nya, semakin baik nilai *fitness*-nya maka semakin besar kemungkinannya untuk terpilih. Contoh probabilitas terpilihnya suatu kromosom dalam roda *roulette* ditunjukkan pada Gambar 1 (Haldurai, 2016).



Gambar 2.1 Contoh seleksi roda *roulette*

Seleksi roda rolet adalah pendekatan seleksi yang paling sederhana. Dalam metode ini semua kromosom (individu) dalam populasi ditempatkan pada roda rolet berdasarkan proporsi nilai fitness (Goldberg, 1989). Masing-masing individu diberi segmen roda rolet. Ukuran masing-masing segmen di roda rolet sebanding dengan nilai fitness individu. Semakin besar nilainya, semakin besar segmennya. Kemudian, roda rolet virtual diputar, Individu yang sesuai dengan segmen di mana rolet roda berhenti kemudian dipilih. Proses diulang sampai jumlah individu yang

diinginkan dipilih. Individu dengan fitness lebih tinggi memiliki peluang lebih banyak untuk terpilih (Kumar, 2012).

2. Seleksi rangking (*rank selection*)

Metode seleksi roda *roulette* akan memiliki masalah ketika terdapat perbedaan *fitness* yang jauh. Sebagai contoh, jika *fitness* kromosom terbaik adalah 90% dari semua roda *roulette* dapat menyebabkan kromosom yang lain memiliki kesempatan yang sangat kecil untuk dapat terpilih (Haldurai, 2016).

Seleksi rangking terlebih dahulu merangking atau mengurutkan kromosom didalam populasi berdasarkan *fitness*-nya kemudian memberi nilai *fitness* baru berdasarkan urutannya. Kromosom dengan *fitness* terburuk akan memiliki *fitness* baru bernilai 1, terburuk kedua bernilai 2 dan seterusnya, sehingga kromosom yang memiliki *fitness* terbaik akan memiliki nilai *fitness* N, dimana N adalah jumlah kromosom di dalam populasi (Haldurai, 2016).

Setelah proses pengurutan dan pemberian nilai *fitness* baru, setiap kromosom akan memiliki kesempatan yang lebih adil untuk terpilih. Namun metode ini dapat menyebabkan konvergensi menjadi lambat, karena kromosom terbaik tidak terlalu berbeda dengan yang lainnya (Haldurai, 2016).

2.5.3 Crossover

Pada proses pindah silang dihasilkan variasi genetik. Dengan suatu skema, dua individu dipilih sebagai orang tua. Setelah didapatkan dua individu orang tua, selanjutnya ditentukan titik pindah silang secara acak. Titik pindah silang adalah titik terjadinya pertukaran gen antar dua individu orang tua. Pertukaran tersebut akan menghasilkan dua anak. Namun, operasi pindah silang tidak selamanya berhasil. Peluang keberhasilan pindah silang dinyatakan dengan probabilitas

pindah silang (Umbarkar, 2015). Berikut adalah contoh dari beberapa *crossover* yang dapat digunakan.

1. Pindah silang satu titik (*single-point crossover*)

Pindah silang ini merupakan skema pindah silang yang paling sederhana. Titik pindah silang hanya satu dengan posisi yang dibangkitkan secara acak. Contoh pindah silang satu titik ditunjukkan pada Gambar 2.2 (Umbarkar, 2015).

Parent 1:	1 0 1 0 1 0 0 1 0
Parent 2:	1 0 1 1 1 0 1 1 0
Offspring 1:	1 0 1 0 1 0 1 1 0
Offspring 2:	1 0 1 1 1 0 0 1 0

Gambar 2.2 Contoh pindah silang satu titik

2. Pindah silang banyak titik (*multi-point crossover*)

Pindah silang ini menggunakan banyak titik untuk pertukarannya. Contoh pindah silang banyak titik ditunjukkan pada Gambar 2.3 (Umbarkar, 2015).

Parent 1:	1 0 1 0 1 0 0 1 0
Parent 2:	1 1 0 0 1 0 1 1 0
Offspring 1:	1 0 0 0 1 0 0 1 0
Offspring 2:	1 1 1 0 1 0 1 1 0

Gambar 2.3 Contoh pindah silang banyak titik

3. Pindah silang pola seragam (*uniform crossover*)

Dengan operasi pindah silang pola seragam maka komposisi gen-gen tertentu pada suatu individu dapat dipertahankan. Hal ini akan memudahkan proses pencarian solusi. Contoh pindah silang pola seragam ditunjukkan pada Gambar 2.4 (Umbarkar, 2015).

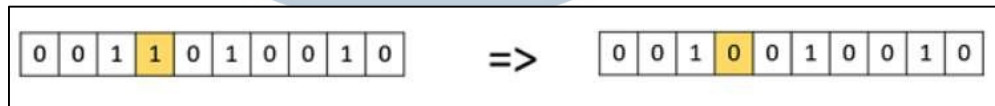
Parent 1:	1 1 1 0 1 0 0 1 0
Parent 2:	1 0 0 0 1 0 1 1 0
Offspring 1:	1 1 0 0 1 0 1 1 0
Offspring 2:	1 0 1 0 1 0 0 1 0

Gambar 2.4 Contoh pindah silang pola seragam

2.5.4 Mutation

Mutasi adalah bagian dari algoritma Genetika yang berhubungan dengan eksplorasi ruang pencarian. Telah dilakukan penelitian bahwa mutase adalah bagian penting dalam algoritma Genetika. Berikut adalah beberapa contoh dari macam-macam mutase pada algoritma Genetika.

1. *Bit Flip Mutation*, yaitu dengan cara memilih satu atau lebih bit dan menukarnya, biasanya digunakan untuk Algoritma Genetika skema pengkodean *binary*. Contoh dari *bit flip mutation* dapat dilihat pada Gambar 2.5



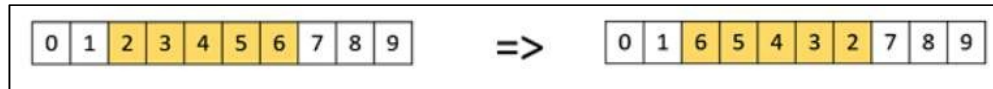
Gambar 2.5 Contoh *Bit Flip Mutation*

2. *Swap Mutation*, yakni dipilih 2 posisi pada kromosom secara acak dan menukar *value* dari 2 kromosom tersebut yang dapat dilihat pada gambar 2.6



Gambar 2.6 Contoh *Swap Mutation*

3. *Scramble Mutation*, yakni dari keseluruhan kromosom, diambil sebuah *subset* dari gen dan mengacak *value*-nya seperti yang ditunjukkan pada Gambar 2.7



Gambar 2.7 Contoh *Scramble Mutation*

2.5.5 Kriteria Penghentian

Terdapat berbagai macam kriteria penghentian yang bisa digunakan, tiga diantaranya adalah (Haldurai, 2016):

1. Memberikan batasan jumlah iterasi. Apabila batas iterasi tersebut dicapai, iterasi dihentikan dan laporkan individu bernilai *fitness* tertinggi sebagai solusi terbaik.
2. Memberikan batasan waktu proses algoritma Genetika. Kriteria ini digunakan pada sistem-sistem waktu nyata (*real time system*), dimana solusi harus ditemukan paling lama, misalkan 12 jam. Dengan demikian, algoritma Genetika bisa dihentikan, ketika proses sudah berlangsung hampir 12 jam.
3. Menghitung kegagalan penggantian anggota populasi yang terjadi secara berurutan sampai jumlah tertentu.

2.6 End User Computing Satisfaction (EUCS)

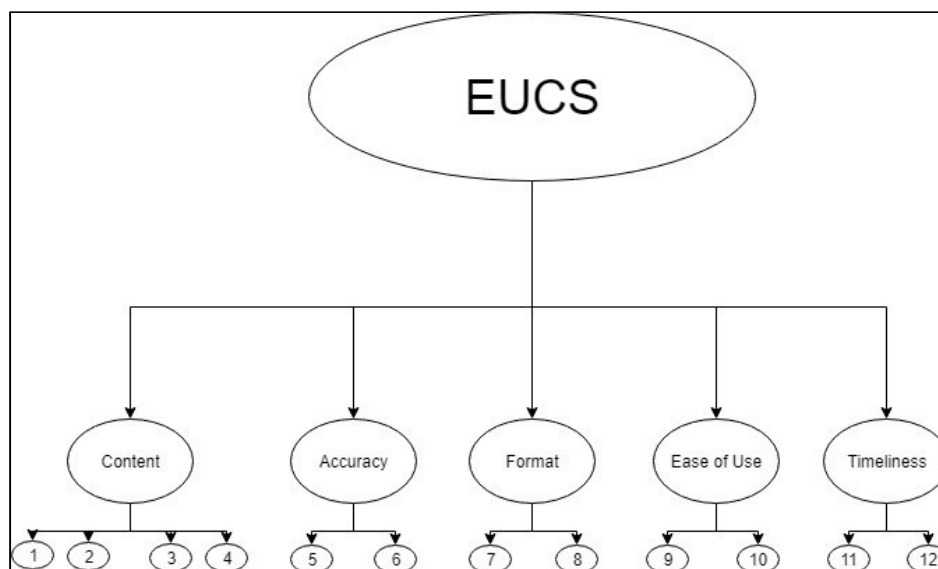
End User Computing Satisfaction (EUCS) merupakan evaluasi secara keseluruhan atas sistem informasi berdasarkan pengalaman pengguna akhir dalam menggunakan sistem (Chin dan Matthew, 2000). Sejumlah studi telah dilakukan untuk menilai keseluruhan evaluasi berdasarkan kepuasan dari pengguna akhir dan juga faktor-faktor penentunya. Tujuan dari evaluasi ini dilakukan adalah untuk memperoleh kesimpulan apakah sistem yang dibuat berguna dan diterima oleh umum. Model evaluasi ini dikembangkan oleh Doll & Torkzadeh (1988) dengan

faktor atau dimensi kepuasan seperti, *Content, Accuracy, Format, Ease of use, Timeliness*. Dalam evaluasi ini, setiap dimensi kepuasan terdapat beberapa pertanyaan. Berdasarkan penelitian dari Etezadi (1991), Setiap faktor kepuasan *Content* tidak memiliki hubungan dengan faktor kepuasan lainnya sehingga tidak ada nilai korelasi antara setiap faktor kepuasan.

Isi EUCS yang direncanakan akan digunakan dalam penelitian ini adalah sebagai berikut:

1. Apakah *web service* sudah menghasilkan jadwal sesuai yang diharapkan?
2. Apakah jadwal yang dihasilkan telah memenuhi kriteria yang ada?
3. Apakah jadwal yang dihasilkan sesuai dengan yang kebutuhan Anda?
4. Apakah *web service* ini memiliki informasi yang cukup?
5. Berapakah tingkat keakuratan *web service* ini?
6. Berapakah tingkat kepuasan Anda terhadap *web service* ini?
7. Apakah *output* yang ditampilkan dalam format yang baik?
8. Apakah informasi yang ditampilkan jelas?
9. Apakah *web service* ini ramah terhadap pengguna?
10. Berapakah tingkat kemudahan *web service* ini?
11. Apakah Anda mendapatkan informasi yang Anda butuhkan tepat pada waktunya?
12. Apakah *web service* ini memberikan informasi yang teraktual?

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 2.8 Model EUCS

Model dari EUCS yang akan digunakan dalam penelitian ini adalah pertanyaan nomor 1 sampai 4 masuk dalam dimensi *Content* dari EUCS. Pertanyaan nomor 5 dan 6 masuk dalam dimensi *Accuracy* dari EUCS. Pertanyaan nomor 7 dan 8 masuk dalam dimensi *Format* dari EUCS. Pertanyaan nomor 9 dan 10 masuk dalam dimensi *Ease of Use* dari EUCS. Pertanyaan nomor 11 dan 12 masuk dalam dimensi *Timeliness* dari EUCS. Dalam penentuan nilai dari EUCS dapat menggunakan Skala Likert. Skala Likert adalah skala yang dirancang untuk mengukur sikap secara ilmiah yang dapat diterima dan divalidasi. Cara untuk mengukurnya adalah dengan menetapkan variabel penelitian yang kemudian dibentuk dalam beberapa pertanyaan atau pernyataan. Jawaban dari pertanyaan atau pernyataan memiliki bobot nilai dari positif hingga negatif (Joshi dkk., 2015).

Sewaktu menanggapi pertanyaan dalam skala Likert, responden menentukan tingkat persetujuan mereka terhadap suatu pernyataan dengan memilih salah satu dari pilihan yang tersedia. Suatu studi empiris menemukan bahwa beberapa karakteristik hasil kuesioner dengan berbagai

jumlah pilihan tersebut ternyata sangat mirip. Skala Likert merupakan metode skala bipolar yang mengukur baik tanggapan positif ataupun negatif terhadap suatu pernyataan. Empat skala pilihan juga kadang digunakan untuk kuesioner skala Likert yang memaksa orang memilih salah satu kutub karena pilihan "netral" tak tersedia (Dawes, 2008).

