



**PENERAPAN ALGORITMA *BLOCPAN* DAN ALGORITMA *CRAFT*  
DALAM PERANCANGAN TATA LETAK RUANG RUMAH SAKIT**

**SKRIPSI**

Oleh

**Wika Anggani  
NIM 071810101052**

**JURUSAN MATEMATIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS JEMBER  
2015**



**PENERAPAN ALGORITMA *BLOCPAN* DAN ALGORITMA *CRAFT*  
DALAM PERANCANGAN TATA LETAK RUANG RUMAH SAKIT**

**SKRIPSI**

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat untuk menyelesaikan Program Studi Matematika (S1) dan mencapai gelar Sarjana Sains

Oleh

**Wika Anggani**  
**NIM 071810101052**

**JURUSAN MATEMATIKA**  
**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**  
**UNIVERSITAS JEMBER**  
**2015**

## PERSEMBAHAN

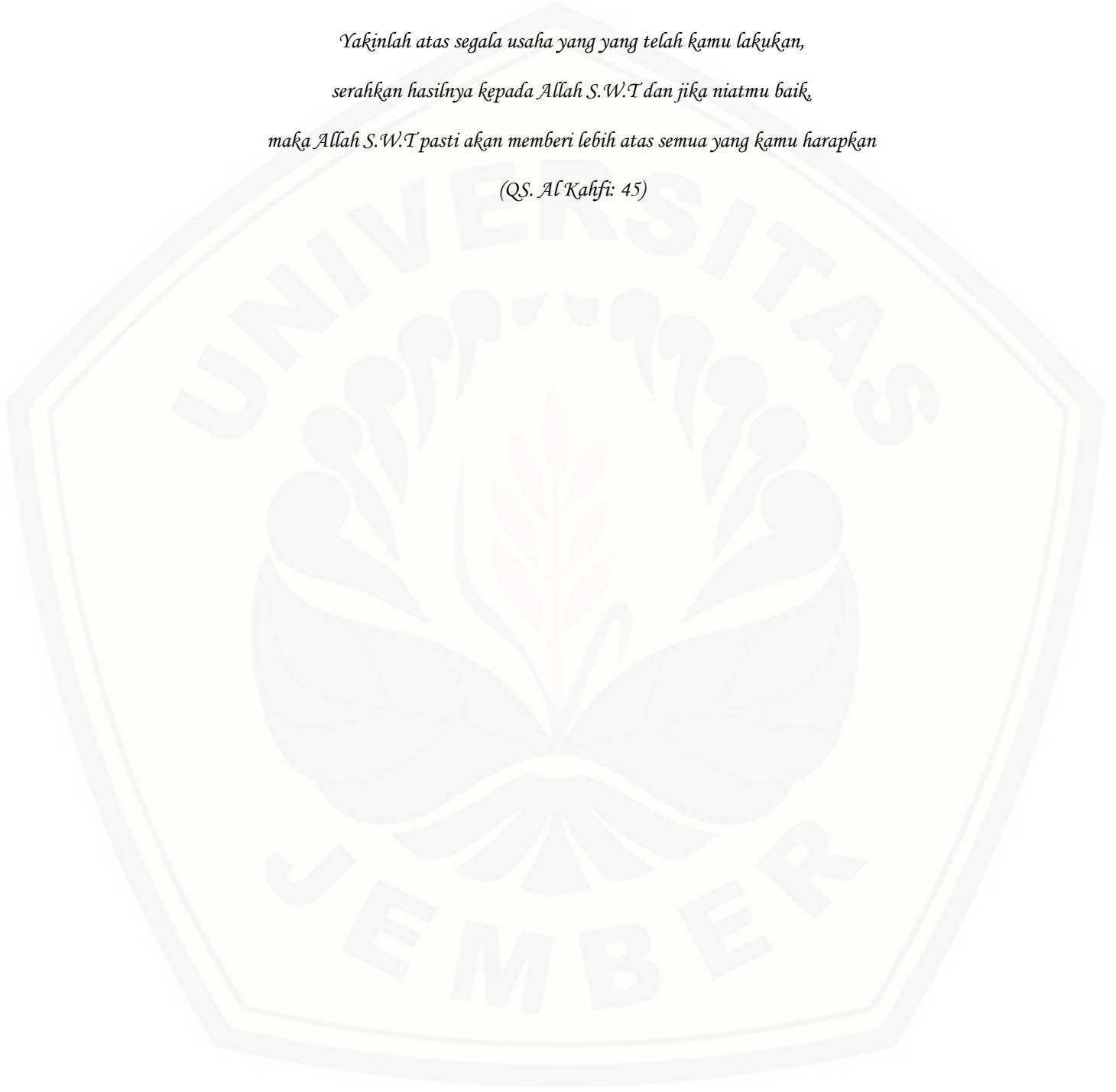
Skripsi ini saya persembahkan untuk:

1. ayahanda Heruyono dan ibunda Ely Mastutik yang tercinta, terimakasih atas kesabarannya dalam mendidik, mendoakan, memberikan kasih sayang serta pengorbanan selama ini;
2. bapak Kusbudiono, S.Si, M.Si., selaku Dosen Pembimbing Utama dan Bapak, Kiswara Agung Santoso, S.Si., M.Kom., selaku Dosen Pembimbing Anggota yang telah meluangkan waktu dan pikiran serta perhatiannya guna memberikan bimbingan dan pengarahan demi terselesaikannya penulisan skripsi ini;
3. adikku Adha Fajar Cahyani dan sahabatku Yulan Isa Puspita yang memberikan semangat dan selalu ada menemaniku dalam kondisi apapun;
4. guru-guru sejak taman kanak-kanak sampai dengan perguruan tinggi;
5. almamater Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

**MOTTO**

*Yakinlah atas segala usaha yang telah kamu lakukan,  
serahkan hasilnya kepada Allah S.W.T dan jika niatmu baik,  
maka Allah S.W.T pasti akan memberi lebih atas semua yang kamu harapkan*

*(QS. Al Kafi: 45)*



**PERNYATAAN**

Saya yang bertanda tangan di bawah ini:

nama : Wika Anggani

NIM : 071810101052

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Penerapan Algoritma *Blocplan* dan Algoritma *Craft* dalam Perancangan Tata Letak Ruang Rumah Sakit” adalah benar-benar hasil karya sendiri, kecuali kutipan yang telah saya sebutkan sumbernya, belum pernah diajukan pada institusi manapun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa ada tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika ternyata dikemudian hari pernyataan ini tidak benar.

Jember, Januari 2015

Yang menyatakan,

Wika Anggani  
NIM 071810101052

**SKRIPSI**

**PENERAPAN ALGORITMA *BLOCPAN* DAN ALGORITMA *CRAFT*  
DALAM PERANCANGAN TATA LETAK RUANG RUMAH SAKIT**

Oleh

Wika Anggani  
NIM 071810101052

Pembimbing

Dosen Pembimbing Utama : Kusbudiono, S.Si., M.Si.

Dosen Pembimbing Anggota : Kiswara Agung Santoso, S.Si., M.Kom.

**PENGESAHAN**

Skripsi berjudul “Penerapan Algoritma *Blocplan* dan Algoritma *Craft* dalam Perancangan Tata Letak Ruang Rumah Sakit” telah diuji dan disahkan pada:

hari, tanggal :

tempat : Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember

Tim Penguji:

Dosen Pembimbing Utama,

Dosen Pembimbing Utama,

Kusbudiono, S.Si., M.Si.

Kiswara Agung Santoso, S.Si., M.Kom.

NIP. 197704302005011001

NIP. 197209071998031003

Penguji I,

Penguji II,

Ika Hesti Agustin, S.Si., M.Si.

Ahmad Kamsyakawuni, S.Si., M.Kom.

NIP. 198408012008012006

NIP. 197211291998021001

Mengesahkan,

Dekan,

Prof. Drs. Kusno, DEA, Ph. D.

NIP. 196101081986021001



## RINGKASAN

### **Penerapan Algoritma *Blocplan* dan Algoritma *Craft* dalam Perancangan Tata Letak Ruang Rumah Sakit;**

Wika Anggani, 071810101052; 2015: 48 halaman; Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Hal paling mendasar yang sering terjadi pada rumah sakit adalah kesalahan pada tata letak ruang rumah sakit yang tidak efisien. Tata letak ruang rumah sakit merupakan salah satu faktor penting yang sangat menentukan seberapa besar kepuasan pasien terhadap pelayanan rumah sakit tersebut, dimana ada beberapa letak ruang yang berbeda dan beberapa dari pasien mengunjungi lebih dari satu letak ruang. Oleh karena itu, tata letak ruang rumah sakit sebaiknya direncanakan sebaik mungkin sehingga dapat memaksimalkan kinerja semua pihak yang bersangkutan di rumah sakit pada umumnya. Salah satu faktor kepuasan seorang pasien adalah kemudahan mengakses ruangan yang terintegrasi di rumah sakit. Kemudahan akses ruangan tersebut erat kaitannya dengan penataan letak ruangan yang berkaitan antara satu dengan lainnya. Satu hal yang dapat dijadikan pertimbangan dalam keterkaitan antar ruangan adalah banyaknya pasien yang berpindah dari satu ruangan ke ruangan yang lain. Sehingga dengan mengetahui ruangan mana yang memiliki banyak keterkaitan kunjungan dapat dilakukan penataan untuk mempermudah akses ruangan di rumah sakit oleh pasien.

Terdapat beberapa algoritma untuk menyelesaikan masalah yang berhubungan dengan menata ulang tata letak suatu ruang. Pada kasus ini untuk menata ulang tata letak ruang rumah sakit yang bertujuan untuk meminimalkan total bobot perpindahan pasien dari satu ruang ke ruang yang lain yaitu dengan menggunakan algoritma *Blocplan* dan *Craft*. Bobot perpindahan adalah jumlah dari hasil perkalian jarak antar poli dan frekuensi pasien yang dikonsulkan dari poli satu ke poli yang lain.



Penelitian ini dilakukan melalui beberapa langkah, yaitu mengumpulkan data tentang frekuensi perpindahan pasien yang dikonsulkan dari poli satu ke poli lain serta jarak antar masing-masing poli di Rumah Sakit Daerah Dr. Soebandi Jember. Kemudian dilakukan penyelesaian tata letak ruang dengan menggunakan algoritma *Blocplan*. Setelah itu hasil dari penyelesaian dengan algoritma *Blocplan* tersebut diuji kembali dengan menggunakan Algoritma *Craft*. Selain hasil algoritma *Blocplan* yang diuji kembali dengan menggunakan algoritma *Craft* terdapat hasil algoritma *Simulated Annealing* dan *Harmony Search* yang akan diuji kembali, dimana hasil dari kedua algoritma tersebut telah didapat dari peneliti sebelumnya Insiyah (2014). Kemudian hasil dari pengujian ketiga algoritma tersebut yaitu algoritma *Blocplan*, *Simulated Annealing* dan *Harmony Search* dibandingkan performa terbaiknya berdasarkan total bobot perpindahan yang paling minimal.

Dari penelitian yang dilakukan didapatkan hasil bahwa penerapan algoritma *Craft* pada tata letak ruang rumah sakit berdasarkan algoritma *Blocplan* lebih baik dari pada penerapan algoritma *Craft* pada tata letak ruang rumah sakit berdasarkan algoritma *Simulated Annealing* dan *Harmony Search*. Penerapan algoritma *Craft* pada tata letak ruang rumah sakit berdasarkan algoritma *Blocplan* memiliki total bobot perpindahan 100142,18 sedangkan penerapan algoritma *Craft* pada tata letak ruang rumah sakit berdasarkan algoritma *Simulated Annealing* dan *Harmony Search* memiliki total bobot perpindahan yang sama yaitu 105553,9.

## PRAKATA

Puji syukur kehadirat Allah SWT atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “Penerapan Algoritma *Blocplan* dan Algoritma *Craft* dalam Perancangan Tata Letak Ruang Rumah Sakit”. Skripsi ini disusun untuk memenuhi salah satu syarat untuk menyelesaikan pendidikan strata satu (S1) pada Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Penyusunan skripsi ini tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis menyampaikan terima kasih kepada:

1. Prof. Drs. Kusno, DEA., Ph.D., selaku Dekan FMIPA Universitas Jember
2. bapak Kusbudiono, S.Si, M.Si., selaku Dosen Pembimbing Utama dan Bapak, Kiswara Agung Santoso, S.Si, M.Kom., selaku Dosen Pembimbing Anggota yang telah meluangkan waktu dan pikiran serta perhatiannya guna memberikan bimbingan dan pengarahan demi terselesaikannya penulisan skripsi ini;
3. ibu Ika Hesti Agustin, S.Si., M.Si., selaku Dosen Penguji I dan Bapak Ahmad Kamsyakawuni, S.Si, M.Si, selaku Dosen Penguji II yang telah memberikan saran dan kritik demi terselesainya penulisan skripsi ini;
4. bapak Prof. Drs. I Made Tirta, M.Sc., Ph.D., selaku Dosen Pembimbing Akademik yang telah membimbing penulis selama menjadi mahasiswa;
5. ayahanda Heruyono dan Ibunda Ely Mastutik yang telah memberikan doa, dorongan semangat, serta nasihat demi terselesaikannya skripsi ini;
6. orang-orang terdekat yang selalu menjadi semangat dalam menyelesaikan kuliah Adha Fajar Cahyani dan Yulan Isa Puspita;
7. semua pihak yang tidak dapat disebutkan satu persatu.

Penulis menerima segala kritik dan saran dari semua pihak demi kesempurnaan skripsi ini. Akhirnya penulis berharap semoga skripsi ini dapat bermanfaat.

Penulis

Jember, Januari 2015

DAFTAR ISI

	Halaman
<b>HALAMAN JUDUL</b> .....	ii
<b>HALAMAN PERSEMBAHAN</b> .....	iii
<b>HALAMAN MOTTO</b> .....	iv
<b>HALAMAN PERNYATAAN</b> .....	v
<b>HALAMAN PEMBIMBING</b> .....	vi
<b>HALAMAN PENGESAHAN</b> .....	vii
<b>RINGKASAN</b> .....	viii
<b>PRAKATA</b> .....	x
<b>DAFTAR ISI</b> .....	xi
<b>DAFTAR TABEL</b> .....	xiii
<b>DAFTAR GAMBAR</b> .....	xiv
<b>DAFTAR LAMPIRAN</b> .....	xv
<b>BAB 1. PENDAHULUAN</b> .....	1
<b>1.1 Latar Belakang</b> .....	1
<b>1.2 Rumusan Masalah</b> .....	2
<b>1.3 Batasan Masalah</b> .....	3
<b>1.3 Tujuan</b> .....	3
<b>1.4 Manfaat</b> .....	4
<b>BAB 2. TINJAUAN PUSTAKA</b> .....	5
<b>2.1 Tata Letak Ruang Rumah Sakit</b> .....	5
<b>2.2 Tingkat Kepuasan Pasien</b> .....	6
<b>2.3 Kaitan Tata Letak Ruang Terhadap Kepuasan Pasien</b> .....	7
<b>2.4 Definisi Algoritma</b> .....	8
<b>2.5 Algoritma <i>Blocplan</i></b> .....	9
<b>2.6 Algoritma <i>Craft</i></b> .....	12

<b>BAB 3. METODE PENELITIAN .....</b>	<b>15</b>
<b>3.1 Data Penelitian .....</b>	<b>15</b>
<b>3.2 Langkah-langkah Penelitian.....</b>	<b>22</b>
<b>BAB 4. PEMBAHASAN .....</b>	<b>24</b>
<b>4.1 Penyelesaian Rancangan Tata Letak Ruang Menggunakan         Algoritma <i>Blocplan</i>.....</b>	<b>24</b>
<b>4.2 Penerapan algoritma <i>Craft</i> Pada Perancangan Tata Letak         Ruang Berdasarkan Algoritma <i>Blocplan</i>, <i>Simulated         Annealing dan Harmony Search</i> .....</b>	<b>30</b>
4.2.1 Penerapan algoritma <i>Craft</i> Pada Perancangan Tata Letak Ruang Berdasarkan Algoritma <i>Blocplan</i> .....	32
4.2.2 Penerapan algoritma <i>Craft</i> Pada Perancangan Tata Letak Ruang Berdasarkan Algoritma <i>Simulated Annealing</i> .....	35
4.2.3 Penerapan algoritma <i>Craft</i> Pada Perancangan Tata Letak Ruang Berdasarkan Algoritma <i>Harmony Search</i> .....	38
4.2.4 Perbandingan Penerapan Algoritma <i>Craft</i> Pada Perancangan Tata Letak Ruang Berdasarkan Algoritma <i>Blocplan</i> , <i>Simulated Annealing dan Harmony Search</i> .....	41
4.2.5 Denah Baru Penerapan Algoritma <i>Craft</i> Pada Perancangan Tata Letak Ruang Berdasarkan Algoritma <i>Blocplan</i> , <i>Simulated Annealing dan Harmony Search</i> .....	42
<b>BAB 5. PENUTUP .....</b>	<b>45</b>
<b>5.1 Kesimpulan .....</b>	<b>45</b>
<b>5.2 Saran .....</b>	<b>47</b>
<b>DAFTAR PUSTAKA.....</b>	<b>48</b>
<b>LAMPIRAN</b>	

DAFTAR TABEL

	Halaman
3.1 Daftar nama poli di rumah Sakit Daerah Dr.Soebandi Jember .....	16
3.2 Frekuensi perpindahan rata-rata pasien per-bulan .....	17
3.3 Jarak antar poli .....	19
4.1 Frekuensi perpindahan antar poli dari yang terbesar ke terkecil .....	26
4.2 Jarak antar ruang ke ruang lain dari terkecil ke terbesar .....	27
4.3 Pertukaran urutan tata letak awal dan urutan tata letak algoritma <i>Blocplan</i> .....	29
4.4 Contoh kemungkinan pertukaran urutan tata letak ruang algoritma <i>Craft</i> berdasarkan algoritma <i>Blocplan</i> .....	31
4.5 Kemungkinan pertukaran urutan tata letak ruang algoritma <i>Craft</i> berdasarkan algoritma <i>Blocplan</i> .....	32
4.6 Kemungkinan pertukaran urutan tata letak ruang algoritma <i>Craft</i> berdasarkan algoritma <i>Simulated Annealing</i> .....	35
4.7 Kemungkinan pertukaran urutan tata letak ruang algoritma <i>Craft</i> Berdasarkan algoritma <i>Harmony Search</i> .....	38



DAFTAR GAMBAR

	Halaman
2.1 Jarak <i>Rectilinier</i> .....	11
2.2 Jarak <i>Euclidean</i> .....	11
3.1 Denah Rumah Sakit.....	15
3.2 Skema Langkah-langkah Penelitian.....	22
4.1 Contoh denah .....	26
4.2 Contoh denah poli persegi panjang.....	26
4.3 Contoh denah baru .....	27
4.4 Tampilan program algoritma <i>Blocplan</i> .....	28
4.5 Tampilan program penerapan algoritma <i>Craft</i> pada tata letak ruang Berdasarkan algoritma <i>Blocplan</i> .....	34
4.6 Tampilan program penerapan algoritma <i>Craft</i> pada tata letak ruang Berdasarkan algoritma <i>BlocplanSimulated Annealing</i> .....	37
4.7 Tampilan program penerapan algoritma <i>Craft</i> pada tata letak ruang Berdasarkan algoritma <i>Harmony Search</i> .....	40
4.8 Denah baru penerapan algoritma <i>Craft</i> pada perancangan tata letak ruang berdasarkan Algoritma <i>Blocplan</i> .....	42
4.9 Denah baru penerapan algoritma <i>Craft</i> pada perancangan tata letak ruang berdasarkan Algoritma <i>Simulated Annealing</i> .....	43
4.10 Denah baru penerapan algoritma <i>Craft</i> pada perancangan tata letak ruang berdasarkan Algoritma <i>Harmony Search</i> .....	41



## BAB 1. PENDAHULUAN

### 1.1 Latar Belakang

Hal paling mendasar yang sering terjadi pada rumah sakit adalah kesalahan pada tata letak ruang rumah sakit yang tidak efisien. Tata letak ruang rumah sakit merupakan salah satu faktor penting yang sangat menentukan seberapa besar kepuasan pasien terhadap pelayanan rumah sakit tersebut, dimana ada beberapa letak ruang yang berbeda dan beberapa dari pasien mengunjungi lebih dari satu letak ruang. Oleh karena itu, tata letak ruang rumah sakit sebaiknya direncanakan sebaik mungkin sehingga dapat memaksimalkan kinerja semua pihak yang bersangkutan di rumah sakit pada umumnya.

Tidak mudah untuk merencanakan tata letak rumah sakit seperti yang diharapkan. Beberapa perancang sering mengalami kesulitan untuk mengatur tata letak suatu ruang sehingga dibutuhkan peran konsultan yang ahli dalam bidang tata letak ruang rumah sakit. Hal ini dikarenakan, satu letak ruang akan mempengaruhi letak ruang yang lain sehingga sangat berperan besar dalam efisiensi kegiatan yang ada pada rumah sakit.

Terdapat beberapa algoritma untuk menyelesaikan masalah yang berhubungan dengan menata ulang tata letak suatu ruang. Pada kasus ini untuk menata ulang tata letak ruang rumah sakit yang bertujuan untuk meminimalkan total jarak perjalanan pasien dari satu letak ruang ke ruang yang lain yaitu dengan menggunakan algoritma *Blocplan* dan *Craft*.

Algoritma *Blocplan* termasuk dalam salah satu algoritma *Hybrid*, dimana algoritma ini pada dasarnya mengklasifikasikan algoritma-algoritma yang memiliki karakteristik optimal dan sub-optimal, atau algoritma yang memiliki karakteristik sebagai algoritma konstruksi dan juga sebagai algoritma perbaikan (Heragu dan Kusiak, 1987). Sedangkan, algoritma *Craft* yang termasuk dalam salah satu algoritma perbaikan, dimana algoritma perbaikan merupakan algoritma yang memodifikasi solusi awal yang merupakan input data hingga biaya material handling dapat

dikurangi. Dalam algoritma perbaikan, yang dipilih secara random untuk kemudian dilakukan pertukaran diantara ruang-ruang dari solusi awal tersebut hingga didapat solusi akhir yang optimal.

Kedua algoritma tersebut, yaitu algoritma *Blocplan* dan *Craft* biasanya digunakan untuk menata ulang tata letak ruang produksi (pabrik). Akan tetapi dalam kasus ini akan digunakan untuk hal yang berbeda yaitu untuk menata ulang tata letak ruang rumah sakit. Penelitian ini dilakukan melalui beberapa langkah, yaitu mengumpulkan data tentang frekuensi perpindahan pasien yang dikonsulkan dari poli satu ke poli lain serta jarak antar masing-masing poli di Rumah Sakit Daerah Dr.Soebandi Jember. Kemudian dilakukan penyelesaian tata letak ruang dengan menggunakan algoritma *Blocplan*. Setelah itu hasil dari penyelesaian dengan algoritma *Blocplan* tersebut diuji kembali dengan menggunakan Algoritma *Craft*. Selain hasil algoritma *Blocplan* yang diuji kembali dengan menggunakan algoritma *Craft* terdapat hasil algoritma *Simulated Annealing* dan *Harmony Search* yang akan diuji kembali, dimana hasil dari kedua algoritma tersebut telah didapat dari peneliti sebelumnya Insyah (2014). Hasil tersebut berupa urutan tata letak ruang, urutan tata letak ruang dari algoritma *Simulated Annealing* adalah (1), (21), (8), (15), (5), (4), (9), (20), (2), (19), (13), (6), (10), (7), (18), (12), (17), (3), (11), (22), (23), (16), (14) dan , urutan tata letak ruang dari algoritma *Harmony Search* adalah (1), (3), (14), (6), (20), (4), (23), (22), (17), (19), (21), (11), (12), (16), (8), (9), (18), (7), (5), (2), (10), (13), (15). Kemudian hasil dari pengujian ketiga algoritma tersebut yaitu algoritma *Blocplan*, *Simulated Annealing* dan *Harmony Search* dibandingkan performa terbaiknya berdasarkan total bobot perpindahan yang paling minimal.

## 1.2 Rumusan Masalah

Permasalahan yang dibahas dalam tugas akhir ini adalah sebagai berikut.

- a. Bagaimana merancang tata letak ruang di instalasi rawat jalan Rumah Sakit Daerah Dr.Soebandi Jember berdasarkan frekuensi perpindahan pasien dan jarak

- antar ruang dengan menggunakan algoritma *Blocplan* sehingga mendapatkan urutan ruang rumah sakit.
- b. Bagaimana hasil penerapan algoritma *Craft* pada perancangan tata letak ruang rumah sakit berdasarkan algoritma *Blocplan*, *Harmony Search* dan *Simmulated Annealing*.
  - c. Bagaimana hasil perbandingan dari penerapan algoritma *Craft* pada perancangan tata letak ruang rumah sakit berdasarkan algoritma *Blocplan*, *Harmony Search* dan *Simmulated Annealing* dengan total bobot dari masing-masing algoritma tersebut sebagai bahan perbandingan.

### 1.3 Batasan Masalah

Permasalahan dalam penulisan skripsi ini dibatasi oleh beberapa hal. Batasan tersebut adalah sebagai berikut:

- a. Penelitian yang dilakukan hanya pada ruang atau poli di instalasi rawat jalan Rumah Sakit Daerah Dr.Soebandi Jember.
- b. Luas area pada semua ruang diabaikan.
- c. Jarak antar poli didapatkan dengan menghitung jarak antar pintu masuk ruang yang satu ke pintu masuk ruang yang lain.
- d. Perancangan tata letak rumah sakit diselesaikan tanpa adanya penambahan ruang baru.

### 1.4 Tujuan

Tujuan dari penelitian ini adalah sebagai berikut.

- a. Mendapatkan hasil dari pengaplikasian algoritma *Blocplan* terhadap perancangan tata letak ruang di instalasi rawat jalan Rumah Sakit Daerah Dr.Soebandi Jember sehingga mendapatkan total bobot perpindahan pasien yang seminimal mungkin.
- b. Mendapatkan hasil penerapan algoritma *Craft* pada perancangan tata letak ruang rumah sakit berdasarkan algoritma *Blocplan*, *Harmony Search* dan *Simmulated Annealing*.

- c. Mendapatkan hasil perbandingan dari penerapan algoritma *Craft* pada perancangan tata letak ruang rumah sakit berdasarkan algoritma *Blocplan*, *Harmony Search* dan *Simmulated Annealing* dengan total bobot dari masing-masing algoritma tersebut sebagai bahan perbandingan.

### **1.5 Manfaat**

Manfaat yang diharapkan dari penulisan tugas akhir ini adalah dapat memberikan kontribusi sebagai bahan pertimbangan dalam perbaikan rancangan tata letak ruang rumah sakit dan fasilitas lain, sehingga mendapatkan tata letak ruang yang efisien.

## BAB 2. TINJAUAN PUSTAKA

### 2.1 Tata Letak Ruang Rumah Sakit

Tata letak ruang merupakan desain dari bagian-bagian dan pusat kerja yang digunakan dalam kehidupan sehari-hari. Tata letak ruang adalah keputusan penting yang menentukan efisiensi sebuah operasi jangka panjang. Hal tersebut dapat dilihat bahwa tata letak ruang dapat menentukan daya saing berbagai perusahaan dalam hal kapasitas, proses, fleksibilitas, biaya, kualitas kerja, kontak pelanggan serta citra perusahaan.

Menurut Permana (2002), untuk memenuhi manfaat dari *setting* terhadap bangunan rumah sakit, maka bangunan rumah sakit itu harus memenuhi persyaratan tata letak ruang rumah sakit diantaranya.

- a. Posisi bangunan satu dengan bangunan lain yang secara fungsi berhubungan harus memiliki akses yang mudah dijangkau, dapat dilihat dan letaknya tidak berjauhan.
- b. Penataan bangunan yang sesuai dengan urutan fungsi agar alur pasien tidak bertabrakan yang akan membuat stagnasi pada ruangan tertentu.
- c. Lahan parkir memadai dan tidak membedakan pelanggan.
- d. Bangunan sebaiknya terintegrasi sesuai dengan fungsinya masing-masing, sehingga pasien tidak kelelahan dalam mencari pelayanan terintegrasi di rumah sakit.
- e. Dapat membentuk citra pelayanan yang memuaskan, intinya adalah bagaimana bangunan tersebut unik, menarik, dan terkesan lega, serta nyaman.
- f. Aman dengan kualitas minimal sesuai standar.
- g. Lantai bangunan harus dirancang sedemikian rupa agar tidak licin, dan nada tangga dari bangunan satu ke bangunan yang lain yang lebih tinggi perlu penataan yang tidak membahayakan pasien serta ada tanda yang memperlihatkan tangga tersebut agar pasien tidak terjatuh.
- h. Bangunan yang baik adalah bangunan yang terkesan lega dan detailnya menarik.



- i. Adanya keserasian perilaku provider dengan area yang ia tempati, dalam arti bangunan dan peralatan yang menunjang penampilan SDM yang berperilaku asertif dan menarik.

## 2.2 Tingkat Kepuasan Pasien

Fokus kualitas adalah kepuasan pelanggan. Oleh karena itu perlu dipahami komponen-komponen yang berkaitan dengan kepuasan pelanggan. Kepuasan pelanggan sangat tergantung kepada persepsi dan ekspektasi mereka. Faktor yang mempengaruhi persepsi dan ekspektasi pelanggan adalah keluhan dan keinginan, pengalaman masa lalu, pengalaman teman-teman dan komunikasi melalui iklan dan pemasaran. Selain itu faktor umur, pendidikan, jenis kelamin, kepribadian, suku dan latar belakang budaya, serta kasus penyakit turut mempengaruhi ekspektasi dan persepsi pelanggan atau pasien (Nelson dan Brown, 1994).

Pasien merupakan individu terpenting di rumah sakit, yaitu sebagai konsumen sekaligus produk rumah sakit. Sebagai pasien yang mempercayakan penyembuhannya kepada rumah sakit yang dipilihnya, mereka mempunyai kebutuhan yang diharapkan dapat diperoleh melalui rumah sakit tersebut. Harapan pasien dari pelayanan rumah sakit dalam proses pengobatan akan menimbulkan suatu kepuasan yang diharapkan dapat mempercepat proses penyembuhan (Soedjadi, 1996).

Berbicara mengenai kepuasan pasien berarti berbicara mengenai mutu pelayanan kesehatan. Kepuasan pasien adalah salah satu indikator kinerja rumah sakit umum (Soedaji, 1996). Kepuasan pasien akan tercapai apabila diperoleh hasil yang optimal bagi setiap pasien. Terdapat beberapa unsur yang dapat mempengaruhi kepuasan pasien terhadap pelayanan rumah sakit yaitu tenaga medis (dokter), tenaga perawat, kondisi lingkungan fisik, penyediaan sarana penunjang medis dan nonmedis serta menu dan makanan.



### 2.3 Kaitan Tata Letak Ruang Terhadap Kepuasan Pasien

Unsur penting yang menentukan kepuasan pasien salah satunya adalah penataan tata ruang. Menurut Permana (2002), tujuan dari pengaturan gedung atau bangunan rumah sakit adalah.

#### a. Kepentingan Konsumen

##### 1) Kemudahan alur lalu lintas pasien

Pihak provider perlu memikirkan desain rumah sakit yang baik, yang tidak membingungkan, posisi bangunan satu dengan bangunan lain yang secara fungsi berhubungan haruslah memiliki akses yang mudah dijangkau, dapat dilihat, dan jaraknya tidak berjauhan. Kemudian diperlukan penataan bangunan yang sesuai dengan urutan fungsi agar alur pasien tidak bertabrakan yang akan menimbulkan stagnasi pada ruangan tertentu.

##### 2) Mengurangi energi pelanggan selama aktifitas dalam mendapatkan pelayanan.

Bangunan sebaiknya terintegrasi sesuai fungsinya masing-masing, sehingga pasien tidak kelelahan dalam mencari pelayanan terintegrasi di rumah sakit. Sebab jika ruangan yang fungsinya saling menunjang kemudian ruangan atau bangunannya berjauhan maka energi pasien akan terkuras habis sebelum pasien tersebut tuntas mendapatkan pelayanan akibatnya pasien akan enggan datang kembali ke rumah sakit tersebut.

##### 3) Menanamkan citra yang baik bukan citra modern

Bangunan yang familiar bisa tampil menarik dan tidak perlu mewah, dapat membentuk citra pelayanan yang memuaskan. Intinya bagaimana bangunan tersebut unik, menarik, dan terkesan lega, dan nyaman.

##### 4) Menimbulkan rasa aman bagi pelanggan

Konstruksi bangunan harus sama dengan kualitas minimal sesuai standar. Lantai bangunan harus dirancang sedemikian rupa agar tidak licin, dan bahkan tangga dari bangunan satu ke bangunan lain yang lebih tinggi perlu penataan yang tidak membahayakan pasien dan ada tanda khusus yang memperlihatkan tangga tersebut agar pasien tidak terjatuh.

#### 5) Menghilangkan kejenuhan konsumen

Bangunan yang baik adalah bangunan yang terkesan lega dan detailnya menarik, warna yang serasi serta variasi lantai yang menarik. Diperlukan desain bangunan yang unik yang bisa menghilangkan kejenuhan, misalnya pada ruang tunggu lantainya bermotif, penempatan bunga pada beberapa sisi bangunan, dan lain sebagainya.

#### 6) Meningkatkan nilai-nilai pelayanan rumah sakit

Saat ini tidak sedikit kelompok konsumen yang berorientasi kepada nilai-nilai yang bisa mereka dapatkan pada saat menerima pelayanan. Bangunan adalah salah satu yang dapat mempertajam nilai tersebut disamping perilaku SDM dan penampilan lainnya yang mampu menggugah rasa bangga pada diri konsumen. Nilai yang terpancar akan lebih menajam dari keserasian perilaku provider dengan area yang ia tempati, dalam arti bangunan dan peralatan yang menunjang penampilan SDM yang berperilaku asertif dan menarik.

#### b. Kepentingan Karyawan

- 1) Menumbuhkan kebanggaan dan keyakinan karyawan
- 2) Mengurangi energi
- 3) Retensi pegawai
- 4) Membentuk budaya profesional
- 5) Menghilangkan kejenuhan karyawan
- 6) Meningkatkan nilai-nilai pelayanan rumah sakit

Namun ruang lingkup dalam penelitian ini tidak membahas pengaruh tata ruang terhadap kepentingan karyawan, hanya melihat pengaruhnya terhadap kepuasan pasien mengenai perpindahan antar ruang yang digunakan oleh pasien.

### 2.4 Definisi Algoritma

Untuk menyelesaikan beberapa masalah perlu adanya suatu solusi yang dapat memecahkan masalah tersebut dengan baik dan benar. Solusi tersebut bisa dalam bentuk algoritma. Algoritma adalah urutan langkah-langkah logis penyelesaian

masalah yang disusun secara sistematis dan logis. Kata logis merupakan kata kunci dalam algoritma. Langkah-langkah dalam algoritma harus logis dan harus dapat ditentukan bernilai salah atau benar (shofwatul'uyun, 2009).

Banyak macam dari algoritma yang terbagi berdasarkan fungsi untuk menyelesaikan suatu masalah yang dihadapi. Dari berbagai macam algoritma tersebut tidak semua algoritma yang tergolong algoritma yang baik. Suatu algoritma yang baik harus memiliki sifat – sifat sebagai berikut :

- a. Masukan (*input*) dari himpunan tertentu.
- b. Keluaran (*output*) pada himpunan tertentu (solusi).
- c. Definisi dari setiap langkah perhitungan.
- d. Kebenaran dari keluaran untuk setiap masukan yang mungkin.
- e. Keberhinggaan dari banyaknya langkah perhitungan.
- f. Keefektifan dari setiap langkah perhitungan.
- g. Keterumuman dalam suatu kelompok permasalahan yang dipecahkan

Secara umum penilaian dalam penggunaan algoritma dapat dilihat dari efisiensi dari algoritma tersebut. Algoritma yang efisien adalah algoritma yang dapat meminimumkan kebutuhan waktu dan ruang. Yang dimaksud waktu (*time*) merupakan banyaknya komputasi elementer dalam algoritma, sedangkan ruang (*space*) adalah banyaknya sel memori yang dibutuhkan oleh algoritma. Ukuran tersebut secara berturut-turut disebut sebagai kompleksitas komputasi (*computational complexity*) dan kompleksitas ruang (*space complexity*).

## **2.5 Algoritma Blocplan**

Algoritma *blocplan* adalah algoritma yang mempertimbangkan pertukaran lokasi departemen berdasarkan keterkaitan pada kerja dan proses, proses ini diulang sampai tidak ada lagi pengurangan ongkos yang berarti (Supardi, 2006). Algoritma ini merupakan model perancangan tata letak fasilitas yang dikembangkan oleh Charles E, Donaghey dan Vanina F Pire pada tahun 1991. Algoritma *blocplan* merupakan salah satu dari algoritma *hybrid* yang menggabungkan metode

pembentukan dan untuk perbaikannya dilakukan dengan menggunakan metode perbaikan. Algoritma *blocplan* mempunyai beberapa kelebihan daripada algoritma pengolah tata letak fasilitas yang lain yaitu mudahnya dalam proses input data yang dilakukan dan hasil yang baik seperti algoritma yang lain. Input data yang digunakan dalam algoritma ini berupa data kualitatif tapi dapat memberikan *output* yang sama baiknya dengan algoritma yang lain yang menggunakan input data kuantitatif. Akan tetapi algoritma *blocplan* juga mempunyai kelemahan yaitu tidak akan menangkap initial *layout* secara akurat, pengembangan tata letak hanya dapat dicari dengan melakukan perubahan atau pertukaran letak departemen satu dengan lainnya (Purnomo, 2004).

Secara umum langkah-langkah algoritma *blocplan* adalah sebagai berikut :

a. Membuat titik awal

Dilakukan dengan menggambar ulang tata letak mesin, peralatan, operator menjadi suatu area atau blok dalam koordinat sumbu x dan y. kemudian ditentukan titik tengah (koordinat) departemen produksi. Titik tengah ini berfungsi menghitung jarak departemen dengan menggunakan perhitungan jarak *rectilinier* atau jarak *euclidean*. Jarak *rectilinier* adalah jarak yang diukur tegak lurus dari pusat fasilitas ke fasilitas yang lain. Rumus dari jarak *rectilinier* adalah sebagai berikut :

$$d_{ij} = |(x_i + x_j)| + |y_i + y_j| \quad (2.1)$$

dimana,  $x_i$  = koordinat x untuk fasilitas i

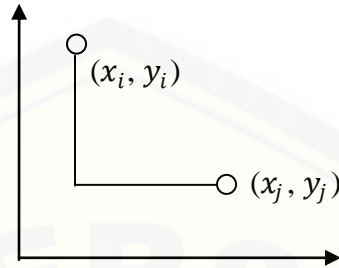
$x_j$  = koordinat x untuk fasilitas j

$y_i$  = koordinat y untuk fasilitas i

$y_j$  = koordinat y untuk fasilitas j

$d_{ij}$  = jarak antar fasilitas i dan j

Gambar dari jarak *rectilinier* adalah sebagai berikut :

Gambar 2.1 Jarak *Rectilinier*

Sedangkan jarak *euclidean* adalah jarak yang diukur lurus dari pusat fasilitas yang satu ke fasilitas yang lain. Rumus dari jarak *euclidean* adalah sebagai berikut

:

$$d_{ij} = [(x_i - x_j)^2 + (y_i - y_j)^2]^{\frac{1}{2}} \quad (2.2)$$

dimana,  $x_i$  = koordinat x untuk fasilitas i

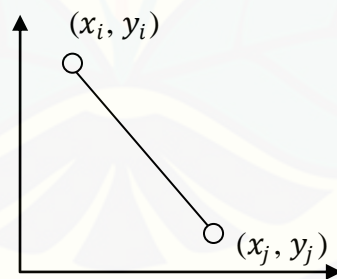
$x_j$  = koordinat x untuk fasilitas j

$y_i$  = koordinat y untuk fasilitas i

$y_j$  = koordinat y untuk fasilitas j

$d_{ij}$  = jarak antar fasilitas i dan j

Gambar dari jarak *euclidean* adalah sebagai berikut :

Gambar 2.2 Jarak *Euclidean*

#### b. Pengukuran jarak

Pengukuran jarak dilakukan dengan menggunakan pengukuran *rectilinier* dan *Euclidean* dan pada pengukuran jarak masing – masing tidak memperhatikan adanya *aisle* (jalan lintasan), sehingga pengukuran dilakukan secara langsung dari



masing-masing titik tengah departemen produksi. Dalam pengukuran biaya *material handling* akan dipengaruhi oleh jenis peralatan yang digunakan, upah tenaga kerja dan jarak yang ditempuh per periode pekerja. Pengukuran jarak *rectilinier* dan jarak *Euclidean* dengan menggunakan rumus (2.1) dan (2.2) yang telah disebutkan sebelumnya.

c. Penentuan frekuensi perpindahan antar departemen

Frekuensi perpindahan ditentukan untuk memperlihatkan banyaknya jumlah aliran perpindahan yang terjadi dalam proses perpindahan antar departemen. Rumus frekuensi perpindahan ini adalah :

$$F = \frac{\text{jarak yang ditempu } h}{\text{jarak yang ditempu } h \text{ rata } 2} \quad (2.3)$$

d. Melakukan pengurutan tata letak ruang

Dari data jarak dan frekuensi perpindahan maka dapat ditentukan urutan tata letak ruang yang baru. Proses pengurutannya yaitu berdasarkan ruang yang memiliki frekuensi perpindahan terbesar menempati ruang yang memiliki perpindahan jarak yang terkecil.

e. Membuat denah baru

Dari urutan tata letak ruang yang didapat tersebut dapat dibuat *layout* atau yang biasa disebut denah yang baru.

## 2.6 Algoritma *Craft*

Algoritma *Craft* (*Computerized Relative Allocation Facilities Technique*) pertama kali diperkenalkan oleh Armour, Buff dan volman pada tahun 1964. Algoritma *craft* ini adalah algoritma program teknik *heuristic* yang berdasarkan pada interpretasi *Quadratic Assignment* dari program proses tata letak. *Quadratic Assignment* digunakan untuk meminimumkan biaya perpindahan material dimana biaya ini digambarkan sebagai fungsi linier dari jarak perpindahan. Algoritma *Craft* ini menggunakan *from to chart* sebagai inputan, biaya *layout* ditentukan berdasarkan jarak *centroid* dan departemen yang dihitung tidak dibatasi dalam bentuk *rectangular*.



Algoritma ini merupakan algoritma *improvement* sehingga memerlukan *initial layout* yang bisa merupakan denah dari departemen yang sudah ada atau hasil dari algoritma lain.

Input yang digunakan dalam algoritma *craft* adalah

a. Denah awal

Layout awal dapat berupa tata letak yang sudah ada, ataupun *layout* awal dari hasil rancangan baru. Di dalam *layout* awal perlu adanya data jumlah fasilitas, luas area masing-masing fasilitas.

b. Pengukuran jarak

Pengukuran jarak pada algoritma *Craft* sama halnya dengan pengukuran jarak pada algoritma *Blocplan*.

c. Melakukan pengurutan tata letak ruang

Pada proses pengurutan tata letak ruang dengan menggunakan algoritma *craft* dibutuhkan syarat pertukaran antar ruang. Pertukaran antar ruang tersebut menurut algoritma ini harus memenuhi salah satu syarat di bawah ini, yaitu :

- 1) Harus mempunyai perbatasan yang sama
- 2) Harus mempunyai ukuran yang sama
- 3) Harus mempunyai kedua perbatasan-perbatasan yang sama pada ketiga departemen.

Cara kerja dari algoritma *craft* adalah dengan mempertimbangkan pertukaran lokasi dari pasangan fasilitas – fasilitas tertentu. Pertimbangan pasangan – pasangan fasilitas tersebut apabila mempunyai area yang sama atau berdekatan. Jika semua fasilitas memiliki area yang sama atau jika setiap pasangan yang tidak berdekatan memiliki area yang sama, maka algoritma *craft* dapat menguji kemungkinan maksimum  $n(n-1)/2$  pertukaran-pertukaran pada setiap iterasi, dimana  $n$  adalah jumlah fasilitas-fasilitas pada masalah layout. Umumnya pada masalah layout, jumlah dari  $n(n-1)/2$  dan hal ini disebabkan tidak semua fasilitas memiliki area yang sama. Pertukaran lokasi fasilitas tersebut akan

menghasilkan denah atau tata letak fasilitas yang mengurangi atau meminimumkan biaya perpindahan bahan dari initial layout atau tata letak awal yang ada.

d. Perhitungan momen perpindahan

Total momen perpindahan dapat ditentukan dengan mengalikan frekuensi perpindahan dari satu departemen ke departemen lainnya dengan jarak antar departemen yang berkaitan. Adapun perhitungan momen perpindahan dapat dihitung dengan menggunakan persamaan :

$$Z_{ij} = \sum_{i=1}^n \sum_{j=1}^n f_{ij} d_{ij} \quad (2.4)$$

Dimana :  $Z_{ij}$  = momen perpindahan stasiun i ke j

$f_{ij}$  = frekuensi perpindahan dari stasiun i ke j

$d_{ij}$  = jarak antar stasiun i dan j

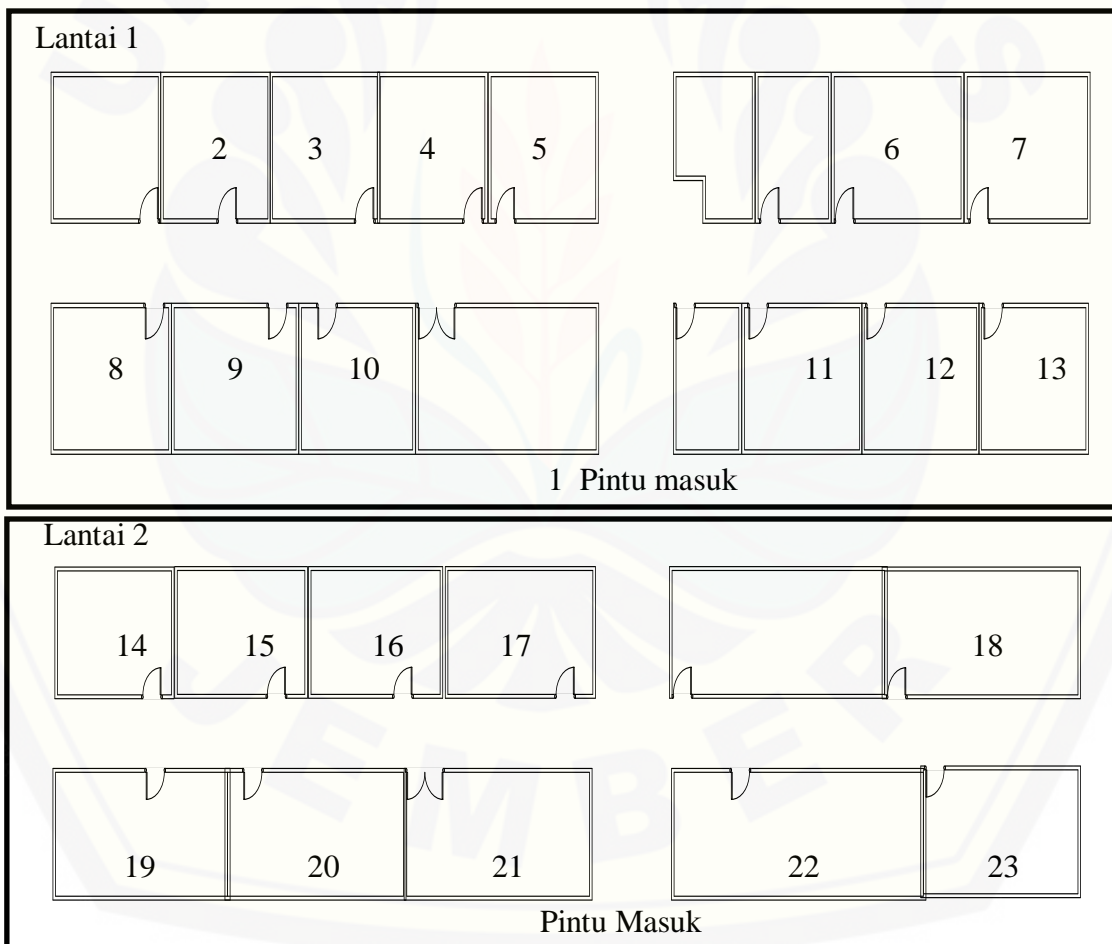
e. Membuat denah baru

Dari urutan tata letak ruang yang didapat tersebut dapat dibuat layout atau yang biasa disebut denah yang baru

### BAB 3. METODE PENELITIAN

#### 3.1 Data Penelitian

Peneliti mengambil data yang sudah digunakan peneliti sebelumnya yaitu data dari Insiyah (2014), yang diperoleh dari Rumah Sakit Daerah Dr.Soebandi Jember khususnya pada instalasi rawat jalan. Data tersebut berupa denah Rumah Sakit Daerah Dr.Soebandi Jember, jarak antar poli satu dengan poli lainnya serta frekuensi perpindahan pasien yang dikonsulkan oleh dokter dari poli satu ke poli lainnya selama 6 bulan. Denah rumah sakit, data jarak serta data frekuensi rata-rata perpindahan pasien tersebut adalah sebagai berikut:



Gambar 3.1 Denah Rumah Sakit

Keterangan denah rumah sakit tersebut ditunjukkan pada Tabel 3.1.

Tabel 3.1 Daftar nama poli di rumah Sakit Daerah Dr.Soebandi Jember

<b>No</b>	<b>Nama Ruang</b>
1	Pintu Masuk
2	Poli Bedah umum
3	Poli bedah urologi
4	Poli bedah syaraf
5	Poli bedah orthopedic
6	Poli gizi
7	Poli hemodialisa
8	Poli kandungan
9	Poli hamil dan laktasi
10	Poli syaraf
11	Poli jantung
12	Poli interna
13	Poli paru
14	Poli VCT
15	Poli kemoterapi
16	Poli anastesi
17	Poli eksekutif
18	Poli gigi dan mulut
19	Poli psikiatri
20	Poli kulit dan kelamin
21	Poli mata
22	Poli anak
23	Poli THT

Tabel 3.2 Frekuensi perpindahan rata-rata pasien per-bulan

Poli	Tujuan											
	1	2	3	4	5	6	7	8	9	10	11	12
<b>1</b>	0	188	71	9	60	0	10	84	115	111	95	136
<b>2</b>	188	0	1	0	6	0	0	4	1	3	60	18
<b>3</b>	71	1	0	0	0	0	2	2	0	4	35	5
<b>4</b>	9	0	0	0	1	0	0	0	0	2	5	0
<b>5</b>	60	6	0	1	0	0	0	0	0	3	11	3
<b>6</b>	0	0	0	0	0	0	0	0	0	1	2	14
<b>7</b>	10	0	2	0	0	0	0	0	0	0	0	0
<b>8</b>	83	4	2	0	0	0	0	0	0	0	27	7
<b>9</b>	115	1	0	0	0	0	0	0	0	1	3	3
<b>10</b>	111	3	4	1	3	1	0	0	1	0	7	5
<b>11</b>	95	60	35	5	11	2	0	27	3	7	0	32
<b>12</b>	136	18	5	0	3	14	0	7	3	5	32	0
<b>13</b>	34	5	3	0	1	0	0	1	2	1	10	9
<b>14</b>	0	1	0	0	0	0	0	0	1	0	0	5
<b>15</b>	4	0	0	0	0	0	0	0	0	0	0	0
<b>16</b>	0	63	27	12	26	0	0	25	9	0	0	0
<b>17</b>	146	0	0	0	0	12	0	0	0	0	25	2
<b>18</b>	60	1	0	0	0	0	0	0	0	2	6	4
<b>19</b>	75	0	0	0	1	0	0	0	2	1	2	0
<b>20</b>	179	5	2	0	0	1	0	0	3	0	2	6
<b>21</b>	281	2	0	2	0	0	0	0	3	3	21	14
<b>22</b>	159	11	1	3	1	0	0	0	0	7	3	0
<b>23</b>	272	6	0	0	0	0	0	0	6	7	13	2



	Poli	Tujuan										
		13	14	15	16	17	18	19	20	21	22	23
A s a l	1	34	0	4	0	146	60	75	179	281	159	272
	2	5	1	0	63	0	1	0	5	2	11	6
	3	3	0	0	27	0	0	2	2	0	1	0
	4	0	0	0	12	0	0	0	0	2	3	0
	5	1	0	0	26	0	0	1	1	0	1	0
	6	0	0	0	0	12	0	0	0	0	0	0
	7	0	0	0	0	0	0	0	0	0	0	0
	8	1	0	0	25	0	0	0	3	0	0	0
	9	2	1	0	9	0	0	2	0	3	0	6
	10	1	0	0	0	0	2	1	2	3	7	7
	11	10	0	0	0	25	6	2	0	21	3	13
	12	9	5	0	0	2	4	0	6	14	0	2
	13	0	2	0	0	0	1	0	1	0	1	3
	14	2	0	0	0	2	0	3	12	0	1	0
	15	0	0	0	0	0	0	0	0	0	0	0
	16	0	0	0	0	13	3	0	0	3	0	6
	17	0	2	0	13	0	0	202	0	25	1	14
	18	1	0	0	3	0	0	0	1	1	2	15
	19	0	3	0	0	202	0	0	1	2	3	1
	20	1	12	0	0	0	1	1	0	1	2	2
	21	0	0	0	3	25	1	0	1	0	4	2
	22	1	1	0	0	1	2	3	2	4	0	9
	23	3	0	0	6	14	15	1	2	2	9	0

Tabel 3.3 Jarak antar poli (meter)

Poli	Tujuan											
	1	2	3	4	5	6	7	8	9	10	11	12
<b>1</b>	<b>0</b>	38.83	34.9	23.4	17.9	28.16	37.85	47.35	41.5	34	13	27
<b>2</b>	38.83	<b>0</b>	7.5	23.5	22	65	71	9.68	8.06	10.96	51.6	65.5
<b>3</b>	34.9	7.5	<b>0</b>	12.5	18	59	65	16.6	9.85	8	46.7	60.5
<b>4</b>	23.4	23.5	12.5	<b>0</b>	6	48	54	26.3	23.1	13.6	36	49.64
<b>5</b>	17.9	22	18	6	<b>0</b>	44	50	32.11	25.06	18.8	30.1	43.73
<b>6</b>	28.16	65	59	48	44	<b>0</b>	8	74.5	67.2	60.5	17	8
<b>7</b>	37.85	71	65	54	50	8	<b>0</b>	82.5	73.2	66.48	17.9	8
<b>8</b>	47.35	9.68	16.6	26.3	32.11	74.5	82.5	<b>0</b>	8.75	13.1	60.1	74.1
<b>9</b>	41.5	8.06	9.85	23.1	25.06	67.2	73.2	8.75	<b>0</b>	6.75	52.7	66.75
<b>10</b>	34	10.96	8	13.6	18.8	60.5	66.48	13.1	6.75	<b>0</b>	46	60
<b>11</b>	13	51.6	46.7	36	30.1	17	17.9	60.1	52.7	46	<b>0</b>	16
<b>12</b>	27	65.5	60.5	49.64	43.73	8	8	74.1	66.75	60	16	<b>0</b>
<b>13</b>	37	76.42	71.4	59.53	54.32	12.8	8.94	85.1	76.75	70	25	10
<b>14</b>	49.7	55.82	50.35	41.9	42.76	83.2	89.1	60.2	56.2	48.7	67.7	82.7
<b>15</b>	44.52	50.64	45.17	36.72	37.58	78.02	83.92	55.02	51.02	43.52	65.52	75.52
<b>16</b>	28.06	34.14	28.71	20.26	21.12	61.65	67.56	38.56	34.56	27.06	46.06	61.06
<b>17</b>	28.6	37.72	29.25	20.8	21.7	62.1	68	39.1	35.1	27.6	46.6	64.6
<b>18</b>	66.53	72.65	57.18	58.73	59.6	100.03	105.93	77.03	73.03	65.53	84.53	99.53
<b>19</b>	45.26	51.37	45.9	37.45	38.31	78.75	84.65	50.75	41.75	44.22	63.23	78.25
<b>20</b>	32	38.12	32.65	24.2	25.06	65.5	71.5	42.5	38.5	31	50	65
<b>21</b>	30	36.12	30.65	22.2	23.06	63.5	69.5	40.5	35.5	29	48	63
<b>22</b>	52	58.12	52.65	44.2	45.02	85.5	91.5	62.5	58.5	51	70	85
<b>23</b>	78	84.12	78.65	70.2	71.06	111.5	117.4	88.5	84.5	77	96	111

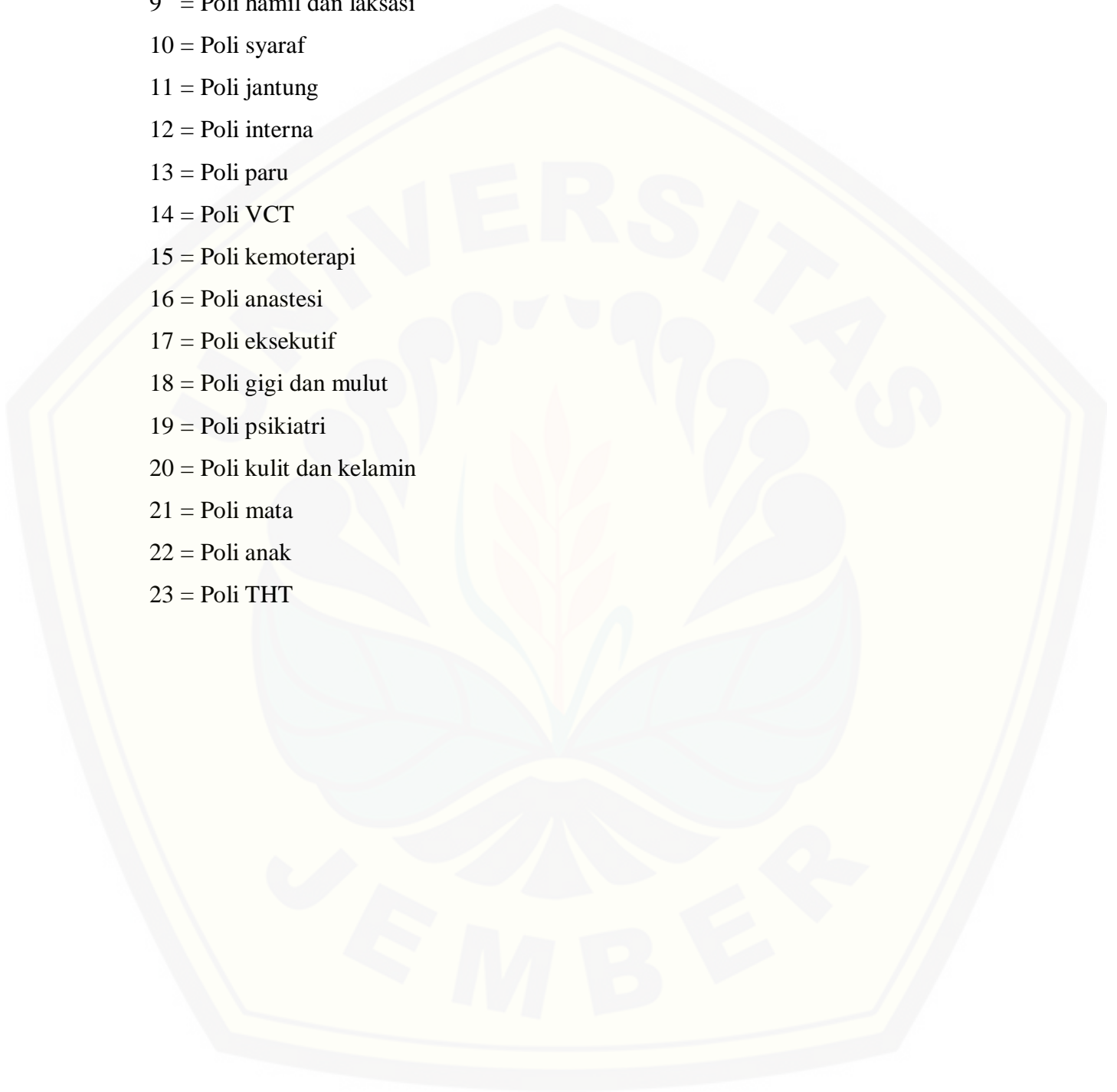
Poli	Tujuan										
	13	14	15	16	17	18	19	20	21	22	23
<b>1</b>	37	49.7	44.52	28.06	28.6	66.53	45.26	32	30	52	78
<b>2</b>	76.42	55.82	50.64	34.14	37.72	72.65	51.37	38.12	36.12	58.12	84.12
<b>3</b>	71.4	50.35	45.17	28.71	29.25	57.18	45.9	32.65	30.65	52.65	78.65
<b>4</b>	59.53	41.9	36.72	20.26	20.8	58.73	37.45	24.2	22.2	44.2	70.2
<b>5</b>	54.32	42.76	37.58	21.12	21.7	59.6	38.31	25.06	23.06	45.02	71.06
<b>6</b>	12.8	83.2	78.02	61.65	62.1	100.03	78.75	65.5	63.5	85.5	111.5
<b>7</b>	8.94	89.1	83.92	67.56	68	105.93	84.65	71.5	69.5	91.5	117.4
<b>8</b>	85.1	60.2	55.02	38.56	39.1	77.03	50.75	42.5	40.5	62.5	88.5
<b>9</b>	76.75	56.2	51.02	34.56	35.1	73.03	41.75	38.5	35.5	58.5	84.5
<b>10</b>	70	48.7	43.52	27.06	27.6	65.53	44.22	31	29	51	77
<b>11</b>	25	67.7	65.52	46.06	46.6	84.53	63.23	50	48	70	96
<b>12</b>	10	82.7	75.52	61.06	64.6	99.53	78.25	65	63	85	111
<b>13</b>	0	92.7	87.52	71.06	71.6	100.53	88.25	75	73	95	121
<b>14</b>	92.7	0	13.5	31	41	75	8.9	17.3	38.09	60	85.6
<b>15</b>	87.52	13.5	0	17.5	27.5	71.5	8.2	12.6	29.1	56.07	80
<b>16</b>	71.06	31	17.5	0	10	54	19.01	8.94	14.4	38.83	64.5
<b>17</b>	71.6	41	27.5	10	0	44	26.5	16.12	8.94	31.04	54.6
<b>18</b>	100.53	75	71.5	54	44	0	69.71	57.6	40.8	11.3	14.4
<b>19</b>	88.25	8.9	8.2	19.01	26.5	69.71	0	13.25	29.25	57.25	81.25
<b>20</b>	75	17.3	12.6	8.94	16.12	57.6	13.25	0	18	44	69
<b>21</b>	73	38.09	29.1	14.4	8.94	40.8	29.25	18	0	28	52
<b>22</b>	95	60	56.07	38.83	31.04	11.3	57.25	44	28	0	24
<b>23</b>	121	85.6	80	64.5	54.6	14.4	81.25	69	52	24	0

Keterangan dari tabel frekuensi rata-rata perpindahan pasien dan jarak Rumah Sakit

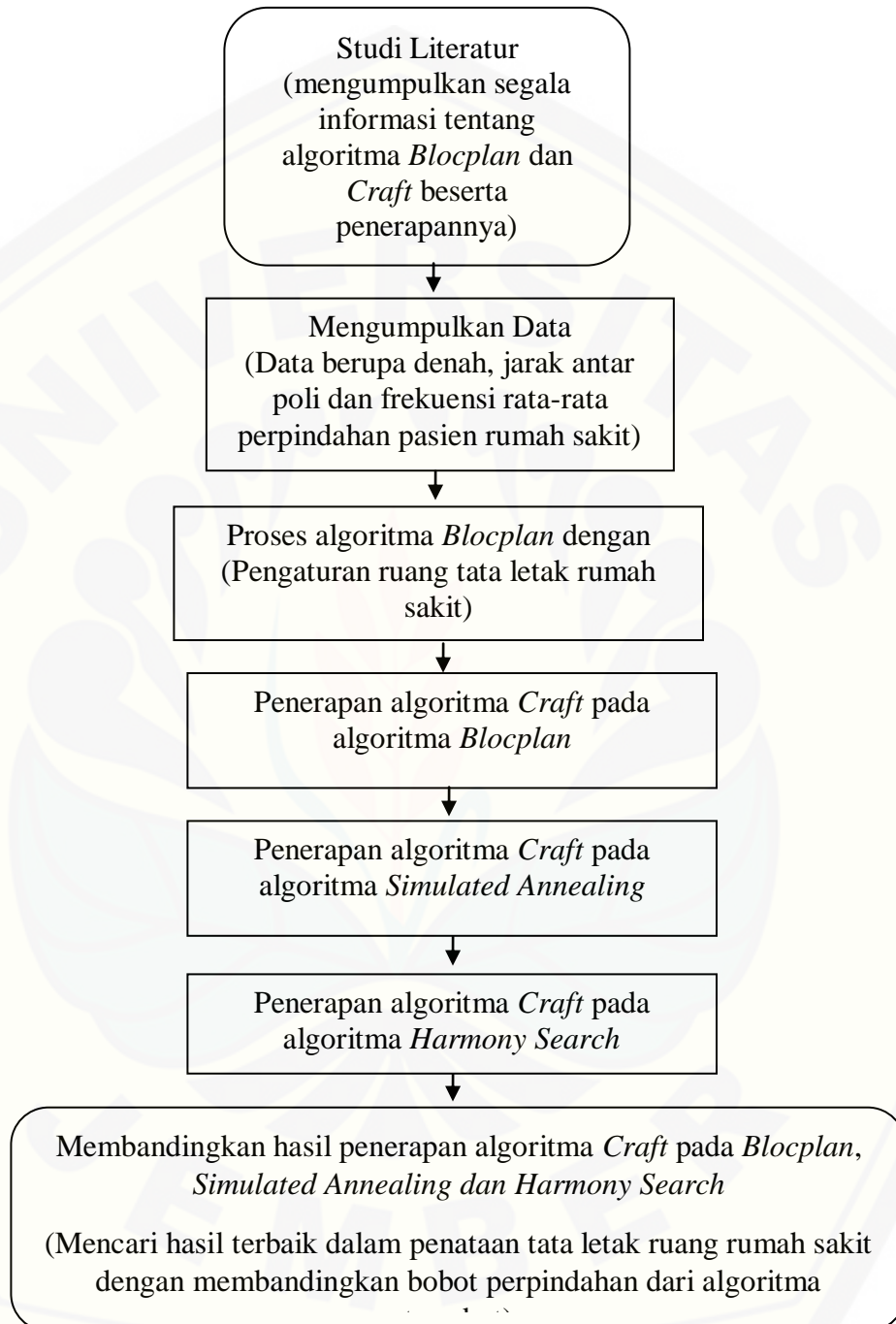
Daerah Dr.Soebandi Jember di atas adalah sebagai berikut.

- 1 = Pintu masuk
- 2 = Poli bedah umum
- 3 = Poli bedah urologi
- 4 = Poli bedah syaraf
- 5 = Poli bedah orthopedik
- 6 = Poli gizi
- 7 = Poli hemodialisa

- 8 = Poli kandungan
- 9 = Poli hamil dan laksasi
- 10 = Poli syaraf
- 11 = Poli jantung
- 12 = Poli interna
- 13 = Poli paru
- 14 = Poli VCT
- 15 = Poli kemoterapi
- 16 = Poli anastesi
- 17 = Poli eksekutif
- 18 = Poli gigi dan mulut
- 19 = Poli psikiatri
- 20 = Poli kulit dan kelamin
- 21 = Poli mata
- 22 = Poli anak
- 23 = Poli THT



### 3.2 Langkah-langkah Penelitian



Gambar 3.2 Skema Langkah-langkah Penelitian



Langkah-langkah yang akan dilakukan dalam menyelesaikan tugas akhir ini adalah sebagai berikut :

a. Studi Literatur

Mengumpulkan literatur tentang algoritma yang diaplikasikan pada perancangan ulang tata letak ruang Rumah Sakit Daerah Dr. Soebandi Jember. Dalam hal ini studi literatur yang dikumpulkan adalah algoritma *Blocplan* dan *Craft*.

b. Pengambilan dan Pengumpulan Data

Seperti yang telah disebutkan di atas, proses pengumpulan data didapat dari peneliti sebelumnya. Data tersebut berupa data dari Rumah Sakit Daerah Dr. Soebandi Jember yang berkaitan dengan jarak antar ruang, denah instalasi rawat jalan serta jumlah pasien yang melakukan kunjungan poli dan konsulan dari satu poli ke poli lainnya.

c. Menerapkan algoritma *Blocplan* dan kemudian menerapkan algoritma *Craft* pada algoritma *Blocplan*, *Craft*, *Harmony Search* dan *Simulated Annealing*.

d. Langkah selanjutnya adalah pembuatan program menggunakan *software* matematika yaitu MATLAB R2009a. Pada langkah ini, penulis akan membuat *script* program dan desain program berdasarkan metode yang digunakan.

e. Dari langkah perhitungan yang telah dilakukan sebelumnya kemudian didapatkan solusi optimal dengan cara menghitung total bobot dari algoritma-algoritma tersebut.

f. Membandingkan hasil dari penerapan algoritma *Craft* pada algoritma *Blocplan*, *Craft*, *Harmony Search* dan *Simulated Annealing*. Dari semua algoritma tersebut dipilih algoritma mana yang mempunyai total bobot perpindahan pasien paling sedikit.

## BAB 4. PEMBAHASAN

### 4.1 Penyelesaian Rancangan Tata Letak Ruang Menggunakan Algoritma *Blocplan*

Terdapat beberapa komponen penting yang diperlukan untuk perancangan tata letak ruang rumah sakit, diantaranya adalah denah rumah sakit, jarak antar poli serta frekuensi perpindahan antara poli yang satu dengan poli yang lain. Perancangan tata letak ruang rumah sakit mempunyai beberapa langkah untuk menyelesaikannya. Langkah paling awal adalah menghitung jarak antar poli dalam rumah sakit tersebut.. Poli Rumah Sakit Daerah Dr. Soebandi Jember yang akan dilakukan penataan yaitu sebanyak 22 poli, akan tetapi dalam proses penyelesaian tata letak ruang menjadi 23 poli, penambahan 1 poli tersebut adalah pintu masuk rumah sakit. Pintu masuk rumah sakit tersebut diasumsikan sebagai ruang atau poli selain 22 poli yang ada di rumah sakit tersebut. Oleh karena Rumah Sakit Daerah Dr. Soebandi Jember memiliki dua lantai dan dua tangga yang menghubungkan lantai satu dan lantai dua maka diasumsikan bahwa setiap pasien yang akan menuju lantai dua melewati satu tangga yang sama. Langkah lain yang harus dilakukan adalah menghitung frekuensi perpindahan dari poli yang satu dengan poli yang lain. Data frekuensi perpindahan pasien yang didapat dari instalasi rawat jalan Rumah Sakit Daerah Dr. Soebandi Jember iniberupa jumlah perpindahan pasiendari atau menuju ke 22 poli tersebut. Data perpindahan pasien tersebut adalah rata-rata perpindahan dari 22 poli selama 6 bulan pada tahun 2014. Oleh karena itu frekuensi perpindahan pasien lebih tepatnya pada kasus ini disebut frekuensi rata-rata perpindahan pasien. Langkah-langkah penyelesaian dengan algoritma *Blocplan* adalah sebagai berikut.

#### 1) Membuat titik awal

langkah awal algoritma *Blocplan* adalah membuat titik awal. Titik awal berlangsungnya suatu kegiatan di rumah sakit terletak pada setiap pintu yang

terdapat pada ruang. Sehingga titik awal yang digunakan pada setiap ruang adalah pada titik pintu ruang tersebut.

## 2) Pengukuran jarak

Pengukuran jarak disini adalah pengukuran jarak antar ruang yang satu ke ruang lain. Pengukuran jarak ditentukan dari pintu masuk antar ruang. Ukuran jarak sudah didapat dari peneliti lain, sehingga untuk pengukuran jarak ini peneliti langsung menggunakan ukuran jarak yang sudah ada tersebut. Jarak antar ruang di instalasi rawat jalan Rumah Sakit Daerah Dr. Soebandi Jember dapat dilihat pada Tabel 3.3.

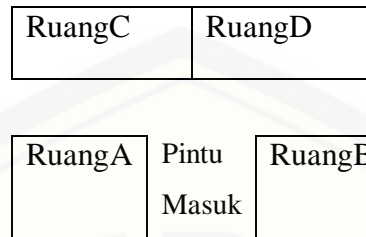
## 3) Penghitungan frekuensi perpindahan pasien

Frekuensi perpindahan pasien merupakan jumlah pasien yang pindah dari ruang satu ke ruang yang lain. Nilai frekuensi perpindahan sudah didapat dari peneliti lain, sehingga untuk penghitungan frekuensi perpindahan ini peneliti langsung menggunakan frekuensi perpindahan yang sudah ada tersebut. Frekuensi perpindahan pasien di instalasi rawat jalan Rumah Sakit Daerah Dr. Soebandi Jember dapat dilihat pada Tabel 3.2.

## 4) Melakukan pengurutan tata letak ruang

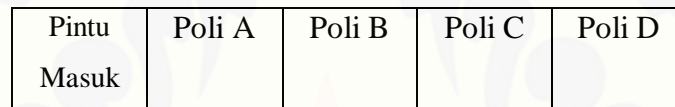
Pada tahap ini yang diperlukan adalah data frekuensi perpindahan pasien dan jarak antar ruang. Pengurutan tata letak ruang pada algoritma *Blocplan* dilakukan dengan cara mengurutkan nilai terbesar dari data frekuensi perpindahan pasien dan jarak antar ruang tersebut hingga mendapatkan nilai yang paling kecil. Kemudian nilai frekuensi perpindahan yang paling besar tersebut ditempatkan pada ruang yang memiliki jarak antarruang paling kecil.

Untuk mengetahui cara kerja algoritma *Blocplan*, berikut ini diberikan contoh penyelesaian tata letak menggunakan sampel data yang kecil berupa denah dengan poli sebanyak 4 ruang seperti pada Gambar 4.1 beserta frekuensi dan jarak antar ruang.



Gambar 4.1 Contoh denah

Jika denah tersebut diasumsikan sebagai denah persegi panjang. Maka denah tersebut menjadi denah seperti Gambar 4.2.



Gambar 4.2 Contoh denah polipersegi panjang

Pintu masuk dianggap sebagai ruang maka pada contoh denah ruang tersebut pintu masuk menjadi ruang (1), Ruang A menjadi ruang (2), ruang B menjadi ruang (3), ruang C menjadi ruang (4) dan ruang D menjadi ruang (5). Sehingga urutan ruang menjadi (1), (2), (3), (4), (5). Data frekuensi perpindahan dan jarak antar ruang telah diurutkan dari yang terbesar hingga yang terkecil seperti berikut.

Tabel 4.1 Frekuensi perpindahan antar poli dari yang terbesar ke terkecil

No	Poli	Frekuensi
1	B – D	15
2	A – D	13
3	B – C	10
4	A – B	5
5	A – C	3
6	C – D	2

Tabel 4.2 jarak antarruang ke ruang lain dari terkecil ke terbesar

No	Ruang	jarak
1	2 – 4	2
2	4 – 5	3
3	2 – 3	4
4	2 – 5	5
5	3 – 5	6
6	3 – 4	7

Dari tabel frekuensi perpindahan dan jarak tersebut ruang yang memiliki frekuensi perpindahan terbesar menempati ruang yang memiliki perpindahan jarak yang terkecil. Sehingga denah menjadi sebagai berikut.



Gambar 4.3 Contoh denah baru

Jika diurutkan menjadi denah persegi panjang urutan ruang menjadi (1), (3), (4), (5), (2) dimana urutan tersebut adalah, pintu masuk, ruang B, ruang C, ruang D, kemudian ruang A.

Untuk tata letak rumah sakit instalasi rawat jalan Rumah Sakit Daerah Dr. Soebandi Jember, pengurutan tata letak ruang yang dilakukan sama dengan langkah-langkah di atas hanya saja menggunakan data frekuensi perpindahan pada tabel 3.2 dan jarak pada tabel 3.3. Hasil yang didapat seperti hasil dari program MATLAB R2009a algoritma *Blocplan* sebagai berikut.





Gambar 4.4 Tampilan program algoritma *Bloclan*

Dari tampilan program di atas dapat dilihat adanya perubahan pengurutan ruang. Urutan awal ruang (1), (2), (3), (4), (5), (6), (7), (8), (9), (10), (11), (12), (13), (14), (15), (16), (17), (18), (19), (20), (21), (22), (23). Setelah dilakukan pengurutan berdasarkan algoritma *Bloclan* di atas didapat urutan tata letak ruang rumah sakit (1), (10), (20), (2), (11), (5), (22), (18), (4), (14), (9), (23), (8), (17), (21), (12), (16), (6), (19), (3), (7), (13), (15). Pertukaran ruang yang dilakukan dapat dilihat pada tabel 4.2.

Tabel 4.3 Pertukaran urutan tata letak awal dan urutan tata letak algoritma *Blocplan*

No	Pertukaran yang dilakukan	Keterangan
1	(2) dengan (10)	ruang (10) berubah menjadi ruang (2)
2	(3) dengan (20)	ruang (20) berubah menjadi ruang (3)
3	(4) dengan (2)	ruang (2) berubah menjadi ruang (4)
4	(5) dengan (11)	ruang (11) berubah menjadi ruang (5)
5	(6) dengan (5)	ruang (5) berubah menjadi ruang (6)
6	(7) dengan (22)	ruang (22) berubah menjadi ruang (7)
7	(8) dengan (18)	ruang (18) berubah menjadi ruang (8)
8	(9) dengan (4)	ruang (4) berubah menjadi ruang (9)
9	(10) dengan (14)	ruang (14) berubah menjadi ruang (10)
10	(11) dengan (9)	ruang (9) berubah menjadi ruang (11)
11	(12) dengan (23)	ruang (23) berubah menjadi ruang (12)
12	(13) dengan (8)	ruang (8) berubah menjadi ruang (13)
13	(14) dengan (17)	ruang (17) berubah menjadi ruang (14)
14	(15) dengan (21)	ruang (21) berubah menjadi ruang (15)
15	(16) dengan (12)	ruang (12) berubah menjadi ruang (16)
16	(17) dengan (16)	ruang (16) berubah menjadi ruang (17)
17	(18) dengan (6)	ruang (6) berubah menjadi ruang (18)
18	(19) dengan (19)	ruang (19) berubah menjadi ruang (19)
19	(20) dengan (3)	ruang (3) berubah menjadi ruang (20)
20	(21) dengan (7)	ruang (7) berubah menjadi ruang (21)
21	(22) dengan (13)	ruang (13) berubah menjadi ruang (22)
22	(23) dengan (15)	ruang (15) berubah menjadi ruang (23)

#### 4.2 Penerapan algoritma *Craft* Pada Perancangan Tata Letak Ruang Berdasarkan Algoritma *Blocplan*, *Simulated Annealing* dan *Harmony Search*

Setelah mendapatkan hasil urutan tata letak ruang dari algoritma lain, maka dari hasil tersebut dilakukan pengurutan kembali dengan algoritma *Craft*. Input yang digunakan dalam algoritma *craft* adalah

a. Denah awal

Denah awal dapat berupa tata letak yang sudah ada ataupun denah awal dari hasil rancangan baru. Pada tata letak ruang rumah sakit ini menggunakan urutan tata letak hasil dari algoritma lain yang sudah telah didapatkan sebelumnya.

b. Pengukuran jarak

Pengukuran jarak pada algoritma *Craft* ditentukan dari pintu masuk antar ruang. Ukuran jarak sudah didapat dari peneliti lain, sehingga untuk pengukuran jarak ini peneliti langsung menggunakan ukuran jarak yang sudah ada tersebut.

c. Melakukan pengurutan tata letak ruang

Oleh karena algoritma *Craft* memerlukan denah dari algoritma lain, maka pada perancangan tata letak rumah sakit ini menggunakan urutan tata letak ruang hasil dari algoritma *Blocplan*, *Simulated Annealing* dan *Harmony Search*. Seperti yang telah disebutkan sebelumnya, pengurutan tata letak ruang pada algoritma *Craft* diberikan beberapa syarat. Syarat tersebut antara lain :

- 1) Harus mempunyai ukuran yang sama
- 2) Harus mempunyai perbatasan yang sama
- 3) Harus mempunyai kedua perbatasan-perbatasan yang sama pada ketiga departemen.

d. Menghitung total bobot

Bobot pasien dihitung berdasarkan perkalian antara frekuensi perpindahan dan jarak antar ruang (Persamaan 2.4). Data frekuensi perpindahan dapat dilihat pada Tabel 4.1 sedangkan data jarak dapat dilihat pada Tabel 3.3. Total bobot

perpindahan adalah jumlah dari bobot-bobot perpindahan tersebut. Kemudian membandingkan total bobot dari masing-masing urutan, nilai total bobot yang paling rendah merupakan urutan tata letak yang paling baik untuk digunakan.

Dari hasil urutan contoh denah perhitungan algoritma *Blocplan* pada Gambar 4.3 dimana didapat urutan (1), (3), (4), (5), (2). Maka penerapan algoritma *Craft* berdasarkan algoritma *Blocplan* adalah sebagai berikut.

Tabel 4.4 Contoh Kemungkinan pertukaran urutan tata ruang algoritma *Craft* berdasarkan algoritma *Blocplan*

Kemungkinan	Urutan	Posisi Ruang yang dirubah	Total Bobot
1	(3), (1), (4), (5), (2)	(3) dengan (1)	-
2	(1), (4), (3), (5), (2)	(3) dengan (4)	163
3	(1), (4), (3), (2), (5)	(3) dengan (4), (5) dengan (2)	176
4	dst		

Dari tabel di atas dapat dilihat bahwa untuk kemungkinan (1) tidak dapat dijadikan solusi sebagai urutan tata letak ruang yang baru, hal ini karena kemungkinan (1) memposisikan pintu masuk tidak berada pada urutan pertama. Kemungkinan (1) dipastikan tidak dipakai sehingga tidak perlu mencari total bobot. Kemudian dari urutan tata letak ruang tersebut diambil total bobot mana yang paling rendah.

e. Membuat denah baru

Berdasarkan hasil dari pengurutan algoritma *Craft* pada tata letak rumah sakit berdasarkan algoritma *Blocplan*, *Simulated Annealing* dan *Harmony Search*, maka langkah terakhir yaitu membuat denah dari hasil pengurutan tersebut.

#### 4.2.1 Penerapan algoritma *Craft* Pada Perancangan Tata Letak Ruang Berdasarkan Algoritma *Blocplan*

Sebelumnya telah didapat urutan dari algoritma *Blocplan* yaitu adalah (1), (10), (20), (2), (11), (5), (22), (18), (4), (14), (9), (23), (8), (17), (21), (12), (16), (6), (19), (3), (7), (13), (15). Dimana urutan tersebut terdiri dari 23 ruang dan 22 poli. Urutan tersebut yaitu ruang 1 = pintu masuk, ruang 2 = poli syaraf, ruang 3 = poli kulit dan kelamin, ruang 4 = poli bedah umum, ruang 5 = poli jantung, ruang 6 = poli bedah orthopedik, ruang 7 = poli anak, ruang 8 = poli gigi dan mulut, ruang 9 = poli bedah syaraf, ruang 10 = poli VCT, ruang 11 = poli hamil dan laktasi, ruang 12 = poli THT, ruang 13 = poli kandungan, ruang 14 = poli eksekutif, ruang 15 = poli mata, ruang 16 = poli interna, ruang 17 = poli anastesi, ruang 18 = poli gizi, ruang 19 = poli psikiatri, ruang 20 = poli bedah urologi, ruang 21 = poli hemodialisa, ruang 22 = poli paru, ruang 23 = poli kemoterapi. Berdasarkan Gambar 3.1 terdapat kemungkinan pertukaran tata letak ruang algoritma *Craft* berdasarkan algoritma *Blocplan* sebagai berikut.

Tabel 4.5 Kemungkinan pertukaran urutan tata letak ruang algoritma *Craft* berdasarkan algoritma *Blocplan*

Kemungkinan	Urutan	Posisi Ruang yang dirubah	Total Bobot
1	(10), (1), (20), (2), (11), (5), (22), (18), (4), (14), (9), (23), (8), (17), (21), (12), (16), (6), (19), (3), (7), (13), (15)	(10) dengan (1)	-
	(1), (20), (10), (2), (11), (5), (22), (18), (4), (14), (9), (23), (8), (17), (21), (12), (16), (6), (19), (3), (7), (13), (15)	(10) dengan (20)	102694,2964



Kemungkinan	Urutan	Posisi Ruang yang dirubah	Total Bobot
3	(1), (20), (10), (2), (11), (5), (22), (4), (18), (14), (9), (23), (8), (17), (21), (12), (16), (6), (19), (3), (7), (13), (15)	(10) dengan (20), (18) dengan (4)	102305,5673
4	(1), (20), (10), (2), (11), (5), (22), (4), (18), (14), (23), (9), (8), (17), (21), (12), (16), (6), (19), (3), (7), (13), (15)	(10) dengan (20), (18) dengan (4), (9) dengan (23)	99556,058
5	dst		

Dari tabel di atas dapat dilihat bahwa untuk kemungkinan (1) tidak dapat dijadikan solusi sebagai urutan tata letak ruang yang baru, hal ini karena kemungkinan (1) memposisikan pintu masuk tidak berada pada urutan pertama. Kemungkinan (1) dipastikan tidak dipakai sehingga tidak perlu mencari total bobot. Kemudian dari urutan tata letak ruang tersebut diambil total bobot mana yang paling rendah. Total bobot perpindahan yang paling rendah itulah yang menjadi solusi terbaik algoritma *Craft*. Hasil dari penerapan algoritma *Craft* pada tata letak ruang berdasarkan Algoritma *Blocplan* dapat dilihat pada hasil perhitungan dengan program MATLAB R2009a pada Gambar 4.4.



Gambar 4.5 Tampilan program penerapan algoritma *Craft* pada tata letak ruang berdasarkan Algoritma *Blocplan*

Dari hasil perhitungan dengan program didapatkan urutan tata letak ruang yang baru yaitu (1), (20), (10), (2), (11), (5), (22), (4), (18), (14), (23), (9), (8), (17), (21), (12), (16), (6), (19), (3), (7), (13), (15) yang merupakan ruang 1 = pintu masuk, ruang 2 = poli kulit dan kelamin, ruang 3 = poli syaraf, ruang 4 = poli bedah umum, ruang 5 = poli jantung, ruang 6 = poli bedah orthopedik, ruang 7 = poli anak, ruang 8 = poli bedah syaraf, ruang 9 = poli gigi dan mulut, ruang 10 = poli VCT, ruang 11 = poli THT, ruang 12 = poli hamil dan laktasi, ruang 13 = poli kandungan, ruang 14 = poli eksekutif, ruang 15 = poli mata, ruang 16 = poli interna, ruang 17 = poli anastesi, ruang 18 = poli gizi, ruang 19 = poli psikiatri, ruang 20 = poli bedah urologi, ruang

21 = poli hemodialisa, ruang 22 = poli paru, ruang 23 = poli kemoterapi dengan total bobot 100142,18.

#### 4.2.2. Penerapan algoritma *Craft* Pada Perancangan Tata Letak Ruang Berdasarkan Algoritma *Simulated Annealing*

Sebelumnya telah didapat urutan dari algoritma *Simulated Annealing* yaitu adalah (1), (21), (8), (15), (5), (4), (9), (20), (2), (19), (13), (6), (10), (7), (18), (12), (17), (3), (11), (22), (23), (16), (14). Dimana urutan tersebut terdiri dari 23 ruang dan 22 poli. Urutan tersebut yaitu ruang 1 = pintu masuk, ruang 2 = poli mata, ruang 3 = poli kandungan, ruang 4 = poli kemoterapi, ruang 5 = poli bedah orthopedik, ruang 6 = poli bedah saraf, ruang 7 = poli hamil dan laktasi, ruang 8 = poli kulit dan kelamin, ruang 9 = poli bedah umum, ruang 10 = poli psikiatri, ruang 11 = poli paru, ruang 12 = poli gizi, ruang 13 = poli syaraf, ruang 14 = poli hemodialisa, ruang 15 = poli gigi dan mulut, ruang 16 = poli interna, ruang 17 = poli eksekutif, ruang 18 = poli bedah urologi, ruang 19 = poli jantung, ruang 20 = poli anak, ruang 21 = poli THT, ruang 22 = poli anastesi, ruang 23 = poli VCT. Penerapan algoritma *craft* pada tata letak ruang berdasarkan algoritma *Simulated Annealing* terdapat beberapa kemungkinan. Diantaranya adalah sebagai berikut.

Tabel 4.6 Kemungkinan pertukaran urutan tata letak ruang algoritma *Craft* berdasarkan algoritma *Simulated Annealing*

Kemungkinan	Urutan	Posisi Ruang yang dirubah	Total Bobot
1	(21), (1), (8), (15), (5), (4), (9), (20), (2), (19), (13), (6), (10), (7), (18), (12), (17), (3), (11), (22), (23), (16), (14).	(21) dengan (1)	-

Kemungkinan	Urutan	Posisi Ruang yang dirubah	Total Bobot
2	(1), (8), (21), (15), (5), (4), (9), (20), (2), (19), (13), (6), (10), (7), (18), (12), (17), (3), (11), (22), (23), (16), (14)	(21) dengan (8)	116938,8301
3	(1), (8), (21), (15), (5), (4), (9), (2), (20), (19), (13), (6), (10), (7), (18), (12), (17), (3), (11), (22), (23), (16), (14)	(21) dengan (8), (20) dengan (2)	117921,6514
4	(1), (8), (21), (15), (5), (4), (9), (2), (20), (19), (6), (13), (10), (7), (18), (12), (17), (3), (11), (22), (23), (16), (14)	(21) dengan (8), (20) dengan (2), (13) dengan (6)	118414,6545
5			

Hasil dari penerapan algoritma *Craft* pada perancangantata letak ruang berdasarkan Algoritma *Simulated Annealing* dapat dilihat pada hasil perhitungan dengan program MATLAB R2009a pada Gambar 4.5.



Gambar 4.6 Tampilan program penerapan algoritma *Craft* pada tata letak ruang berdasarkan Algoritma *Simulated Annealing*

Dari hasil perhitungan dengan program didapatkan urutan tata letak ruang yang baru yaitu (1), (14), (3), (6), (20), (4), (23), (22), (17), (19), (21), (11), (12), (16), (8), (9), (18), (7), (5), (2), (10), (13), (15) yang merupakan ruang 1 = pintu masuk, ruang 2 = poli VCT, ruang 3 = poli bedah urologi, ruang 4 = poli gizi, ruang 5 = poli kulit dan kelamin, ruang 6 = poli bedah syaraf, ruang 7 = poli THT, ruang 8 = poli anak, ruang 9 = poli eksekutif, ruang 10 = poli psikiatri, ruang 11 = poli mata, ruang 12 = poli jantung, ruang 13 = poli interna, ruang 14 = poli anastesi, ruang 15 = poli kandungan, ruang 16 = poli hamil dan laktasi, ruang 17 = poli gigi dan mulut, ruang 18 = poli hemodialisa, ruang 19 = poli bedah orthopedik, ruang 20 = poli bedah umum, ruang 21 = poli syaraf, ruang 22 = poli paru, ruang 23 = poli kemoterapi dengan total bobot 105553,9.



#### 4.2.3 Penerapan algoritma *Craft* Pada Perancangan Tata Letak Ruang Berdasarkan Algoritma *Harmony Search*

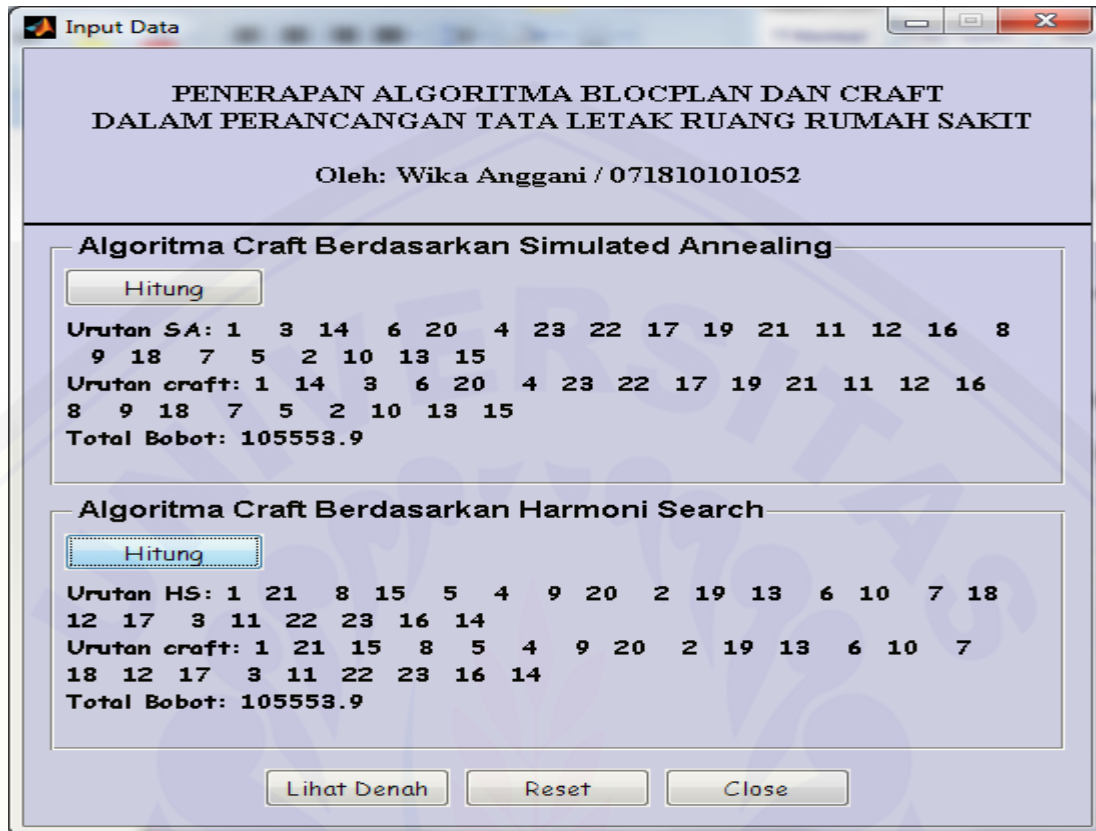
Sebelumnya telah didapat urutan dari algoritma *Harmony Search* yaitu adalah (1), (3), (14), (6), (20), (4), (23), (22), (17), (19), (21), (11), (12), (16), (8), (9), (18), (7), (5), (2), (10), (13), (15). Dimana urutan tersebut terdiri dari 23 ruang dan 22 poli. Urutan tersebut yaitu ruang 1 = pintu masuk, ruang 2 = poli bedah urologi, ruang 3 = poli VCT, ruang 4 = poli gizi, ruang 5 = poli kulit dan kelamin, ruang 6 = poli bedah syaraf, ruang 7 = poli THT, ruang 8 = poli anak, ruang 9 = poli eksekutif, ruang 10 = poli psikiatri, ruang 11 = poli mata, ruang 12 = poli jantung, ruang 13 = poli interna, ruang 14 = poli anastesi, ruang 15 = poli kandungan, ruang 16 = poli hamil dan laktasi, ruang 17 = poli gigi dan mulut, ruang 18 = poli hemodialisa, ruang 19 = poli bedah orthopedik, ruang 20 = poli bedah umum, ruang 21 = poli syaraf, ruang 22 = poli paru, ruang 23 = poli kemoterapi. Penerapan algoritma *craft* pada tata letak ruang berdasarkan algoritma *Harmony Search* terdapat beberapa kemungkinan. Diantaranya adalah sebagai berikut.

Tabel 4.7 Kemungkinan pertukaran urutan tata letak ruang algoritma *Craft* berdasarkan algoritma *Harmony Search*

Kemungkinan	Urutan	Posisi Ruang yang dirubah	Total Bobot
1	(3), (1), (14), (6), (20), (4), (23), (22), (17), (19), (21), (11), (12), (16), (8), (9), (18), (7), (5), (2), (10), (13), (15)	(3) dengan (1)	-
2	(1), (14), (3), (6), (20), (4), (23), (22), (17), (19), (21), (11), (12), (16), (8), (9), (18), (7), (5), (2), (10), (13), (15)	(3) dengan (14)	104910.6522

Kemungkinan	Urutan	Posisi Ruang yang dirubah	Total Bobot
3	(1), (14), (3), (6), (20), (4), (23), (17), (22), (19), (21), (11), (12), (16), (8), (9), (18), (7), (5), (2), (10), (13), (15)	(3) dengan (4), (22) dengan (17)	106448,2641
4	(1), (14), (3), (6), (20), (4), (23), (17), (22), (19), (11), (21), (12), (16), (8), (9), (18), (7), (5), (2), (10), (13), (15)	(3) dengan (4), (22) dengan (17), (21) dengan (11)	107423,1531
5	dst		

Hasil dari penerapan algoritma *Craft* pada tata letak ruang berdasarkan Algoritma *Harmony Search* dapat dilihat pada hasil perhitungan dengan program MATLAB R2009a pada Gambar 4.6



Gambar 4.7 Tampilan program penerapan algoritma *Craft* pada tata letak ruang berdasarkan Algoritma *Harmony Search*

- a. Dari hasil perhitungan dengan program didapatkan urutan tata letak ruang yang baru yaitu (1), (21), (15), (8), (5), (4), (9), (20), (2), (19), (13), (6), (10), (7), (18), (12), (17), (3), (11), (22), (23), (16), (14) yang merupakan ruang 1 = pintu masuk, ruang 2 = poli mata, ruang 3 = poli kemoterapi, ruang 4 = poli kandungan, ruang 5 = poli bedah orthopedik, ruang 6 = poli bedah syaraf, ruang 7 = poli hamil dan laktasi, ruang 8 = poli kulit dan kelamin, ruang 9 = poli bedah umum, ruang 10 = poli psikiatri, ruang 11 = poli paru, ruang 12 = poli gizi, ruang 13 = poli syaraf, ruang 14 = poli hemodialisa, ruang 15 = poli gigi dan mulut, ruang 16 = poli interna, ruang 17 = poli eksekutif, ruang 18 = poli bedah urologi, ruang 19 = poli

jantung, ruang 20 = poli anak, ruang 21 = poli THT, ruang = 22 poli anastesi, ruang 23 = poli VCT dengan total bobot 105553,9.

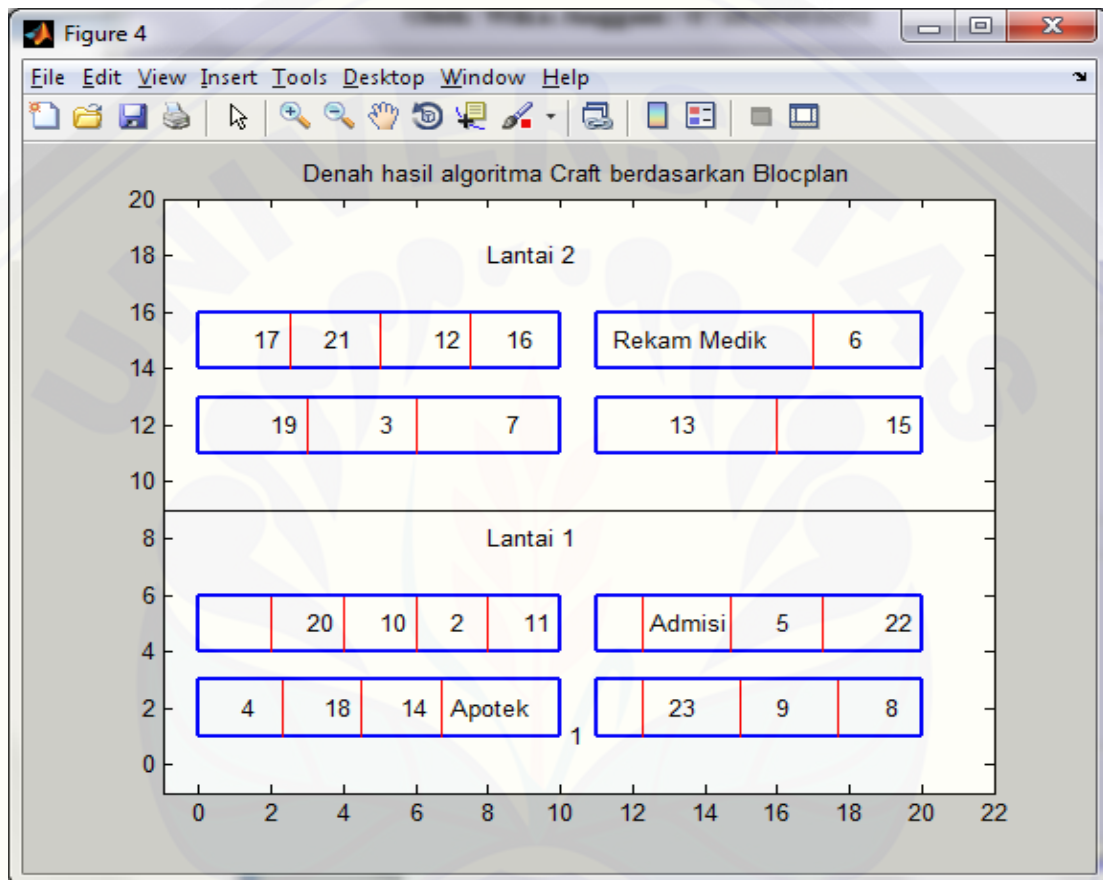
#### 4.2.4 Perbandingan Penerapan Algoritma *Craft* Pada Perancangan Tata Letak Ruang Berdasarkan Algoritma *Blocplan*, *Simulated Annealing* dan *Harmony Search*

Dari hasil total bobot yang didapat dari masing-masing algoritma di atas, maka dapat diketahui algoritma mana yang paling baik untuk penerapan algoritma *Craft*. Telah diketahui bahwa total bobot penerapan algoritma *Craft* pada tata letak ruang berdasarkan algoritma *Blocplan* adalah 100142,18, serta berdasarkan algoritma *Simulated Annealing* dan *Harmony Search* yang sama-sama mempunyai total bobot sebesar 105553,9. Sehingga dapat dikatakan bahwa penerapan algoritma *Craft* pada tata letak ruang berdasarkan algoritma *Blocplan* lebih baik dari pada penerapan algoritma *Craft* pada tata letak ruang berdasarkan algoritma *Simulated Annealing* dan *Harmony Search*.

#### 4.2.5 Denah Baru Penerapan Algoritma *Craft* Pada Perancangan Tata Letak Ruang

Berdasarkan Algoritma *Blocplan*, *Simulated Annealing* dan *Harmony Search*

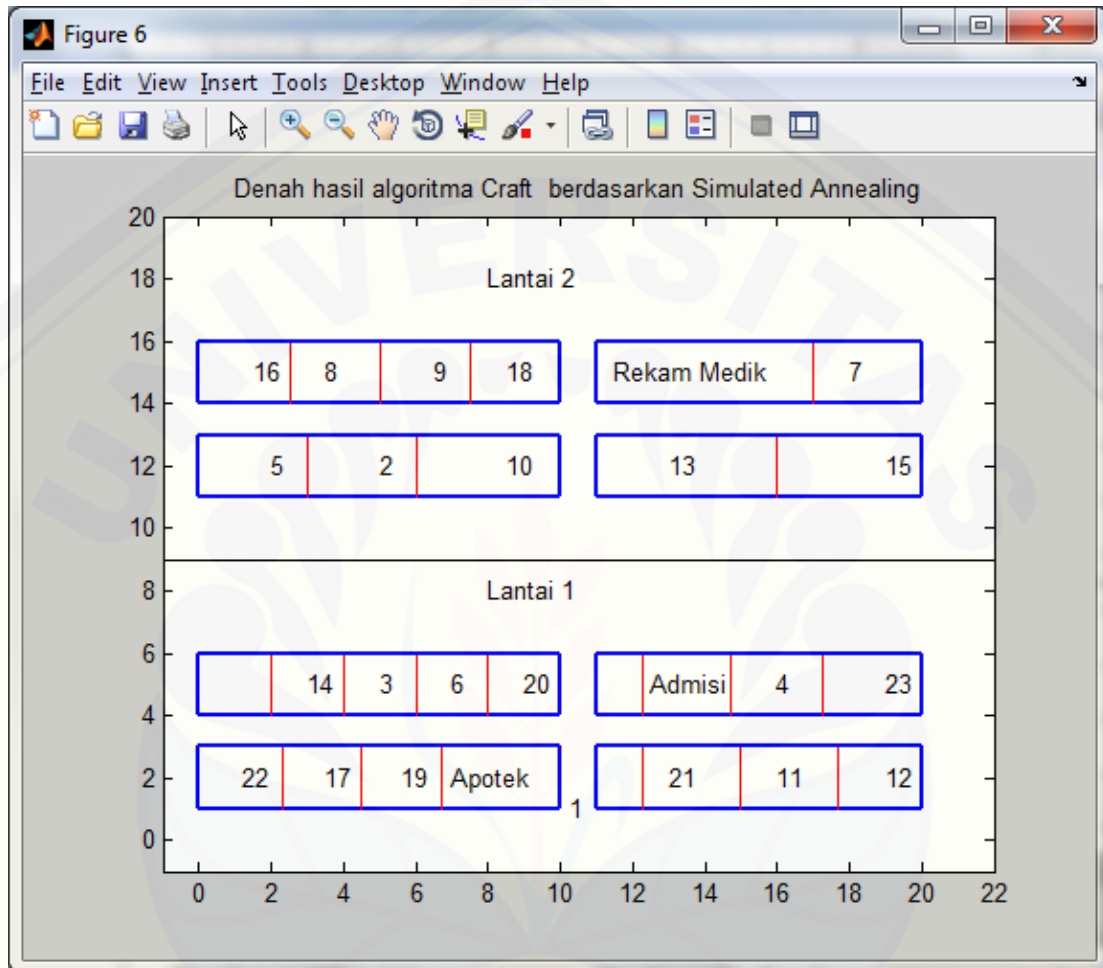
- a. Denah baru penerapan algoritma *Craft* pada perancangan tata letak ruang berdasarkan algoritma *Blocplan*



Gambar 4.8 Denah baru penerapan algoritma *Craft* pada perancangan tata letak ruang berdasarkan Algoritma *Blocplan*

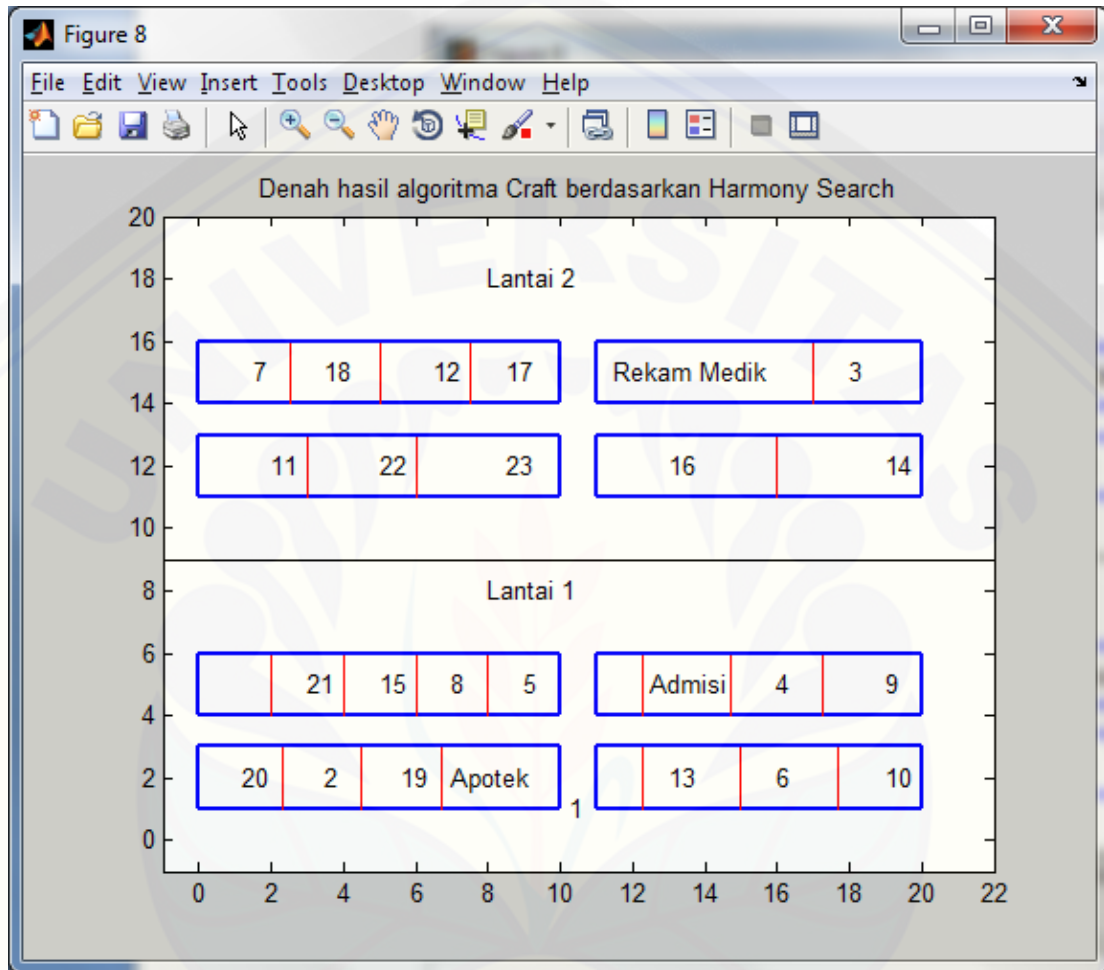


- b. Denah baru penerapan algoritma *Craft* padaperancangan tata letak ruang berdasarkan algoritma *Simulated Annealing*



Gambar 4.9 Denah baru penerapan algoritma *Craft* padaperancangan tata letak ruang berdasarkan Algoritma *Simulated Annealing*

- c. Denah baru penerapan algoritma *Craft* pada perancangan tata letak ruang berdasarkan algoritma *Simulated Annealing*



Gambar 4.10 Denah baru penerapan algoritma *Craft* perancangan tata letak ruang berdasarkan Algoritma *Harmony Search*

## BAB 5. PENUTUP

### 5.1 Kesimpulan

Dari hasil pembahasan pada bab sebelumnya, didapatkan hasil tata letak ruang Rumah Sakit Daerah Dr.Soebandi sebagai berikut.

- a. Penataan tata letak ruang dengan algoritma *Blocplan* didapatkan urutan tata letak (1), (10), (20), (2), (11), (5), (22), (18), (4), (14), (9), (23), (8), (17), (21), (12), (16), (6), (19), (3), (7), (13), (15). Dimana urutan tersebut terdiri dari 23 ruang dan 22 poli. Urutan tersebut yaitu ruang 1 = pintu masuk, ruang 2 = poli syaraf, ruang 3 = poli kulit dan kelamin, ruang 4 = poli bedah umum, ruang 5 = poli jantung, ruang 6 = poli bedah orthopedik, ruang 7 = poli anak, ruang 8 = poli gigi dan mulut, ruang 9 = poli bedah syaraf, ruang 10 = poli VCT, ruang 11 = poli hamil dan laktasi, ruang 12 = poli THT, ruang 13 = poli kandungan, ruang 14 = poli eksekutif, ruang 15 = poli mata, ruang 16 = poli interna, ruang 17 = poli anastesi, ruang 18 = poli gizi, ruang 19 = poli psikiatri, ruang 20 = poli bedah urologi, ruang 21 = poli hemodialisa, ruang 22 = poli paru, ruang 23 = poli kemoterapi.
- b. Penerapan algoritma *Craft* pada perancangan tata letak ruang rumah sakit berdasarkan algoritma *Blocplan* mempunyai urutan (1), (20), (10), (2), (11), (5), (22), (4), (18), (14), (23), (9), (8), (17), (21), (12), (16), (6), (19), (3), (7), (13), (15) yang merupakan ruang 1 = pintu masuk, ruang 2 = poli kulit dan kelamin, ruang 3 = poli syaraf, ruang 4 = poli bedah umum, ruang 5 = poli jantung, ruang 6 = poli bedah orthopedik, ruang 7 = poli anak, ruang 8 = poli bedah syaraf, ruang 9 = poli gigi dan mulut, ruang 10 = poli VCT, ruang 11 = poli THT, ruang 12 = poli hamil dan laktasi, ruang 13 = poli kandungan, ruang 14 = poli eksekutif, ruang 15 = poli mata, ruang 16 = poli interna, ruang 17 = poli anastesi, ruang 18 = poli gizi, ruang 19 = poli psikiatri, ruang 20 = poli bedah urologi, ruang 21 = poli hemodialisa, ruang 22 = poli paru, ruang 23 = poli kemoterapi dengan total bobot 100142,18. Sedangkan penerapan algoritma *Craft* pada perancangan tata letak

ruang rumah sakit berdasarkan algoritma *Simulated Annealing* mempunyai urutan (1), (14), (3), (6), (20), (4), (23), (22), (17), (19), (21), (11), (12), (16), (8), (9), (18), (7), (5), (2), (10), (13), (15) yang merupakan ruang 1 = pintu masuk, ruang 2 = poli VCT, ruang 3 = poli bedah urologi, ruang 4 = poli gizi, ruang 5 = poli kulit dan kelamin, ruang 6 = poli bedah syaraf, ruang 7 = poli THT, ruang 8 = poli anak, ruang 9 = poli eksekutif, ruang 10 = poli psikiatri, ruang 11 = poli mata, ruang 12 = poli jantung, ruang 13 = poli interna, ruang 14 = poli anastesi, ruang 15 = poli kandungan, ruang 16 = poli hamil dan laktasi, ruang 17 = poli gigi dan mulut, ruang 18 = poli hemodialisa, ruang 19 = poli bedah orthopedik, ruang 20 = poli bedah umum, ruang 21 = poli syaraf, ruang 22 = poli paru, ruang 23 = poli kemoterapi dengan total bobot 105553,9. Kemudian untuk penerapan algoritma *Craft* pada perancangan tata letak ruang rumah sakit berdasarkan algoritma *Harmony Search* mempunyai urutan (1), (21), (15), (8), (5), (4), (9), (20), (2), (19), (13), (6), (10), (7), (18), (12), (17), (3), (11), (22), (23), (16), (14) yang merupakan ruang 1 = pintu masuk, ruang 2 = poli mata, ruang 3 = poli kemoterapi, ruang 4 = poli kandungan, ruang 5 = poli bedah orthopedik, ruang 6 = poli bedah syaraf, ruang 7 = poli hamil dan laktasi, ruang 8 = poli kulit dan kelamin, ruang 9 = poli bedah umum, ruang 10 = poli psikiatri, ruang 11 = poli paru, ruang 12 = poli gizi, ruang 13 = poli syaraf, ruang 14 = poli hemodialisa, ruang 15 = poli gigi dan mulut, ruang 16 = poli interna, ruang 17 = poli eksekutif, ruang 18 = poli bedah urologi, ruang 19 = poli jantung, ruang 20 = poli anak, ruang 21 = poli THT, ruang = 22 poli anastesi, ruang 23 = poli VCT dengan total bobot 105553,9.

- c. Dari hasil total bobot tersebut, dapat dilihat bahwa penerapan algoritma *Craft* pada perancangan tata letak ruang rumah sakit berdasarkan algoritma *Blocplan* lebih baik dari pada penerapan algoritma *Craft* pada perancangan tata letak ruang rumah sakit berdasarkan algoritma *Simulated Annealing* dan *Harmony Search*.

## 5.2 Saran

Perancangan tata letak ruang rumah sakit menggunakan algoritma *Blocplan* dan *Craft* dapat dikembangkan lebih jauh lagi. Dengan menggunakan program lain atau dengan menggunakan algoritma lain yang fungsinya untuk penataan tata letak ruang, baik untuk rumah sakit atau untuk fasilitas yang lain.





## LAMPIRAN

### Script Program

#### A. Riset

```
clc; % membersihkan command window
clear all;% membersihkan semua data
close all; % menutup semua figure

set(0,'Units','points')
Screen = get(0,'screensize'); % mengambil ukuran layar Laptop
pos=[0 0 400 380+44]; % ukuran gui panjang=400 x lebar=380+36

data1=[];% inialisai data
data2=[];% idem

win1=figure(...
'units','points',...
'position',[Screen(3:4)/2-pos(3:4)/2 pos(3:4)],...
'color',[.8 .8 .9],...
'resize','off',...
'menubar','none',...
'toolbar','none',...
'numbertitle','off',...
'name','Input Data');
% posisi:[dari_kiri dari_bawah panjang lebar]
%color:[merah hijau biru] ; interval:0-1
%=====
labell=uicontrol('parent',win1,...
'units','points',...
'position',[10 310+36 380 75],...
'style','text',...
'string',{'';'PENERAPAN ALGORITMA BLOCPAN DAN CRAFT';' DALAM
PERANCANGAN TATA LETAK RUANG RUMAH SAKIT';'';'Oleh: Wika Anggani /
071810101052'},...
'fontname','times new roman','BackgroundColor',[.8 .8 .9],...
'fontsize',12,'fontweight','bold');

labell=uicontrol('parent',win1,...
'units','points',...
'position',[0 305+36 400 1],...
'style','text',...
'fontname','times new roman','BackgroundColor',[0 0 0],...
'fontsize',12,'fontweight','bold');

%=====
hp0 = uipanel('parent',win1,...
'Title','Input data','FontSize',12,...
'units','points',...
'fontweight','bold',...
'BackgroundColor',[.8 .8 .9],...
```

```
        'Position',[10 205+24 380 95+12]);

proses1=uicontrol('parent',win1,...
    'units','points',...
    'position',[15 265+36 75 15],...
    'style','Pushbutton',...
    'callback','openfilejarak',...
    'string','File Jarak: ',...
    'fontname','comic',...
    'fontsize',10);
label1=uicontrol('parent',win1,...
    'units','points',...
    'position',[90 265+36 295 15],...
    'style','text','horizontalalignment','left',...
    'string',' ',...
    'fontname','comic','BackgroundColor',[.8 .8 .9],...
    'fontsize',10);
proses2=uicontrol('parent',win1,...
    'units','points',...
    'position',[15 245+36 75 15],...
    'style','Pushbutton',...
    'callback','openfilefrekwensi',...
    'string','File Frekwensi: ',...
    'fontname','comic',...
    'fontsize',10,'enable','off');
% enable off: biar g bisa di klik

label2=uicontrol('parent',win1,...
    'units','points',...
    'position',[90 245+36 295 15],...
    'style','text','horizontalalignment','left',...
    'string',' ',...
    'fontname','comic','BackgroundColor',[.8 .8 .9],...
    'fontsize',10);
%=====
label3=uicontrol('parent',win1,...
    'units','points',...
    'position',[15 210+24 360 42],...
    'style','text','horizontalalignment','left',...
    'string',{'Urutan Awal: '},...
    'fontname','comic','BackgroundColor',[.8 .8 .9],...
    'fontsize',10,'fontweight','bold');
%=====
hp0 = uipanel('parent',win1,...
    'Title','Algoritma Blocplan','FontSize',12,...
    'units','points',...
    'fontweight','bold',...
    'BackgroundColor',[.8 .8 .9],...
    'Position',[10 125+12 380 80+12]);

proses4=uicontrol('parent',win1,...
    'units','points',...
    'position',[15 165+24 75 20],...
    'style','Pushbutton',...
```

```
'callback','blockplan',...
'string',' Hitung ',...
'fontname','comic',...
'fontsize',10);
label4=uicontrol('parent',win1,...
'units','points',...
'position',[15 130+12 295 30+12],...
'fontweight','bold',...
'style','text','horizontalalignment','left',...
'string',{'Urutan:'},...
'fontname','comic','BackgroundColor',[.8 .8 .9],...
'fontsize',10);

%=====
hp0 = uipanel('parent',win1,...
'Title','Algoritma Craft Berdasarkan Algoritma
Blocplan','FontSize',12,...
'units','points',...
'fontweight','bold',...
'BackgroundColor',[.8 .8 .9],...
'Position',[10 40 380 80+12]);

proses5=uicontrol('parent',win1,...
'units','points',...
'position',[15 80+12 75 20],...
'style','Pushbutton',...
'callback','craft_BP',...
'string',' Hitung ',...
'fontname','comic',...
'fontsize',10);
label5=uicontrol('parent',win1,...
'units','points',...
'position',[15 45 295 30+12],...
'fontweight','bold',...
'style','text','horizontalalignment','left',...
'string',{'Urutan:','Total Bobot:'},...
'fontname','comic','BackgroundColor',[.8 .8 .9],...
'fontsize',10);

%=====
proses6=uicontrol('parent',win1,...
'units','points',...
'position',[165 10 70 20],...
'style','Pushbutton',...
'callback','RASET',...
'string',' Reset ',...
'fontname','comic',...
'fontsize',10);
proses7=uicontrol('parent',win1,...
'units','points',...
'position',[240 10 70 20],...
'style','Pushbutton',...
'callback','Close',...
'string',' Close ',...
'fontname','comic',...
'fontsize',10);
proses8=uicontrol('parent',win1,...
```

```
'units','points',...
'position',[90 10 70 20],...
'style','Pushbutton',...
'callback','Peta',...
'string',' Lihat Denah ',...
'fontname','comic',...
'fontsize',10,'enable','off');
=====
menu11=uimenu('parent',win1,...
'Label',' Hitung Bobot
Manual','callback','manual','enable','off');
menu12=uimenu('parent',win1,...
'Label',' Algoritma SA dan
HS','callback','RISET2','enable','off');
```

## B. Riset 2

```
clc; % membersihkan command window
% clear all;% membersihkan semua data
close all; % menutup semua figure

set(0,'Units','points')
Screen = get(0,'screensize'); % mengambil ukuran layar Laptop
pos=[0 0 400 380+29]; % ukuran gui panjang=400 x lebar=380+36

% data1=[];% inisialisasi data
% data2=[];% idem

win21=figure(...
'units','points',...
'position',[Screen(3:4)/2-pos(3:4)/2 pos(3:4)],...
'color',[.8 .8 .9],...
'resize','off',...
'menubar','none',...
'toolbar','none',...
'numbertitle','off',...
'name','Input Data');
=====
label1=uicontrol('parent',win21,...
'units','points',...
'position',[10 285+36 380 85],...
'style','text',...
'string',{'';'PENERAPAN ALGORITMA BLOCPAN DAN CRAFT';' DALAM
PERANCANGAN TATA LETAK RUANG RUMAH SAKIT';'';'Oleh: Wika Anggani /
071810101052'},...
'fontname','times new roman','BackgroundColor',[.8 .8 .9],...
'fontsize',12,'fontweight','bold');

label1=uicontrol('parent',win21,...
'units','points',...
'position',[0 280+36 400 1],...
'style','text',...
'fontname','times new roman','BackgroundColor',[0 0 0],...
'fontsize',12,'fontweight','bold');
```

```
%=====
%=====
hp0 = uipanel('parent',win21,...
            'Title','Algoritma Craft Berdasarkan Simulated
Annealing','FontSize',12,...
            'units','points',...
            'fontweight','bold',...
            'BackgroundColor',[.8 .8 .9],...
            'Position',[10 125+12+(30+12) 380 80+12+(30+12)]);

proses4=uicontrol('parent',win21,...
                'units','points',...
                'position',[15 165+24+(30+12)*2 75 20],...
                'style','Pushbutton',...
                'callback','craft_SA',...
                'string',' Hitung ',...
                'fontname','comic',...
                'fontsize',10);
label4=uicontrol('parent',win21,...
                'units','points',...
                'position',[15 130+12+(30+12) 355 (30+12)*2],...
                'fontweight','bold',...
                'style','text','horizontalalignment','left',...
                'string',{'Urutan SA:','Urutan Craft: ','Total Bobot:'},...
                'fontname','comic','BackgroundColor',[.8 .8 .9],...
                'fontsize',10);

%=====
hp0 = uipanel('parent',win21,...
            'Title','Algoritma Craft Berdasarkan Harmoni
Search','FontSize',12,...
            'units','points',...
            'fontweight','bold',...
            'BackgroundColor',[.8 .8 .9],...
            'Position',[10 40 380 80+12+(30+12)]);

proses5=uicontrol('parent',win21,...
                'units','points',...
                'position',[15 80+12+(30+12) 75 20],...
                'style','Pushbutton',...
                'callback','craft_HS',...
                'string',' Hitung ',...
                'fontname','comic',...
                'fontsize',10);
label5=uicontrol('parent',win21,...
                'units','points',...
                'position',[15 45 355 (30+12)*2],...
                'fontweight','bold',...
                'style','text','horizontalalignment','left',...
                'string',{'Urutan HS:','Urutan Craft: ','Total Bobot: '},...
                'fontname','comic','BackgroundColor',[.8 .8 .9],...
                'fontsize',10);

%=====
proses6=uicontrol('parent',win21,...
                'units','points',...
                'position',[165 10 70 20],...
```



```
'style','Pushbutton',...
'callback','RISET',...
'string',' Reset ',...
'fontname','comic',...
'fontsize',10);
proses7=uicontrol('parent',win21,...
'units','points',...
'position',[240 10 70 20],...
'style','Pushbutton',...
'callback','Close',...
'string',' Close ',...
'fontname','comic',...
'fontsize',10);
proses8=uicontrol('parent',win21,...
'units','points',...
'position',[90 10 70 20],...
'style','Pushbutton',...
'callback','Peta2',...
'string',' Lihat Denah ',...
'fontname','comic',...
'fontsize',10,'enable','off');
```

## C. Algoritma *Blocplan*

```
% pake data 3
urutan1=[];
urutan1=1;
data13=data2;
data13(:,1)=zeros(Job1,1);
for i=2:Job1
    n=length(urutan1);
    % legth= panjang/ukuran matrik
    frek=[];
    indek_fre=[];
    for j=1:n
        [frek(j) indek_fre(j)]=max(data13(urutan1(j),:));
    end
    [frek_max indek]=max(frek);
    urutan1(i)=indek_fre(indek(1));

    % mengganti nol
    data13(:,urutan1(i))=zeros(Job1,1);
end
%=====
[data_urut indek_jarak]=sort(data1(1,:));
urutan0=urutan1;
for i=1:Job1
    urutan1(i)=urutan0(indek_jarak(i));
end
% -----
```

```
% for i=1:Job1
%     urutan_01(i)=urutan1(urutan_jarak(i));
% end
% urutan1=urutan_01;
% -----
% k=1;
% for i=1:Job1-1
%
%     indek_brs(k:(k+Job1-i))=i:Job1-1;
%     indek_klm(k:(k+Job1-i))=i*ones(1,1:(Job1-i));
%     data_x(k:(k+Job1-i))=data2(i,(i+1):(Job1));
%     k=k+Job1-i;
% end

% urutan1=[(1), (21), (8), (15), (5), (4), (9), (20), (2), (19),
(13), (6), (10), (7), (18), (12), (17), (3), (11), (22), (23),
(16), (14)];
% update data2 jadi sesuai urutan yangbaru
data12=[];
for i=1:Job1

    for j=1:Job1
        a=urutan1(i);
        b=urutan1(j);
        data12(i,j)=data2(a,b);
    end
end
data4=data12.*data1;
momentum1=sum(sum(data4));

set(label4,'string',{'Urutan: ' num2str(urutan1)});
```

## D. Algoritma Craft

```
unction [urutan2 momentum2]=craft(data2, Job1, urutan1, data1)
% pake data 3
urutan2=urutan1;
data14=data2;
k=0;
for i=1:17 % mulai urutan ke 2 sampai job-1
    urut=[];
    if i==1
        indek1=[1 3 2 4:Job1];
    elseif i==2
        indek1=[1 3 2 4:10 12 11 13:Job1];
    elseif i==3
        indek1=[1 2 4 3 5:Job1];
    elseif i==4
        indek1=[1:4 6 5 7:Job1];
    elseif i==5
        indek1=[1:6 8 7 9:Job1];
    elseif i==6
        indek1=[1 3 2 4:7 9 8 10:Job1];
```

```
elseif i==7
    indek1=[1:7 9 8 10:Job1];
elseif i==8
    indek1=[1:9 11 10 12:Job1];
elseif i==9
    indek1=[1:10 12 11 13:Job1];
elseif i==10
    indek1=[1:12 14 13 15:Job1];
elseif i==11
    indek1=[1 3 2 4:7 9 8 10 12 11 13:Job1];
elseif i==12
    indek1=[1:13 15 14 16:Job1];
elseif i==13
    indek1=[1:4 6 5 7:13 15 14 16:Job1];
elseif i==14
    indek1=[1:14 16 15 17:Job1];
elseif i==15
    indek1=[1:18 20 19 21:Job1];
elseif i==16
    indek1=[1:19 21 20 22 23];
elseif i==17
    indek1=[1:21 23 22];

elseif i==18
    indek1=[ ];

end

for j=1:Job1
   urut(j)=urutan2(indek1(j));
end

% update data2 jadi sesuai urutan yangbaru
data15=[];
for i=1:Job1

    for j=1:Job1
        a=urut(i);
        b=urut(j);
        data15(i,j)=data2(a,b);
    end
end
data5=data15.*data1;
k=k+1;
urutan02(k,:)=urut;
momentum02(k)=sum(sum(data5));

end
[momentum2 indek]=min(momentum02);
urutan2=urutan02(indek,:);
```

## E. Penerapan Algoritma *Craft* berdasarkan algoritma *Blocplan*

```
[urutan2 momentum2]=craft(data2, Job1, urutan1, data1);

set(label5,'string',{['Urutan: ' num2str(urutan2)];['Total Bobot: '
' num2str(momentum2/2)]});

if isempty(urutan1)==0
    set(proses8,'enable','on');
end
```

## F. Penerapan Algoritma *Craft* berdasarkan algoritma *Simulated Annealing*

```
urutan_sa=[1 3 14 6 20 4 23 22 17 19 21 11 12 16 8 9 18 7 5 2 10
13 15];
% update data2 jadi sesuai urutan yangbaru
data22=[];
for i=1:Job1
    for j=1:Job1
        a=urutan_sa(i);
        b=urutan_sa(j);
        data22(i,j)=data2(a,b);
    end
end
data_sa=data22.*data1;
momentum_sa=sum(sum(data_sa));

%=====
%craft
[urutan_csa momentum_csa]=craft(data2, Job1, urutan_sa, data1);

set(label4,'string',{['Urutan SA: ' num2str(urutan_sa)];['Total
Bobot: ' num2str(81392)];['Urutan craft: '
num2str(urutan_csa)];['Total Bobot: ' num2str(momentum_csa)]});
set(label4,'string',{['Urutan SA: ' num2str(urutan_sa)];['Urutan
craft: ' num2str(urutan_csa)];['Total Bobot: '
num2str(momentum_csa/2)]});
```

## G. Penerapan Algoritma *Craft* berdasarkan algoritma *Harmony Search*

```
urutan_hs=[1 21 8 15 5 4 9 20 2 19 13 6 10 7 18 12 17 3 11 22 23
16 14];
% update data2 jadi sesuai urutan yangbaru
data22=[];
for i=1:Job1
    for j=1:Job1
        a=urutan_hs(i);
        b=urutan_hs(j);
        data22(i,j)=data2(a,b);
    end
end
```

```
data_hs=data22.*data1;
momentum_hs=sum(sum(data_hs));

%=====
%craft
[urutan_chs momentum_chs]=craft(data2, Job1, urutan_hs, data1);

set(label5,'string',{'Urutan HS: ' num2str(urutan_hs)};['Total
Bobot: ' num2str(95897)];['Urutan craft: '
num2str(urutan_chs)];['Total Bobot: ' num2str(momentum_csa)]});
set(label5,'string',{'Urutan HS: ' num2str(urutan_hs)};['Urutan
craft: ' num2str(urutan_chs)];['Total Bobot: '
num2str(momentum_csa/2)]});
set(proses8,'enable','on')

DENAH BLOCLPLAN DAN CRAFT
for i=2:4
figure(i)
clf

plot([0 10],[1 1],'linewidth',2);hold on
plot([0 10],[3 3],'linewidth',2);
plot([0 0],[1 3],'linewidth',2);
plot([10 10],[1 3],'linewidth',2);
plot([2.3 2.3],[1 3],'r',[4.5 4.5],[1 3],'r',[6.7 6.7],[1 3],'r');

%-----
plot([0 10],[4 4],'linewidth',2);hold on
plot([0 10],[6 6],'linewidth',2);
plot([0 0],[4 6],'linewidth',2);
plot([10 10],[4 6],'linewidth',2);
plot([2 2],[4 6],'r',[4 4],[4 6],'r',[6 6],[4 6],'r',[8 8],[4
6],'r');

%-----
plot([11 20],[1 1],'linewidth',2);hold on
plot([11 20],[3 3],'linewidth',2);
plot([11 11],[1 3],'linewidth',2);
plot([20 20],[1 3],'linewidth',2);
plot([12.3 12.3],[1 3],'r',[15 15],[1 3],'r',[17.7 17.7],[1
3],'r');

%-----
plot([11 20],[4 4],'linewidth',2);hold on
plot([11 20],[6 6],'linewidth',2);
plot([11 11],[4 6],'linewidth',2);
plot([20 20],[4 6],'linewidth',2);
plot([12.3 12.3],[4 6],'r',[14.7 14.7],[4 6],'r',[17.25 17.25],[4
6],'r');

%-----
plot([-5 30],[9 9],'k')

%-----
```



```
plot([0 10],[11 11],'linewidth',2);hold on
plot([0 10],[13 13],'linewidth',2);
plot([0 0],[11 13],'linewidth',2);
plot([10 10],[11 13],'linewidth',2);
plot([3 3],[11 13],'r',[6 6],[11 13],'r');

%-----
plot([0 10],[14 14],'linewidth',2);hold on
plot([0 10],[16 16],'linewidth',2);
plot([0 0],[14 16],'linewidth',2);
plot([10 10],[14 16],'linewidth',2);
plot([2.5 2.5],[14 16],'r',[5 5],[14 16],'r',[7.5 7.5],[14
16],'r');

%-----
plot([11 20],[11 11],'linewidth',2);hold on
plot([11 20],[13 13],'linewidth',2);
plot([11 11],[11 13],'linewidth',2);
plot([20 20],[11 13],'linewidth',2);
plot([16 16],[11 13],'r');

%-----
plot([11 20],[14 14],'linewidth',2);hold on
plot([11 20],[16 16],'linewidth',2);
plot([11 11],[14 16],'linewidth',2);
plot([20 20],[14 16],'linewidth',2);
plot([17 17],[14 16],'r');

%-----
text(8,8,'Lantai 1');
text(8,18,'Lantai 2');
text(7,2,'Apotek');
text(12.5,5,'Admisi');
text(11.5,15,'Rekam Medik');
axis([-1 22 -1 20]);

if i==2
    kata=urutan0;
    title('Denah Awal');
elseif i==3
    kata=urutan1;
    title('Denah hasil algoritma Blockplan');
else
    kata=urutan2;
    title('Denah hasil algoritma Craft berdasarkan Blocplan');
end
a=[];
a=kata;
x=[10.25 3 5 7 9 16 19 1.2 3.5 5.6 13 16 19 1.5 3.5 6.5 8.5 18 2 5
8.5 13 19];
y=[1 5 5 5 5 5 2 2 2 2 2 15 15 15 15 12 12 12 12 12 ];
for j=1:length(x)
text(x(j),y(j),num2str(a(j)));
end
```

end

## DENAH ALGORITMA SIMULATED ANNEALING DAN HARMONY SEACRH

```
for i=5:8
figure(i)
clf

plot([0 10],[1 1],'linewidth',2);hold on
plot([0 10],[3 3],'linewidth',2);
plot([0 0],[1 3],'linewidth',2);
plot([10 10],[1 3],'linewidth',2);
plot([2.3 2.3],[1 3],'r',[4.5 4.5],[1 3],'r',[6.7 6.7],[1 3],'r');

%-----
plot([0 10],[4 4],'linewidth',2);hold on
plot([0 10],[6 6],'linewidth',2);
plot([0 0],[4 6],'linewidth',2);
plot([10 10],[4 6],'linewidth',2);
plot([2 2],[4 6],'r',[4 4],[4 6],'r',[6 6],[4 6],'r',[8 8],[4
6],'r');

%-----
plot([11 20],[1 1],'linewidth',2);hold on
plot([11 20],[3 3],'linewidth',2);
plot([11 11],[1 3],'linewidth',2);
plot([20 20],[1 3],'linewidth',2);
plot([12.3 12.3],[1 3],'r',[15 15],[1 3],'r',[17.7 17.7],[1
3],'r');

%-----
plot([11 20],[4 4],'linewidth',2);hold on
plot([11 20],[6 6],'linewidth',2);
plot([11 11],[4 6],'linewidth',2);
plot([20 20],[4 6],'linewidth',2);
plot([12.3 12.3],[4 6],'r',[14.7 14.7],[4 6],'r',[17.25 17.25],[4
6],'r');

%-----
plot([-5 30],[9 9],'k')

%-----
plot([0 10],[11 11],'linewidth',2);hold on
plot([0 10],[13 13],'linewidth',2);
plot([0 0],[11 13],'linewidth',2);
plot([10 10],[11 13],'linewidth',2);
plot([3 3],[11 13],'r',[6 6],[11 13],'r');

%-----
plot([0 10],[14 14],'linewidth',2);hold on
plot([0 10],[16 16],'linewidth',2);
plot([0 0],[14 16],'linewidth',2);
plot([10 10],[14 16],'linewidth',2);
```

```
plot([2.5 2.5],[14 16],'r',[5 5],[14 16],'r',[7.5 7.5],[14
16],'r');

%-----
plot([11 20],[11 11],'linewidth',2);hold on
plot([11 20],[13 13],'linewidth',2);
plot([11 11],[11 13],'linewidth',2);
plot([20 20],[11 13],'linewidth',2);
plot([16 16],[11 13],'r');

%-----
plot([11 20],[14 14],'linewidth',2);hold on
plot([11 20],[16 16],'linewidth',2);
plot([11 11],[14 16],'linewidth',2);
plot([20 20],[14 16],'linewidth',2);
plot([17 17],[14 16],'r');

%-----
text(8,8,'Lantai 1');
text(8,18,'Lantai 2');
text(7,2,'Apotek');
text(12.5,5,'Admisi');
text(11.5,15,'Rekam Medik');
axis([-1 22 -1 20]);

if i==5
    kata=urutan_sa;
    title('Denah hasil Simulated Annealing');
elseif i==6
    kata=urutan_csa;
    title('Denah hasil algoritma Craft berdasarkan Simulated
Annealing');
elseif i==7
    kata=urutan_hs;
    title('Denah hasil algoritma Harmony Search');
elseif i==8
    kata=urutan_chs;
    title('Denah hasil algoritma Craft berdasarkan Harmony
Search');
end
a=[];
a=kata;
x=[10.25 3 5 7 9 16 19 1.2 3.5 5.6 13 16 19 1.5 3.5 6.5 8.5 18 2 5
8.5 13 19];
y=[1 5 5 5 5 5 5 2 2 2 2 2 2 15 15 15 15 15 12 12 12 12 12 ];
for j=1:length(x)
text(x(j),y(j),num2str(a(j)));

end
end
```