

BAB I PENDAHULUAN

1.1 Latar Belakang

Penjadwalan adalah bagian yang sangat penting dalam proses produksi sebuah perusahaan karena penjadwalan merupakan bagian dari perencanaan dan pengendalian produksi. Banyak masalah yang dapat terjadi karena penjadwalan tidak menggunakan algoritma yang tepat, melainkan menggunakan pengalaman dari masa lalu maupun menggunakan intuisi. Jika penjadwalan produksinya tidak optimal, maka akan terjadi penumpukan pekerjaan sehingga berakibat pada ruginya sebuah perusahaan. Hal ini terjadi karena dalam suatu sistem produksi yang kompleks, selalu melibatkan banyak proses, banyak mesin dan waktu pemrosesan yang bervariasi.

Salah satu proses produksi yang kompleks adalah *flowshop*. Penjadwalan produksi *flowshop* merupakan kegiatan perencanaan produksi yang terdapat pada perusahaan manufaktur dimana sistem untuk penjadwalan produksinya hanya untuk item dengan pesanan khusus atau proyek yang melibatkan produk atau jasa dengan sejumlah proses didalamnya yang harus diselesaikan dengan mesin tertentu. Permasalahan penjadwalan *flowshop* adalah untuk menentukan urutan pengerjaan *job* dari sejumlah kombinasi urutan pengerjaan yang mungkin. Artinya, dalam penjadwalan produksi terdapat n *job* dan m mesin dalam proses produksi akan memiliki kombinasi urutan pengerjaan sejumlah $n!$ (n faktorial). Oleh karena itu diperlukan suatu metode yang mampu menentukan solusi yang optimal dari banyaknya kemungkinan yang ada tanpa harus menghitung semua urutan yang mungkin terlebih dahulu. Salah satu jenis perusahaan yang termasuk dalam produksi *flowshop* adalah perusahaan *furniture*.

Perusahaan *furniture* adalah perusahaan yang memproduksi perangkat pengisi suatu interior yang berfungsi sebagai aksesoris pelengkap ruangan. *Furniture* juga biasa disebut sebagai mebel (berasal dari kata *moveble*) yang artinya bergerak [SUN-10]. Dalam produksi *furniture*, setidaknya terdapat 3 proses utama dalam menghasilkan suatu barang atau produk *furniture*, yaitu

pemotongan bahan dasar (kayu), pengerjaan (Perakitan) dan *finishing*. Ketiga proses tersebut tentunya menggunakan mesin yang berbeda dan bersifat paralel dalam merangkai suatu produk dan tentu memiliki waktu penyelesaian yang berbeda dari masing-masing mesin.

Salah satu algoritma yang digunakan dalam permasalahan penjadwalan produksi *flowshop* adalah aturan Johnson. Aturan Johnson (*Johnson's Rule*) adalah suatu aturan untuk meminimumkan *makespan* 3 mesin yang disusun paralel dan saat ini menjadi dasar teori penjadwalan [ARY-07]. Aturan Johnson mampu memberikan solusi optimal dalam pengaturan penjadwalan produksi dibandingkan dengan penjadwalan menggunakan pengalaman produksi tahun-tahun sebelumnya atas dasar pengalaman perusahaan karena kondisi atau pesanan produksi yang ada tidaklah sama, sehingga aturan Johnson mampu mengurangi menumpuknya pekerjaan dalam sebuah produksi dan mampu meminimalkan *makespan* sebagai tujuan utamanya.

Penelitian sebelumnya tentang penjadwalan produksi *flowshop* pernah dilakukan oleh Raditya (2005) di PT. Alakasa Extrusindo menggunakan algoritma CDS (Campbells Dudek & Smith) yang merupakan pengembangan dari aturan Johnson dan membandingkan hasilnya dengan metode yang digunakan oleh perusahaan yaitu metode FCFS (*First Come First Serve*). Dari hasil penelitian yang dilakukan menunjukkan bahwa unjuk kerja algoritma CDS dalam meminimalkan *makespan* memberikan hasil yang lebih baik dibandingkan dengan metode FCFS yang digunakan perusahaan.

Dalam skripsi ini akan digunakan aturan Johnson dalam simulasi penjadwalan produksi *flowshop* pada perusahaan *furniture* untuk meminimalkan *makespan*. Hasil penjadwalan dari aturan Johnson selanjutnya dilakukan evaluasi kinerja sehingga bisa memberi gambaran sejauh mana efektifitas dan kelayakan dari aturan Johnson dalam penjadwalan produksi *flowshop* di perusahaan *furniture*. Unjuk kerja dari aturan Johnson akan diukur dengan empat parameter yaitu keseluruhan waktu penyelesaian *job* (*makespan*), rata-rata waktu penyelesaian tiap *job* (*average flowtime*), utilitas dari tiap mesin dan rata-rata jumlah *job* dalam sistem (*WIP-Inventory*).

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, maka permasalahan dalam skripsi ini dapat dirumuskan sebagai berikut :

1. Bagaimana menerapkan aturan Johnson dalam simulasi penjadwalan produksi *flowshop* di perusahaan *furniture*.
2. Bagaimana evaluasi kinerja aturan Johnson pada simulasi penjadwalan produksi *flowshop* di perusahaan *furniture*.

1.3 Batasan Masalah

Batasan masalah dari skripsi ini adalah sebagai berikut :

1. Banyaknya *job* yang digunakan pada skripsi ini maksimal 20 *job*.
2. Pola kedatangan *job* bersifat statis, artinya tidak ada *job* baru yang datang saat sistem produksi sedang menyelesaikan *job-job* yang telah diberikan selama periode tertentu.

1.4 Tujuan

Tujuan yang ingin dicapai dalam skripsi ini adalah sebagai berikut :

1. Menerapkan aturan Johnson pada simulasi penjadwalan produksi *flowshop* di perusahaan *furniture*.
2. Melakukan evaluasi kinerja aturan Johnson dalam penjadwalan produksi *flowshop* di perusahaan *furniture*.

1.5 Manfaat

Manfaat yang dapat diperoleh dari skripsi ini yaitu :

1. Menghindari menumpuknya pekerjaan karena kurang optimalnya penjadwalan produksi sehingga menyebabkan ruginya sebuah perusahaan *furniture*.
2. Menghasilkan penjadwalan produksi *flowshop* yang optimal menggunakan aturan Johnson.
3. Mengetahui evaluasi kinerja aturan Johnson dalam penjadwalan produksi *flowshop* di perusahaan *furniture*.

1.6 Sistematika Penulisan

Skripsi ini disusun dengan sistematika penulisan sebagai berikut:

1. BAB I PENDAHULUAN

Bab ini berisi uraian dasar pemikiran dan latar belakang yang melandasi pembuatan skripsi ini, perumusan masalah, batasan permasalahan serta tujuan skripsi dan metodologi yang diterapkan dalam proses penyelesaian skripsi ini.

2. BAB II TINJAUAN PUSTAKA

Bab ini menjelaskan tentang teori-teori penunjang dalam perancangan dan pembuatan simulasi penjadwalan produksi *flowshop* menggunakan aturan Johnson. Pembahasan meliputi dasar-dasar perusahaan *furniture*, penjadwalan produksi, produksi *flowshop* dan aturan Johnson.

3. BAB III METODOLOGI DAN PERANCANGAN

Bab ini membahas perancangan sistem penjadwalan produksi *flowshop* di perusahaan *furniture* menggunakan aturan Johnson serta perancangan pengukuran kinerja dari aturan Johnson yang meliputi perancangan data, perancangan proses dan desain antarmuka (*interface*).

4. BAB IV IMPLEMENTASI

Bab ini berisi tentang penjelasan implementasi sistem dari simulasi penjadwalan produksi *flowshop* menggunakan aturan Johnson pada perusahaan *furniture*.

5. BAB V PENGUJIAN DAN ANALISA

Bab ini berisi tentang pembahasan, ujicoba sistem serta evaluasi kinerja dari simulasi penjadwalan produksi *flowshop* yang dihasilkan dari ujicoba menggunakan aturan Johnson.

6. BAB V PENUTUP

Bab ini berisi kesimpulan dari sistem yang telah dirancang dan diteliti serta saran pengembangan dari keseluruhan tahapan pembuatan skripsi ini maupun pembuatan perangkat lunaknya.

BAB II

TINJAUAN PUSTAKA

2.1 Konsep Dasar Penjadwalan Produksi

2.1.1 Pengertian Penjadwalan

Penjadwalan adalah aspek yang penting dalam pengendalian operasi baik dalam industri manufaktur ataupun industri jasa. Dalam usaha meningkatkan pasar dan *volume* produksi, penjadwalan yang efektif mampu meningkatkan keuntungan dalam fungsi operasi di waktu yang akan datang. Oleh sebab itu, penjadwalan yang optimal dalam suatu proses produksi haruslah sesuai dengan faktor-faktor yang mempengaruhi penjadwalan tersebut dengan tujuan meningkatkan kepuasan terhadap konsumen.

Penjadwalan adalah suatu proses pengambilan keputusan yang memainkan peranan penting dalam kebanyakan bidang manufaktur dan pelayanan industri dimana penjadwalan digunakan sebagai pengadaan bahan dan produksi dalam bidang transportasi dan distribusi serta proses informasi dan komunikasi. Penjadwalan merupakan alat ukur yang baik bagi perencanaan agregat, untuk jangka pendek dalam rentang periode beberapa hari sampai satu bulan, perusahaan harus melakukan penjadwalan produksi untuk memenuhi permintaan konsumen. Penjadwalan tersebut untuk melaksanakan rencana agregat dan jadwal induk produksi yang telah dibuat. Order aktual adalah dasar untuk penjadwalan sumber daya produksi (fasilitas, tenaga kerja dan peralatan) selanjutnya pengurutan kerja pada setiap unit produksi sehingga dicapai optimalitas utilitas dari kapasitas yang ada atau tujuan lain [TEG-02].

Penjadwalan didefinisikan sebagai *n job* pekerjaan yang akan diproses dan memiliki masing-masing memiliki waktu *set-up*, waktu proses dan *due tue* (tenggang waktu), agar dapat diselesaikan, setiap pekerjaan harus diproses pada beberapa mesin. Maka dibutuhkan suatu urutan pekerjaan-pekerjaan pada mesin-mesin yang ada agar diperoleh performansi optimal [ELS-85].

Penjadwalan didefinisikan sebagai penugasan waktu mulai dan selesainya tugas-tugas di setiap stasiun kerja atau departemen [FOG-91].

Penjadwalan sebagai proses pemilihan, pengorganisasian dan pemberian waktu dalam penggunaan sumber daya untuk melaksanakan aktifitas yang diperlukan agar menghasilkan keluaran (*output*) yang diinginkan dan memenuhi waktu serta kendala yang ada [MOR-93].

Penjadwalan didefinisikan sebagai proses pengalokasian sumber-sumber atau mesin-mesin yang ada untuk menjalankan sekumpulan tugas dalam jangka waktu tertentu [BAK-74]. Dalam definisinya dinyatakan bahwa penjadwalan berfungsi sebagai :

- a) Alat pengambil keputusan yaitu menetapkan jadwal.
- b) Penjadwalan juga berarti suatu teori yang terdiri dari kumpulan prinsip-prinsip dasar, model, teknik dan kesimpulan-kesimpulan logis dalam proses pengambilan keputusan.

Penjadwalan merupakan sesuatu yang akan memerlukan informasi yang lebih banyak daripada tipe informasi yang diperlukan untuk membuat rencana produksi. Hal ini perlu, untuk mengetahui kemampuan dari setiap mesin, setiap pengikat, setiap operator, setiap departemen dan lain-lain. Dalam kemampuan ini termasuk ukuran dan tipe dari komponen yang dapat dipakai, waktu untuk operasi, waktu yang tersedia, waktu pemeliharaan, waktu untuk menukar hubungan-hubungan dan banyak lainnya yang mungkin kecil, tetapi tidak kurang pentingnya sebagai bahan informasi untuk mendayagunakan pekerjaan ini, pembuat jadwal tersebut harus mengenal aspek-aspek pengerjaan di dalamnya [BIE-74].

Penjadwalan produksi adalah proses pengurutan pembuatan produk secara menyeluruh pada sejumlah mesin tertentu. Pengurutan (*sequencing*) berarti pembuatan produk pada satu mesin tertentu [CON-67].

Penjadwalan produksi didefinisikan sebagai rencana pengaturan urutan kerja serta pengalokasian sumber, baik waktu maupun fasilitas untuk setiap operasi yang harus diselesaikan [VOL-98].

Dari beberapa definisi diatas, maka dapat ditarik satu definisi penjadwalan produksi yaitu suatu kegiatan perancangan berupa pengalokasian sumber daya baik mesin maupun tenaga kerja untuk menjalankan *job* sesuai prosesnya dalam jangka waktu tertentu.

2.1.2 Istilah Dalam Penjadwalan Produksi

Beberapa istilah penting dalam penjadwalan produksi adalah sebagai berikut : [ARY-07]

- *Processing Time* (t_i) adalah waktu yang dibutuhkan untuk mengerjakan suatu pekerjaan. Dalam waktu proses ini sudah termasuk waktu yang dibutuhkan untuk persiapan dan pengaturan (*setup*) selama proses berlangsung.
- *Due-date* (d_i) adalah batas waktu yang diberikan untuk menyelesaikan suatu tugas atau operasi terakhir dari suatu pekerjaan. Apabila tugas tersebut tidak terselesaikan hingga batas waktu, maka terjadi keterlambatan.
- *Completion Time* (C_i) adalah waktu yang dibutuhkan untuk menyelesaikan pekerjaan mulai dari saat tersedianya pekerjaan ($t = 0$) sampai dengan tugas ke- n selesai.
- *Lateness* (L_i) adalah selisih waktu penyelesaian tugas (*Completion time*) dengan batas waktunya (*Due date*). Apabila tugas diselesaikan sebelum batas waktu (*due date*) maka terjadi nilai keterlambatan positif. Apabila tugas diselesaikan setelah batas waktu, maka terjadi keterlambatan negatif.
- *Slack time* (SL_i) adalah waktu tersisa yang muncul akibat dari waktu proses (*time processing*) lebih kecil dari batas waktunya (*Due date*).
- *Tardiness* (T_i) adalah ukuran waktu terlambat yang bernilai positif jika suatu pekerjaan dapat diselesaikan lebih cepat dari *due date*.
- *Flowtime* (F_i) adalah waktu yang dibutuhkan oleh suatu pekerjaan dari saat pekerjaan tersebut masuk ke dalam suatu tahap proses sampai pekerjaan yang bersangkutan selesai dikerjakan. Dengan kata lain, *flowtime* adalah waktu proses ditambah dengan waktu menunggu sebelum diproses.
- *Makespan* (M) adalah total waktu penyelesaian pekerjaan-pekerjaan mulai urutan pertama yang dikerjakan pada mesin atau *work center* pertama sampai urutan pekerjaan terakhir *work center* terakhir.

2.1.3 Tujuan Penjadwalan Produksi

Dalam penjadwalan produksi, terdapat banyak faktor yang mempengaruhi karena sistem produksi merupakan suatu masalah yang sangat kompleks. Dengan adanya penjadwalan tersebut, diharapkan kinerja dari masing-masing sistem bisa optimal. Berikut adalah tujuan dengan adanya penjadwalan produksi : [SUK-03]

- Meningkatkan utilitas sumber daya yang dimiliki, yaitu penggunaan sumber-sumber daya secara efektif untuk mengurangi waktu menganggur dari sumber-sumber daya.
- Respon yang cepat terhadap permintaan.
- Mengurangi persediaan barang setengah jadi (*work in process inventory*) untuk mengurangi rata-rata jumlah tugas yang menunggu dalam antrian satu mesin karena mesin tersebut sibuk.
- Mengurangi waktu keterlambatan, yaitu berusaha tidak melewati batas waktu (*due date*) yang telah ditetapkan dalam penyelesaian pekerjaan.
- Meningkatkan produktifitas mesin, yaitu penyelesaian tugas yang diterima terhadap karakteristik sumber daya yang dimiliki.

Dengan tercapainya tujuan pengaturan dan penjadwalan ini diharapkan nantinya dapat meningkatkan produktivitas perusahaan secara keseluruhan dalam menghasilkan produk dan memenuhi order yang diterima.

2.1.4 Kriteria Dalam Penjadwalan Produksi

Di dalam pengambilan keputusan tentang penjadwalan, banyak kriteria yang ditampilkan sebagai evaluasi dari penjadwalan sejumlah *job* yang diproses dalam sejumlah mesin yang merupakan fungsi dari sekumpulan waktu penyelesaian. Misalkan ada sebanyak n *job* yang akan dijadwalkan, maka kriteria penjadwalan dapat berupa: [ARY-07]

1. Minimasi *makespan*

Total waktu proses yang dibutuhkan untuk menyelesaikan suatu kumpulan *job*.

2. Minimasi waktu penyelesaian rata-rata (*mean flowtime*)

$$\bar{F} = \frac{1}{n} \sum_{j=1}^n F_j, \text{ dimana } F_j = t_i + w_i \quad (2.1)$$

Dimana n : Banyak *job*

\bar{F} : Rata-rata *flowtime*

F_j : *Flowtime* mesin ke $-j$

t_i : Waktu proses mesin ke $-i$

W_i : Waktu tunggu mesin ke $-i$

3. Minimasi *mean lateness*

$$\bar{L} = \frac{1}{n} \sum_{j=1}^n L_j \quad (2.2)$$

Dimana n : Banyak *job*

\bar{L} : Rata-rata *lateness*

L_j : *Lateness* mesin ke $-j$

4. Minimasi *mean tardiness*

$$\bar{T} = \frac{1}{n} \sum_{j=1}^n T_j \quad (2.3)$$

Dimana n : Banyak *job*

\bar{T} : Rata-rata *tardiness*

T_j : *Tardiness* mesin ke $-j$

5. Maksimasi utilitas rata-rata mesin (U_m)

$$U_m = \frac{\sum_{j=1}^n t_j}{F_{max}} \quad (2.4)$$

Dimana U_m : Utilitas mesin

t_j : Waktu pemrosesan mesin j

F_{max} : Total *Flowtime* mesin ke $-j$

6. WIP - *Inventory*

$$U_m = \frac{F_{max}}{\sum_{j=1}^n t_j} \quad (2.5)$$

Dimana U_m : Utilitas mesin

t_j : Waktu pemrosesan mesin j

F_{max} : Total *Flowtime* mesin ke $-j$

Utilitas mesin (U_m) merupakan rasio dari seluruh waktu proses yang dibebankan pada mesin dengan rentang waktu untuk menyelesaikan seluruh tugas pada mesin tersebut. Kriteria penjadwalan dapat pula dibagi berdasarkan waktu, ongkos maupun kombinasi dari keduanya. Berdasarkan waktu, maka kriteria ini dapat dibedakan atas minimisasi *makespan* dan pemenuhan *due date*. *Makespan* merupakan jangka waktu penyelesaian dalam suatu penjadwalan yang mana merupakan jumlah dari seluruh waktu proses.

Due date, seperti yang telah di utarakan sebelumnya, merupakan batas waktu penyerahan produk oleh produsen yang ditetapkan oleh konsumen. Produsen selalu berusaha untuk memenuhi *due date* tersebut, terutama untuk produk-produk kritis, misalnya produk yang akan di produksi oleh perusahaan lain dan produsen bertindak sebagai *supplier* bagi perusahaan lain, maka keterlambatan yang terjadi akan menyebabkan *idle* bagi perusahaan lain dan ini akan berdampak negatif pada hilangnya kepercayaan perusahaan pada produsen.

Kriteria lain yang berkaitan dengan ongkos terutama ditujukan ke biaya produksi. Kriteria ini tidak memperhatikan kriteria waktu yang ada sehingga dengan suatu penjadwalan produksi tertentu diharapkan mendapatkan ongkos produksi yang minimal. Kriteria gabungan (kombinasi) merupakan jenis yang menggabungkan beberapa kriteria yang ada sehingga menjadi suatu model penjadwalan dengan multi-kriteria.

2.2 Penjadwalan *Flowshop*

Penjadwalan *flowshop* dicirikan *job* yang cenderung memiliki kesamaan untuk proses operasi (*routing*) untuk semua *job*. *Flowshop* dibedakan atas *pure flowshop*, yaitu penjadwalan yang memiliki jalur produksi yang sama untuk semua tugas dan *general flowshop*, yaitu *flowshop* yang memiliki pola aliran proses yang berbeda. Ini disebabkan adanya variasi dalam pengerjaan tugas yang datang tidak harus dikerjakan pada semua mesin. Jadi mungkin saja suatu proses dilewati.

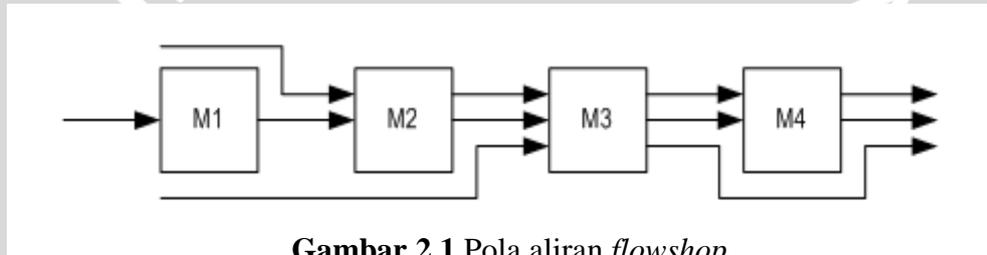
Penjadwalan *flowshop* terdiri dari : [UTT-08]

1. *Pure flowshop*

Flowshop yang memiliki jalur produksi yang sama untuk semua tugas. *Pure flowshop* dengan teknologikal terjadi dijadwalkan dengan tiap *job* harus diproses pada M_k sebelum M_i untuk semua *job*. Semua *job* memerlukan dan dikerjakan dimana operasi dikerjakan pada satu mesin.

2. *General flowshop*

Flowshop yang memiliki pola aliran berbeda. Ini disebabkan adanya variasi dalam pengerjaan tugas, sehingga tugas yang datang tidak dikerjakan pada semua mesin. Jadi mungkin saja suatu proses dilewati. Penjadwalan dilakukan dengan membagi permasalahan kedalam beberapa tipe.



Gambar 2.1 Pola aliran *flowshop*

Penjadwalan *flowshop* merupakan penjadwalan yang dilakukan berdasarkan suatu aliran produksi, dimana mesin-mesin yang ada disusun sesuai dengan proses produksinya (seri) dan setiap *job* harus melalui urutan mesin yang sama dalam waktu tertentu. Setiap operasi berikutnya berasal dari satu operasi yang mendahuluinya dan operasi kedua dari terakhir mempunyai satu operasi yang mengikutinya disebut juga *linear precedence diagram*. Penjadwalan *flowshop* dapat disebut sebagai *permutation flowshop* apabila urutan *job* yang ada pada tiap mesin tidak berubah [RON-03].

Karakteristik dasar penjadwalan *flowshop* sebagai berikut : [ARI-07]

- Terdapat n *job* yang tersedia dan siap diproses pada waktu $t = 0$.
- Waktu *set up independent* terhadap urutan pengerjaan.
- Terdapat m mesin berbeda yang tersedia secara terus-menerus.
- Operasi-operasi individual tidak dapat dipecah-pecah.

Penjadwalan *flowshop* memiliki beberapa asumsi, antara lain :

- Proses produksi dari masing-masing *job* sudah diketahui secara jelas.
- Tidak terdapat *pre-emption* (interupsi untuk mengerjakan produk lain di tengah-tengah pengerjaan suatu produk).
- Setiap *job* memerlukan *m* mesin dan setiap proses memerlukan mesin yang berbeda.
- Waktu *set-up* bersifat *independent* dan termasuk dalam waktu proses.
- Semua *job* mempunyai *ready time* yang sama.

Sistem produksi *flowshop* umumnya terdiri dari beberapa mesin (*m*) dan beberapa *job* (*n*) yang harus dikerjakan, serta memiliki waktu proses untuk setiap *job i* pada mesin *j* yaitu t_{ij} (untuk $i = 1, 2, \dots, n; j = 1, 2, \dots, m$) maka :

$$T_{ij} = t_{ij} \times d_i \quad (2.5)$$

Dengan d_i adalah jumlah permintaan untuk *job i* dan T_{ij} adalah total proses (sesuai *demand*) *job i* pada mesin *j* serta dimulainya *job i* pada mesin *j* (S_{ij}) dan saat selesainya *job i* pada mesin *j* (E_{ij}). Sedangkan $T_{[i]j}$ merupakan total waktu proses *job* posisi ke-*i* (dalam *sequence*) pada mesin *j* maka saat dimulainya *job* urutan ke-*i* pada *j* ($S_{[i]j}$) dan saat selesainya *job* urutan ke-*i* pada mesin *j* ($E_{[i]j}$) dapat dirumuskan sebagai berikut.

1. Untuk $j = 1$

$$S_{[1]1} = 0 \quad \rightarrow E_{[1]1} = T_{[1]1}$$

$$S_{[i]1} = E_{[i-1]1} \quad \rightarrow E_{[i]1} = S_{[i]1} + T_{[i]1}$$

dimana $[i] = 2, 3, 4, \dots, n$

2. Untuk $j = 2$

$$S_{[1]1} = E_{[1]1} + E_{[1]2} \quad \rightarrow E_{[1]2} = S_{[1]2} + T_{[1]2}$$

Jika $E_{[i-1]1} \leq E_{[1]1}$ maka

$$S_{[i]2} = E_{[i]1} \quad \rightarrow E_{[i]2} = S_{[i]2} + T_{[i]2}$$

3. Untuk $j = 2, 3, 4, \dots, m$ sama dengan formulasi $j = 2$.

2.3 Aturan Johnson (*Johnson's rule*)

Johnson's rule adalah suatu aturan untuk meminimumkan makespan 2 atau 3 mesin yang disusun paralel dan saat ini menjadi dasar teori penjadwalan. Permasalahan Johnson diformulasikan dengan *job* j yang diproses pada 2 mesin dengan t_{j1} adalah waktu proses pada mesin 1 dan t_{j2} adalah waktu proses mesin 2.

Algoritma *Johnson's rule* adalah sebagai berikut : [ARY-07]

1. Tentukan $\min_i \{t_{i,1}, t_{i,2}\}$
2. Jika waktu proses minimum terdapat pada mesin pertama (misal $t_{i,1}$) maka tempatkan *job* tersebut pada awal deret penjadwalan.
3. Bila waktu proses minimum didapat pada mesin kedua (misal $t_{i,2}$), maka *job* tersebut ditempatkan pada posisi akhir dari deret penjadwalan.
4. Pindahkan *job-job* tersebut dari daftarnya dan susun dalam bentuk deret penjadwalan. Jika masih ada *job* yang tersisa ulangi kembali ke langkah 1, sebaliknya bila tidak ada lagi *job* yang tersisa, berarti penjadwalan sudah selesai.

Penjadwalan *flowshop* dengan aturan Johnson hanya bisa dilakukan untuk *job* dengan dua atau tiga mesin. Aturan Johnson ini bisa digunakan untuk meminimasi waktu pemrosesan untuk mengurutkan suatu kelompok pekerjaan melalui dua fasilitas. Aturan Johnson juga meminimasi total waktu menganggur pada mesin. Aturan Johnson meliputi empat tahap, yaitu : [ARI-07]

1. Semua pekerjaan harus dicantumkan dan masing-masing waktu yang dibutuhkan oleh sebuah mesin harus ditunjukkan.
2. Pilihlah pekerjaan dengan waktu aktivitas yang paling pendek. Jika waktu yang paling pendek terdapat pada mesin pertama, maka pekerjaan dijadwalkan pertama kali. Jika waktu paling pendek terdapat pada mesin kedua, pekerjaan dijadwalkan paling akhir. Kemacetan dalam waktu aktivitas bisa dibagi-bagi secara arbitrer.
3. Sekali suatu pekerjaan telah dijadwalkan, sisihkan pekerjaan itu.
4. Terapkan tahap 2 dan tahap 3 ke pekerjaan yang tersisa, bekerja ke arah pusat urutan itu.

2.3.1 Peta Gantt (Gantt Chart)

Gantt Chart dikembangkan oleh Henry I Gantt pada masa perang dunia I. Gantt Chart merupakan representasi grafis dari pekerjaan-pekerjaan yang harus diselesaikan, digambarkan dalam bentuk susunan blok-blok batang analog dengan waktu penyelesaian pekerjaan-pekerjaan tersebut.

Keuntungan menggunakan Gantt Chart : [TUA-10]

1. Semua pekerjaan diperlihatkan secara grafis dalam satu peta yang mudah dipahami.
2. Kemajuan pekerjaan mudah diperiksa setiap waktu karena sudah tergambar dengan jelas.
3. Dalam situasi keterbatasan sumber, penggunaan Gantt Chart memungkinkan evaluasi lebih awal mengenai penggunaan sumber seperti yang telah direncanakan.

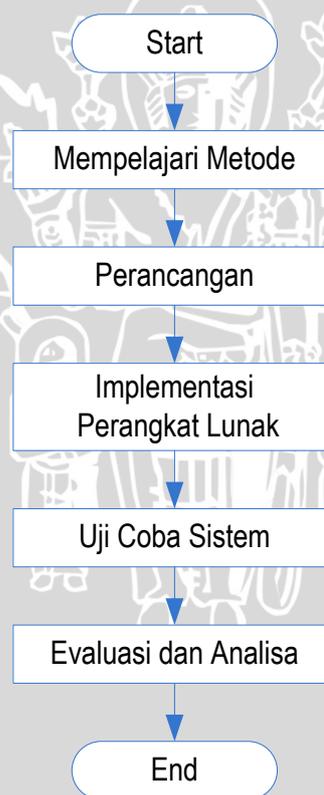


BAB III

METODOLOGI DAN PERANCANGAN

Bab ini akan membahas mengenai perancangan perangkat lunak untuk simulasi penjadwalan produksi *flowshop* menggunakan aturan Johnson di perusahaan *furniture* serta evaluasi kinerja dari metode ini. Pembahasan ini meliputi deskripsi sistem secara umum, yaitu perancangan data, perancangan proses dan perancangan antarmuka.

Tahapan penelitian yang dilakukan penulis sehingga diperoleh suatu perangkat lunak dalam simulasi penjadwalan produksi *flowshop* menggunakan aturan Johnson di perusahaan *furniture* adalah sebagai berikut.



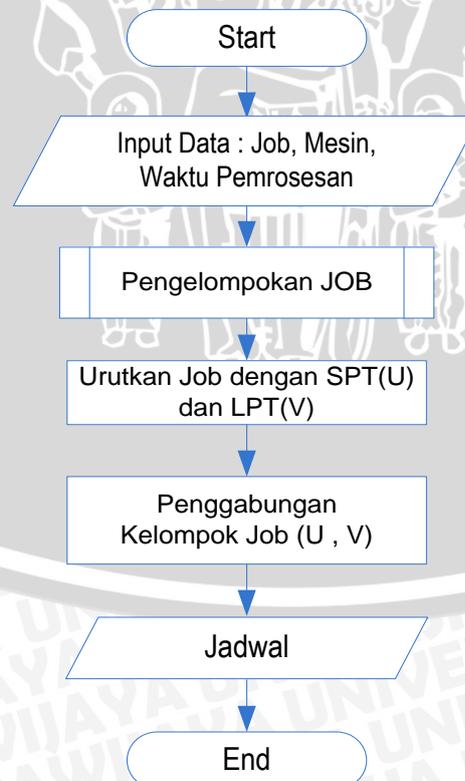
Gambar 3.1 Tahapan penelitian simulasi penjadwalan produksi *flowshop* menggunakan aturan Johnson

Penelitian dilakukan dengan tahapan sebagai berikut :

- 1) Mempelajari metode yang digunakan dari berbagai sumber atau referensi seperti yang telah dijelaskan pada Bab II (Tinjauan Pustaka).
- 2) Merancang perangkat lunak dengan aturan Johnson untuk simulasi penjadwalan produksi *flowshop* di perusahaan *furniture*.
- 3) Membuat perangkat lunak berdasarkan perancangan yang dilakukan pada langkah 2.
- 4) Uji coba penjadwalan *flowshop* menggunakan perangkat lunak yang telah dibuat.
- 5) Evaluasi dan analisa kinerja hasil uji coba.

3.1 Diagram Alir Aturan Johnson

Diagram alir digunakan untuk mengetahui alur kerja dari aturan Johnson sehingga dalam pembuatan perangkat lunak simulasi penjadwalan produksi *flowshop* di perusahaan *furniture* bisa diketahui arah kerja yang efektif dan efisien. Gambar 3.2 menunjukkan diagram alir aturan Johnson.



Gambar 3.2 Diagram alir Aturan Johnson

Tahapan proses diagram alir aturan Johnson Gambar 3.2 diatas dapat dijelaskan sebagai berikut :

Langkah 1 : Input data masukan (*job*, mesin, waktu pemrosesan).

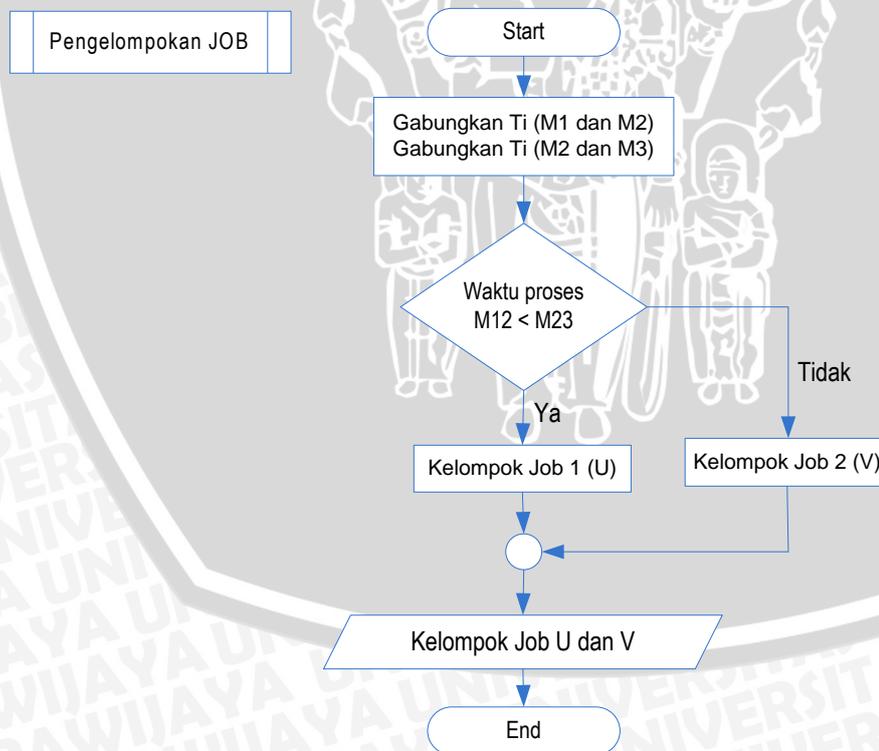
Langkah 2 : *Job* dikelompokkan berdasarkan waktu pemrosesan dari masing-masing mesin. Langkah 2 masih berupa *procedure*.

Langkah 3 : Kelompok *job* 1 (U) diurutkan dengan metode SPT (*Shortest Processing Time*) dan yang kelompok kedua (V) diurutkan dengan LPT (*Longest Processing Time*).

Langkah 4 : Penggabungan kelompok *job* dimulai dari kelompok *job* dengan urutan SPT dilanjutkan dengan LPT (U, V).

Langkah 5 : Cetak jadwal produksi

Dari *flowchart* Gambar 3.2, terdapat *procedure* pengelompokan *job* pada penjadwalan produksi *flowshop* dengan aturan Johnson. Gambar 3.3 dibawah menunjukkan *procedure* pengelompokan *job*.



Gambar 3.3 Diagram alir *procedure* pengelompokan *job*

Langkah 1 : Gabungkan waktu pemrosesan mesin 1 dan mesin 2 (M12) serta mesin 2 dan mesin 3 (M23)

Langkah 2 : Seleksi terhadap masing-masing waktu pemrosesan mesin. Jika waktu pemrosesan M12 kurang dari M23, maka *job* dikelompokkan dalam kelompok *job* 1 (U), jika tidak dikelompokkan kedalam kelompok *job* 2 (V).

Langkah 3 : Data dari *procedure* berupa kelompok *job*.

3.2 Perancangan Basis Data

3.2.1 Tabel Basis Data

Dalam pembuatan aplikasi simulasi penjadwalan produksi *flowshop* pada perusahaan *furniture* digunakan beberapa tabel. Tabel-tabel tersebut adalah sebagai berikut.

- Tabel DATA

Tabel 3.1 Tabel DATA

<i>Field</i>	<i>Type</i>	<i>Ukuran</i>	<i>Keterangan</i>
ID_DATA	<i>Integer</i>	-	<i>Primary key</i>
Jumlah_Job	<i>Integer</i>	-	
Jumlah_Mesin	<i>Integer</i>	-	

- Tabel JOB

Tabel 3.2 Tabel JOB

<i>Field</i>	<i>Type</i>	<i>Ukuran</i>	<i>Keterangan</i>
ID_JOB	<i>Integer</i>	-	<i>Primary key</i>
ID_DATA	<i>Integer</i>	-	

- Tabel PROSES

Tabel 3.3 Tabel PROSES

<i>Field</i>	<i>Type</i>	<i>Ukuran</i>	<i>Keterangan</i>
ID_JOB	<i>Integer</i>	-	
ID_MESIN	<i>Text</i>	50	
ID_DATA	<i>Integer</i>	-	
TIME_PROCESSING	<i>Integer</i>	-	

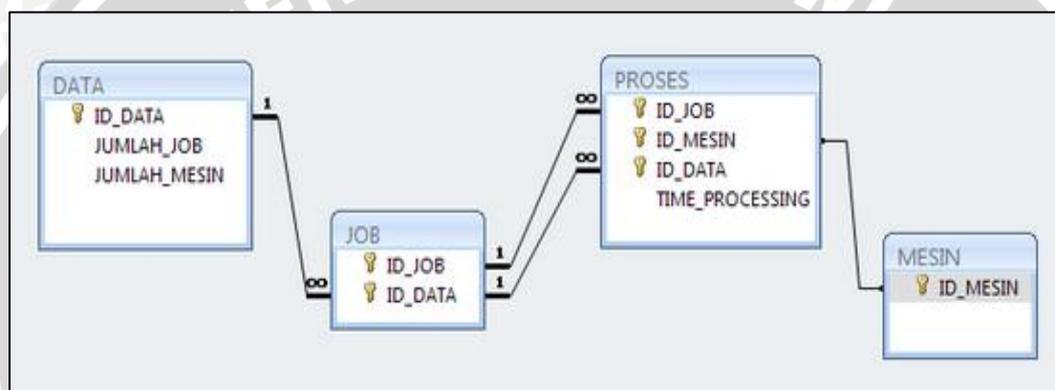
- Tabel MESIN

Tabel 3.4 Tabel MESIN

Field	Type	Ukuran	Keterangan
ID_MESIN	Text	50	Primary key

3.2.2 Entity Relational Database (ERD)

Subbab 3.2.1 menyatakan tabel-tabel yang akan digunakan dalam simulasi penjadwalan produksi *flowshop* di perusahaan *furniture*. Selanjutnya ditentukan relasi antar tabel untuk mengetahui hubungan antar tabel tersebut seperti Gambar 3.4 berikut.



Gambar 3.4 ERD simulasi penjadwalan *flowshop*

3.3 Pengolahan Data Manual

Pengolahan data manual digunakan untuk mengetahui kebenaran jalannya aturan Johnson yang digunakan dalam simulasi penjadwalan produksi *flowshop* di perusahaan *furniture*. Dalam simulasi ini, terdapat tiga mesin yang digunakan dengan tugas yang berbeda dari tiap mesin, namun setiap *job* pasti akan melewati proses di tiga mesin tersebut, yaitu :

- Mesin A : Pemotong Bahan Dasar (Kayu)
- Mesin B : Pengerjaan (Proses)
- Mesin C : *Finishing*

Dalam contoh kasus berikut, akan diberikan proses produksi furniture yang diselesaikan dengan aturan Johnson. Selanjutnya akan di cari evaluasi kinerja dari jadwal yang sudah terbentuk.

Tabel 3.5 Contoh penjadwalan produksi *flowshop* dengan 3 mesin

<i>Job</i>	<i>Time processing (T_i)</i>		
	Mesin A	Mesin B	Mesin C
1	62	65	53
2	84	93	49
3	6	50	57
4	18	2	55
5	89	27	91

Langkah-langkah penyelesaian kasus diatas adalah :

- Untuk setiap *job*, jumlahkan waktu pemrosesan dari mesin A dan mesin B yang selanjutnya disebut mesin AB, kemudian waktu pemrosesan dari mesin B dan mesin C selanjutnya disebut mesin BC.

Tabel 3.6 Penggabungan pemrosesan 3 mesin menjadi 2 mesin

<i>Job</i>	<i>Time Processing (T_i)</i>	
	Mesin AB	Mesin BC
1	127	118
2	177	142
3	56	107
4	20	57
5	116	118

- Jika waktu pemrosesan Mesin AB < Mesin BC, masukkan *job* dalam kelompok *a*. jika waktu pemrosesan Mesin AB > Mesin BC, masukkan *job* dalam kelompok *b*.

$$a = \{3, 4, 5\}$$

$$b = \{1, 2\}$$

- Urutkan *job* pada kelompok *a* berdasarkan waktu pemrosesan secara SPT (*Shortest Processing Time*) dengan melihat waktu pemrosesan terkecil. sedangkan kelompok *b* diurutkan secara LPT (*Longest Processing Time*), sehingga diperoleh :

$$a = \{4, 3, 5\}$$

$$b = \{2, 1\}$$

- Gabungkan keseluruhan kelompok jadwal *a* dan kelompok *b*.

$$\text{Jadwal} = \{4, 3, 5, 2, 1\}$$

Urutan-urutan tugas ilustrasi aturan Johnson dalam simulasi penjadwalan produksi *flowshop* di perusahaan *furniture* dapat disajikan seperti Tabel 3.7 berikut.

Tabel 3.7 Urutan tugas ilustrasi aturan Johnson

Stage	Job-job Terjadwal	Penugasan	Jadwal Parsial
1	1 2 3 4 5	4 = [20]	4 x x x x
2	1 2 3 5	3 = [56]	4 3 x x x
3	1 2 5	5 = [116]	4 3 5 x x
4	1 2	1 = [118]	4 3 5 x 1
5	1	2 = [142]	4 3 5 2 1

Dari jadwal yang sudah terbentuk, maka dilakukan evaluasi kerja terhadap aturan Johnson sehingga kriteria penjadwalan optimal bisa tercapai. Berikut parameter-parameter pembentuk penjadwalan optimal.

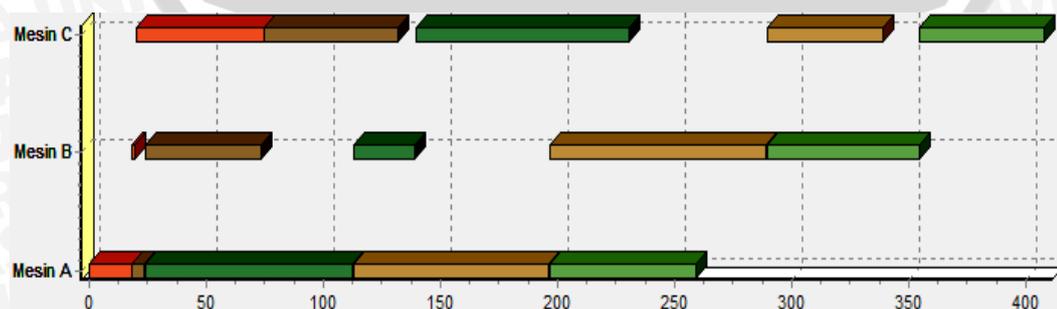
- Menghitung besarnya *makespan*

Tabel 3.8 Perhitungan *Makespan* contoh kasus Tabel 3.5

Job	t_i (A)	t_i (B)	t_i (C)	W_i (A)	W_i (B)	W_i (C)	F_i (A)	F_i (B)	F_i (C)
4	18	2	55	0	18	20	18	20	75
3	6	50	57	18	24	75	24	74	132
5	89	27	91	24	113	140	113	140	231
2	84	93	49	113	197	290	197	290	339
1	62	65	53	197	290	355	259	355	408

Keterangan t_i : Waktu penyelesaian *job*
 W_i : Waktu tunggu *job*
 F_i : *Flowtime job*

Dari Tabel 3.9 di atas maka Gantt Chart untuk memudahkan pembacaan data dapat digambarkan seperti Gambar 3.5 berikut.



Gambar 3.5 Gantt Chart contoh kasus Tabel 3.5

$$\begin{aligned}
 \text{Makespan} &= \max \{F_i, i = 1, 2, 3\} \\
 &= \max \{18, 24, 113, 197, 259, 20, 74, 140, 290, 355, 75, 132, \\
 &\quad 231, 339, 408\} \\
 &= \mathbf{408}
 \end{aligned}$$

Jadi besar *makespan* hasil penjadwalan produksi *flowshop* 4-3-5-2-1 dalam contoh kasus diatas adalah **408**.

- Minimasi *Average flowtime* (Minimasi rata-rata *flowtime*)
Semakin kecil rata-rata *flowtime* maka produktivitas perusahaan semakin baik.

Tabel 3.9 Waktu pemrosesan dan waktu tunggu

Job	t_i (A)	t_i (B)	t_i (C)	W_i (A)	W_i (B)	W_i (C)
4	18	2	55	0	18	20
3	6	50	57	18	24	75
5	89	27	91	24	113	140
2	84	93	49	113	197	290
1	62	65	53	197	290	355
Jumlah	259	237	305	352	642	880

Average flowtime mesin A

$$\bar{F}_A = \frac{1}{n} (259 + 352) = \frac{1}{5} (611) = 122.2$$

Average flowtime mesin B

$$\bar{F}_B = \frac{1}{n} (237 + 642) = \frac{1}{5} (879) = 175.8$$

Average flowtime mesin C

$$\bar{F}_C = \frac{1}{n} (305 + 880) = \frac{1}{5} (1185) = 237$$

- **Utilitas Mesin A**

$$U_A = \frac{\sum t_i}{\sum F_A} \times 100\% = \frac{259}{611} \times 100\% = 42.38\%$$

Utilitas Mesin B

$$U_B = \frac{\sum t_i}{\sum F_B} \times 100\% = \frac{237}{879} \times 100\% = 56.43\%$$

Utilitas Mesin C

$$U_C = \frac{\sum t_i}{\sum F_C} \times 100\% = \frac{305}{1185} \times 100\% = 67.59\%$$

- **WIP – Inventory Mesin A**

$$WIP_A = \frac{\sum F_A}{\sum t_i} = \frac{611}{259} = 2.36$$

WIP – Inventory Mesin B

$$WIP_B = \frac{\sum F_B}{\sum t_i} = \frac{879}{237} = 1.71$$

WIP – Inventory Mesin C

$$WIP_C = \frac{\sum F_C}{\sum t_i} = \frac{1185}{305} = 1.48$$

3.4 Rancangan Uji Coba

Rancangan uji coba dilakukan guna mengetahui kebenaran aturan Johnson dalam menyelesaikan masalah simulasi penjadwalan produksi *flowshop*, dalam hal ini adalah penjadwalan produksi pada perusahaan *furniture*. Rancangan evaluasi ini menggunakan 15 percobaan dengan jumlah *job* yang berbeda waktu pemrosesannya dan jumlah mesin yang digunakan adalah 3 (mesin pemotong, mesin proses, dan mesin *finishing*). Rancangan uji coba aturan Johnson dalam menyelesaikan masalah simulasi penjadwalan produksi *flowshop* di perusahaan *furniture* tampak seperti Tabel 3.10 berikut.

Tabel 3.10 Rancangan evaluasi kinerja aturan Johnson

Kriteria	Aturan Johnson
Jadwal	
<i>Makespan</i>	
<i>Average Flowtime job</i> Mesin	
Utilitas Mesin A	
Utilitas Mesin B	
Utilitas Mesin C	
WIP – <i>inventory</i> Mesin A, B, C	

Keterangan :

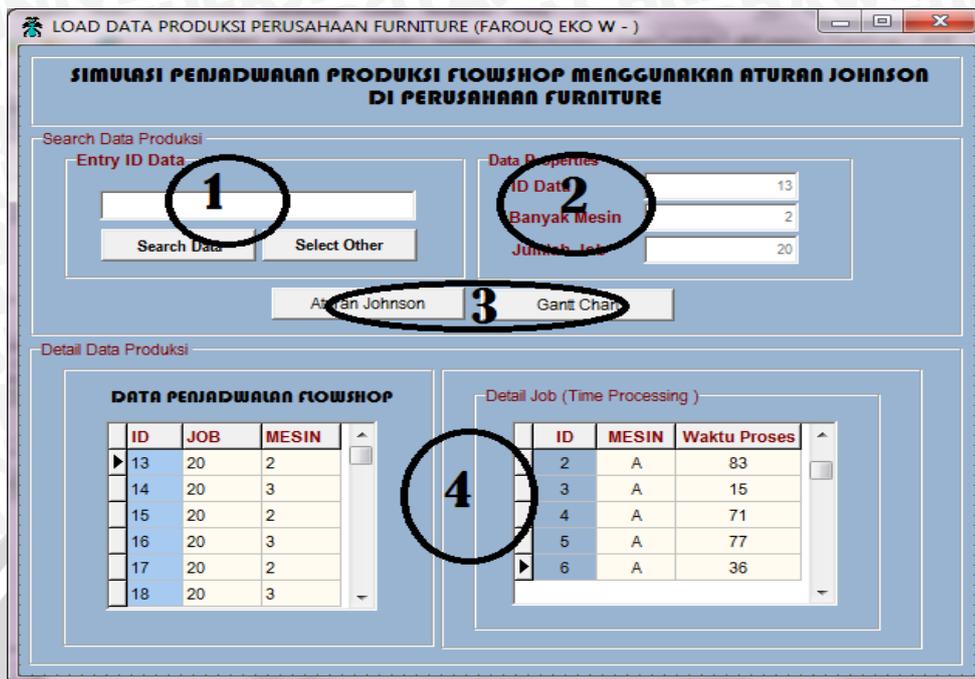
- **Makespan** : Waktu penyelesaian seluruh *job*. Semakin kecil nilai *makespan* maka produktivitas perusahaan akan semakin baik.
- **Average Flowtime** : Digunakan untuk mengetahui rata-rata waktu penyelesaian setiap *job*. Semakin kecil rata-rata *flowtime*, maka produktivitas perusahaan semakin baik.
- **Utilitas Mesin** : Digunakan untuk menilai apakah sebuah mesin sudah digunakan dengan optimal atau tidak. Semakin besar nilai utilitas mesin, maka produktivitas perusahaan semakin baik (dalam %).
- **Work-in-process (WIP) Inventory** : Evaluasi ini dilakukan dengan menentukan rata-rata jumlah *job* dalam sistem. Hal ini berkaitan erat dengan *inventory*. Semakin sedikit jumlah *job* dalam sistem, maka nilai *inventory*-nya akan semakin kecil.

3.5 Desain Interface

Rancangan tampilan untuk aplikasi simulasi penjadwalan produksi *flowshop* menggunakan aturan Johnson di perusahaan *furniture* terdiri dari 3 bagian utama, yaitu :

- Load data simulasi penjadwalan *flowshop* di perusahaan *furniture*
- Hasil penjadwalan dengan aturan Johnson
- Laporan (Gantt Chart) simulasi penjadwalan dengan aturan Johnson

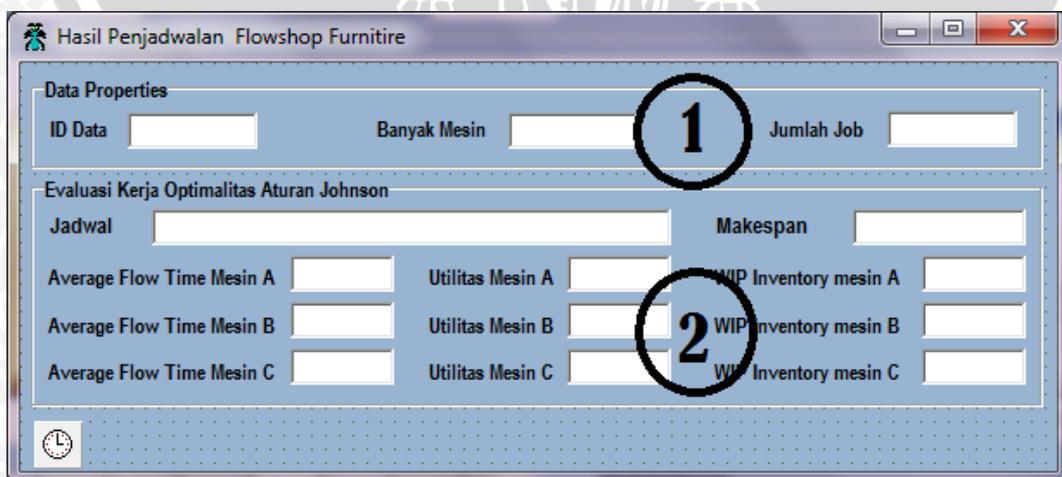
Laod data simulasi penjadwalan *flowshop* di perusahaan *furniture* terdiri dari 4 bagian utama, yaitu mencari data yang digunakan untuk mencari data simulasi penjadwalan yang terdiri dari banyak *job* dan banyak mesin, detail dari data yang dicari , tombol proses penjadwalan dan Gantt Chart dari data simulasi serta detail data simulasi berupa banyak *job*, banyak mesin serta waktu proses dari masing-masing data. Load data simulasi penjadwalan merupakan data masukan awal untuk proses penjadwalan produksi *flowshop* menggunakan aturan Johnson. Berikut adalah tampilan utama load data simulasi penjadwalan dengan aturan Johnson pada perusahaan *furniture*.



Gambar 3.6 Halaman utama load data simulasi *flowshop*

Keterangan gambar:

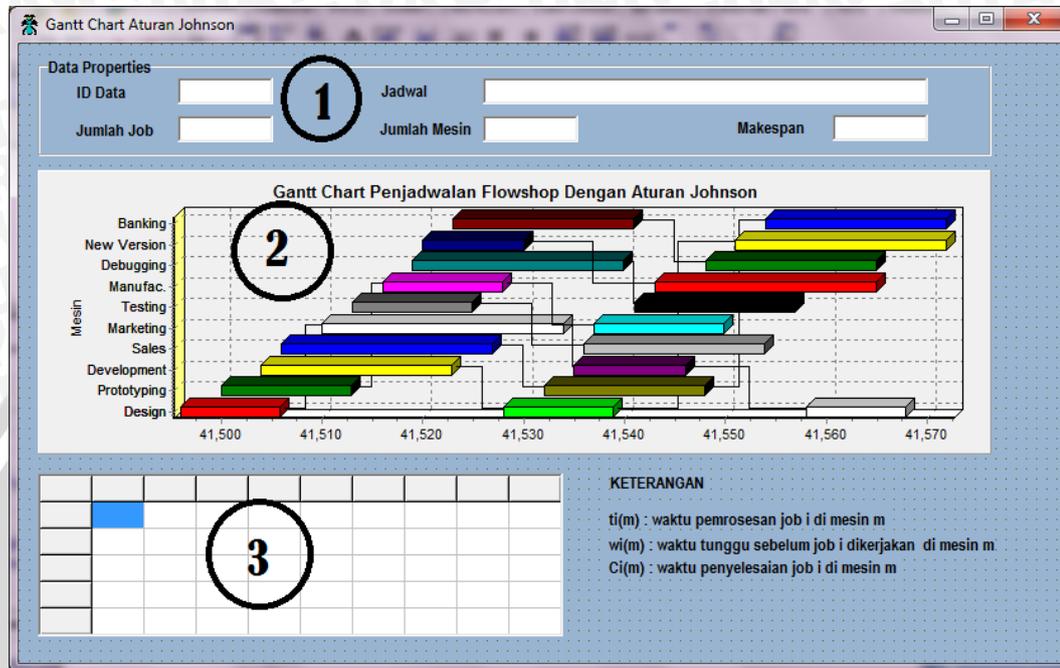
1. Mencari data simulasi dengan memasukkan id data.
2. Data jumlah mesin dan jumlah *job* hasil pencarian.
3. Tombol proses penjadwalan dengan aturan Johnson dan Gantt chart hasil penjadwalan *flowshop*.
4. Detail data dari pencarian (ID, Banyak *Job*, banyak mesin, waktu proses).



Gambar 3.6 Hasil proses penjadwalan *flowshop*

Keterangan gambar:

1. Data properties dari *job* yang akan di jadwalkan
2. Hasil proses penjadwalan dari aturan Johnson



Gambar 3.7 Gantt Chart hasil penjadwalan dengan aturan Johnson

Keterangan gambar:

1. Data properties dari *job* yang dijadwalkan
2. *Gantt Chart* dari data yang dijadwalkan dengan aturan Johnson
3. Data waktu proses, waktu tunggu dan waktu penyelesaian *job* dengan aturan Johnson

BAB IV IMPLEMENTASI

Pada bab ini dilakukan implementasi simulasi penjadwalan produksi *flowshop* menggunakan aturan Johnson di perusahaan *furniture*.

4.1 Lingkungan Implementasi

Lingkungan implementasi yang akan dijelaskan dalam Subbab 4.1 meliputi lingkungan implementasi perangkat keras dan lingkungan implementasi perangkat lunak. Spesifikasi lingkungan perangkat keras dan perangkat lunak yang digunakan dalam penjadwalan produksi *flowshop* menggunakan aturan Johnson adalah sebagai berikut.

Tabel 4.1 Lingkungan Implementasi

Lingkungan Implementasi	Spesifikasi
Perangkat Keras	Intel(r) Atom CPU N270 @1,6GHz, RAM 1GB, Hardisk 120 GB, Monitor, <i>Keyboard</i> , <i>Mouse</i>
Perangkat Lunak	Operating Sistem (OS) Windows 7 Ultimate Software Development : Borland Delphi 7 Database : Microsoft Office Acces

4.2 Implementasi Program

Berdasarkan rancangan proses yang telah dibuat, maka pada subbab ini akan dijelaskan mengenai implementasi proses-proses tersebut dalam pengembangan perangkat lunak. Secara umum, simulasi penjadwalan produksi *flowshop* menggunakan aturan Johnson pada perusahaan *furniture* dikelompokkan menjadi beberapa tahap, yaitu :

- 1) Proses penerimaan data masukan
- 2) Proses penjadwalan dengan aturan Johnson
- 3) Proses pengukuran kinerja dan pembuatan Gantt Chart.

4.2.1 Proses Penerimaan Data Masukan

Pada awal penggunaan aplikasi pengguna akan memilih salah satu data yang tersedia dalam data simulasi (*database*) pada perusahaan *furniture*. Data tersebut selanjutnya diolah dengan mengelompokkan waktu pemrosesan menjadi kelompok-kelompok *array* sesuai dengan jumlah mesin yang digunakan yaitu 3 mesin sehingga terbentuk tiga *array*. Proses tersebut ditunjukkan dalam potongan program pada *Source Code* 4.1 berikut.

```

job:=strtoint(dbjob.Text);
mesin:=strtoint(dbmesin.Text);
for i:=1 to mesin do
  KodeMesin[i]:=uppercase(chr(96+i));
for k:=1 to job do
  begin
    cb1:='select TIME_PROCESSING from PROSES where
ID_DATA='+iddata.text+' '+'and'+
'+ID_JOB='+inttostr(k)+' '+'and'+
'+ID_MESIN='+'''+KodeMesin[1]+'''';
    cb2:='select TIME_PROCESSING from PROSES where
ID_DATA='+iddata.text+' '+'and'+
'+ID_JOB='+inttostr(k)+' '+'and'+
'+ID_MESIN='+'''+KodeMesin[2]+'''';
    adodataset2.close;
    adodataset2.CommandText:=cb1;
    adodataset2.open;
    A[k]:=adodataset2.FieldValues['TIME_PROCESSING'];
    adodataset2.close;
    adodataset2.CommandText:=cb2;
    adodataset2.open;

    B[k]:=adodataset2.FieldValues['TIME_PROCESSING'];
    if (mesin=3) then
      begin

        cb3:='select TIME_PROCESSING from PROSES where
ID_DATA='+iddata.text+' '+'and'+
'+ID_JOB='+inttostr(k)+' '+'and'+
'+ID_MESIN='+'''+KodeMesin[3]+'''';
        adodataset2.close;
        adodataset2.CommandText:=cb3;
        adodataset2.open;
        C[k]:=adodataset2.FieldValues['TIME_PROCESSING'];
      end
    end;
  end;

```

Source Code 4.1 Proses penerimaan data masukan

4.2.2 Proses Penjadwalan Dengan Aturan Johnson

Proses penerimaan data masukan pada tahap awal selanjutnya akan menghasilkan jadwal produksi. Jadwal yang dihasilkan akan mengikuti aturan-aturan dari metode Johnson sehingga menghasilkan jadwal yang efektif dan efisien serta mampu mengurangi menumpuknya pekerjaan di perusahaan *furniture*. *Source code* tahapan proses penjadwalan dengan aturan Johnson tampak seperti dibawah *source code* 4.2 berikut.

```
j:=strtoint(input.DBJob.Text);
for i:=1 to j do
begin m1[i]:=input.A[i];m2[i]:=input.B[i];
m3[i]:=input.C[i];end;
if input.DBMesin.text='2' then
begin
//grouping jd U&V
index1:=1;index2:=1;a1:=1;
while a1<>j+1 do
begin
if m1[a1]<m2[a1] then
begin
Utp[index1]:=m1[a1];Uindex[index1]:=a1; end
index1:=index1+1;a1:=a1+1;
Else begin
Vtp[index2]:=m2[a1];Vindex[index2]:=a1;
index2:=index2+1;a1:=a1+1;
end
end
end
else
begin
//gabung 3 jadi 2
a2:=1;
while a2<>j+1 do
begin
m12[a2]:=m1[a2]+m2[a2];m23[a2]:=m2[a2]+m3[a2];
a2:=a2+1;
end;
//grouping jd U&V
index1:=1;index2:=1;a1:=1;
while a1<>j+1 do
begin
if m12[a1]<m23[a1] then
begin
Utp[index1]:=m12[a1];Uindex[index1]:=a1;
```

```

        index1:=index1+1;a1:=a1+1;
        end else begin
            Vtp[index2]:=m23[a1];Vindex[index2]:=a1;
            index2:=index2+1;a1:=a1+1;
        end end end;
//Sorting SPT
for p:=1 to index1-1 do
begin
    for q:=p+1 to index1-1 do
    begin
        if Utp[p]>Utp[q] then
        begin
            Utemp:=Utp[p];Utp[p]:=Utp[q];
            Utp[q]:=Utemp;ind1:=Uindex[p];
            Uindex[p]:=Uindex[q];Uindex[q]:=ind1;
        end
    end
end;
//Sorting LPT
for s:=1 to index2-1 do
begin
    for t:=s+1 to index2-1 do
    begin
        if Vtp[s]<Vtp[t] then
        begin
            Vtemp:=Vtp[s];Vtp[s]:=Vtp[t];
            Vtp[t]:=Vtemp;ind2:=Vindex[s];
            Vindex[s]:=Vindex[t]; Vindex[t]:=ind2;
        end
    end
end
//Gabung hasil sorting
x:=1;a3:=1;a4:=1;end
end;
while
x<>(strtoint(input.DBMesin.text)*strtoint(input.DBJob.
b.text))+1 do
begin
    if x<index1 then
    begin
        seq[x]:=Uindex[a3];
        x:=x+1;a3:=a3+1;
    end
    else
    begin
        seq[x]:=Vindex[a4];
        x:=x+1;
        a4:=a4+1;
    end
end;
end;

```

Source Code 4.2 Penjadwalan produksi *flowshop* dengan aturan Johnson

4.2.3 Evaluasi Kinerja Optimalitas Aturan Johnson

Hasil penjadwalan yang diperoleh selanjutnya akan dievaluasi sehingga tingkat optimalitas dari metode tersebut bisa terlihat. Evaluasi dilakukan dengan empat parameter yaitu *makespan*, rata-rata *flowime*, utilitas mesin dan *WIP-Inventory*. *Source code* evaluasi kinerja aturan Johnson dalam simulasi penjadwalan produksi *flowshop* di perusahaan *furniture* menggunakan aturan Johnson pada perusahaan *furniture* seperti *Source Code* 4.3 berikut.

```
function Tinput.Optimal(A,B,C,seq: array of integer;
flag:integer) :double;
var
e,f,maxi,numj,z,totft,totfta,totftb,totftc,tottpA,tottpB,
tottpC : integer;
ctime,mtime : array [1..20] of integer;
begin
totfta:=0;
totftb:=0;
totftc:=0;
for z:=1 to 20 do
begin
ctime[z]:=0; mtime[z]:=0;
end;
e:=0;
while e<(strtoint(DBJob.text)) do
begin
numj:=seq[e];
if DBMesin.text='2' then
begin
f:=1;
while f<>(strtoint(DBMesin.text))+1 do
begin
if f=1 then
begin
maxi:= max(ctime[numj],mtime[f]);
ctime[numj]:=maxi+A[numj-1];
mtime[f]:=ctime[numj];
totfta:=totfta+mtime[f];
f:=f+1;
end
else
begin
maxi:= max(ctime[numj],mtime[f]);
ctime[numj]:=maxi+B[numj-1];
mtime[f]:=ctime[numj];
totftb:=totftb+mtime[f];
f:=f+1;
end
end
end
end
end
```

```
else
begin
  f:=1;
  while f<>(strtoint(DBMesin.text))+1 do
  begin
    maxi:= max(ctime[numj],mtime[f]);
    if f=1 then
    begin
      ctime[numj]:=maxi+A[numj-1];
      mtime[f]:=ctime[numj];
      totfta:=totfta+mtime[f];
      f:=f+1;
    end
    else if f=2 then
    begin
      ctime[numj]:=maxi+B[numj-1];
      mtime[f]:=ctime[numj];
      totftb:=totftb+mtime[f];
      f:=f+1;
    end
    else
    begin
      ctime[numj]:=maxi+C[numj-1];
      mtime[f]:=ctime[numj];
      totftc:=totftc+mtime[f];
      f:=f+1;
    end
  end
end;
e:=e+1;
end;
//cari total flow time
totft:=totfta+totftb+totftc;
z:=1;
tottpA:=0;
tottpB:=0;
tottpC:=0;
while z<>(strtoint(DBJob.text))+1 do
begin
  tottpA:=tottpA+A[z-1];
  tottpB:=tottpB+B[z-1];
  tottpC:=tottpC+C[z-1];
  z:=z+1;
  if (flag=0) then//average flow time A
    result:=roundto(totfta/(strtoint(DBJob.text)),-2)
  else if (flag=1) then//WIP A
    result:=roundto((totfta/tottpA),-2)
  else if (flag=2) then//WIP B
    result:=roundto((totftb/(tottpA+tottpB)),-2)
  else if (flag=3) then//WIP C
  begin
    if DBMesin.text='2' then
      result:=0
    else
      result:=roundto((totftc/(tottpA+tottpB+tottpC)),-2)
    end
  end
end
```

```

else if (flag=4) then//utilitas A
  result:=roundto(((tottpA/totfta)*100),-2)
else if (flag=5) then//utilitas B
  result:=roundto((((tottpA+tottpB)/totftb)*100),-2)
else if (flag=6) then//utilitas C
begin
  if DBMesin.text='2' then
    result:=0
  else
result:=roundto((((tottpA+tottpB+tottpC)/totftc)*100),-2)
end
else if (flag=7) then //average flow time
  result:=mtime[strtoint(DBMesin.text)]
else if (flag=8) then //average flow time
  result:=roundto(totftb/(strtoint(DBJob.text)),-2)
else
begin
  if DBMesin.text='2' then
    result:=0
  else
    result:=roundto(totftc/(strtoint(DBJob.text)),-2)
end
end;

```

Source code 4.3 Evaluasi kinerja aturan Johnson dalam penjadwalan

4.2.4 Proses Pembuatan Gantt Chart

Implementasi pembuatan Gantt Chart bertujuan untuk mempermudah pembacaan hasil penjadwalan produksi *flowshop* menggunakan aturan Johnson pada perusahaan *furniture*. Pada tahapan ini terdapat jadwal dengan *makespan* paling optimal dari proses penjadwalan. Terdapat juga tabel rangkuman penjelasan *Gantt Chart* yang berisi waktu pemrosesan, waktu tunggu, dan waktu penyelesaian seluruh *job* di setiap mesin sesuai dengan jadwal yang dihasilkan.

Source code pembuatan Gantt Chart hasil penjadwalan dengan aturan Johnson seperti *source code* 4.4 berikut.

```

procedure Treport.FormCreate(Sender: TObject);
var
  i,j,x,y,z,num : integer;
begin
edit3.Text:=input.iddata.Text;
edit9.Text:=input.DBMesin.Text;
edit10.Text:=input.DBJob.Text;
edit2.Text:=floattostr(coba.ZJ);
edit1.Text:='-';
for i:=1 to strtoint(input.DBjob.text) do
  edit1.Text:=edit1.Text+inttostr(coba.QJ[i])+'-';

```

```

//menghitung flow time tiap proses
for i:=1 to strtoint(input.DBjob.text) do
  for j:=1 to strtoint(input.DBMesin.text) do
    if (i-1=0) and (j-1=0) then
      begin
        if j=1 then
          ft[i,j]:=input.A[coba.QJ[i]]
        else if j=2 then
          ft[i,j]:=input.B[coba.QJ[i]]
        else
          ft[i,j]:=input.C[coba.QJ[i]]
        end
      else if (i-1=0) then
        begin
          if j=1 then
            ft[i,j]:=input.A[coba.QJ[i]]+ft[1,j-1]
          else if j=2 then
            ft[i,j]:=input.B[coba.QJ[i]]+ft[1,j-1]
          else
            ft[i,j]:=input.C[coba.QJ[i]]+ft[1,j-1]
          end
        else if (j-1=0) then
          ft[i,j]:=input.A[coba.QJ[i]]+ft[i-1,1]
        else
          begin
            if j=1 then
              ft[i,j]:=max(ft[i-1,j],ft[i,j-1])+input.A[coba.QJ[i]]
            else
              if j=2 then
                ft[i,j]:=max(ft[i-1,j],ft[i,j-1])+input.B[coba.QJ[i]]
              else
                ft[i,j]:=max(ft[i-1,j],ft[i,j-1])+input.C[coba.QJ[i]];
            end;
          //membentuk Gantt Chart
          with series1 do
            begin Clear;
            for y:=1 to strtoint(input.DBMesin.Text)do
              begin
                if y=1 then
                  begin
                    num:=0;
                    for x:=1 to strtoint(input.DBJob.Text) do
                      begin
                        AddGanttColor( num,ft[x,1],0,'Mesin
A', $123456+(595354*x) );
                        num:=ft[x,1];
                      end;
                    end
                  else if y=2 then
                    begin
                      num:=ft[1,1];
                      for x:=1 to strtoint(input.DBJob.Text) do
                        begin

```

```

AddGanttColor( num,ft[x,y],1,'Mesin B', $123456+(595354*x) );
    num:=ft[x+1,y]-input.B[coba.QJ[x+1]];
end;end
else begin
    num:=ft[1,2];
    for x:=1 to strtoint(input.DBJob.Text) do
    begin
        AddGanttColor( num,ft[x,y],2,'Mesin
C', $123456+(595354*x) );
        num:=ft[x+1,3]-input.C[coba.QJ[x+1]];
    end;end end;
end;
//membuat tabel report
stringgrid1.Cells[1,0]='ti(A)';
stringgrid1.Cells[2,0]='ti(B)';
stringgrid1.Cells[3,0]='ti(C)';
stringgrid1.Cells[4,0]='wi(A)';
stringgrid1.Cells[5,0]='wi(B)';
stringgrid1.Cells[6,0]='wi(C)';
stringgrid1.Cells[7,0]='Ci(A)';
stringgrid1.Cells[8,0]='Ci(B)';
stringgrid1.Cells[9,0]='Ci(C)';
for i:=1 to strtoint(input.DBjob.text) do
    stringgrid1.Cells[0,i]='Job'+ ' '+inttostr(coba.QJ[i]);
for i:=1 to strtoint(input.DBMesin.text) do
    for j:=1 to strtoint(input.DBjob.text) do
        if i=1 then begin
stringgrid1.Cells[i,j]:=inttostr(input.A[coba.QJ[j]];
stringgrid1.Cells[7,j]:=inttostr(ft[j,i]);
stringgrid1.Cells[4,j]:=inttostr(strtoint(stringgrid1.Cel
ls[7,j])-strtoint(stringgrid1.Cells[i,j]));end
        else if i=2 then begin
stringgrid1.Cells[i,j]:=inttostr(input.B[coba.QJ[j]];
stringgrid1.Cells[8,j]:=inttostr(ft[j,i]);
stringgrid1.Cells[5,j]:=inttostr(strtoint(stringgrid1.Cel
ls[8,j])-strtoint(stringgrid1.Cells[i,j]));
            end
            else
            begin
                if (strtoint(input.DBMesin.text)=3) then
                begin
stringgrid1.Cells[i,j]:=inttostr(input.C[coba.QJ[j]];
stringgrid1.Cells[9,j]:=inttostr(ft[j,i]);
stringgrid1.Cells[6,j]:=inttostr(strtoint(stringgrid1.Cel
ls[9,j])-strtoint(stringgrid1.Cells[i,j]));
                End else begin
                    stringgrid1.Cells[i,j]='#';
                    stringgrid1.Cells[9,j]='#';
                    stringgrid1.Cells[6,j]='#';
                end end;
            string grid1.RowCount:=strtoint(input.DBjob.text)+1;
            stringgrid1.Height:=((strtoint(input.DBjob.text)+1)*20)+1
            0;end;end.

```

Source Code 4.4 Proses pembuatan Gantt Chart hasil penjadwalan

4.3 Implementasi Antarmuka

Berdasarkan perancangan pada subbab 3.5, maka dibuat antarmuka perangkat lunak simulasi penjadwalan produksi *flowshop* menggunakan aturan Johnson pada perusahaan *furniture*. Antarmuka dalam aplikasi ini terdiri dari 3 bagian yaitu antarmuka utama, antarmuka proses penjadwalan dan antarmuka laporan (Gantt Chart) hasil penjadwalan.

4.3.1 Antarmuka Utama

Antarmuka utama digunakan pengguna (*user*) untuk memilih *job* dalam data simulasi yang ingin dijadwalkan. Setelah memilih ID data, detail data dalam data simulasi yang terdiri dari banyak *job*, banyak mesin dan waktu pemrosesan akan tampil. Berikut contoh proses pencarian data dari halaman utama seperti Gambar 4.1 berikut.



DATA PENJADWALAN FLOWSHOP

ID	JOB	MESIN
87	5	3

Detail Job (Time Processing)

ID	MESIN	Waktu Proses
1	B	33
2	B	78
3	B	41
4	B	12
5	B	11

Gambar 4.1 Antarmuka utama proses penjadwalan *flowshop*

4.3.2 Antarmuka Proses

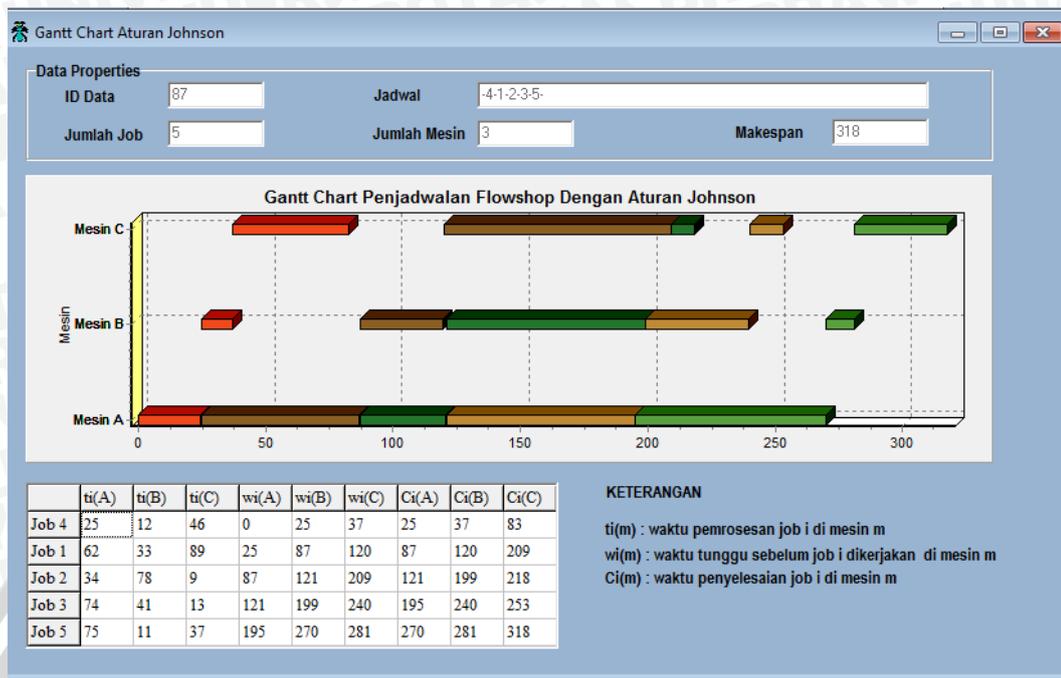
Antarmuka proses digunakan untuk melakukan penjadwalan produksi *flowshop* dengan aturan Johnson dengan data simulasi pada perusahaan *furniture*. Dalam antarmuka proses ini sekaligus ditunjukkan evaluasi kinerja berupa nilai makespan, rata-rata *flowtime* dari hasil penjadwalan, utilitas mesin dan WIP-Inventory sehingga kinerja dari aturan Johnson dalam hal penjadwalan produksi bisa diketahui tingkat efektivitasnya. Berikut tampilan antarmuka proses dan evaluasi kinerja dari aturan Johnson.

Data Properties		
ID Data	87	Banyak Mesin
		3
		Jumlah Job
		5
Evaluasi Kerja Optimalitas Aturan Johnson		
Jadwal	4.1.2.3.5	Makespan
		318
Average Flow Time Mesin A	139.6	Utilitas Mesin A
		38.68%
Average Flow Time Mesin B	175.4	Utilitas Mesin B
		50.74%
Average Flow Time Mesin C	216.2	Utilitas Mesin C
		59.11%
		WIP Inventory mesin A
		2.59
		WIP Inventory mesin B
		1.97
		WIP Inventory mesin C
		1.69

Gambar 4.2 Antarmuka proses penjadwalan dengan aturan Johnson

4.3.3 Antarmuka Laporan (Gantt Chart)

Proses ketiga adalah pembacaan data hasil penjadwalan menggunakan Gantt Chart. Tujuan utamanya adalah memberikan kemudahan dalam melakukan pembacaan data yang dihasilkan sehingga dapat dengan mudah menyimpulkan hasil penjadwalan dengan aturan Johnson dalam simulasi penjadwalan pada perusahaan *furniture*. Selain itu juga disertakan kesimpulan atau rangkuman proses keseluruhan mulai dari waktu pemrosesan, waktu tunggu, waktu penyelesaian secara keseluruhan job yang akan dijadwalkan dengan mesin yang ada pada perusahaan *furniture*. Gambar 4.3 adalah gantt chart penjadwalan produksi *flowshop* menggunakan aturan Johnson.



Gambar 4.3 Antarmuka Laporan (Gantt Chart) hasil penjadwalan



BAB V

ANALISA DAN PEMBAHASAN

5.1 Proses Pengujian

Sistem yang telah dibuat selanjutnya harus mengalami proses pengujian sehingga sistem tersebut dapat memberi bukti bahwa sistem tersebut layak untuk digunakan dalam penjadwalan produksi *flowshop* di perusahaan *furniture*. Proses pengujian juga berfungsi sebagai tolak ukur dalam melihat kinerja Aturan Johnson dalam penjadwalan produksi.

5.1.1 Data Pengujian

Data yang digunakan dalam proses pengujian adalah data simulasi *job*, *time processing* serta banyak mesin yang digunakan dengan total data sebanyak 40 data. Data tersebut memiliki variasi *job* mulai 5 sampai dengan 20 *job* dengan jumlah mesin yang digunakan 3 mesin.

5.1.2 Hasil Pengujian

Dalam ujicoba ini akan diambil 8 data simulasi *job* dengan variasi banyak *job* 5, 10, 15 dan 20 *job* yang akan diimplementasikan pada sistem yang telah dibuat sehingga dapat diambil sebuah kesimpulan aturan Johnson memiliki kinerja yang optimal dalam penjadwalan produksi di perusahaan *furniture*. Berikut ini adalah tabel pengukuran kinerja dari data *job* yang memberikan hasil terbaik dari keseluruhan data *job* dalam proses uji coba berdasarkan *id data* dalam basisdata.

- 1. ID Data : 84
- Jumlah Job : 5
- Jumlah Mesin : 3

Tabel 5.1 Data *job* dengan ID Data 84

<i>Time Processing</i>	<i>Job</i>				
	1	2	3	4	5
Mesin A	20	48	35	39	91
Mesin B	39	32	47	32	49
Mesin C	64	15	57	30	97

Tabel 5.2 Pengukuran kinerja ID Data 84

Kriteria	Algoritma Johnson
Jadwal	1,3,5,4,2
Makespan	337
Average flowtime job mesin A	127.8
Average flowtime job mesin B	170.4
Average flowtime job mesin C	250.8
Utilitas Mesin A	36.46%
Utilitas Mesin B	50.7%
Utilitas Mesin C	55.42%
WIP Inventory Mesin A	2.74
WIP Inventory Mesin B	1.97
WIP Inventory Mesin C	1.8

2. **ID Data** : 88
Jumlah Job : 5
Jumlah Mesin : 3

Tabel 5.3 Data job dengan ID Data 88

Time Processing	Job				
	1	2	3	4	5
Mesin A	40	2	12	88	18
Mesin B	14	41	46	23	81
Mesin C	49	13	37	87	16

Tabel 5.4 Pengukuran kinerja ID Data 88

Kriteria	Algoritma Johnson
Jadwal	2,1,3,4,5
Makespan	268
Average flowtime job mesin A	80
Average flowtime job mesin B	122.8
Average flowtime job mesin C	165
Utilitas Mesin A	40%
Utilitas Mesin B	59.45%
Utilitas Mesin C	68.73%
WIP Inventory Mesin A	2.5
WIP Inventory Mesin B	1.68
WIP Inventory Mesin C	1.46

3. ID Data : 64
 Jumlah Job : 10
 Jumlah Mesin : 3

Tabel 5.5 Data *job* dengan ID Data 64

Time Processing	Job									
	1	2	3	4	5	6	7	8	9	10
Mesin A	94	43	36	78	58	86	13	5	64	91
Mesin B	80	94	23	58	86	57	80	9	62	65
Mesin C	43	18	23	92	96	36	25	82	46	66

Tabel 5.6 Pengukuran kinerja ID Data 64

Kriteria	Algoritma Johnson
Jadwal	8,7,4,5,10,1,2,9,6,3
Makespan	691
Average flowtime <i>job</i> mesin A	278.5
Average flowtime <i>job</i> mesin B	361.4
Average flowtime <i>job</i> mesin C	419.4
Utilitas Mesin A	20.39%
Utilitas Mesin B	32.71%
Utilitas Mesin C	40.75%
WIP Inventory Mesin A	4.9
WIP Inventory Mesin B	3.06
WIP Inventory Mesin C	2.45

4. ID Data : 69
 Jumlah Job : 10
 Jumlah Mesin : 3

Tabel 5.7 Data *job* dengan ID Data 69

Time Processing	Job									
	1	2	3	4	5	6	7	8	9	10
Mesin A	99	49	36	85	58	24	84	4	96	71
Mesin B	85	22	2	67	41	66	7	50	4	74
Mesin C	73	27	85	51	33	8	95	3	42	92

Tabel 5.8 Pengukuran kinerja ID Data 69

Kriteria	Algoritma Johnson
Jadwal	3,7,10,1,4,5,6,8,2,9
Makespan	690
Average flowtime <i>job</i> mesin A	347.9
Average flowtime <i>job</i> mesin B	412.4
Average flowtime <i>job</i> mesin C	467.8

Utilitas Mesin A	17.42%
Utilitas Mesin B	24.83%
Utilitas Mesin C	32.77%
WIP Inventory Mesin A	5.74
WIP Inventory Mesin B	4.03
WIP Inventory Mesin C	3.05

5. **ID Data** : 46
Jumlah Job : 15
Jumlah Mesin : 3

Tabel 5.9 Data *job* dengan ID Data 46

<i>Time Processing</i>	<i>Job</i>									
	1	2	3	4	5	6	7	8	9	10
Mesin A	96	44	64	46	48	51	85	67	81	49
	11	12	13	14	15					
	65	79	77	84	85					
Mesin B	1	2	3	4	5	6	7	8	9	10
	51	56	38	66	27	90	36	81	72	15
	11	12	13	14	15					
Mesin C	55	10	80	93	64					
	1	2	3	4	5	6	7	8	9	10
	26	31	44	31	82	63	69	18	49	18
	11	12	13	14	15					
	1	37	79	21	46					

Tabel 5.10 Pengukuran kinerja ID Data 46

Kriteria	Algoritma Johnson
Jadwal	5,6,13,9,14,15,7,8,4,2,3,1,11,12,10
<i>Makespan</i>	1054
<i>Average flowtime job mesin A</i>	544.93
<i>Average flowtime job mesin B</i>	612.13
<i>Average flowtime job mesin C</i>	653.6
Utilitas Mesin A	12.49%
Utilitas Mesin B	20.2%
Utilitas Mesin C	25.19%
WIP Inventory Mesin A	8.01
WIP Inventory Mesin B	4.95
WIP Inventory Mesin C	3.97

6. ID Data : 48
 Jumlah Job : 15
 Jumlah Mesin : 3

Tabel 5.11 Data *job* dengan ID Data 48

Time Processing	Job									
	1	2	3	4	5	6	7	8	9	10
Mesin A	43	94	12	24	67	71	19	27	88	74
	11	12	13	14	15					
	94	41	71	47	69					
Mesin B	1	2	3	4	5	6	7	8	9	10
	16	73	58	55	3	4	60	48	48	86
	11	12	13	14	15					
Mesin C	15	94	31	32	32					
	1	2	3	4	5	6	7	8	9	10
	91	89	25	1	1	75	87	72	59	49
	11	12	13	14	15					
	95	15	61	34	1					

Tabel 5.12 Pengukuran kinerja ID Data 48

Kriteria	Algoritma Johnson
Jadwal	1,3,6,8,7,11,2,10,12,9,13,14,4,15,5
Makespan	845
Average flowtime <i>job</i> mesin A	418.8
Average flowtime <i>job</i> mesin B	475.47
Average flowtime <i>job</i> mesin C	566.73
Utilitas Mesin A	13.39%
Utilitas Mesin B	20.98%
Utilitas Mesin C	26.48%
WIP Inventory Mesin A	7.47
WIP Inventory Mesin B	4.77
WIP Inventory Mesin C	3.78

7. ID Data : 43
 Jumlah Job : 15
 Jumlah Mesin : 3

Tabel 5.13 Data *job* dengan ID Data 43

Time Processing	Job									
	1	2	3	4	5	6	7	8	9	10
Mesin A	66	55	31	9	98	4	87	96	73	88
	11	12	13	14	15					
	93	7	5	79	77					
Mesin B	1	2	3	4	5	6	7	8	9	10
	84	70	64	32	95	45	31	22	77	14
	11	12	13	14	15					
Mesin C	48	14	43	39	81					
	1	2	3	4	5	6	7	8	9	10
	98	32	11	58	25	60	1	95	30	22
	11	12	13	14	15					
	10	91	30	34	11					

Tabel 5.14 Pengukuran kinerja ID Data 43

Kriteria	Algoritma Johnson
Jadwal	12,4,13,6,1,5,8,9,2,15,3,14,11,10,7,
Makespan	900
Average flowtime <i>job</i> mesin A	357.2
Average flowtime <i>job</i> mesin B	433.2
Average flowtime <i>job</i> mesin C	497.2
Utilitas Mesin A	16.2%
Utilitas Mesin B	25.04%
Utilitas Mesin C	29.97%
WIP Inventory Mesin A	6.17
WIP Inventory Mesin B	3.99
WIP Inventory Mesin C	3.34

8. **ID Data** : 14
Jumlah Job : 20
Jumlah Mesin : 3

Tabel 5.15 Data *job* dengan ID Data 14

<i>Time Processing</i>	<i>Job</i>									
	1	2	3	4	5	6	7	8	9	10
Mesin A	16	89	49	15	89	45	60	23	57	64
	11	12	13	14	15	16	17	18	19	20
	7	1	63	41	63	47	26	75	77	40
Mesin B	1	2	3	4	5	6	7	8	9	10
	66	58	31	68	78	91	13	59	49	85
	11	12	13	14	15	16	17	18	19	20
Mesin C	85	9	39	41	56	40	54	77	51	31
	1	2	3	4	5	6	7	8	9	10
	58	56	20	85	53	35	53	41	69	13
	11	12	13	14	15	16	17	18	19	20
	86	72	8	49	47	87	58	18	68	28

Tabel 5.16 Data *job* dengan ID Data 14

Kriteria	Aturan Johnson
Jadwal	12,17,1,14,8,4,16,11,9,5,6,19,2,15,10,1 8,7,20,3,13,
<i>Makespan</i>	1112
<i>Average flowtime job mesin A</i>	404
<i>Average flowtime job mesin B</i>	593.75
<i>Average flowtime job mesin C</i>	666.8
Utilitas Mesin A	11.72
Utilitas Mesin B	17.08
Utilitas Mesin C	22.74
WIP <i>Inventory</i> Mesin A	8.53
WIP <i>Inventory</i> Mesin B	5.86
WIP <i>Inventory</i> Mesin C	4.4

9. ID Data : 24
 Jumlah Job : 3
 Jumlah Mesin : 20

Tabel 5.17 Data *job* dengan ID Data 24

Time Processing	Job									
	1	2	3	4	5	6	7	8	9	10
Mesin A	61	25	52	72	89	75	60	28	94	95
	11	12	13	14	15	16	17	18	19	20
	18	73	40	61	68	75	37	13	65	7
Mesin B	1	2	3	4	5	6	7	8	9	10
	21	8	5	8	58	59	85	35	84	97
	11	12	13	14	15	16	17	18	19	20
Mesin C	93	60	99	29	94	41	51	87	97	11
	1	2	3	4	5	6	7	8	9	10
	91	13	7	95	20	69	45	44	29	32
	11	12	13	14	15	16	17	18	19	20
	94	84	60	49	49	65	85	52	8	58

Tabel 5.18 Data *job* dengan ID Data 24

Kriteria	Aturan Johnson
Jadwal	20,8,4,1,17,18,11,12,13,15,7,10,6,9,16,19,14,5,2,3
Makespan	1279
Average flowtime <i>job</i> mesin A	516.1
Average flowtime <i>job</i> mesin B	696.1
Average flowtime <i>job</i> mesin C	760.6
Utilitas Mesin A	10.73%
Utilitas Mesin B	16.02%
Utilitas Mesin C	21.56%
WIP Inventory Mesin A	9.32
WIP Inventory Mesin B	6.24
WIP Inventory Mesin C	4.64

10. ID Data : 30
 Jumlah Job : 20
 Jumlah Mesin : 3

Tabel 5.19 Data *job* dengan ID Data 30

<i>Time Processing</i>	<i>Job</i>									
	1	2	3	4	5	6	7	8	9	10
Mesin A	65	94	96	47	35	34	84	3	60	34
	11	12	13	14	15	16	17	18	19	20
	70	57	8	74	13	37	87	71	57	70
Mesin B	1	2	3	4	5	6	7	8	9	10
	70	3	43	14	26	83	26	65	47	94
	11	12	13	14	15	16	17	18	19	20
Mesin C	75	30	1	71	46	87	78	76	75	55
	1	2	3	4	5	6	7	8	9	10
	94	98	63	83	19	79	54	78	29	8
	11	12	13	14	15	16	17	18	19	20
	38	97	61	10	37	16	78	96	9	91

Tabel 5.20 Data *job* dengan ID Data 30

Kriteria	Aturan Johnson
Jadwal	13,15,4,8,12,2,20,6,1,18,17,11,3,16,10,19,14,7,9,5
<i>Makespan</i>	1214
<i>Average flowtime job mesin A</i>	515.3
<i>Average flowtime job mesin B</i>	598.35
<i>Average flowtime job mesin C</i>	739.6
Utilitas Mesin A	10.82%
Utilitas Mesin B	18.22%
Utilitas Mesin C	22.43%
WIP <i>Inventory</i> Mesin A	9.24
WIP <i>Inventory</i> Mesin B	5.49
WIP <i>Inventory</i> Mesin C	4.46

5.2 Perbandingan Perhitungan Manual dan Perhitungan Komputasi

Perhitungan manual aturan Johnson bertujuan untuk mengetahui kebenaran jalannya aturan Johnson dalam penjadwalan produksi *flowshop*. Dari contoh kasus perhitungan manual Subbab 3.3 pada Tabel 3.5 memberikan hasil sebagai berikut.

Tabel 5.21 Hasil Perhitungan Manual Contoh kasus Tabel 3.5

Kriteria	Aturan Johnson
Jadwal	4, 3, 5, 2, 1
Makespan	408
Average Flowtime job Mesin A	122.2
Average Flowtime job Mesin B	175.8
Average Flowtime job Mesin C	237
Utilitas Mesin A	42.39 %
Utilitas Mesin B	56.43 %
Utilitas Mesin C	67.59 %
WIP – inventory Mesin A	2.36
WIP – inventory Mesin B	1.77
WIP – inventory Mesin C	1.48

Selanjutnya hasil perhitungan manual contoh kasus diatas dibandingkan dengan perhitungan komputasi dari perangkat lunak yang sudah dibuat. Gambar 5.1 berikut menunjukkan hasil penyelesaian dari perangkat lunak contoh kasus Subbab 3.3 Tabel 3.5.

Hasil Penjadwalan Flowshop Furniture

Data Properties

ID Data: 92 Banyak Mesin: 3 Jumlah Job: 5

Evaluasi Kerja Optimalitas Aturan Johnson

Jadwal: 4,3,5,2,1 Makespan: 408

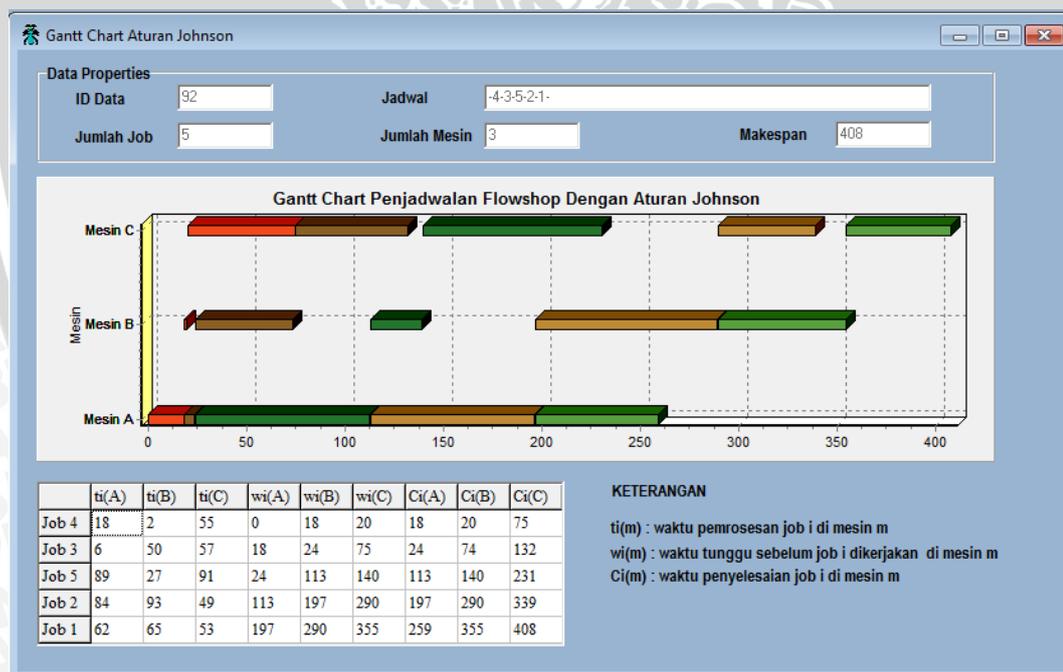
Average Flow Time Mesin A	122,2	Utilitas Mesin A	42,39%	WIP Inventory mesin A	2,36
Average Flow Time Mesin B	175,8	Utilitas Mesin B	56,43%	WIP Inventory mesin B	1,77
Average Flow Time Mesin C	237	Utilitas Mesin C	67,59%	WIP Inventory mesin C	1,48

MESIN A : MESIN PEMOTONG BAHAN (KAYU)
 MESIN B : MESIN PROSES
 MESIN C : MESIN FINISHING

Gambar 5.1 Hasil Penyelesaian Perangkat Lunak Contoh Kasus Tabel 3.5

Dari perhitungan komputasi perangkat lunak yang telah dibuat memberikan hasil yang sama dengan hasil perhitungan manual contoh kasus pada Tabel 3.5. Keseluruhan parameter yang di uji pada simulasi penjadwalan produksi *flowshop* menggunakan aturan Johnson mulai dari nilai makespan, rata-rata *flowtime* tiap mesin, Utilitas mesin dan *WIP-Inventory* memberikan hasil yang sama. Hal ini membuktikan bahwa jalannya algoritma pada metode Johnson dalam menyelesaikan permasalahan penjadwalan produksi *flowshop* dengan perangkat lunak yang telah dibuat sudah sesuai dengan tahapan-tahapan yang sudah ditentukan oleh algoritma tersebut.

Waktu tunggu (W_i) dan *flowtime* (F_i) dari masing-masing mesin yang sudah diselesaikan dalam tahapan perhitungan manual pencarian *makespan* pada contoh kasus Tabel 3.5 memberikan hasil seperti Subbab 3.3 Tabel 3.8. Sedangkan dari perangkat lunak yang telah dibuat memberikan hasil seperti Gambar 5.2 berikut.



Gambar 5.2 Hasil penyelesaian waktu tunggu (W_i) dan *flowtime* (F_i) perangkat lunak contoh kasus Tabel 3.5

Hasil perhitungan perangkat lunak dan perhitungan manual penyelesaian waktu tunggu dan *flowtime* memberikan hasil yang sama.

5.3 Analisa Hasil Uji Coba

Dari pengujian yang telah dilakukan menggunakan aturan Johnson dalam simulasi penjadwalan pada perusahaan *furniture* dengan banyak data pengujian adalah 8 (ID : 84, 88, 64, 69, 43, 46, 48, 14, 24, 30) memberikan beberapa hasil yang berbeda. Jumlah *job* yang digunakan ujicoba simulasi penjadwalan adalah 5 *job*, 10 *job*, 15 *job*, dan 20 *job*. Sedangkan jumlah mesin yang digunakan adalah 3 mesin (mesin pemotong bahan, mesin proses dan mesin *finishing*).

Tabel 5.22 Evaluasi kerja hasil penjadwalan dengan aturan Johnson

ID	Jumlah Job	M	AFT (A)	AFT (B)	AFT (C)	UM- (A) (%)	UM (B) (%)	UM (C) (%)
84	5	337	127.8	170.4	250.8	36.46	50.7	55.42
88	5	268	80	122.8	165	40	59.45	68.73
64	10	691	278.5	361.4	419.4	20.39	32.71	40.75
69	10	690	347.9	412.4	467.8	17.42	24.83	32.77
43	15	900	357.2	433.2	497.2	16.2	25.04	29.97
46	15	1054	544.93	612.1	653.6	12.49	20.2	25.19
48	15	845	418.8	475.4	566.7	13.39	20.98	26.48
14	20	1112	404	593.7	666.8	11.72	17.08	22.74
24	20	1279	516.1	696.1	760.6	10.73	16.02	21.56
30	20	1241	515.3	598.3	739.6	10.82	18.22	22.43

Dari data Tabel 5.22, untuk ID data 84 (Jumlah *job* 5) memberikan nilai *makespan* 337 dan ID data 88 memberikan nilai *makespan* 268. Rata-rata *flowtime* untuk masing-masing mesin dengan ID data 84 menghasilkan 127.8 (mesin A), 170.4 (mesin B) dan 250.8 (mesin C), sedangkan ID data 88 menghasilkan rata-rata *flowtime* 80 (mesin A), 122.8 (mesin B) dan 165 untuk mesin C. Untuk utilitas dari mesin A dengan ID data 84 adalah 36.46%, 50.7% untuk mesin B dan 55.42% untuk mesin C. Sedangkan untuk data dengan ID 88 memberikan hasil utilitas mesin A sebesar 40%, 59.45% untuk mesin B dan 68.73% untuk mesin C. Dari hasil ini, dengan jumlah *job* 5, maka unjuk kerja dari aturan Johnson cukup efektif dalam melakukan penjadwalan produksi *flowshop* pada perusahaan *furniture*. Ini terbukti dengan utilitas dari masing-masing mesin dengan rata-rata nilai lebih dari 50% dalam penjadwalan dengan 5 *job*. Sedangkan untuk rata-rata *flowtime* selalu dibawah *makespan* yang dihasilkan.

Data dengan ID 64 dan 69 adalah data simulasi dengan jumlah *job* 10 dengan banyak mesin proses 3, yaitu mesin pemotong bahan, mesin proses dan mesin *finishing*. Dengan jumlah *job* 10, *makespan* yang dihasilkan adalah 691 untuk ID data 64 dan 690 *makespan* untuk ID data 69. Sedangkan rata-rata *flowtime* yang dihasilkan untuk tiap mesin memberikan hasil yang cukup besar dengan nilai rata-rata *flowtime* tertinggi adalah 467.8 dan rata-rata *flowtime* terendah adalah 278.5. sedangkan utilitas mesin dari masing-masing bagian memiliki nilai yang sangat rendah yaitu utilitas mesin tertinggi hanya 40.75% untuk mesin *finishing*, sedangkan utilitas terendah adalah 17.42% untuk mesin pemotong bahan. Hal ini juga berlaku pada mesin dengan jumlah *job* 15.

Utilitas mesin terlihat sangat rendah pada saat uji simulasi dengan jumlah *job* sebanyak 20 *job*. Rata-rata utilitas mesin untuk mesin A, mesin B dan mesin C tidak lebih dari 20%. Sedangkan rata-rata *flowtime* yang dihasilkan sangatlah tinggi, artinya produktifitas perusahaan cukup kecil dengan penjadwalan yang ada. Jika jumlah *job* semakin banyak. Rata-rata *flowtime* yang dihasilkan tertinggi adalah 760.6 sedangkan yang terendah adalah 513.3, meskipun *makespan* yang dihasilkan cukup baik dengan waktu proses yang cukup tinggi yaitu 1112 untuk data dengan ID data 14 dan 1279 untuk data dengan ID data 24.

Hasil uji coba di atas memberikan kesimpulan bahwa semakin banyak *job* yang akan dijadwalkan dengan aturan Johnson dalam perusahaan *furniture* mampu memberikan nilai *makespan* yang cukup optimal dari jadwal yang diperoleh. Namun dari utilitas mesin yang ada, persentasenya sangatlah rendah. Hal ini akan menandakan bahwa optimalitas dari penggunaan mesin yang ada (mesin pemotong kayu, mesin proses dan mesin *finishing*) sangat kecil, sehingga mampu mengurangi produktifitas perusahaan dalam pengerjaan pesananan dalam perusahaan *furniture*.

BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

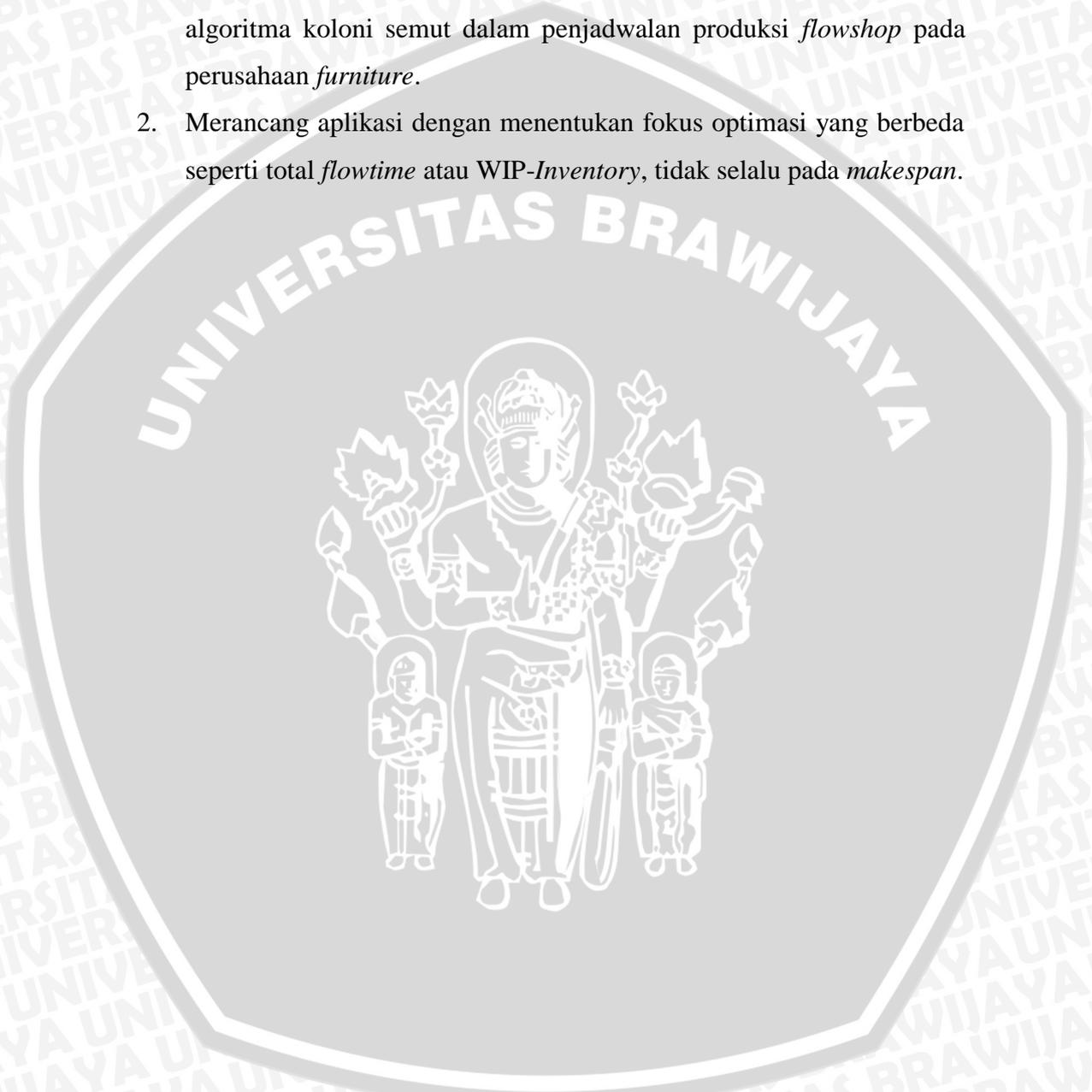
Berdasarkan hasil analisa dan pengujian yang telah dilakukan, dapat disimpulkan beberapa hal sebagai berikut.

1. Aturan Johnson dapat diterapkan dalam penjadwalan produksi *flowshop* seperti permasalahan pada perusahaan *furniture*. Parameter yang digunakan untuk menguji optimalitas dari aturan Johnson dalam penjadwalan adalah nilai *makespan*, *average flowtime*, utilitas mesin dan *WIP-Inventory*. Data simulasi yang diterapkan pada aturan Johnson berupa jumlah *job*, jumlah mesin dan waktu penyelesaian tiap mesin. Langkah penyelesaian permasalahan *flowshop* dengan aturan Johnson yaitu dengan penggabungan waktu penyelesaian tiap *job*, pengelompokan *job* berdasarkan waktu penyelesaian tiap mesin, dimana selanjutnya kelompok *job* diurutkan berdasarkan waktu pemrosesan dari masing-masing kelompok baik secara *Shortest Time Processing* (SPT) dan *Longest Time Processing* (LPT). Waktu pemrosesan terkecil dari mesin pertama selalu diletakkan didepan sehingga diperoleh urutan *job* yang optimal. Hasil penjadwalan produksi *flowshop* yang diperoleh menggunakan aturan Johnson selanjutnya dievaluasi dengan menghitung nilai *makespan*, *average flowtime*, utilitas mesin dan *WIP-Inventory*.
2. Semakin kecil nilai *makespan* hasil penjadwalan, semakin optimal jadwal yang dihasilkan, sebaliknya jika nilai *makespan* yang dihasilkan semakin besar maka jadwal yang dihasilkan kurang optimal.
3. Nilai *makespan* yang minimum tidak selalu menghasilkan utilitas mesin yang optimum.

6.2 Saran

Sebagai penelitian lebih lanjut, beberapa saran yang dapat menjadi pertimbangan untuk dilakukan adalah:

1. Menggunakan metode *heuristic* lain seperti algoritma genetika atau algoritma koloni semut dalam penjadwalan produksi *flowshop* pada perusahaan *furniture*.
2. Merancang aplikasi dengan menentukan fokus optimasi yang berbeda seperti total *flowtime* atau *WIP-Inventory*, tidak selalu pada *makespan*.



DAFTAR PUSTAKA

- [ARI-07] Arida, Luh. 2007. Penjadwalan *Flowshop* Menggunakan Algoritma Koloni Semut. Jurusan Teknik Informatika ITS. Surabaya.
- [ARY-07] Aryetta, Risky. 2007. Penjadwalan Mesin Dengan Metode CDS (Campbell, Dudek & Smith) dan Metode Palmer Pada Bagian Casting Shop di PT. Indonesia Asahan Aluminium (Inalum). Fakultas Teknik Universitas Sumatera Utara. Medan.
- [DIA-06] Dianalatif, Eddy. 2006. Penjadwalan Model *Flowshop* N Job M Mesin Dengan Tujuan Untuk Meminimasi Total Waktu Pengerjaan (*Makespan*). Fakultas Teknik dan Ilmu Komputer Unikom. Bandung.
- [HEI-00] Heizer, J., Render, B. 2000. *Principles of Operations Management*. Prentice hall.
- [NUR-08] Nuryanto. 2008. Implementasi Algoritma Johnson Untuk Penjadwalan Produksi Komponen Bodi Mobil. Jurusan Ilmu Komputer Universitas Gajahmada. Yogyakarta.
- [RAD] Raditya, Mohammad. Analisis Penjadwalan Produksi *Flowshop* Untuk Meminimasi *Makespan* di Departemen Die Shop PT. Alakasa Extrusindo. Fakultas Ilmu Teknik dan Komputer Universitas Bina Nusantara. Jakarta.
- [SUK-03] Sukmono, Wisnu W. 2003. Sistem Penjadwalan (*Schedulling*) Single Mesin pada Line 800 BPM (*Boottle Per Minute*) di PT. Coca-Cola Bottling Indonesia Jawa Tengah. Jurusan Matematika Universitas Diponegoro. Semarang.

[TUA-10] Tualaka, Jovan M. 2010. Penjadwalan *Flowshop* dengan *Ant Colony Optimization* Untuk Meminimasi *Makespan* di Departemen SPINING III PT. Grandtex Textile Indonesia. Jurusan teknik industri Unikom. Bandung.

[UTT-08] Uttari, Satria. 2008. Produk Main Frame Pada Mesin Punch Exentrik di PT. BETON PERKASA WIJAKSANA. Fakultas Teknik Universitas Pembangunan Nasional. Jakarta.

