



BAB V

BAB V

IMPLEMENTASI PERANGKAT LUNAK

5.1 Implementasi

Tujuan dari tahap implementasi ini untuk memastikan perangkat lunak yang dibuat dapat bekerja secara efektif dan efisien sesuai yang diinginkan, untuk itu pada tahap implementasi ini akan menjelaskan, antara lain, bahasa pemrograman yang dipakai dan lingkungan pengembangan. Berikut implementasi dari sistem perangkat lunak untuk masalah penentuan lokasi fasilitas dengan menggunakan metode *fuzzy c-means*.

5.1.1 Pemilihan Bahasa Pemrograman

Sistem perangkat lunak yang akan dibuat adalah sebuah perangkat lunak yang berisi banyak perhitungan, peraturan logika, pengembangan visual, pola desain dan pemakaian pada lingkup kerjanya, dan fleksibilitas. Maka dibutuhkan sebuah bahasa pemrograman yang terstruktur, handal, praktis, mudah digunakan dan juga mendukung batasan sistem operasi secara umum. Borland Delphi 6.0 adalah salah satu bahasa pemrograman yang memiliki semua persyaratan yang dibutuhkan dalam proses pembuatan perangkat lunak ini.

5.1.2 Implementasi Antarmuka

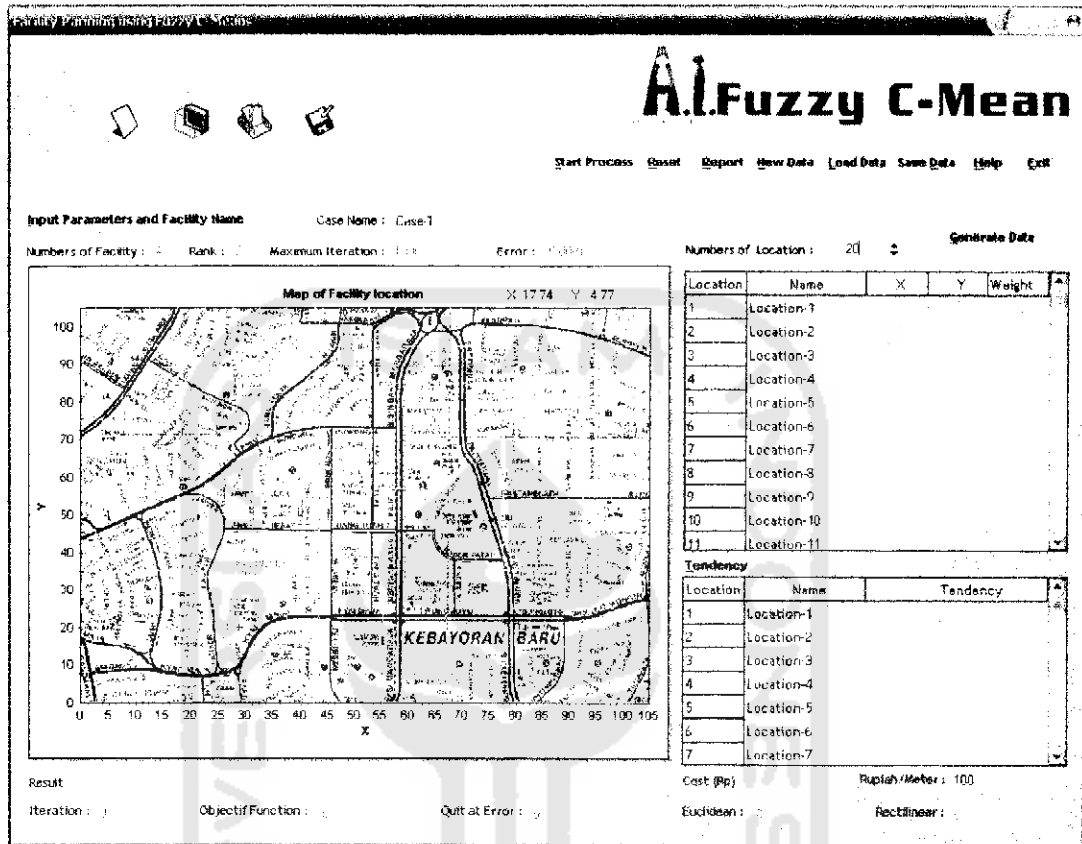
Berikut ini adalah gambaran antarmuka-antarmuka yang terdapat pada sistem perangkat lunak menentukan lokasi fasilitas menggunakan *fuzzy c-means*.

5.1.2.1 Antarmuka Menu Utama

Antarmuka ini merupakan menu utama untuk melakukan proses. Terdapat beberapa komponen pada antarmuka menu utama ini:

1. Grafik untuk peta lokasi fasilitas: untuk menampilkan lokasi fasilitas yang optimal.
2. *Input Parameter dan Facility Name*: merupakan cara penginputan *numbers of facility, maximum iterasi, error, rank* dan untuk mengubah nama fasilitas yang akan menandakan menandakan sebagai pusat fasilitas pada peta lokasi fasilitas.
3. *Input data lokation* : untuk menentukan banyak *location* yang kita ingin kita clusterkan
4. *Edit data coordinate dan weight*: untuk mengubah nilai *coordinate* dan *weight* yang sudah ada pada stringgrid.
5. Tombol *Load data* : untuk pemanggilan *file* yang berisi data *coordinate* dan *weight*
6. Tombol *Save data* : untuk menyimpan *coordinate* dan *weight*
7. Tobol *Generate data* : untuk merandom *coordinate* dan *weight* yang akan ditampilkan pada stringgrid, untuk menentukan lokasi pada peta lokasi fasilitas.

8. Tombol *Start Process* : untuk menjalankan proses dalam menentukan lokasi fasilitas yang optimal.
9. Tombol *report* : informasi hasil proses perhitungan.
10. Tombol *Help* : untuk membantu seputar penggunaan sistem.
11. Tombol *Exit* : untuk keluar dari sistem.
12. Tombol *Insert Map* : untuk mengambil gambar map yang akan diakan ditampilkan pada latar grafik.
13. Tombol *Clear Map* : untuk membersihkan tampilan map pada latar grafik.
14. Tombol *Load Graphics* : untuk memanggil *file* grafik yang pernah disimpan sebelumnya.
15. Tombol *Save Graphics* : untuk menyimpan grafik.
16. Tombol *Full Screen* : Untuk membesarkan tampilan pada grafik agar grafik tampak lebih jelas dalam menginputkan lokasi fasilitas.
17. Tombol *Terminate* : untuk menghentikan proses iterasi yang sedang berjalan.
18. Tombol *Reset* : untuk mengulang proses dan siap untuk melakukan proses kembali.
19. Tombol *Tendency* : untuk menampilkan informasi tendency yang didapat.
20. Tombol *Coordinate of Facility Center* : untuk menampilkan informasi *Coordinate of Facility Center*.



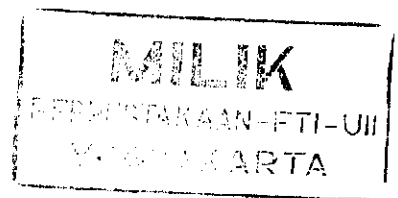
Gambar 5.1 Antarmuka Menu Utama.

5.1.2.2 Antarmuka Peta Full Screen

Antarmuka ini digunakan untuk memperbesar peta atau full Screen, agar peta terlihat lebih jelas dan besar. Gambar (5.2) adalah gambar antarmuka Peta Full Screen.

5.1.2.3 Antarmuka *Edit Input Data Parameter*

Antarmuka ini digunakan untuk membuat atau *edit input data*. Pada antarmuka ini data yang diinputkan adalah *numbers of facility*, *maximum iterasi*, *rank*, *error* dan nama fasilitas yang akan menandakan sebagai pusat fasilitas pada peta lokasi fasilitas. Gambar (5.3) adalah gambar antarmuka *Edit input data*.



Facility	Name
1	Facility-1
2	Facility-2
3	Facility-3
4	Facility-4

Gambar 5.3 Antarmuka *Edit input Data Parameter*.

5.1.2.4 Antarmuka *Edit coordinate dan weight*

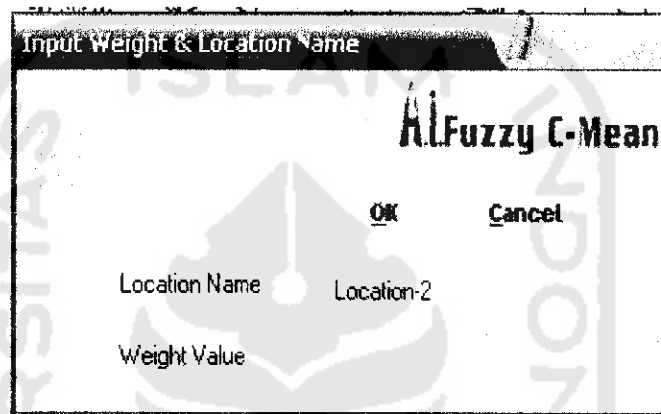
Antarmuka ini digunakan untuk merubah nilai yang berada pada *coordinate* dan *weight* pada stringgrid, yang berfungsi sebagai derajat keanggotaan lokasi.

Gambar (5.4) adalah gambar antarmuka *Edit coordinate dan weight*.

Gambar 5.4 Antarmuka *Edit coordinate dan weight*.

5.1.2.5 Antarmuka *Input Weight* dan *Location Name*

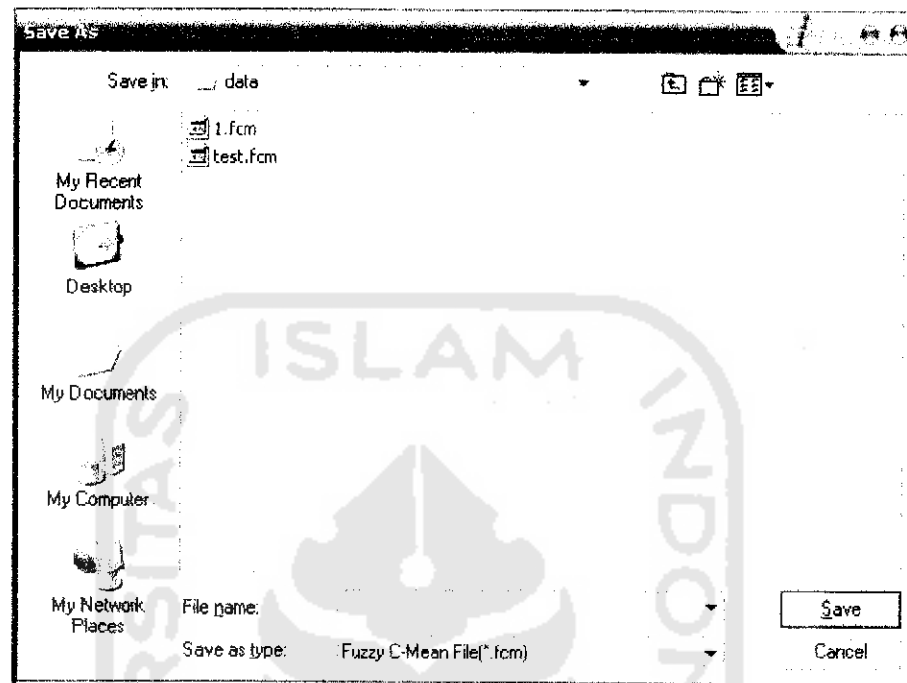
Antarmuka ini digunakan untuk menginputkan *Location Name* dan *Weight* saat menentukan lokasi pada peta secara manual, maka akan muncul form input weight dan location name. Gambar (5.5) Antarmuka *input Weight* dan *Location Name*.



Gambar 5.5 Antarmuka *input Weight* dan *Location Name*.

5.1.2.6 Antarmuka *Save Data*

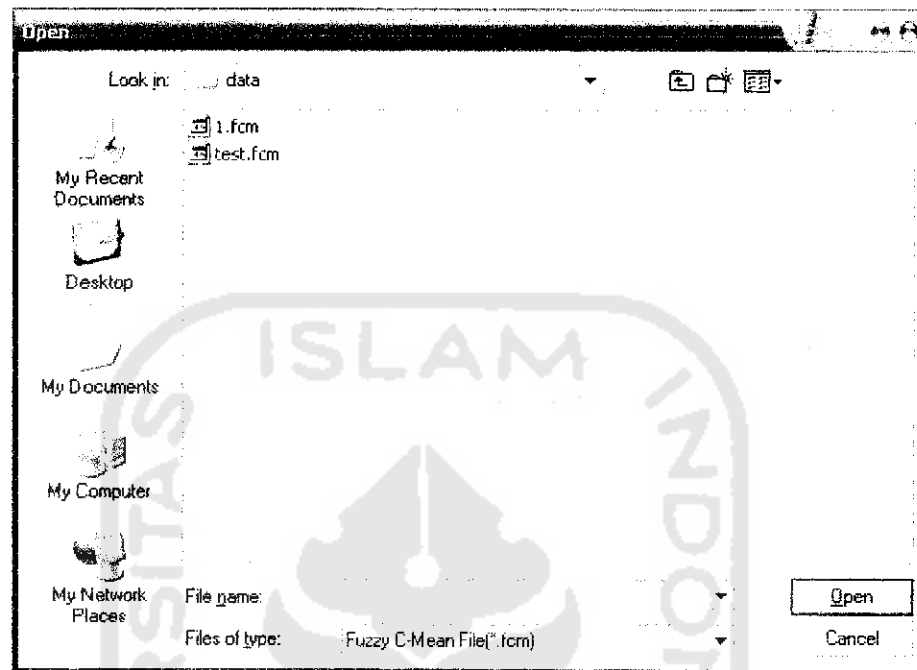
Antarmuka ini digunakan untuk menyimpan data *locations*, *coordinate*, dan *weight* yang berada pada stringgrid. *file* yang digunakan untuk menyimpan hasil dari proses penentuan lokasi fasilitas adalah berupa file yang berekstensi *.fcm* (5.6) adalah Antarmuka *Save data*.



Gambar 5.6 Antarmuka *Save Data*.

5.1.2.7 Antarmuka *Load Data*

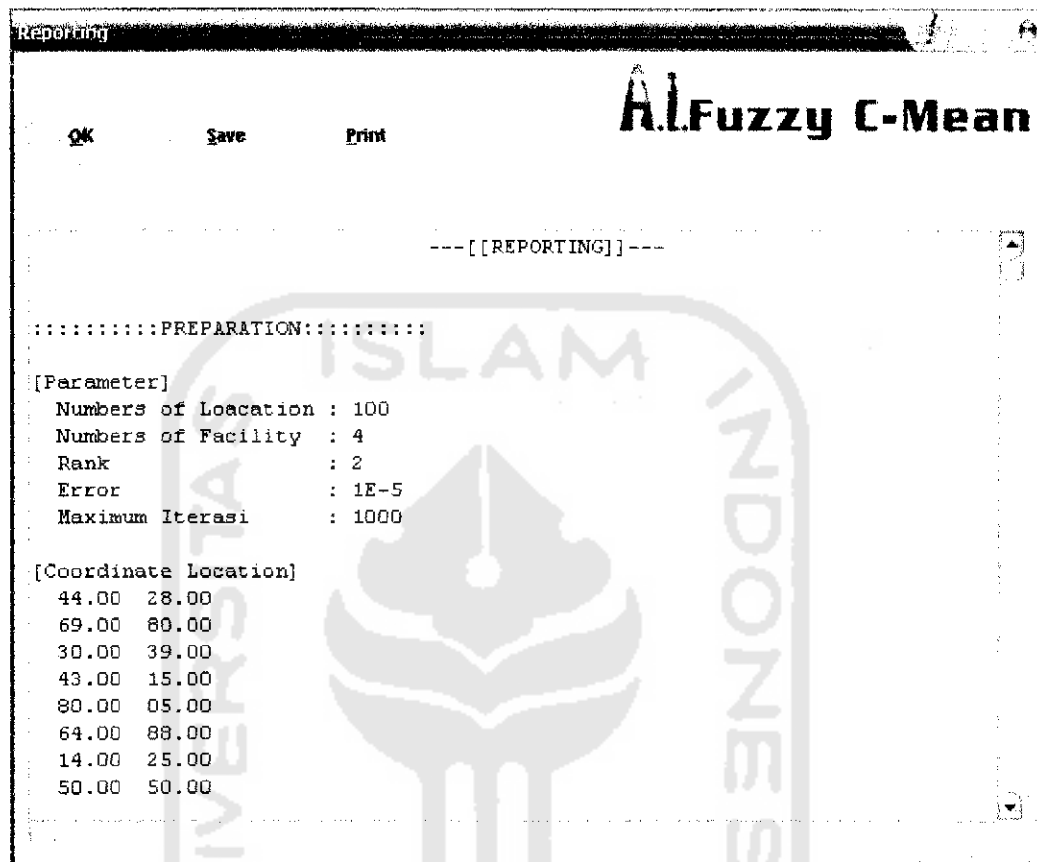
Antarmuka ini digunakan untuk mengambil data *coordinate* dan *weight* yang sebelumnya sudah pernah disimpan. *file* yang digunakan untuk mengambil data hasil dari penyimpanan sebelumnya untuk lokasi fasilitas berupa *file* yang berekstensi *.fcm*. Gambar (5.7) adalah Antarmuka Load data



Gambar 5.7 Antarmuka *Load Data*.

5.1.2.8 Antarmuka *Report*

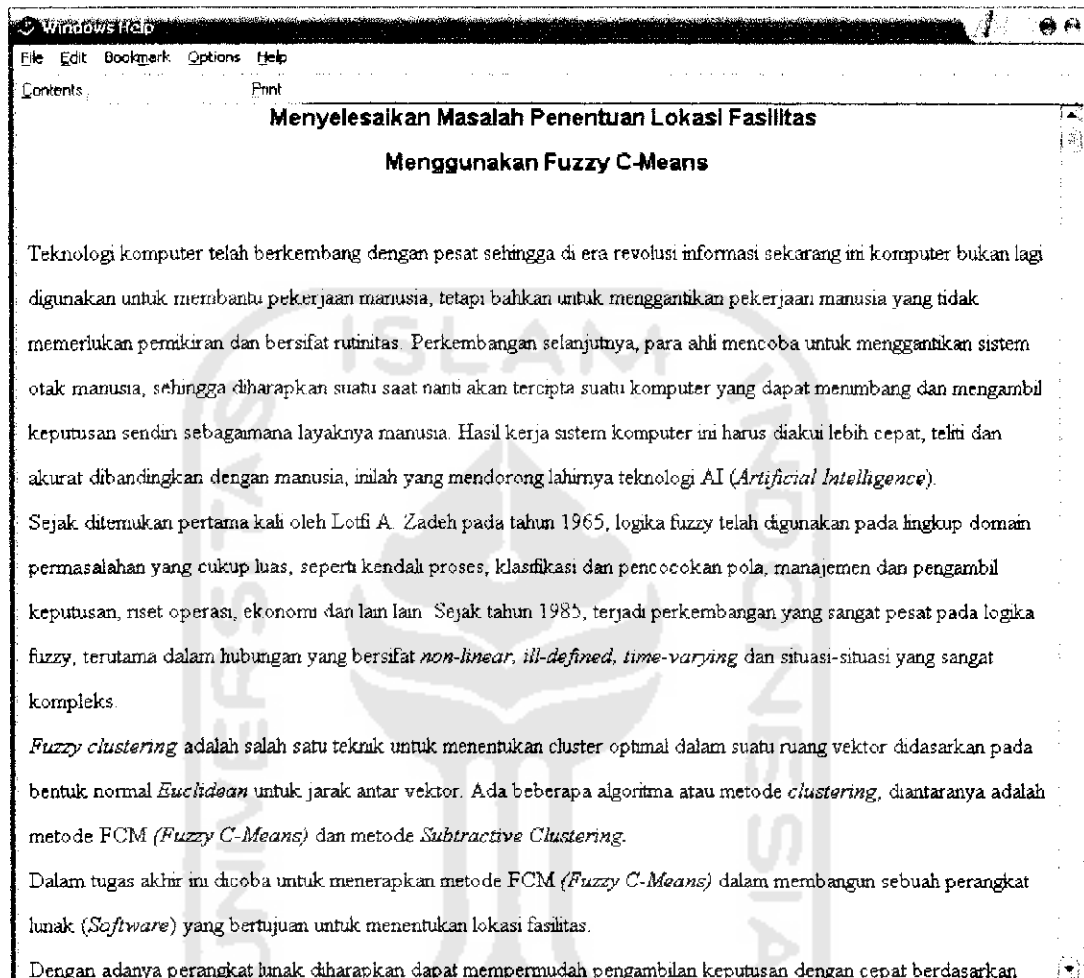
Antarmuka ini digunakan untuk memberikan informasi hasil proses perhitungan sebanyak iterasi perulangannya, dalam menyelesaikan masalah penentuan lokasi fasilitas menggunakan *fuzzy C-Means*. Gambar (5.8) adalah antarmuka *Report*.



Gambar 5.8 Antarmuka *Report*.

5.1.2.9 Antarmuka *Help*

Antarmuka ini digunakan untuk membantu user dalam menggunakan sistem dan beberapa informasi lainnya yang berhubungan dengan penentuan lokasi fasilitas menggunakan *fuzzy c-means*. Gambar (5.9) adalah Antarmuka *Help*



Gambar 5.9 Antarmuka *Help*.

5.2 Perancangan Procedure

Pada perancangan Perocedur ini adalah proses perancangan pada penentuan lokasi fasilitas menggunakan *fuzzy c-means* secara keseluruhan.

5.2.1 Perancangan Procedure Fuzzy C-Means

Dari diagram alir Gambar (4.2) dapat dijabarkan procedurnya sebagai berikut:

Prosedure Matrik Partisi Awal

```

Var
Jumlah lokasi                :Integer
Jumlah fasilitas             :Integer
I                             :Integer
J                             :Integer
Matrix partisi               :array[1..50,1..500] of real
Sigma matrix partisi         :array[1..50] of real
Begin
for i=1 to jumlah_fasilitas do
  begin
    for j=1 to jumlah_lokasi do
      begin
        matrix partisi (i,j)=random;
        Sigma matrix Partisi [i]= Sigma matrix partisi (i)+matrix partisi [i,j]
      end
    end
  end
for i=1 to jumlah_fasilitas do
  begin
    for j=1 to jumlah_lokasi do
      begin
        matrix partisi(i,j)=matrix partisi (i,j)/Sigma matrix partisi (i)
      end
    end
  end
end

```

Procedure Pusat Fasilitas

```

Var
Jumlah lokasi                :Integer;
Jumlah fasilitas             :Integer;
I                             :Integer
J                             :Integer
K                             :Integer
Matrix partisi               :Array[1..50,1..500] of real
Matrik partisi pangkat       :Array[1..50,1..500] of real
Sigma partisi pangkat        :Array[1..50] of real
Kordinat                    :Array[1..2,1..50] of real
Sigma partisi pangkat kordinat:Array[1..2,1..50] of real
Pusat fasilitas              :Array[1..2,1..50] of real
Begin
for i=1 to jumlah_fasilitas do
  begin
    Sigma matrix partisi pangkat(i)=0
  end
  for j=1 to jumlah_lokasi do
    begin
      for i =1 to jumlah_fasilitas do
        begin
          matrix partisi pangkat(i,j):=(exp(ln(matrix partisi(i,j))*pangkat))
          sigma partisi pangkat(i)= sigma partisi pangkat(i)+matrix partisi
            pangkat(i,j)
        end
      end
    end
  end
for i=1 to jumlah_fasilitas do
  begin
    for k=1 to 2 do
      begin
        temp=0
        for j=1 to jumlah_lokasi do
          begin
            temp=temp+(matrix partisi pangkat(i,j)*kordinat (k,j))
          end
        end
      end
    end
  end
end

```

```

        end
        Sigma partisi pangkat kordinat(k,i)=temp
        Pusat fasilitas (k,i)=sigam partisi pangkat koordinat(k,i)/sigma Partisi
        pangkat(i)
    end
end
end

```

Procedure Fungsi Objektif

```

Var
Jumlah lokasi           :Integer;
Jumlah fasilitas        :Integer;
I                       :Integer;
J                       :Integer;
K                       :Integer;
Matrik partisi pangkat  :Array[1..50,1..500] of real;
Matrix L objektif       :Array[1..50,1..500] of real;
Pusat fasilitas         :Array[1..2,1..50] of real;
Kordinat                :Array[1..2,1..50] of real;
Fungsi_objektif_1       :Real;
Fungsi_objektif         :Real;
Begin
for i=1 to jumlah fasilitas do
begin
for j=1 to jumlah lokasi do
begin
temp=0
for k=1 to 2 do
begin
temp=temp+(sqr(koordinat(k,j)-pusat_fasilitas(k,i)))
end
matrix L obyektif(i,j)=temp*matrix Partisi pangkat(i,j)
end
end
end
fungsi_objektif_1=fungsi objektif
fungsi_objektif=0
for j=1 to jumlah lokasi do
begin
temp=0
for i=1 to jumlah fasilitas do
begin
temp=temp+matrix L obyektif(i,j)
end
fungsi objektif=fungsi objektif+temp
end
end
end

```

Procedure Perubahan Partisi

```

Var
Jumlah lokasi           :Integer;
Jumlah fasilitas        :Integer;
I                       :Integer;
J                       :Integer;
Matrix L update partisi :Array[1..50,1..500] of real;
Matrix LT update partisi :Array[1..50,1..500] of real;
Matrik partisi          :Array[1..50,1..500] of real;
Begin
for i=1 to jumlah fasilitas do
begin

```

```

    for j=1 to jumlah lokasi do
    begin
        matrix L update partisi(i,j):=(exp(ln(temp)*(-1/(pangkat-1))))
    end
    end
end
for j=1 to jumlah lokasi do
begin
    matrix LT update partisi(j)=0
end
for j=1 to jumlah lokasi do
begin
    for i=1 to jumlah fasilitas do
    begin
matrix LT update partisi(j)=matrix LT update partisi(j)+matrix L update partisi(i,j)
    end
    end
end
for j=1 to jumlah lokasi do
begin
    for i=1 to jumlah fasilitas do
    begin
        matrix partisi(i,j)=matrix L update partisi(i,j)/matrix LT update partisi(j)
    end
    end
end
end
end

```

Procedure Informasi Kelompok Fasilitas

```

var
Jumlah lokasi           :Integer
Jumlah fasilitas        :Integer
I                       :Integer
J                       :Integer
Matrix partisi          :Array[1..50,1..500] of real
Begin
for j=1 to jumlah lokasi do
begin
    maksimum=-1000000000
    for i=1 to jumlah fasilitas do
    begin
        if maksimum<matrix partisi(i,j) then
        begin
            lokasi(j)=i
            maksimum=matrix partisi(i,j)
        end
    end
    end
end
end
end
end

```

Procedure Informasi Total biaya

```

Var
Jumlah lokasi           :Integer
I                       :Integer
Pusat fasilitas         :Array[1..2,1..50] of real
Kordinat                :Array[1..2,1..50] of real
matrix bobot            :Array[1..50] of real
rectilinear             :Real
euclidean               :Real
Bagin
rectilinear=0
euclidean=0

```

```

for i=1 to jumlah lokasi do
  begin
    euclidean=euclidean+(sqrt(sqr(kordinat(1,i)-pusat_fasilitas(i)))+
      sqr(kordinat(2,i)-pusat_fasilitas(i)))*matrix_bobot(i))
    rectilinear=rectilinear+((abs(kordinat(1,i)-pusat_fasilitas(1,lokasi(i)))+
      abs(kordinat(2,i)-pusat_fasilitas(2,lokasi(i)))*matrix_bobot(i))
  end
Euclidean=euclidean*rupiah
Rectilinear=rectilinear*Rupiah
end

```

