//fungsi kamar

function kamar(Input kelas:string)🡪string

{I.S. : kelas sudah terdefinisi}

{F.S. : menghasilkan fungsi Kamar}

Kamus:

Algoritma:

if (kelas = 'VIP')

then

kamar 🡨 'Anggrek'

else

if (kelas = '1')

then

kamar 🡨 'Sakura'

else

if (kelas = '2')

then

kamar 🡨 'Mawar'

else

if (kelas = '3')

then

kamar 🡨 'Flamboyan'

endif

endif

endif

endif

endfunction

//fungsi harga

function harga(Input kelas:string)🡪integer

{I.S. : kelas sudah terdefinisi}

{F.S. : menghasilkan fungsi Harga kamar per hari}

Kamus:

Algoritma:

if (kelas = 'VIP')

then

harga 🡨 200000

else

if (kelas = '1')

then

harga 🡨 125000

else

if (kelas = '2')

then

harga 🡨 75000

else

if (kelas = '3')

then

harga 🡨 35000

endif

endif

endif

endif

endfunction

//prosedure Menu Pilihan

procedure MenuPilihan (Output Menu: integer)

{I.S. : User memilih salah satu menu}

{F.S. : Menampilkan hasil sesuai menu yang dipilih}

Kamus:

Algoritma:

Output('|=================================================|')

Output('|============== PROGRAM RAWAT INAP ===============|')

Output('|=================================================|')

Output('| MENU PILIHAN |')

Output('|-------------------------------------------------|')

Output('| 1. ISI DATA PASIEN |')

Output('| 2. CARI BERDASARKAN ID |')

Output('| 3. CARI BERDASARKAN NAMA |')

Output('| 4. CARI BERDASARKAN KAMAR |')

Output('| 5. TAMPIL DATA KESELURUHAN YANG SUDAH TERURUT |')

Output('| 0. Keluar |')

Output('|-------------------------------------------------|')

Output('| PILIHAN ANDA? |')

Output('|=================================================|')

Input(Menu)

//validasi Menu Pilihan

while (Menu < 0) or (menu > 5) do

Output('Pilihan Hanya Boleh 0/1/2/3/4/5 ulangi Tekan Enter !')

Input(Menu)

endwhile

endprocedure

//prosedure buka file

procedure buka\_file(Input i: integer)

Kamus:

Algoritma:

Open(file\_pasien, Nama\_file)

{$I-}

reSet(file\_pasien)

{$I+}

if (ioresult ≠ 0) then

Output('Terjadi Kesalahan File!!')

rewrite(file\_Pasien)

end

else

I 🡨 1

N 🡨 filesize(file\_pasien)

seek(file\_Pasien,0)

while not eof(file\_pasien) do

seek(file\_pasien,i-1)

Fread(file\_pasien,pasien[i])

I 🡨 i+1

endwhile

endif

endprocedure

//prosedure Memasukan data pasien

procedure IsiDataPasien (Output jumlah:integer, Output pasien:arr\_pasien)

{I.S. : User mamasukan Jumlah pasien & array pasien (1:Jumlah)}

{F.S. : Menghasilakan array pasien (1:jumlah)}

Kamus:

i : integer

procedure buka\_file(Input i: integer)

function kamar(Input kelas:string)🡪string

Algoritma:

buka\_file(i) //memanggil prosedur buka file

rewrite(file\_pasien)

Output('Maksimal data = ',Maks\_pasien)

Output('Banyaknya Data Pasien : ')

Input(jumlah)

while (jumlah<1) or (jumlah>Maks\_pasien) do

Output('Banyaknya Data Pasien Hanya Boleh Antara 1-200 !')

Input(jumlah)

endwhile

Output('DAFTAR PASIEN')

Output('===============')

Output('======================================================')

Output('| ID | NAMA PASIEN | KELAS | KAMAR |')

Output('------------------------------------------------------')

for i 🡨 1 to jumlah do

Output('| | | | |')

Input(Pasien[i].id\_pasien)

Input(Pasien[i].Nama\_pasien)

Input(pasien[i].kelas)

while (uppercase(pasien[i].kelas) ≠ 'VIP') and

(pasien[i].kelas ≠ '1') and

(pasien[i].kelas ≠ '2') and

(pasien[i].kelas ≠ '3') do

Output('Kelas hanya boleh tersedia VIP/1/2/3, ulangi tekan Enter!')

Input(pasien[i].kelas)

endwhile

pasien[i].kamar 🡨 kamar(pasien[i].kelas)

Output(pasien[i].kamar)

seek(file\_pasien, filesize(file\_pasien))

Output(file\_pasien,Pasien[i])

endfor

Output('======================================================')

Output('Lanjutkan tekan Enter..')

close(file\_pasien)

endprocedure

//prosedure Memasukan data pasien

procedure tambahDataPasien (Output n:integer, Output pasien:arr\_pasien)

{I.S. : User mamasukan Jumlah pasien & array pasien (1:Jumlah)}

{F.S. : Menghasilakan array pasien (1:jumlah)}

Kamus:

i,batas : integer

procedure buka\_file(Input i: integer)

function kamar(Input kelas:string)🡪string

Algoritma:

buka\_file(i) //memanggil prosedur buka file

batas 🡨 Maks\_pasien - n

Output('Maksimal data yg bisa ditambahkan = ',batas)

Output('Banyaknya Data Pasien : ')

Input(jumlah)

while (jumlah<1) or (jumlah>batas) do

Output('Banyaknya Data Pasien Hanya Boleh Antara 1-',batas,'!')

Input(jumlah)

endwhile

Output('DAFTAR PASIEN')

Output('===============')

Output('======================================================')

Output('| ID | NAMA PASIEN | KELAS | KAMAR |')

Output('------------------------------------------------------')

for i 🡨 1 to jumlah do

Output('| | | | |')

Input(Pasien[i].id\_pasien)

Input(Pasien[i].Nama\_pasien)

Input(pasien[i].kelas)

while (uppercase(pasien[i].kelas) ≠ 'VIP') and

(pasien[i].kelas ≠ '1') and

(pasien[i].kelas ≠ '2') and

(pasien[i].kelas ≠ '3') do

Output('Kelas hanya boleh tersedia VIP/1/2/3, ulangi tekan Enter!')

Input(pasien[i].kelas)

endwhile

pasien[i].kamar 🡨 kamar(pasien[i].kelas)

Output(pasien[i].kamar)

seek(file\_pasien, filesize(file\_pasien))

Fwrite(file\_pasien,Pasien[i])

endfor

Output('======================================================')

Output('Lanjutkan tekan Enter..')

close(file\_pasien)

endprocedure

//menu isi data

procedure menu\_isi\_data(Input pilihan : integer)

{I.S. : User memilih salah satu menu}

{F.S. : Menampilkan hasil sesuai menu yang dipilih}

Kamus:

procedure IsiDataPasien (Output jumlah:integer, Output pasien:arr\_pasien)

procedure tambahDataPasien (Output n:integer, Output pasien:arr\_pasien)

procedure MenuPilihan (Output Menu: integer)

Algoritma:

Output('Menu Isi Data')

Output('=============')

Output('1. Isi Data (Mengulang dari awal)')

Output('2. Tambah Data')

Output('0. Kembali')

Output('Pilihan Anda?')

Input(pilihan)

//validasi pilihan

while (pilihan<0) or (pilihan>2) do

Output('pilihan hanya 1/2, ulangi tekan Enter!!')

Input(pilihan)

endwhile

depend on(pilihan)

1 : isidatapasien(n,pasien)

2 : tambahdatapasien(n,pasien)

0 : menupilihan(menu)

enddepend

endprocedure

//prosedure Mengurutkan data berdasarkan id dengan minimum sort secara ascending

procedure UrutID\_asc (Input N:integer, Output Pasien:arr\_pasien)

{I.S. : Jumlah pasien & array pasien sudah terdefinisi}

{F.S. : Menghasilkan array pasien yang sudah tersusun secara ascending}

Kamus:

i,j,min : integer

temp : dt\_pasien

procedure buka\_file(Input i: integer)

Algoritma:

buka\_file(i) // memanggil prosedur buka file

for i 🡨 1 to (N-1) do

min 🡨 i

for j 🡨 i + 1 to N do

if(Pasien[j].id\_pasien < pasien[min].id\_pasien)

then

min 🡨 j

endif

endfor

temp 🡨 Pasien[i]

Pasien[i] 🡨 Pasien[min]

Pasien[min] 🡨 temp

endfor

close(file\_pasien)

endprocedure

//prosedure Mengurutkan data berdasarkan id dengan minimum sort secara descending

procedure UrutID\_dsc (Input N:integer, Output Pasien:arr\_pasien)

{I.S. : Jumlah pasien & array pasien sudah terdefinisi}

{F.S. : Menghasilkan array pasien yang sudah tersusun secara descending}

Kamus:

i,j,min : integer

temp : dt\_pasien

procedure buka\_file(Input i: integer)

Algoritma:

buka\_file(i) // memanggil prosedur buka file

for i 🡨 1 to (N-1) do

min 🡨 i

for j 🡨 i + 1 to N do

if(Pasien[j].id\_pasien > pasien[min].id\_pasien)

then

min 🡨 j

endif

endfor

temp 🡨 Pasien[i]

Pasien[i] 🡨 Pasien[min]

Pasien[min] 🡨 temp

endfor

close(file\_pasien)

endprocedure

//prosedure Mengurutkan data berdasarkan nama dengan maximum sort (ascending)

procedure UrutNama\_asc (Input N:integer, Output Pasien:arr\_pasien)

{I.S. : Jumlah pasien & array pasien sudah terdefinisi}

{F.S. : Menghasilkan array pasien yang sudah tersusun secara ascending}

Kamus:

i,j,max : integer

temp : dt\_pasien

procedure buka\_file(Input i: integer)

Algoritma:

buka\_file(i) // memanggil prosedur buka file

for i 🡨 1 to (N-1) do

max 🡨 i

for j 🡨 i + 1 to N do

Algoritma:

if(Pasien[j].nama\_pasien < pasien[max].nama\_pasien)

then

max 🡨 j

endif

endfor

temp 🡨 Pasien[i]

Pasien[i] 🡨 Pasien[max]

Pasien[max] 🡨 temp

endfor

close(file\_pasien)

endprocedure

//prosedure Mengurutkan data berdasarkan nama dengan maximum sort (descending)

procedure UrutNama\_dsc (Input N:integer, Output Pasien:arr\_pasien)

{I.S. : Jumlah pasien & array pasien sudah terdefinisi}

{F.S. : Menghasilkan array pasien yang sudah tersusun secara descending}

Kamus:

i,j,max : integer

temp : dt\_pasien

procedure buka\_file(Input i: integer)

Algoritma:

buka\_file(i) // memanggil prosedur buka file

for i 🡨 1 to (N-1) do

max 🡨 i

for j 🡨 i + 1 to N do

if(Pasien[j].nama\_pasien > pasien[max].nama\_pasien)

then

max 🡨 j

endif

endfor

temp 🡨 Pasien[i]

Pasien[i] 🡨 Pasien[max]

Pasien[max] 🡨 temp

endfor

close(file\_pasien)

endprocedure

//prosedure mengurutkan data berdasarkan Kelas dengan bubble sort (ascending)

procedure UrutKelas\_asc (Input N:integer, Output Pasien:arr\_pasien)

{I.S. : Jumlah pasien & array pasien sudah terdefinisi}

{F.S. : Menghasilkan array pasien yang sudah tersusun secara ascending}

Kamus:

i,j : integer

temp : dt\_pasien

procedure buka\_file(Input i: integer)

Algoritma:

buka\_file(i) // memanggil prosedur buka file

for i 🡨 1 to N-1 do

for j 🡨 N downto (i+1) do

if (Pasien[j].kelas < pasien[j-1].kelas)

then

temp 🡨 pasien[j]

pasien[j] 🡨 pasien[j-1]

pasien[j-1] 🡨 temp

endif

endfor

endfor

close(file\_pasien)

endprocedure

//prosedure mengurutkan data berdasarkan Kelas dengan bubble sort (descending)

procedure UrutKelas\_dsc (Input N:integer, Output Pasien:arr\_pasien)

{I.S. : Jumlah pasien & array pasien sudah terdefinisi}

{F.S. : Menghasilkan array pasien yang sudah tersusun secara descending}

Kamus:

i,j : integer

temp : dt\_pasien

procedure buka\_file(Input i: integer)

Algoritma:

buka\_file(i) // memanggil prosedur buka file

for i 🡨 1 to N-1 do

for j 🡨 N downto (i+1) do

if (Pasien[j].kelas > pasien[j-1].kelas)

then

temp 🡨 pasien[j]

pasien[j] 🡨 pasien[j-1]

pasien[j-1] 🡨 temp

endif

endfor

endfor

close(file\_pasien)

endprocedure

//prosedure mengurutkan data berdasarkan Kamar dengan bubble sort (ascending)

procedure UrutKamar\_asc (Input N:integer, Output Pasien:arr\_pasien)

{I.S. : Jumlah pasien & array pasien sudah terdefinisi}

{F.S. : Menghasilkan array pasien yang sudah tersusun secara ascending}

Kamus:

i,j : integer

temp : dt\_pasien

procedure buka\_file(Input i: integer)

Algoritma:

buka\_file(i) // memanggil prosedur buka file

for i 🡨 1 to N-1 do

for j 🡨 N downto (i+1) do

if (Pasien[j].kelas < pasien[j-1].kelas)

then

temp 🡨 pasien[j]

pasien[j] 🡨 pasien[j-1]

pasien[j-1] 🡨 temp

endif

endfor

endfor

close(file\_pasien)

endprocedure

//prosedure mengurutkan data berdasarkan Kamar dengan bubble sort (descending)

procedure UrutKamar\_dsc (Input N:integer, Output Pasien:arr\_pasien)

{I.S. : Jumlah pasien & array pasien sudah terdefinisi}

{F.S. : Menghasilkan array pasien yang sudah tersusun secara descending}

Kamus:

i,j : integer

temp : dt\_pasien

procedure buka\_file(Input i: integer)

Algoritma:

buka\_file(i) // memanggil prosedur buka file

for i 🡨 1 to N-1 do

for j 🡨 N downto (i+1) do

if (Pasien[j].kelas > pasien[j-1].kelas)

then

temp 🡨 pasien[j]

pasien[j] 🡨 pasien[j-1]

pasien[j-1] 🡨 temp

endif

endfor

endfor

close(file\_pasien)

endprocedure

//prosedure Menampilkan data pasien

procedure TampilDataPasien (Input N:integer, Input Pasien:arr\_pasien)

{I.S. : Jumlah pasien & array pasien sudah terdefinisi}

{F.S. : Menampilkan data-data pasien}

Kamus:

i:integer

procedure buka\_file(Input i: integer)

function harga(Input kelas:string)🡪integer

Algoritma:

buka\_file(i) // memanggil prosedur buka file

Output('DAFTAR PASIEN')

Output('===============')

Output('======================================================================')

Output('| ID | NAMA PASIEN | KELAS | KAMAR | HARGA KAMAR |')

Output('----------------------------------------------------------------------')

for i 🡨 1 to N do

Output('| | | | | |')

Output(Pasien[i].id\_pasien)

Output(pasien[i].Nama\_pasien)

Output(pasien[i].kelas)

Output(pasien[i].kamar)

pasien[i].harga 🡨 harga(pasien[i].kelas)

Output(pasien[i].harga,' /hari')

endfor

Output('======================================================================')

endprocedure

procedure Menu\_Urut\_asc(Input pilihan:integer)

{I.S. : User memilih salah satu menu}

{F.S. : Menampilkan hasil sesuai menu yang dipilih}

Kamus:

procedure UrutID\_asc (Input N:integer, Output Pasien:arr\_pasien)

procedure UrutNama\_asc (Input N:integer, Output Pasien:arr\_pasien)

procedure UrutKelas\_asc (Input N:integer, Output Pasien:arr\_pasien)

procedure UrutKamar\_asc (Input N:integer, Output Pasien:arr\_pasien)

procedure TampilDataPasien (Input N:integer, Input Pasien:arr\_pasien)

Algoritma:

Output('Menu Urut')

Output('=========')

Output('1. Urut Berdasar ID')

Output('2. Urut Berdasar Nama')

Output('3. Urut Berdasar Kelas')

Output('4. Urut Berdasar Kamar')

Output('0. Kembali')

Output('Pilihan Anda?')

Input(pilihan)

//validasi pilihan

while (pilihan<0) or (pilihan>4) do

Output('pilihan hanya 1/2/3/4, ulangi tekan Enter!!')

Input(pilihan)

endwhile

depend on(pilihan)

1 : UrutID\_asc(n,pasien)

tampildatapasien(n,pasien)

2 : UrutNama\_asc(n,pasien)

tampildatapasien(n,pasien)

3 : UrutKelas\_asc(n,pasien)

tampildatapasien(n,pasien)

4 : UrutKamar\_asc(n,pasien)

tampildatapasien(n,pasien)

0 : menupilihan(menu)

enddepend

endprocedure

procedure Menu\_Urut\_dsc(Input pilihan:integer)

{I.S. : User memilih salah satu menu}

{F.S. : Menampilkan hasil sesuai menu yang dipilih}

Kamus:

procedure UrutID\_dsc (Input N:integer, Output Pasien:arr\_pasien)

procedure UrutNama\_dsc (Input N:integer, Output Pasien:arr\_pasien)

procedure UrutKelas\_dsc (Input N:integer, Output Pasien:arr\_pasien)

procedure UrutKamar\_dsc (Input N:integer, Output Pasien:arr\_pasien)

procedure TampilDataPasien (Input N:integer, Input Pasien:arr\_pasien)

Algoritma:

Output('Menu Urut')

Output('=========')

Output('1. Urut Berdasar ID')

Output('2. Urut Berdasar Nama')

Output('3. Urut Berdasar Kelas')

Output('4. Urut Berdasar Kamar')

Output('0. Kembali')

Output('Pilihan Anda?')

Input(pilihan)

//validasi pilihan

while (pilihan<0) or (pilihan>4) do

Output('pilihan hanya 1/2/3/4, ulangi tekan Enter!!')

Input(pilihan)

endwhile

depend on(pilihan)

1: UrutID\_dsc(n,pasien)

tampildatapasien(n,pasien)

2: UrutNama\_dsc(n,pasien)

tampildatapasien(n,pasien)

3: UrutKelas\_dsc(n,pasien)

tampildatapasien(n,pasien)

4: UrutKamar\_dsc(n,pasien)

tampildatapasien(n,pasien)

0: menupilihan(menu)

enddepend

endprocedure

procedure Menu\_Urut(Input pilihan:integer)

{I.S. : User memilih salah satu menu}

{F.S. : Menampilkan hasil sesuai menu yang dipilih}

Kamus:

procedure Menu\_Urut\_asc(Input pilihan:integer)

procedure Menu\_Urut\_dsc(Input pilihan:integer)

procedure MenuPilihan (Output Menu: integer)

Algoritma:

Output('Menu Urut')

Output('=========')

Output('1. Urut Secara Ascending(kecil ke besar)')

Output('2. Urut Secara Descending(besar ke kecil)')

Output('0. Kembali')

Output('Pilihan Anda?')

Input(pilihan)

//validasi pilihan

while (pilihan<0) or (pilihan>2) do

Output('pilihan hanya 1/2, ulangi tekan Enter!!')

Input(pilihan)

endwhile

depend on(pilihan)

1: Menu\_urut\_asc(pilihan)

2: menu\_Urut\_dsc(pilihan)

0: menupilihan(menu)

enddepend

endprocedure

//prosedure menyimpan ke file

procedure simpan\_file(Input i:integer)

Kamus:

Algoritma:

rewrite(file\_pasien)

for i 🡨 1 to N do

seek(file\_pasien,i-1)

Output(file\_pasien,pasien[i])

endfor

close(file\_pasien)

endprocedure

procedure Cariid (Input N : integer,Input pasien : arr\_pasien)

{I.S. : Elemen array yg terurut sudah terdefinisi}

{F.S. : Menampilkan data yang di cari ditemukan atau tidak ditemukan}

Kamus:

ia,ib,k : integer

ketemu : boolean

id : string

Algoritma:

buka\_file(i) // memanggil prosedur buka file

Output('id Pasien yang Dicari : ')

Input(id)

Ia 🡨 1

Ib 🡨 N

Ketemu 🡨 false

while (not ketemu) and (Ia ≤ Ib) do

K 🡨 (Ia + Ib) div 2

if (Pasien[K].id\_pasien = id)

then

ketemu 🡨 true

else

if (pasien[K].id\_pasien < id)

then

Ia 🡨 K + 1 //pencarian dilanjutkan ke kanan

else

Ib 🡨 K - 1 //pencarian dilanjutkan ke kiri

endif

endif

endwhile

if (ketemu)

then

Output('DAFTAR PASIEN')

Output('===============')

Output('Pasien dengan ID : ',id)

Output('======================================================')

Output('| ID | Nama Pasien | Kelas | Kamar |')

Output('------------------------------------------------------')

Output('| | | | |')

Output(id)

Output(Pasien[k].Nama\_pasien)

Output(Pasien[k].kelas)

Output(pasien[k].kamar)

Output('======================================================')

pasien[k].harga 🡨 harga(pasien[k].kelas)

else

Output('id Pasien ',id,' tidak ada!')

endif

close(file\_pasien)

endprocedure

//precedure cari berdasarkan nama

Procedure CariNama (Input N:integer,Input pasien:arr\_pasien,Input NmP:string)

{I.S. : Elemen array sudah terdefinisi}

{F.S. : Menampilkan data yang di cari ditemukan atau tidak ditemukan}

Kamus:

i,j,baris : integer

temp : arr\_pasien

Algoritma:

buka\_file(i) // memanggil prosedur buka file

Output('Nama Pengarang yang dicari: ')

Input(Nmp)

i 🡨 1

j 🡨 0

while (i ≤ N) do

if (pos(NmP, Pasien[i].Nama\_pasien) > 0 )

then

j 🡨 J + 1

Temp[j] 🡨 Pasien[i]

endif

i 🡨 i+1

endwhile //endwhile

if (j>0)

then

Output('DATA PASIEN')

Output('=============')

Output('Pasien dengan Nama : ',NmP)

Output('==========================================================')

Output('|NO | Nama Pasien | ID | Kelas | Kamar |')

Output('----------------------------------------------------------')

baris 🡨 8

for i 🡨 1 to J do

Output('| | | | | |')

Output(i)

Output(temp[i].Nama\_pasien)

Output(temp[i].id\_pasien)

Output(temp[i].kelas)

Output(temp[i].kamar)

baris 🡨 baris + 1

endfor

Output('==========================================================')

else

Output('Pasien dengan Nama ',Nmp,' tidak ditemukan!')

endif

endprocedure

//prosedure cari berdasarkan kamar

procedure carikamar(Input N:integer,Input pasien:arr\_pasien,Input NmK:string)

{I.S. : Elemen array sudah terdefinisi}

{F.S. : Menampilkan data yang di cari ditemukan atau tidak ditemukan}

Kamus:

i,j,baris : integer

temp : arr\_pasien

Algoritma:

buka\_file(i) // memanggil prosedur buka file

Output('Nama Kamar yang dicari: ')

Input(NmK)

i 🡨 1

j 🡨 0

while(i ≤ N) do

if (pos(NmK,Pasien[i].Kamar) > 0 )

then

j 🡨 J + 1

Temp[j] 🡨 Pasien[i]

endif

i 🡨 i+1

endwhile //endwhile

if (j>0)

then

Output('DATA PASIEN')

Output('=============')

Output('Kamar dengan Nama : ',NmK)

Output('==========================================================')

Output('|NO | Kamar | Nama Pasien | ID | Kelas |')

Output('----------------------------------------------------------')

baris 🡨 8

for i 🡨 1 to J do

Output('| | | | | |')

Output(i)

Output(temp[i].kamar)

Output(temp[i].Nama\_pasien)

Output(temp[i].id\_pasien)

Output(temp[i].kelas)

baris 🡨 baris + 1

endfor

Output('==========================================================')

else

Output('Kamar ',Nmk,' tidak ada!')

endif

endprocedure

**{Algoritma utama}**

PasienRawatInap

Kamus:

const

nama\_file = 'data\_pasien.txt'

Maks\_pasien = 200

type

dt\_pasien = record

id\_pasien, Nama\_pasien : string,

kelas, kamar : string,

harga,total : integer

endrecord

arr\_pasien = array [1..Maks\_pasien] of dt\_pasien

Pasien : arr\_pasien

N,Menu,pilihan : integer

file\_pasien : file of dt\_pasien

NmP,NmK : string //NmP = Nama pasien, NmK = Nama kamar

i,jumlah : integer

Algoritma:

n 🡨 0

repeat

buka\_file(i) //memanggil prosedure buka file

simpan\_file(i) //memanggil prosedure simpan ke file

menupilihan(menu)

depend on (menu)

1: menu\_isi\_data(pilihan)

2: urutID\_asc(n, pasien)

simpan\_file(i)

cariid(n, pasien)

3: carinama(n,pasien,nmp)

close(file\_pasien)

4: carikamar(n,pasien,nmk)

close(file\_pasien)

5: menu\_urut(pilihan)

simpan\_file(i)

enddepend

until (menu = 0)

Output('Terima Kasih Telah Menggunakan Program Ini')