

## BAB III

### ANALISA DAN PERANCANGAN SISTEM

#### 3.1 Deskripsi Umum

Suatu sistem yang digunakan untuk membantu Kepala Ruang untuk menentukan alternatif dalam pembuatan jadwal shift perawat. Sistem dibuat dengan menerapkan Metode *Constraint Satisfaction*, dimana yang akan menggunakan sistem ini adalah kepala ruang. Kepala ruang yang bertugas menginput beberapa data yang dibutuhkan, yaitu identitas perawat dan jadwal shift perawat. Pembuatan alternatif penjadwalan shift perawat ini dikhususkan untuk Rumah Sakit Bhayangkara Hasta Brata Kota Batu.

#### 3.2 Analisa Data

Penelitian dilakukan dengan mengambil *sample* data pada Rumah Sakit Bhayangkara Hasta Brata Kota Batu. *Sample* yang dijadikan acuan sebagai penjadwalan shift perawat adalah penjadwalan shift yang masih manual yang disusun oleh kepala ruang selama 2 bulan terakhir.

Data – data yang dijadikan acuan untuk analisa dan perancangan sistem pada penjadwalan shift perawat antara lain :

- Jumlah Perawat

Jumlah perawat yang ada pada Rumah Sakit Hasta Brata Kota Batu ada 11 orang, 1 orang kepala ruang. Jadi total perawat di Rumah Sakit Hasta Brata Kota Batu ada 12 orang.

- Jumlah Ruangan Rawat Inap

Pada Rumah Sakit Hasta Brata ini memiliki setidaknya 2 ruang rawat inap untuk pasien, yaitu Ruang Bugenvil dan Ruang Mawar, pada 2 ruang rawat inap tersebut memiliki jumlah kamar sebanyak 16 kamar untuk ruang rawat inap pasien. Ruang rawat inap tersebut terdiri dari 6 kamar kelas 1 dan 10 kamar kelas 2.

- Slot Waktu

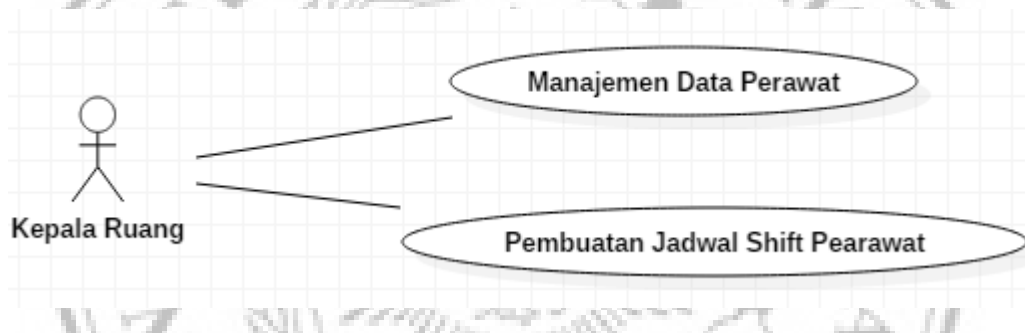
Slot waktu untuk shift kerja pada penjadwalan shift perawat pada Rumah Sakit Hasta Brata Kota Batu ada 3 shift jaga, yaitu : pagi, sore, dan malam, kemudian 1 shift off yaitu libur. Setiap perawat setidaknya mendapatkan semua giliran shift dalam satu bulan,

kemudian pada saat shift jaga minimal ada 3 perawat yang bertugas. Dalam 1 hari ada 3 shift jaga yang diterapkan, jadi setiap hari minimal ada 9 perawat yang bertugas. Jadi dalam satu bulan, ( 1 x 12 perawat ) x 30 hari = 360 slot waktu dalam satu bulan.

### 3.3 Perancangan Sistem

#### 3.3.1. Use Case Diagram

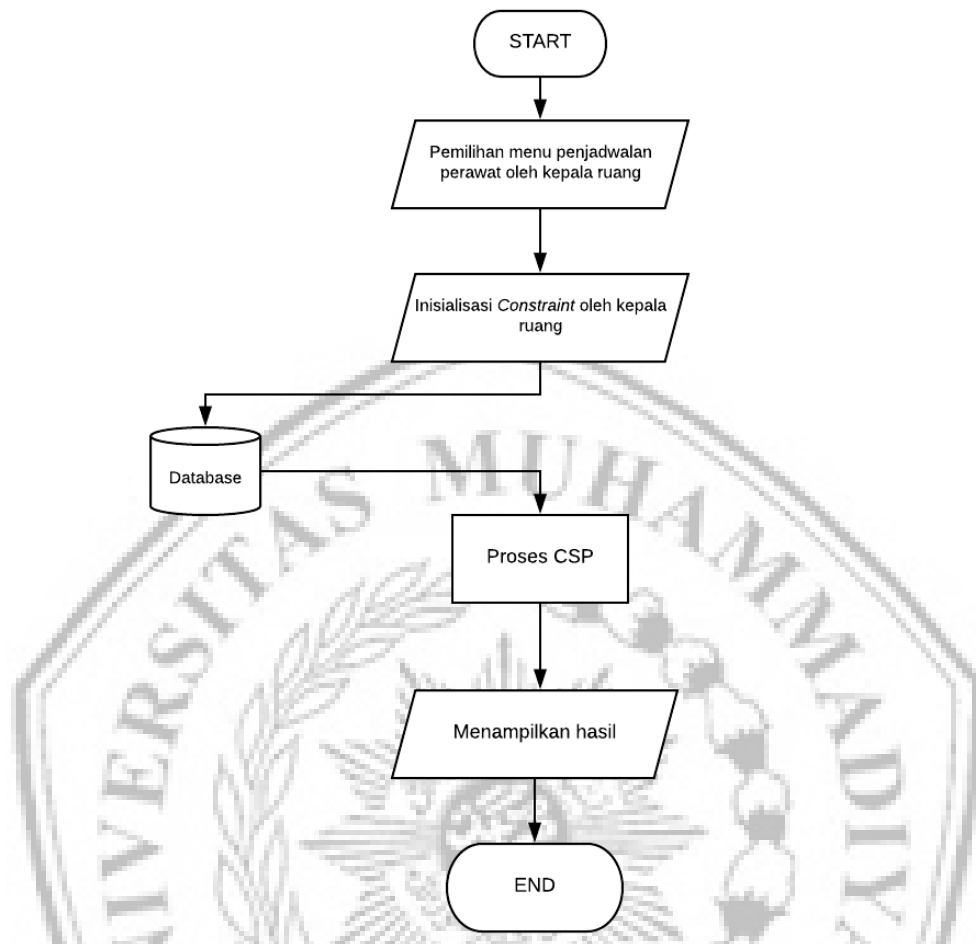
Gambar 3.1 merupakan *use case* untuk kepala ruang yang bertindak sebagai admin. Pada gambar tersebut admin memiliki 2 hak akses, yaitu yang pertama admin memiliki hak akses untuk me-manajemen data perawat yang meliputi create, read, update dan delete pada data perawat dan hak akses yang kedua adalah membuat jadwal shift perawat. Gambar 3.1 menunjukan *UseCase* diagram dalam sistem ini.



*Gambar 3. 1 Use Case Diagram untuk Admin*

#### 3.3.2. Design Perangkat Lunak

Aplikasi penjadwalan shift perawat ini diasumsikan hanya di akses oleh satu orang user saja, yaitu kepala ruang. Dimana kepala ruang tersebut yang bertugas untuk menyusun jadwal perawat secara keseluruhan. Penjelasan design perangkat lunak yang akan dibuat disajikan kedalam gambar 3.2.



**Gambar 3. 2** Flowchart sistem penjadwal shift perawat

Pada gambar 3.2 dijelaskan alur dari sistem pembuatan jadwal shift perawat adalah sebagai berikut :

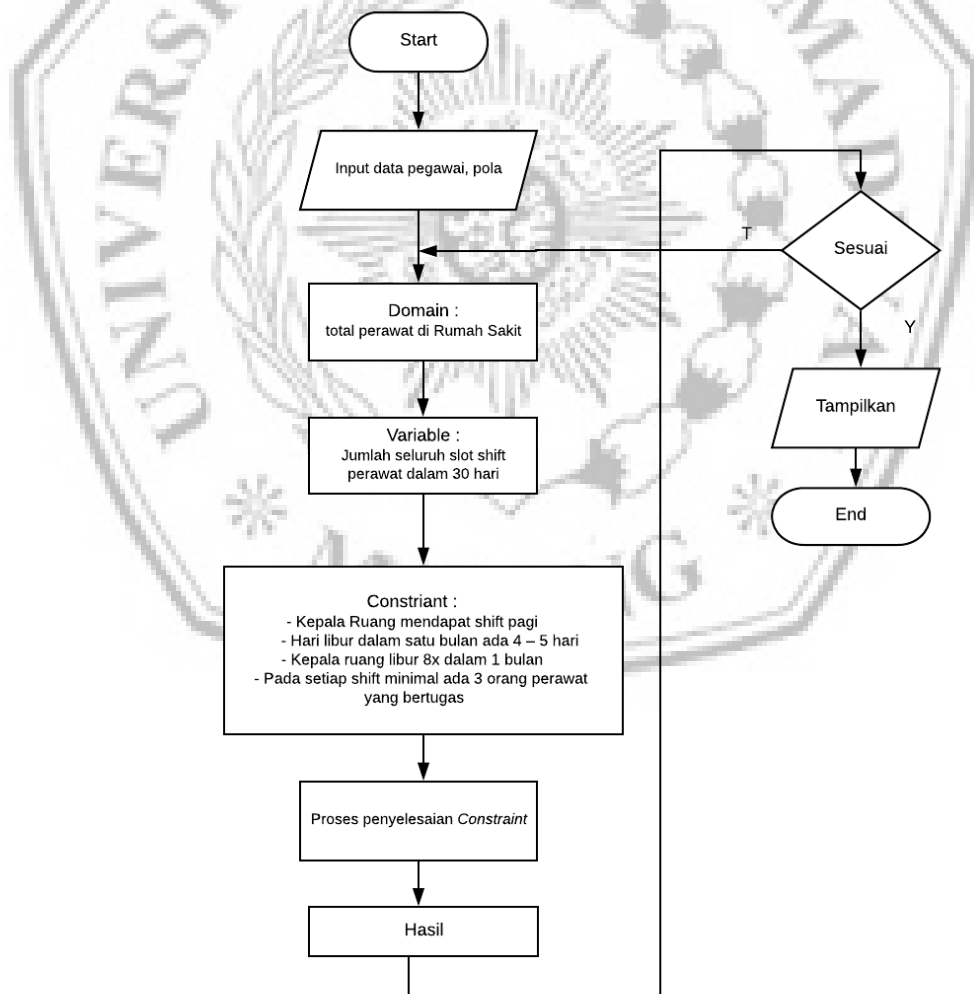
1. Yang bertindak sebagai *user* adalah kepala ruang, Kepala Ruang memilih menu untuk penjadwalan shift perawat.
2. Kemudian Kepala Ruang menginisialisasi *constraint* yang ingin ditambahkan. Sistem menyimpan *soft constraint* yang telah diinisialisasikan kedalam database.
3. Sistem mengambil data dari database untuk menganalisa elemen – elemen dasar dalam penyusunan *Constraint Satisfaction Problem*.
4. Menampilkan hasil penyelesaian dari *Constraint Satisfaction Problem* berupa jadwal shift perawat.

- Inisialisai *Constraint*

*Constraint Satisfaction* yang diinisialisasikan oleh kepala ruang yaitu *domain*, *variable*, dan *constraintnya* berupa *soft constraint*, memang definisi *soft constraint* disini adalah batasan-batasan tambahan yang bisaanya merupakan permintaan pribadi elemen penjadwalan. Oleh sebab itu, ada beberapa data yang memang harus diinputkan sendiri oleh *admin*. Data-data *soft constraint* yang perlu diinputkan user tersebut antara lain :

1. Kepala Ruang mendapat shift pagi
2. Hari libur dalam satu bulan ada 4 – 5 hari / ada 1x libur diantara 7 hari kerja.
3. Kepala ruang libur 8x dalam 1 bulan.
4. Pada setiap shift minimal ada 3 orang perawat yang bertugas.

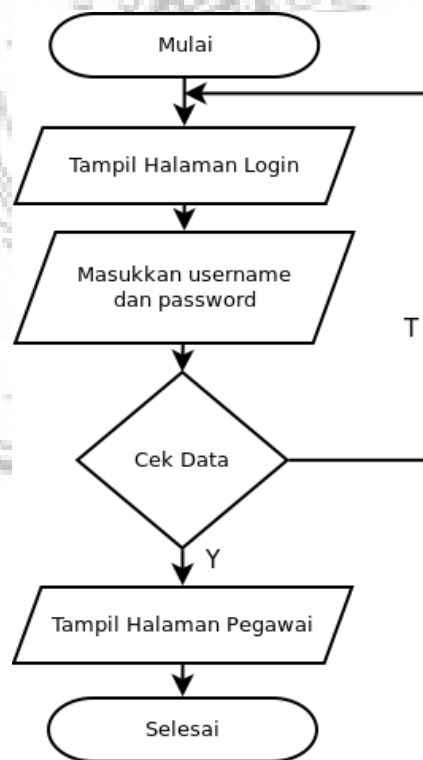
- Flowchart *Constraint*



**Gambar 3. 3** Flowchart *Constraint*

Pada gambar 3.3 ini proses dari *Constraint* pada sistem pembuatan jadwal akan dijelaskan sebagai berikut:

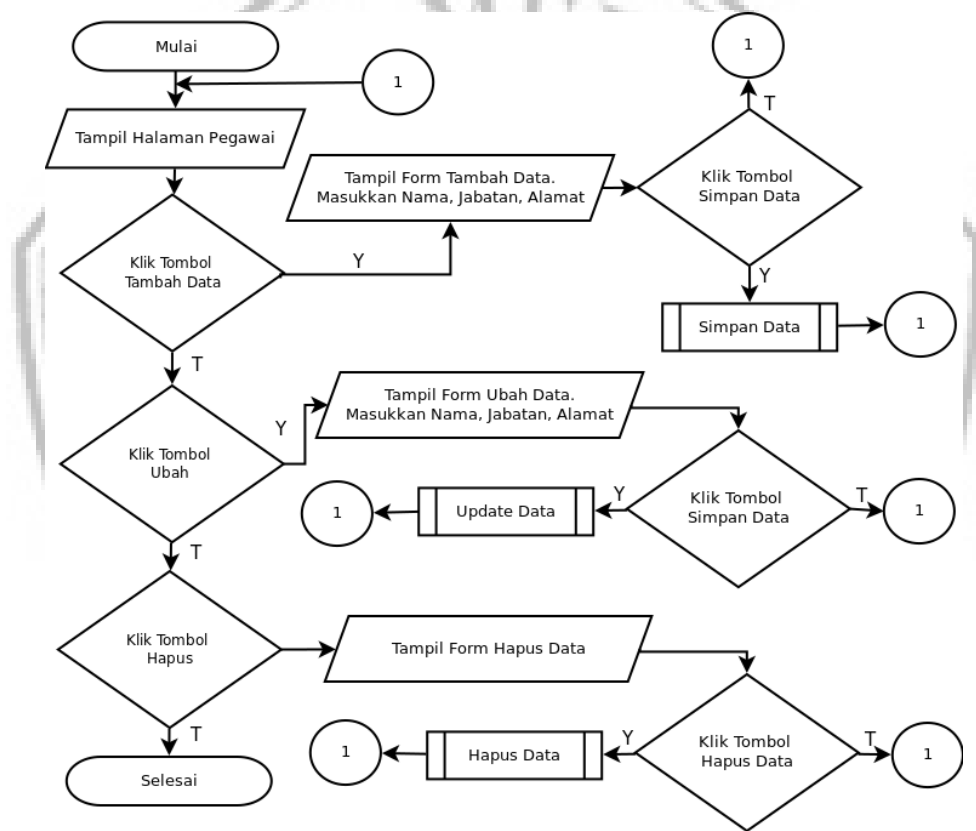
1. Kepala ruang menginput data pegawai dan pola yang akan digunakan.
2. Menginisialisasikan *domain*, yang berisi total perawat yang ada di Rumah Sakit Bhayangkara Hasta Brata.
3. Menginisialisasikan *variable*, yang diisi oleh jumlah seluruh slot shift perawat dalam 30 hari.
4. Menginisialisasikan *constraint*.
5. Setelah semua sudah diinisialisasikan maka sistem akan memproses *domain*, *variable*, dan *constraint*.
6. Kemudian sistem akan menampilkan hasil dari proses penginisialisasian.
7. Apabila hasilnya tidak sesuai dengan *constraint* yang diinisialisasikan maka akan kembali ke pemrosesan di poin 2.
8. Setelah *variable*, *domain*, dan *constraint* sesuai maka sistem akan menampilkan hasil penjadwalan yang siap dicetak.



**Gambar 3. 4** Flowchart Login

Flowchart pada gambar 3.4 merupakan alur proses login admin pada aplikasi yang dapat di jelaskan sebagai berikut :

1. Terlebih dahulu, buka halaman login aplikasi.
2. Masukkan username dan password pada form login.
3. Setelah semua form login teriisi, selanjutnya klik tombol login.
4. Aplikasi akan mengecek apakah data username dan password yang dimasukkan ada apa tidak. Jika data tidak ada maka akan kembali lagi ke halaman login, jika data ada maka akan tampil halaman pegawai.

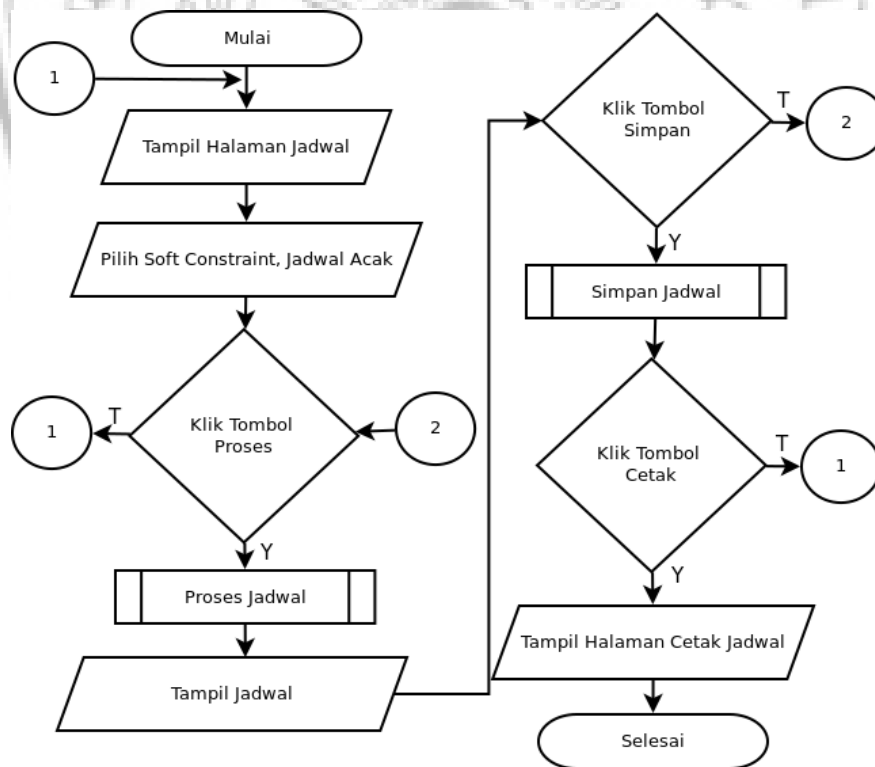


**Gambar 3. 5** Flowchart Halaman Pegawai

Gambar 3.5 adalah flowchart pada halaman pegawai, flowchart ini menjelaskan alur proses halaman pegawai pada aplikasi yang dapat di jelaskan sebagai berikut :

1. Masuk ke halaman pegawai.
2. Setelah tampil halaman pegawai, terdapat tabel data pegawai serta tombol tambah data, ubah, dan hapus.

3. Jika ingin menambah data pegawai, klik tombol tambah data. Maka akan muncul form tambah data pegawai.
4. Masukkan data pegawai berupa nama, jabatan, dan alamat pada form tambah data.
5. Klik tombol simpan data untuk menyimpan data pegawai baru, jika tidak maka kembali ke halaman data pegawai.
6. Jika ingin merubah data pegawai, maka klik tombol ubah pada data pegawai di tabel. Akan muncul form ubah data pegawai.
7. Ubah data pegawai pada form ubah data, jika sudah klik tombol simpan data untuk menyimpan perubahan data pegawai, jika tidak maka kembali ke halaman data pegawai.
8. Jika ingin menghapus data pegawai, maka klik tombol hapus pada data pegawai di tabel. Akan muncul data pegawai yang di pilih pada form hapus.
9. Klik tombol hapus data untuk menghapus data pegawai, jika tidak maka kembali ke halaman data pegawai.



**Gambar 3. 6** Flowchart Halaman Jadwal

Pada gambar 3.6 merupakan alur proses halaman jadwal pada aplikasi yang dapat di jelaskan sebagai berikut:

1. Masuk ke halaman jadwal.
2. Kemudian di klik pada soft constraint untuk melakukan proses soft constraint dan jadwal acak untuk mendapatkan jadwal yang acak.
3. Klik tombol proses untuk mendapatkan jadwal pegawai.
4. Jika jadwal yang ditampilkan tidak sesuai dengan yang diinginkan maka kembali klik tombol proses untuk mendapatkan jadwal yang lain.
5. Jika jadwal yang diinginkan sudah sesuai maka klik tombol simpan untuk menyimpan jadwal.
6. Setelah data tersimpan, jika ingin mencetak jadwal tersebut, maka klik tombol cetak.
7. Setelah klik tombol cetak maka akan muncul halaman cetak jadwal.

### 3.3.3. Proses Pengolahan Data

**Tabel 3.1** Daftar Nama Pegawai

No	Nama	Jabatan
1	Ahmad Kholil	Kepala Ruang
2	Diah Fatmawati	Perawat
3	Rinawati	Perawat
4	Fendi Setyo Utomo	Perawat
5	Hasan Khumaidi	Perawat
6	Angga Gigih	Perawat
7	M. Fatchurahman	Perawat
8	M. Faizin	Perawat
9	Ferika Irma	Perawat
10	Taufik Hidayat	Perawat



11	Wulan Vitasari	Perawat
12	Siti Afifah	Perawat

Pada tabel 3.1 dilampirkan daftar nama pegawai yang digunakan oleh kepala ruang untuk mendapatkan sebuah jadwal yang diinginkan, kemudian sistem akan melakukan beberapa proses sehingga jadwal yang disajikan bisa sesuai dengan keinginan. Kemudian untuk menyusun penjadwalan, digunakan pola penjadwalan sebanyak 7 pola, yaitu :

- P, P, S, S, M, M, X
- P, P, M, M, X, S, S
- M, M, X, S, S, P, P
- M, M, X, P, P, S, S
- S, S, P, P, M, M, X
- S, S, M, M, X, P, P
- P, S, M, M, X, S, P

Pola tersebut nantinya akan digunakan untuk membuat penjadwalan shift perawat, pola tersebut dibuat setelah melakukan analisa dari hasil penjadwalan yang dilakukan secara maual oleh kepala ruang selama 2 bulan terakhir, dimana pola tersebut setidaknya telah memperkecil kemungkinan pegawai melakukan pelanggaran pada *constraint* dalam hasil penjadwalannya. *Constraint* yang ditetapkan dalam penjadwalan ini akan dijelaskan pada sub bab berikutnya.

Pada proses pembuatan jadwal, sistem akan membuat variable untuk menampung nilai random, kemudian setelah variable random tersebut terisi nilai hasil dari random, maka nilai tersebut digunakan untuk memilih pola yang ada. Misal nilai random bernilai 2, maka pola yang digunakan untuk mencari jadwal kepala ruang dan perawat adalah pola nomer 2. Setelah melakukan proses pemilihan pola, selanjutnya yaitu mengecek pilihan dari *soft constraint*. Pengaktifan *soft constraint* atau tidak hanya akan berpengaruh pada jadwal kepala ruang saja, tapa berpengaruh kepada jadwal perawat.

Jika *soft constraint* diaktifkan, maka buat sebuah variabel index pola yang bernilai 1, kemudian dilakukan perulangan sebanyak 30 kali. Pada perulangan ini jadwal yang di dapat hanya jadwal kepala ruang. Jadwal kepala ruang bernilai “P” (pagi) jika nilai variabel index pola  $\leq 5$ , jika tidak maka jadwal kepala ruang adalah “X” (libur). Lakukan *increment* pada variabel index pola dan cek apakah nilai index pola sama dengan 8, jika sama dengan

8 maka atur nilai variabel index pola menjadi 1. Lakukan proses perulangan hingga selesai, jika selesai maka didapatkanlah jadwal kepala ruang.

Tetapi jika *soft constraint* tidak diaktifkan maka buat sebuah variable index pola bernilai 0. Selanjutnya lakukan perulangan sebanyak 30 kali, sesuai dengan jumlah banyaknya hari dalam satu bulan kalender system, pada perulangan tersebut kepala ruang akan mendapatkan jadwal sesuai dengan pola random yang dilakukan pada langkah sebelumnya, serta lakukan *increment* variable index pola. Cek apakah nilai index pola sama dengan jumlah pola hasil dari random, jika ya maka atur variable index pola menjadi 0. lakukan proses di atas hingga perulangan selesai, jika sudah selesai maka akan mendapatkan jadwal kepala ruang.

Kemudian untuk mendapatkan jadwal perawat, maka harus dilakukan proses sebelumnya. Untuk mendapatkan jadwal perawat lakukan proses perulangan sebanyak data perawat dan atur variable index pola menjadi 0. Kemudian setiap proses perulangan data perawat, dilakukan perulangan sebanyak 30 kali. Pada perulangan tersebut perawat akan mendapat jadwal sesuai dengan pola random yang dilakukan pada langkah sebelumnya, serta dilakukan *increment* variable index pola. Cek apakah nilai index pola sama dengan jumlah pola hasil dari random, jika ya maka atur variable index pola menjadi 0. Lakukan proses di atas hingga perulangan selesai, jika sudah selesai maka akan mendapatkan jadwal perawat.

Setelah kepala ruang dan semua perawat sudah mendapatkan jadwal shift selama 30 hari, maka didapatkan hasil dari penjadwalan selama 30 hari untuk semua pegawai berupa tabel.

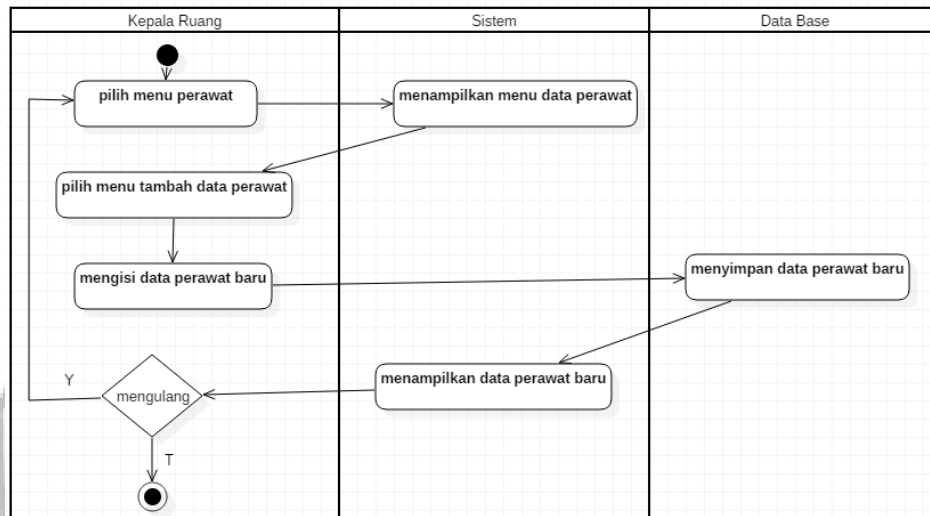
**Tabel 3.2** Contoh hasil jadwal shift perawat

Nama	Tanggal								
	1	2	3	4	5	6	7	....	30
K. Ruang	P	P	P	P	P	X	X	....	P
Perawat	P	P	S	S	M	M	X	...	P
Perawat	S	S	M	M	X	P	P	....	S
....	....	....	....	....	....	....	....	....	....
Perawat	M	M	X	P	P	S	S	....	M

### 3.3.4. Activity Diagram

Diagram yang menggambarkan berbagai aliran aktifitas dalam sistem yang sedang dirancang, bagaimana masing-masing aliran berawal, *decision* yang mungkin terjadi, dan bagaimana aktifitas itu berakhir.

#### 3.3.4.1 Create Data Perawat

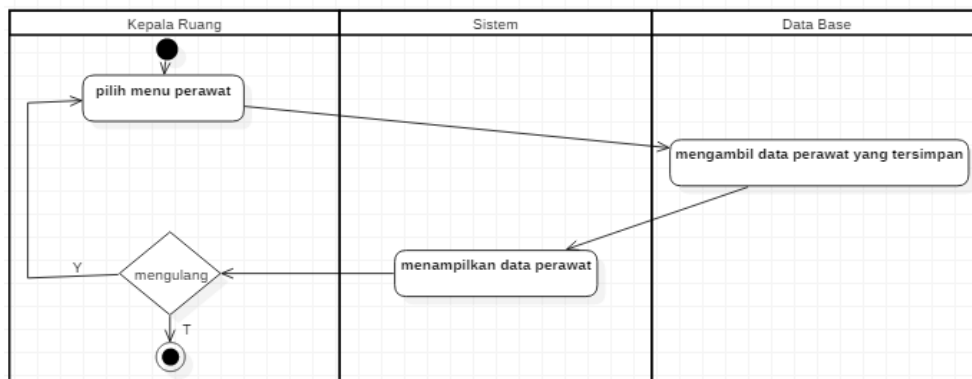


**Gambar 3. 7** Activity diagram Create Data Perawat

Keterangan :

Admin memilih menu perawat dan memilih tambah data perawat dan mengisi data perawat, kemudian database menyimpan inputan dari admin kemudian sistem menampilkan data perawat baru yang diinputkan.

#### 3.3.4.2 Read Data Perawat

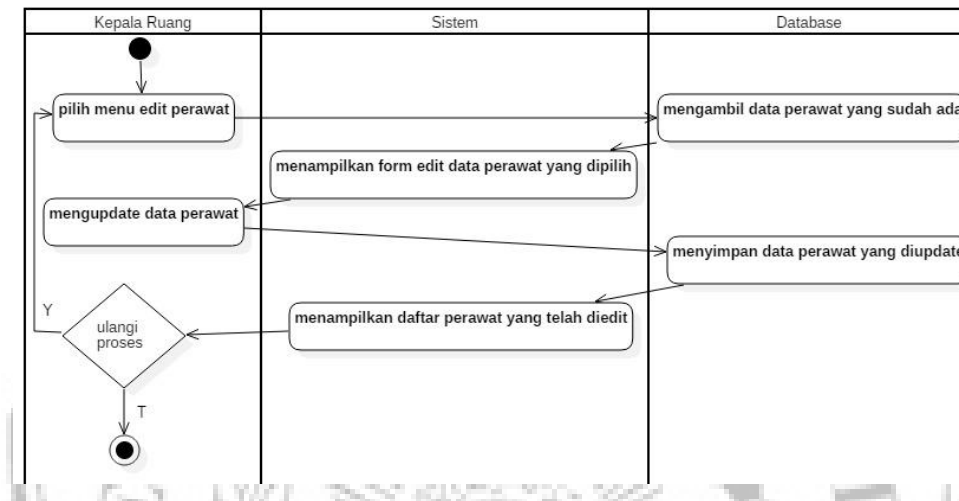


**Gambar 3. 8** Activity diagram Read Data Perawat

Keterangan :

admin memilih menu perawat kemudian sistem menampilkan pilihan menu dan admin kemudian memilih data perawat, kemudian sistem menampilkan data perawat yang sudah tersimpan.

### 3.3.4.3 Update Data Perawat

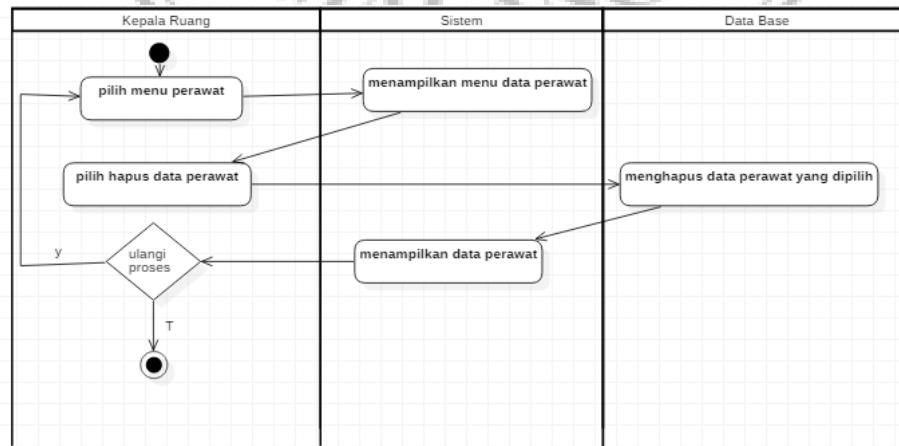


Gambar 3. 9 Activity diagram Update Data Perawat

Keterangan :

admin memilih menu update perawat, kemudian sistem menampilkan data perawat yang tersimpan pada database, kemudian admin memilih data perawat mana yang akan di update, setelah admin merubah data perawat dan kemudian disimpan, sistem akan menyimpan perubahan pada database.

### 3.3.4.4 Delete Data Perawat

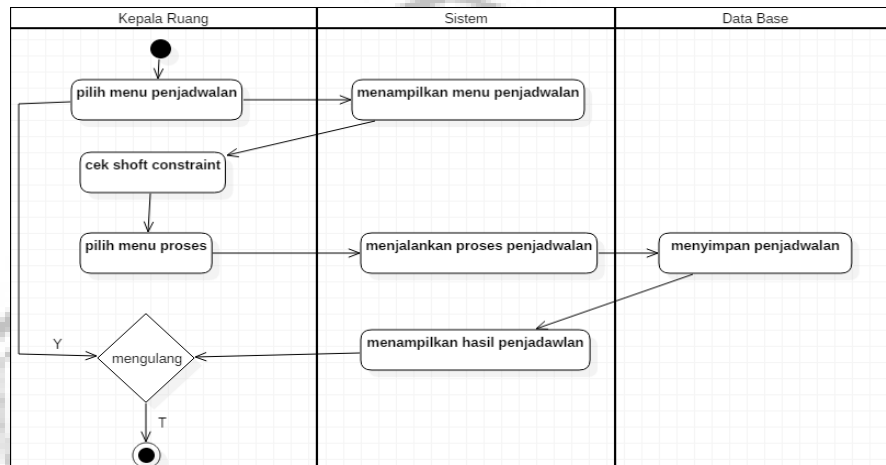


Gambar 3. 10 Activity diagram Delete Data Perawat

Keterangan :

admin memilih menu delete data perawat, kemudian sistem menampilkan data perawat yang tersimpan pada database, kemudian admin memilih data perawat mana yang akan di hapus, setelah admin menghapus data perawat dan kemudian disimpan, sistem akan menyimpan perubahan pada database.

#### 3.3.4.5 Penjadwalan Shift Perawat



Gambar 3. 11 Activity diagram Penjadwalan Perawat

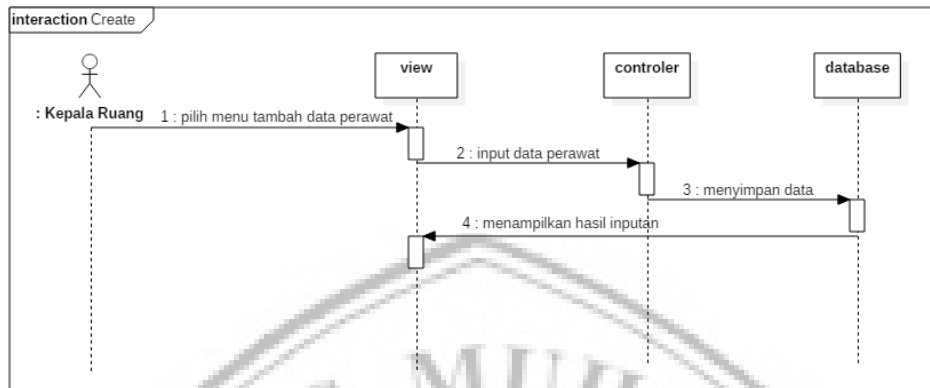
Keterangan :

Admin memilih menu pembuatan jadwal shift perawat, kemudian mengecek *soft constraint*, lalu klik proses kemudian sistem akan menjalankan proses penjadwalan shift perawat dan mengirimkan nya ke database, setelah selesai pada data maka sistem akan menampilkan sistem penjadwalan.

#### 3.3.5. Sequence Diagram

Sequence diagram digunakan untuk menggambarkan arus pekerjaan, pesan yang disampaikan dan bagaimana elemen-elemen di dalamnya bekerja sama dari waktu ke waktu untuk mencapai suatu hasil.

### 3.3.5.1 Create

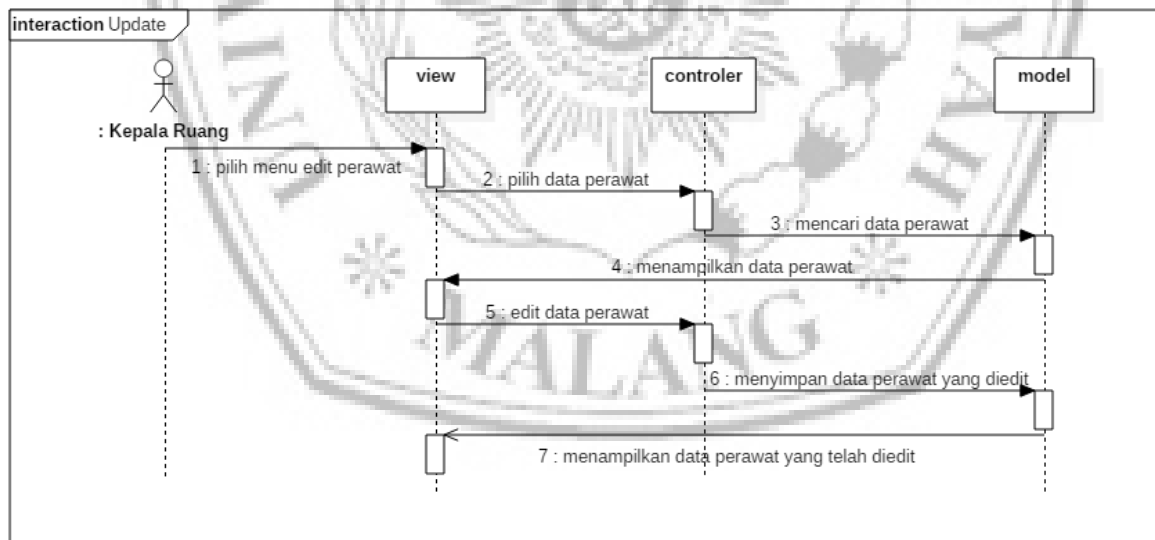


Gambar 3. 12 Sequence diagram Create Data Perawat

#### Keterangan :

Pada gambar 3.12 ini menjelaskan *sequence diagram* dari create data perawat yang lebih terperinci dalam menggambarkan proses dari create data perawat.

### 3.3.5.2 Update

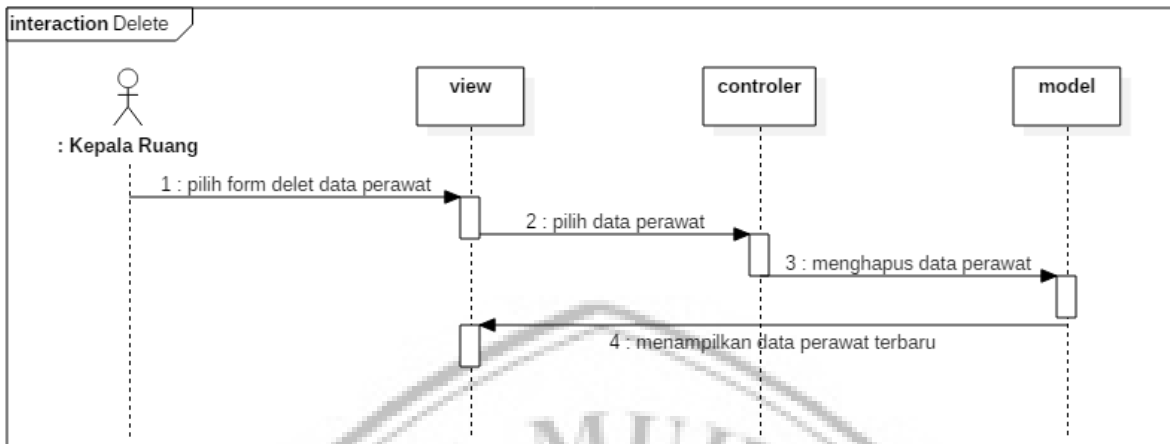


Gambar 3. 13 Sequence diagram Update Data Perawat

#### Keterangan :

Pada gambar 3.13 ini menjelaskan *sequence diagram* dari update data perawat yang lebih terperinci dalam menggambarkan proses dari update data perawat.

### 3.3.5.3 Delete

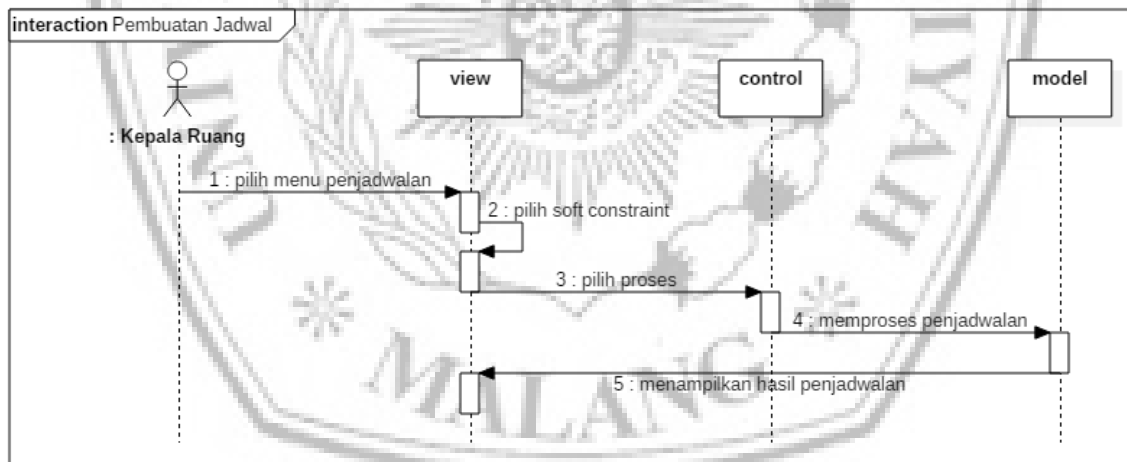


Gambar 3. 14 Sequence diagram Delete Data Perawat

Keterangan :

Pada gambar 3.14 ini menjelaskan *sequence diagram* dari delete data perawat yang lebih terperinci dalam menggambarkan proses dari delete data perawat.

### 3.3.5.4 Pembuatan jadwal shift perawat



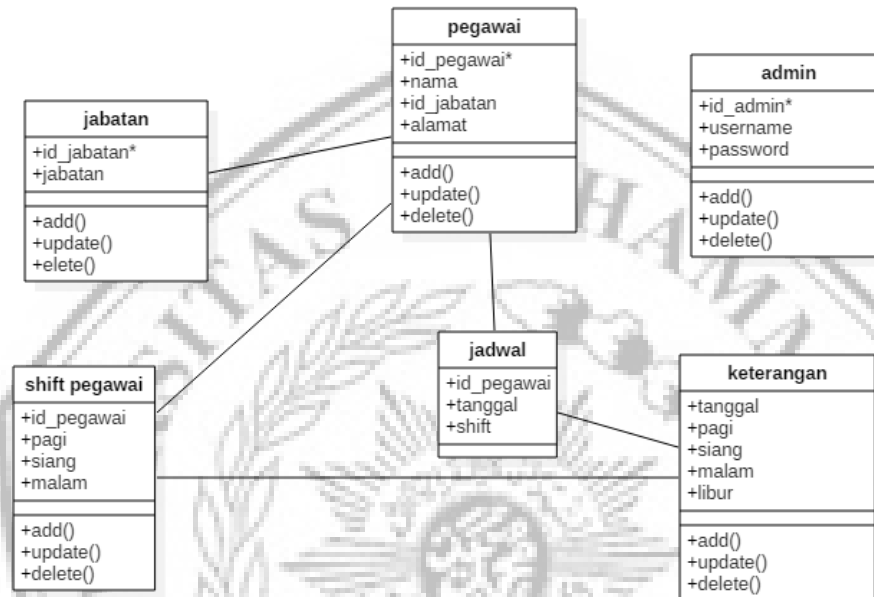
Gambar 3. 15 Sequence diagram Penjadwalan Perawat

Keterangan :

Pada gambar 3.15 ini menjelaskan *sequence diagram* dari pembuatan jadwal perawat yang lebih terperinci dalam menggambarkan proses penjadwalan perawat.

### 3.3.6. Class Diagram

*Class* diagram adalah diagram yang digunakan untuk menampilkan beberapa kelas serta paket-paket yang ada dalam sistem/perangkat lunak yang sedang dikembangkan dimana diagram ini memberikan gambaran (diagram statis) tentang sistem/perangkat lunak dan relasi-relasi yang ada di dalamnya. *Class* diagram perawat yang dihasilkan terdapat dalam gambar 3.16.

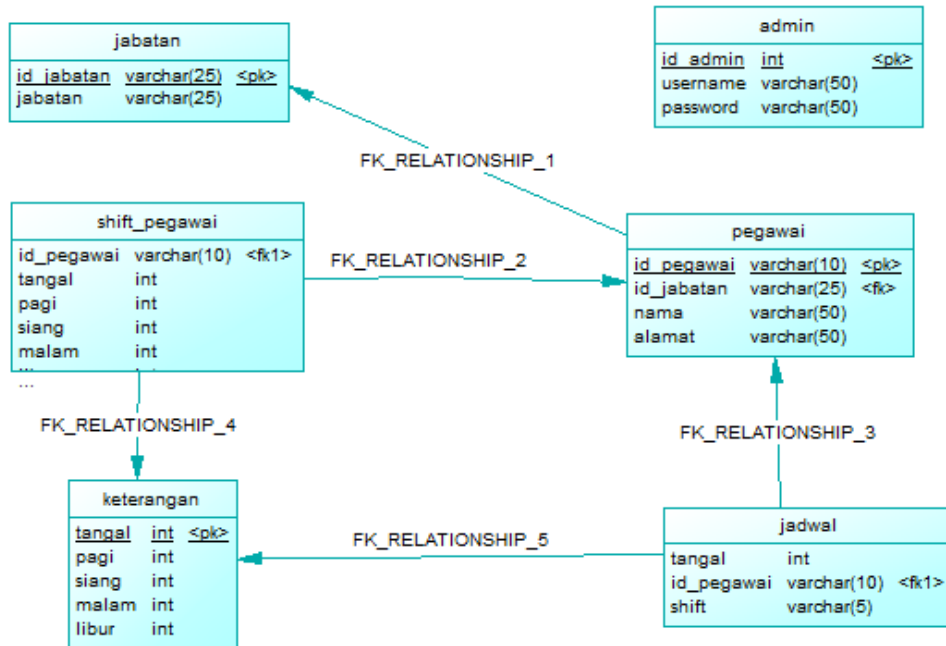


Gambar 3. 16 Class Diagram Perawat

### 3.3.7. Physical Data Model (PDM)

Sebuah *Physical Data Model* (PDM) menggambarkan secara detail konsep rancangan struktur basis data yang dirancang untuk suatu aplikasi. PDM merupakan hasil *generate* dari CDM.[12] Pada PDM ini terdapat 6 (enam) entitas (tabel), entitas tersebut terdiri atas *jabatan*, *shift pegawai*, *keterangan*, *admin*, *pegawai*, dan *jadwal*. Untuk lebih jelasnya, PDM dapat dilihat pada Gambar 3.17.





Gambar 3. 17 PDM Pegawai

### 3.3.8. Analisa Algoritma *Constraint*

Pada bagian ini dijelaskan tentang analisa aplikasi dan penggunaan dari algoritma dalam pendekatan *Constraint Satisfaction*. Pada pendekatan ini, yang dianggap sebagai *variable* adalah jumlah seluruh slot shift perawat yang bertugas selama 30 hari, kemudian *domain*-nya adalah seluruh karyawan yang bertugas di ruang tersebut. Hal dibawah ini merupakan aturan yang sudah dikelompokkan kedalam *Hard Constraint* dan *Soft Constraint*.

- *Hard Constraint* :
  - a. Pada Ruang Inap Rumah Sakit Bhayangkara Hasta Brata Batu ada beberapa kategori karyawan yaitu, Kepala Ruang, perawat.
  - b. Pada jadwal ada tiga shift dalam satu hari yaitu, pagi, siang dan malam.
  - c. Pada jadwal Kepala Ruang hanya ada shift pagi.
  - d. Pada jadwal perawat, setelah 2 hari shift malam berurutan mendapat libur 1 hari dikeesokan harinya.
  - e. Setelah shift malam keesokan harinya tidak boleh shift pagi.

- *Soft Constraint* :

- a. Pada jadwal Kepala Ruang hanya ada 5 hari kerja berurutan dalam satu minggu.
- b. Hari libur dalam satu bulan ada 4 – 5 hari / ada 1x libur diantara 7 hari kerja.
- c. Kepala ruang mendapatkan 8x libur dalam 1 bulan.
- d. Pada setiap shift minimal ada 3 orang perawat yang bertugas.

Pada *Soft Constraint* diatas, ada suatu prioritas yang utama, yaitu adanya perawat yang bertugas 3 orang disetiap shiftnya. Diharapkan dengan adanya 3 orang perawat disetiap shifnya dapat memberikan pelayanan yang efektif kepada pasien dan para perawat juga diharapkan bisa mendapat jam kerja yang ideal.

### 3.3.9. Design Prototype Interface

Rancangan desain prototype *interface* merupakan rancangan awal dalam menentukan tampilan sistem yang akan dibuat. Tujuan dari penggunaan *design interface* ini adalah untuk membuat interaksi pengguna sesederhana dan seefisien mungkin. Bagaimana user berinteraksi dengan komputer menggunakan tampilan antarmuka (*interface*) yang ada pada layar komputer.

*Design Interface* ini sangat penting karena akan sangat berpengaruh pada user dalam menggunakan atau berkomunikasi dengan komputer. Apabila suatu program sulit untuk digunakan, maka hal ini akan memaksa user untuk melakukan suatu kesalahan saat menggunakan program tersebut.

#### 3.3.9.1 Tampilan Login



The image shows a login form with a header section containing a circular logo of RS Bhayangkara Hata Brata Kota Batu. Below the logo, the text 'Form Login' and 'RS Bhayangkara Hata Brata Kota Batu' is displayed. The main body of the form contains two input fields: 'Username' and 'Password', each preceded by a small circular icon. Below these fields is a 'LOG IN' button.

**Gambar 3. 18** Halaman Login

### 3.3.9.2 Tampilan Form Tambah Data Perawat

**FORM TAMBAH DATA PEGAWAI**

**ID\_Pegawai**

**Nama**

**Jabatan**

**Alamat**

BATAL

SIMPAN

*Gambar 3. 19* Halaman Tambah Data Perawat

### 3.3.9.3 Tampilan Form Edit Data Perawat

**FORM EDIT DATA PEGAWAI**

**ID\_Pegawai**

**Nama**

**Jabatan**

**Alamat**

BATAL

SIMPAN

*Gambar 3. 20* Halaman Edit Data Perawat

#### 3.3.9.4 Tampilan Form Data Perawat

APLIKASI PENJADWALAN PERAWAT

PEGAWAI

JADWAL

LOG OUT

DATA PEGAWAI

TAMBAH DATA

NO	ID PEGAWAI	NAMA PEGAWAI	JABATAN	ALAMAT	
1	P01	Ahmad Kholil	Kepala Ruang	Malang	<div>UBAH</div> <div>TAMBAH</div>
2	P02	Diah Fatmawati	Perawat	Batu	<div>UBAH</div> <div>TAMBAH</div>

**Gambar 3. 21** Halaman Data Perawat

### 3.3.9.5 Tampilan Form Penjadwalan

[illegible]

**Gambar 3. 22** Halaman Penjadwalan Shift Perawat