

ALGORITMA BRANCH AND BOUND  
UNTUK PENJADWALAN PELAYANAN DI RESTORAN

Disusun Untuk Memenuhi Tugas Akhir  
Mata Kuliah Strategi Algoritma  
Semester Genap 2019



**STIKI**

Disusun Oleh :

Panji Iman B.	– 171 111 023
Risky Dewantara	– 171 111 086
Fery Eka M.	– 171 111 117
Luqman Al-Hakim	– 171 111 118

PROGRAM STUDI TEKNIK INFORMATIKA  
SEKOLAH TINGGI INFORMATIKA DAN KOMPUTER INDONESIA  
2019

## **Kata Pengantar**

Penulis mengucapkan puji syukur kehadiran Allah SWT yang telah melimpahkan rahmat kepada kita semua, dan berkat itu semua penulis dapat menyelesaikan proposal yang berjudul “Algoritma Branch and Bound untuk Penjadwalan Pelayanan di Restoran”.

Proposal ini disusun untuk memenuhi tugas mata kuliah Strategi Algoritma. Harapan penulis adalah proposal ini dapat menambah pengetahuan dan wawasan. Penulis selalu mengharapkan adanya saran dan masukan dari berbagai pihak.

Pada kesempatan ini, penulis mengucapkan terima kasih kepada:

1. Rakhmad Maulidi, M.Kom. selaku Dosen Strategi Algoritma yang telah memberi bimbingan dan materi proposal.
2. Orang tua yang telah memberi dorongan serta doa.
3. Pihak yang telah bekerja sama membantu proses pembuatan proposal.

Harapan penulis semoga proposal ini dapat memberikan manfaat, khususnya bagi penulis dan umumnya bagi seluruh pihak yang membaca.

Malang, 9 Juli 2019

Tim Penulis

## DAFTAR ISI

KATA PENGANTAR .....	
DAFTAR ISI .....	
BAB I PENDAHULUAN	
1.1. Latar Belakang .....	1
1.2. Rumusan Masalah .....	1
1.3. Tujuan Penelitian .....	1
1.4. Batasan Masalah .....	2
1.5. Manfaat Penelitian .....	2
1.6. Metodologi Penelitian.....	2
BAB II LANDASAN TEORI	
2.1. Algoritma Branch and Bound .....	4
BAB III PEMBAHASAN MASALAH	
3.1. Prosedur Perhitungan .....	7
3.2. Kasus .....	7
BAB IV PENUTUP	
4.1. Kesimpulan .....	13
4.2. Saran .....	13
DAFTAR PUSTAKA .....	26

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Penjadwalan merupakan rencana pengaturan urutan kerja. Penjadwalan pada kasus ini dilakukan dengan pengurutan kerja pada tiap-tiap proses sehingga tercapai hasil yang optimal. Pada penjadwalan ini digunakan untuk pelayanan di sebuah restoran.

Menurut Haming dan Nurnajamuddin penjadwalan adalah pengalokasian sumber daya dari waktu ke waktu untuk menunjang pelaksanaan dan penyelesaian suatu aktivitas pengerjaan spesifik. Penentuan sumber daya (sumber daya manusia, sumber daya kapasitas dan peralatan, dan waktu) ditunjukkan untuk mewujudkan penggunaan sumber daya secara efektif dan efisien, dan menghasilkan output yang tepat jumlah, tepat waktu, dan tepat kualitas.

Masalah utama yang dihadapi oleh restoran adalah menentukan pelayanan yang minimal. Tujuan penulisan laporan ini adalah untuk menguraikan salah satu metode dalam meminimalkan waktu pelayanan memasak di sebuah restoran. Restoran menggunakan metode FIFO dalam proses pelayanan pesanan, yang berarti pekerjaan yang dikerjakan terlebih dahulu adalah pesanan yang pertama datang. Artinya, penyelesaian pekerjaan akan membutuhkan waktu yang lama, karena pelayanan pesanan tidak dilakukan sesuai waktu kebutuhannya. Atas dasar permasalahan tersebut restoran memerlukan sebuah metode penjadwalan yang baik untuk pelaksanaan pelayanannya. Berdasarkan permasalahan di atas dan juga pada penelitian-penelitian sebelumnya perlu dilakukan suatu penjadwalan pelayanan agar bisa meminimalkan waktu penyelesaian pesanan, sehingga restoran dapat melayani pesanan sesuai dengan waktu yang ditentukan. Pada penelitian ini, metode penjadwalan produksi yang digunakan adalah branch and bound.

### **1.2 Rumusan Masalah**

Menentukan urutan pelayanan pesanana makanan pada restoran dengan menggunakan algoritma Branch and Bound untuk penjadwalan

### **1.3 Tujuan Penelitian**

Membangun sebuah algoritma branch and bound yang dapat digunakan untuk penjadwalan pada restoran

#### **1.4 Batasan Masalah**

Batasan masalah dalam laporan ini adalah sebagai berikut.

- a. Mengurutkan pesanan berdasarkan prioritas
- b. Prioritas berupa keanggotaan pemesan, lama masakan yang dipesan dan waktu pesanan dibuat
- c. Program akan dijalankan jika pesanan lebih dari ambang batas 100 pesanan
- d. Pesanan dibawah ambang batas hanya akan memperhatikan beberapa batasan

#### **1.5 Manfaat Penelitian**

Adapun manfaat yang diharapkan dalam perhitungan ini

##### **1.5.1 Bagi Pengembang**

Pengembang dapat mengetahui estimasi waktu yang dibutuhkan untuk makanan yang dipesan di restoran tersebut saat pesanan makanan sedang ramai di sebuah restoran

##### **1.5.2 Bagi Restoran**

Manfaat untuk restoran adalah dapat menjadwalkan dengan tepat dan lebih cepat setiap pesanan yang telah dibuat oleh pemesan sehingga tidak terjadi kesalahan atau keterlambatan pelayanan di restoran tersebut.

#### **1.6 Metodologi Penelitian**

Metode yang digunakan dalam perhitungan ini

##### **1.6.1 Tempat dan Waktu Penelitian**

Lokasi yang digunakan dalam penelitian perhitungan ini berada di Kota Malang. Waktu penelitian dilakukan selama 1 hari saja, yaitu pada tanggal 7 Juli 2019.

##### **1.6.2 Alat dan Bahan Penelitian**

Alat yang digunakan dalam penelitian ini adalah

Hardware : Laptop / Komputer

### **1.6.3 Pengumpulan Data dan Informasi**

Survei singkat, Melakukan survei kepuasan pada beberapa responden yang ada untuk mengetahui pengalaman mereka saat melakukan pesanan di restoran.

### **1.6.4 Prosedur Penelitian**

1. Pengumpulan Data dan Informasi
2. Analisis
3. Pemilihan Metode Perhitungan
4. Proses Perhitungan
5. Evaluasi
6. Pembuatan Laporan

## **BAB II**

### **LANDASAN TEORI**

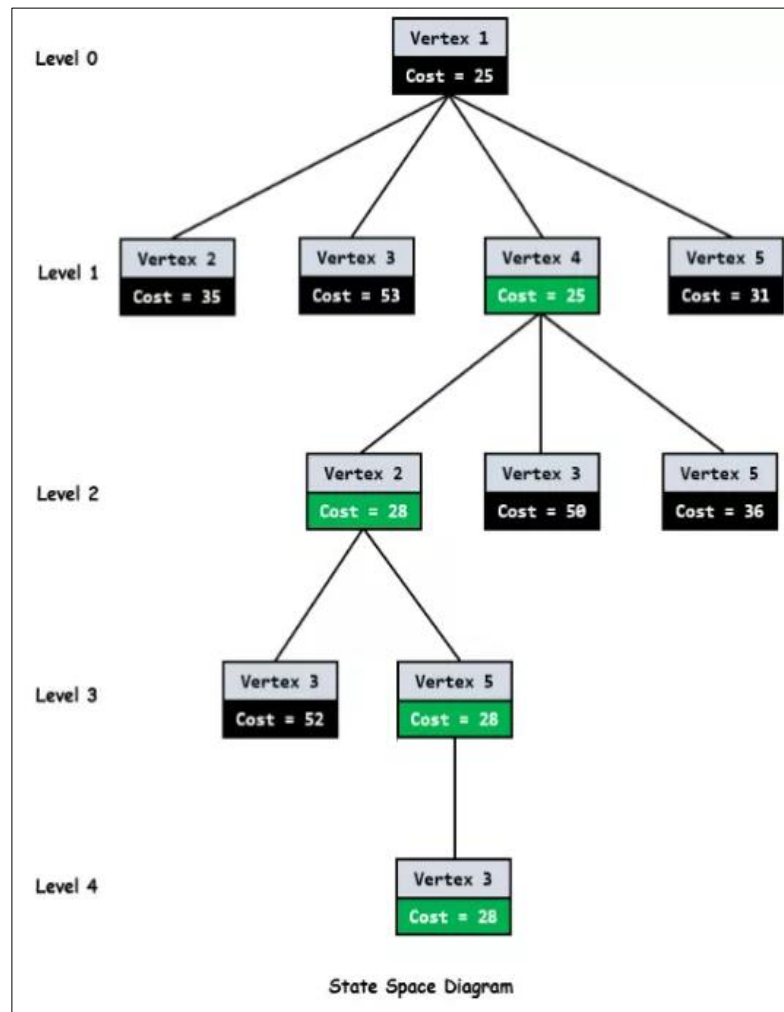
#### **2.1 Algoritma Branch and Bound**

Algoritma Branch and Bound, atau yang biasa disingkat dengan B&B merupakan sebuah metode pencarian penyelesaian tercepat dan efisien dari sebuah masalah yang diimplementasikan ke dalam sebuah pohon atau *tree*. Pada algoritma ini, sebuah masalah digambarkan dalam bentuk *tree* dimana masing-masing *branch* menggambarkan urutan yang saling berhubungan.

Dalam masalah yang dibahas di sini, n-buah pesanan akan dimasukkan ke dalam sistem dan oleh sistem akan diurutkan dengan memperhatikan beberapa batasan selain urutan pemesanan. Oleh karena itu, algoritma Branch and Bound dikembangkan untuk menentukan pelayanan yang tercepat dan optimal. Branch and Bound adalah suatu prosedur yang paling umum untuk mencari solusi optimal pada masalah optimasi seperti masalah penjadwalan.

Di dalam algoritma Branch and Bound, terdapat tiga buah bagian utama, yaitu batas bawah (lower bound), strategi pencarian dan pencabangan (branching). Di dalam prosedur ini, suatu masalah dipecah menjadi beberapa submasalah yang merepresentasikan pembagian kerja secara sebagian. Simpul-simpul bercabang lebih jauh sampai diperoleh solusi yang lengkap. Prosedur ini terus diulang sampai pencarian pada pohon berakhir dan solusi optimal ditemukan.

Algoritma ini bisa direpresentasikan dengan sebuah *tree* atau *binary tree* yang setiap nodenya mewakili setiap nilai atau variabel yang akan dioptimasi pengerjaannya. Pemilihan node di setiap cabang berdasarkan pada *bound* atau batasan yang telah ditentukan. Setiap node dari tree tersebut akan dikonstruksi oleh algoritma branch and bound, akan tetapi hanya ada beberapa cabang yang akan dilewati dan dibaca oleh algoritma ini. Hal ini terjadi karena letak optimasi pada algoritma ini adalah ketika sebuah solusi telah ditemukan dan sudah dilewati maka selanjutnya jika solusi tersebut ditemukan kembali, solusi tersebut akan diabaikan dan akan berlanjut pada solusi selanjutnya.



Model diagram diatas adalah sebuah permasalahan yang diselesaikan dengan B&B yang mengguakan bound atau parameter batasan dimana setiap tahapan yang dipilih harus memiliki nilai atau *weight* lebih kecil atau sama dengan 28. Pada diagram diatas setiap vertex yang tidak terpilih sebagai tahapan solusi nilainya akan berubah secara acak, sehingga bisa diselesaikan dengan persamaan matematika seperti di bawah ini,

$$V_{xa} = \{V1, V2, V3, V4, V5\}$$

Himpunan diatas dapat diselesaikan dengan persmaan pada himpunan di bawah dengan himpunan untuk penyelesaian level 0.

$$V_{x'0} = \{V1,\} \quad \rightarrow \quad V_{x0} = \{V2, V3, V4, V5\}$$

Pada solusi level 0, tahapan dengan solusi terpilih akan dimasukkan ke dalam himpunan solusi baru  $V_{x'0}$  yang merupakan solusi awal tahap 1 pada permasalahan ini. Kemudian pada  $V_{x0}$  sisa tahapan node yang tidak digunakan dikembalikan dan menjadi



cabang baru pada solusi selanjutnya yaitu  $V_{x'1}$ . Penyelesaian selanjutnya akan dituliskan dengan persamaan sebagai berikut

$$V_{x'1} = \{V1, V4\} \quad \rightarrow \quad V_{x1} = \{V2, V3, V5\}$$

$$V_{x'2} = \{V1, V4, V2\} \quad \rightarrow \quad V_{x2} = \{V3, V5\}$$

$$V_{x'3} = \{V1, V4, V2, V5\} \quad \rightarrow \quad V_{x3} = \{V3\}$$

$$V_{x'4} = \{V1, V4, V2, V5, V3\} \rightarrow V_{x4} = \{ \}$$

Dapat disimpulkan pada persamaan di atas terdapat 5 level dengan 1 level inisialisasi dan 4 level sebagai tahapan solusi untuk permasalahan tersebut yang sesuai dengan bound yang telah ditentukan sebelumnya.

Langkah-langkah dari metode branch and bound untuk menentukan solusi optimal, diringkas sebagai berikut.

1. Mencari solusi optimal untuk model pemrograman linear dengan bilangan batasan bilangan bulat
2. Pada node 1, buat solusi awal menjadi batas atas (upper bound) dan sisanya menjadi batas bawah (lower bound).
3. Pilih variabel dengan bagian pecahan terbesar untuk percabangan. Buat 2 garis atau relasi baru untuk variabel ini yang mewakili nilai bilangan (integer) yang dibagi. Hasilnya akan menjadi 2 relasi yang baru.
4. Buat 2 node baru, masing-masing node untuk masing-masing relasi
5. Selesaikan rancangan hasil model pemrograman linear dengan relasi yang baru yang ditambahkan pada masing-masing node.
6. Solusi awal adalah batas atas (upper bound) pada tiap node, dan solusi bilangan (integer) maksimum yang ada (pada tiap node) adalah batas bawah (lower bound).
7. Jika pada proses menghasilkan solusi bilangan (integer) yang layak dengan nilai batas atas (upper bound) terbesar pada tiap node akhir, solusi bilangan (integer) optimal telah tercapai. Jika solusi bilangan (integer) yang layak tidak muncul, cabang dari node dengan batas atas (upper bound) terbesar.
8. Ulangi lagi langkah pada nomor 3 (tiga) sampai hasil memenuhi parameter yang telah ditetapkan

## BAB III

### PEMBAHASAN MASALAH

#### 3.1 Prosedur Perhitungan

Proses perhitungan pada algoritma branch and bound atau B&B bisa direpresentasikan sebagai himpunan  $P_n$  yang merupakan sebuah himpunan tahapan dalam sebuah solusi. Tahapan direpresentasikan dengan sebuah node yang saling terhubung dari awal B&B diinisialisasi. Dalam merekonstruksi sebuah tree untuk B&B, digunakan beberapa metode seperti halnya metode greedy ataupun hanya proses rekursif acak untuk menghasilkan sebuah tree baru. Dalam kasus yang kami ambil, data bersifat acak sesuai dengan kapan data tersebut diinputkan. Pada tree yang dibentuk setiap levelnya merupakan sebuah himpunan tahapan yang akan menjadi sebuah solusi baru. Setiap node memiliki nilai atau *weight* yang akan menjadi pertimbangan ketika memilih tahapan tersebut menjadi solusi baru. Berikut adalah diagram yang menggambarkan bagaimana bentuk dari algoritma B&B pada kasus yang kami ambil.

Setiap level memiliki percabangan yang berbeda-beda dengan solusi yang berbeda pula. Dalam penentuan cabang selanjutnya “bound” sangat berpengaruh. Menggunakan bound yang tepat dapat mengurangi waktu pencarian solusi paling optimal dan membuat kerja algoritma ini semakin cepat. Dalam persamaan matematika dapat dituliskan dengan

$$P_a = \{P_1, P_2, P_3, \dots, P_{100}, P_n\}$$

Dimana  $P_a$  adalah solusi awal yang belum diproses menggunakan algoritma B&B. Untuk awal proses, himpunan  $P_a$  akan diurutkan menggunakan algoritma greedy atau juga bisa diurutkan berdasarkan ukuran (*weight*). Kasus yang kami gunakan memiliki beberapa batasan awal sebelum inputan diproses harus sesuai, yaitu jika jumlah pesanan makanan tidak mencapai 100 maka jumlah parameter yang digunakan akan berkurang dan hanya menggunakan pengkondisian biasa tanpa menggunakan B&B untuk penjadwalan antrean pesanan.

#### 3.2 Kasus (Penjadwalan Antrean Pesanan di Resto)

Pada kasus ini dengan batasan awal yang telah dibuat sebelumnya yaitu jumlah pemesanan yang dibuat agar pengolahan inputan bisa menjadi lebih memberikan dampak pada hasil perhitungan. Jika tidak sesuai dengan batas bawah permasalahan tersebut dapat

diselesaikan dengan pengkondisian biasa dengan parameter yang sudah dikurangi. Berikut beberapa parameter yang digunakan pada algoritma ini ;

Batasan	Nilai	Keterangan
Jumlah pesanan	$\geq 100$	Batasan pesanan minimum
Urutan pemesanan	Integer	-
Waktu Pemesanan	Time	Jam pesanan dibuat
Lama Masak	Integer	Estimasi waktu rata-rata dari menu tersebut
Keanggotaan pemesan	Member / bukan member	-
Jenis masakan	Basah / Kering / snack	-
Banyak pesan	Integer	-
Jumlah Juru Masak	Integer	
Jenis pesanan	Di tempat/bawa pulang	

Batasan yang telah disebutkan sengaja dibuat untuk menguji algoritma yang dibangun. Dalam kondisi sesungguhnya beberapa batasan dengan nilai tetap dapat berubah-ubah. Semisal  $W_t \leq 45$  menit, jika resto memiliki lebih banyak juru masak, waktu tunggu bisa dikurangi.  $W_t$  sendiri didapatkan dari waktu pemesanan dan dikurangi estimasi dari setiap pesanan yang sudah terlebih dahulu diproses jika melebihi batas maka pesanan tersebut akan dinaikkan prioritasnya dan akan diproses terlebih dahulu, jika pesanan tersebut merupakan sebuah pesanan dari jenis makanan atau pemesan yang sama pesanan tersebut akan segera digabungkan dengan pesanan lain agar dapat segera diolah.

Batasan tersebut dibuat berdasarkan sebuah data yang terdiri dari menu makanan dan estimasi waktu masak dari tiap makanan. Berikut adalah data estimasi waktunya;

Menu Masakan	Jenis Makanan	Estimasi waktu masak
Cumi Wijen	Makanan Utama (kering)	8 Menit
Udang Wijen	Makanan Utama (kering)	10 Menit
Bakwan Jagung	Makanan Utama (kering)	10 Menit
Tahu Isi	Makanan Utama (kering)	8 Menit
Cumi Goreng Kering Spesial	Makanan Utama (kering)	10 Menit
Bebek Goreng Krispi	Makanan Utama (kering)	15 Menit

Nasi Bebek Goreng Kering	Makanan Utama (kering)	20 Menit
Bebek Betutu Goreng	Makanan Utama (kering)	15 Menit
Bebek Betutu Kuah	Makanan Utama (kuah)	18 Menit
Nasi Bebek Betutu	Makanan Utama (kering)	16 Menit
Sop Ayam kuah Bening	Makanan Utama (basah)	20 Menit
Sup Tom Yam Seafood	Makanan Utama (basah)	15 Menit
Sup Lontong Kikil	Makanan Utama (basah)	20 Menit
Sup Asparagus	Makanan Utama (basah)	20 Menit
Sup Jagung Kepiting	Makanan Utama (basah)	22 Menit
Bakmi Anglo Kuah	Makanan Utama (basah)	23 Menit
Nasi Timbel Bakar	Makanan Utama (kering)	18 Menit
Nasi Kampoeng	Makanan Utama (kering)	10 Menit
Nasi Buk Madura	Makanan Utama (kering)	14 Menit
Sego Ndeso	Makanan Utama (kering)	10 Menit
French Fries	Snack Pembuka (kering)	5 Menit
Nasi Goreng Kepiting	Makanan Utama (kering)	14 Menit
Ayam Mentega	Makanan Utama (kering)	18 Menit
Bebek Lada Hitam	Makanan Utama (kering)	20 Menit
Pisang Goreng Pasir	Snack Pembuka (kering)	6 Menit
Brokoli Garlic	Snack Pembuka (kering)	5 Menit

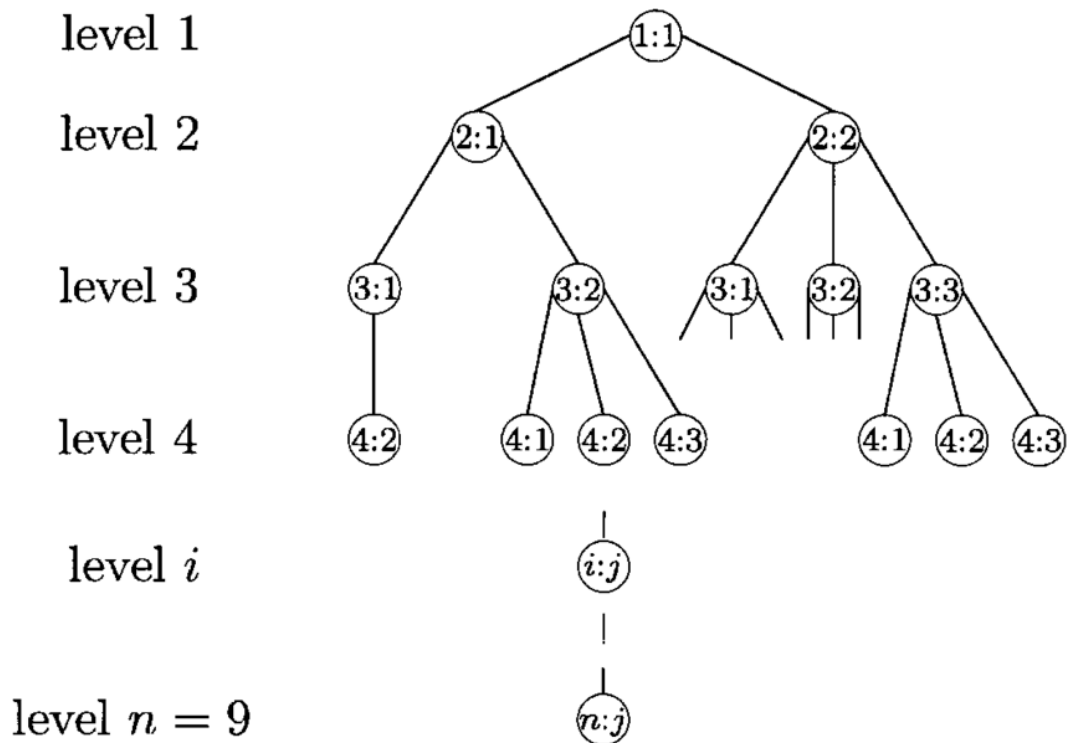
Dengan data yang ada bisa disimpulkan beberapa hal salah satunya adalah pola dari estimasi waktu dalam menyiapkan menu, setiap makanan memiliki estimasi waktu yang berbeda tergantung pada tingkat kesulitan pengolahan dan persiapannya sebelum dihidangkan. Jenis masakan juga berpengaruh pada pengolahan dan waktu yang dibutuhkan, pada makanan berkuah waktu yang dibutuhkan cenderung bertambah karena tingkat kesulitan juga naik.

Selanjutnya dalam aplikasi ini nantinya ada 3 tahapan utama agar dapat berjalan seperti yang sudah dituliskan di atas, tahapan tersebut adalah ;

- a. Filtrasi / peyaringan inputan sesuai parameter awal pesanan di mana  $\sum P \geq 100$ ;  $P$  adalah variabel pesanan

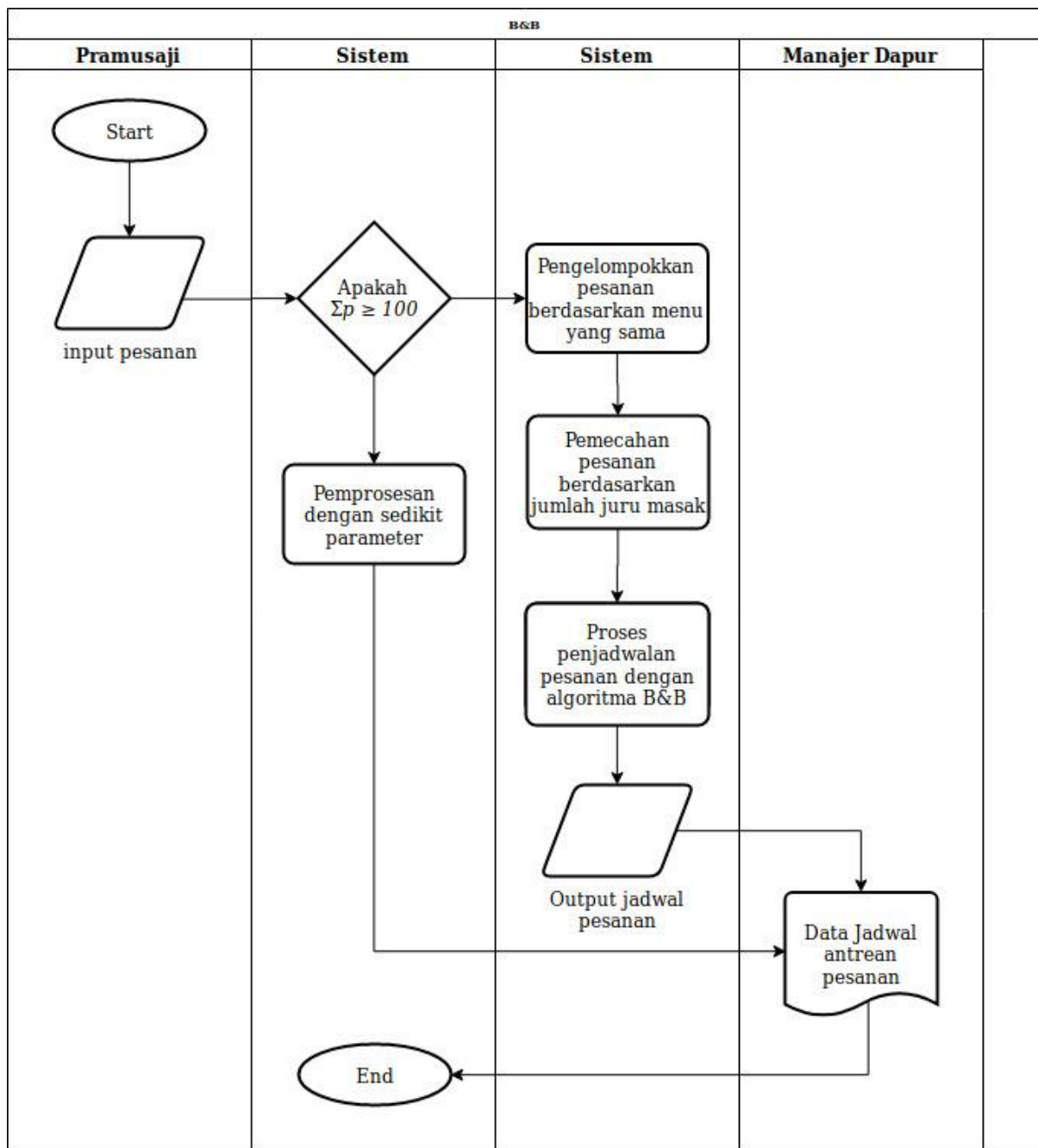
- b. Pengelompokkan / Grouping inputan berdasarkan menu makanan yang sama. Hal ini dilakukan karena dari  $\geq 100$  pesanan yang ada, memiliki kemungkinan untuk beberapa pesanan yang sama. Sebagai gantinya pesanan tersebut akan disatukan kedalam sebuah tahapan dalam tree dan estimasinya bertambah sesuai dengan rasionya
- c. Pada tahapan pesanan yang sudah dikelompokkan akan dibagi lagi menjadi beberapa tahapan terpisah sesuai dengan jumlah juru masak yang ada dan berapa porsi yang bisa ditangani
- d. Setelah di kelompokkan dan dipisah. Selanjutnya proses tersebut dimasukkan ke dalam tahapan level yang berbeda secara berurutan dalam bentuk tree. Jadi penggabungan tadi bukan berarti banyak pesanan menjadi satu, hanya dalam proses pemasakannya saja masakan menjadi satu. Untuk proses antrean dan pemesanan tetap direpresentasikan ke dalam node tahapan masing-masing
- e. Kemudian proses Branch and Bound dilakukan berdasarkan pada rules/bound yang sudah ditentukan sebelumnya.
- f. Hasil akan terus dibandingkan hingga solusi optimal ditemukan. Semua level pada branch and bound pasti dibentuk/direkonstruksi, namun tidak semua node di setiap level dijelajahi. Hal ini dilakukan agar solusi yang dihasilkan dapat menjadi lebih optimal.

Pada algoritma yang memiliki tujuan untuk mencari solusi optimal dari sebuah permasalahan parameter atau batasan merupakan aspek yang harus diperhatikan. Dalam algoritma B&B semakin banyak parameter yang digunakan maka semakin optimal solusi yang ditemukan. Untuk masalah kecepatan algoritma ini terletak pada proses branching yang lebih efektif dengan hanya melewati node tertentu dengan bound yang sudah ditentukan dari awal, tanpa melewati semua node yang ada sehingga waktu yang dibutuhkan lebih sedikit.



Dalam prosesnya sering kali terdapat 2 atau bahkan lebih solusi yang hampir sama dan memiliki penyelesaian yang serupa namun terdapat perbedaan pada beberapa poin. Sehingga pada akhirnya setiap solusi yang ditemukan harus dibandingkan ulang untuk menemukan solusi paling efektif dari sebuah permasalahan.

Dalam kasus ini selain branch and bound, juga digunakan beberapa parameter tambahan, pengkondisian dan algoritma pengelompokkan. Hal ini bertujuan agar data yang diproses pada algoritma ini dapat lebih terstruktur dengan baik dan menghasilkan solusi yang lebih optimal. Penggunaan algoritma, parameter dan kondisi tambahan diletakkan pada bagian awal dan akhir dari algoritma ini. Untuk selanjutnya proses – proses tersebut akan digambarkan menggunakan flowchart yang ada di bawah;



Flowchart di atas menjelaskan bagaimana program ini nantinya bekerja. Secara proses bisnis hanya ada 3 komponen di restoran yang terlibat ke dalam program ini, yaitu sistem ini sendiri, pramusaji dan manajer dapur yang mengurus hubungan pemesanan dan juru masak. Dengan begini semua proses algoritma yang sudah dituliskan diatas telah bekerja dan sesuai dengan apa yang telah digambarkan.

## **BAB IV**

### **PENUTUP**

#### **4.1 Kesimpulan**

- a. Program penjadwalan pelayanan di restoran memudahkan para pekerja untuk memanage waktu pada saat pemrosesan dan mempercepat pemesanan yang di pesan oleh customer atau pelanggan.
- b. Algoritma Branch And Bound digunakan untuk mempermudah dalam proses pengurutan penjadwalan pelayanan di restoran. Urutan penjadwalan di implementasikan menggunakan pohon tree. Jadi seperti sebuah anak pohon dan akan bercabang ke bawah dan di setiap cabangnya akan menghasilkan sebuah solusi – solusi optimal dan di setiap simpul akan di pilih dan di proses kembali sampai solusi paling optimal atau seoptimal mungkin.
- c. Pada pembahasan ini, aspek yang dibahas adalah proses pengurutan penjadwalan pelayanan di restoran dengan memperhatikan batasan pelayanan di restoran yang ada di Kota Malang.

#### **4.2 Saran**

##### **4.2.1 Bagi Pengembang**

Saran penulis untuk penelitian selanjutnya yang berkaitan dengan Algoritma Branch And Bound dalam system pengurutan penjadwalan pelayanan di restoran adalah sebagai berikut :

1. Menguasai teori terlebih dahulu sebelum ke tahap implementasi
2. Mempelajari Algoritma Branch And Bound secara detail dan terperinci.
3. Jangan pernah merasa cepat puas dengan hasil yang telah di capai

##### **4.2.2 Bagi Pengguna**

Saran untuk pengguna :

1. Mengembangkan metode yang telah dijelaskan dengan sebaik – baiknya
2. Mengembangkan metode tersebut ke tahap selanjutnya
3. Gunakan waktu sebaik mungkin



## **DAFTAR PUSTAKA**

<https://www.geeksforgeeks.org/implementation-of-0-1-knapsack-using-branch-and-bound/>

[https://en.wikipedia.org/wiki/Branch\\_and\\_bound](https://en.wikipedia.org/wiki/Branch_and_bound)