

BAB V

IMPLEMENTASI PERANGKAT LUNAK

5.1. Implementasi Secara Umum

Analisis Perbandingan Metode Penyelesaian Persoalan Nilai Eigen dan Vektor Eigen Sebagai Penyelesaian Persamaan Homogen Berukuran Besar dengan metode Faddeev-Leverrier – Bairstow dan metode Power – Deflation diimplementasikan dengan bahasa pemrograman Borland Delphi 6.0. Implementasi merupakan salah satu tahap di mana sistem dioperasikan pada keadaan yang sebenarnya, sehingga akan diketahui apakah sistem yang dibuat benar-benar sesuai dengan yang diharapkan.

Sebelum program diimplementasikan, maka program harus bebas dari kesalahan. Tetapi apabila sistem dalam kondisi yang tidak baik, kesalahan pun tidak dapat dihindari. Kesalahan program yang mungkin terjadi antara lain karena kesalahan penulisan (*coding*), kesalahan proses, atau kesalahan logika.

5.2. Tahap Pembuatan Perangkat Lunak

Dalam pembuatan perangkat lunak ini terdapat beberapa tahapan yang dapat dibedakan menjadi 3 tahap, yaitu :

a. Tahap Pemrograman Visual

Pada tahap ini yang dilakukan adalah merancang form yang akan digunakan program serta kontrol-kontrol yang diperlukan. Perancangan tersebut ditangani oleh paket-paket yang disediakan oleh Borland Delphi 6.0.

b. Tahap Penulisan Kode (*Coding*)

Pada tahap ini yang dilakukan adalah menuliskan kode-kode yang diimplementasikan dengan procedure untuk selanjutnya ditempatkan atau dipanggil pada kontrol-kontrol (*object*) yang dipakai. Penulisan kode ini dilakukan dengan menggunakan *Text Editor* milik Borland Delphi 6.0.

c. Tahap *Debugging*

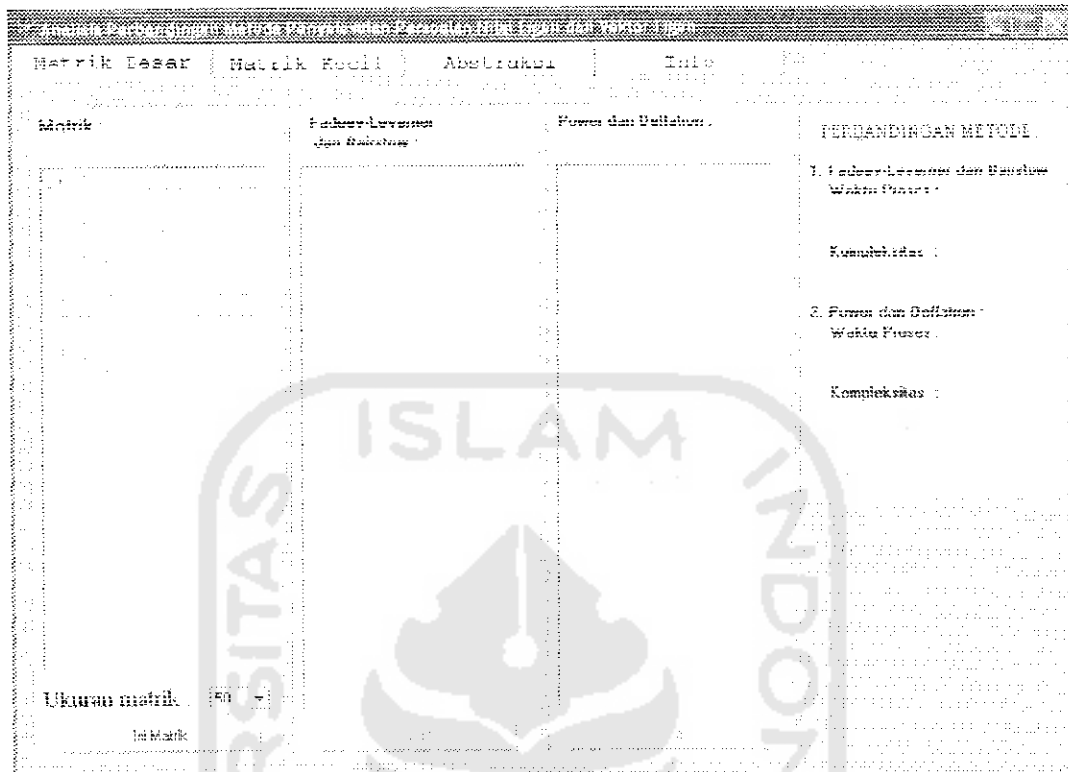
Pada tahap ini dilakukan pengujian terhadap perangkat lunak yang dibuat. Metode pengujian yang digunakan ialah metode *Try and Error*, dimana setiap langkah yang menghasilkan output diteliti kembali keabsahannya, sehingga hasil yang didapatkan benar-benar akurat dengan metode yang digunakan.

5.3. Implementasi Antarmuka

5.3.1. Menu Matrik Besar

Tampilan menu matrik besar berisi fasilitas untuk membandingkan keempat metode dari data yang diambil secara random. Pada menu ini, pengguna dapat menginputkan ukuran matrik yang diinginkan dan dapat melihat hasil dari perhitungan dari setiap metode yang diinginkan.

Kemudian pengguna dapat melihat dengan jelas perbedaan dari hasil perhitungan untuk masing-masing metode berdasarkan waktu komputasi dan kompleksitas algoritma

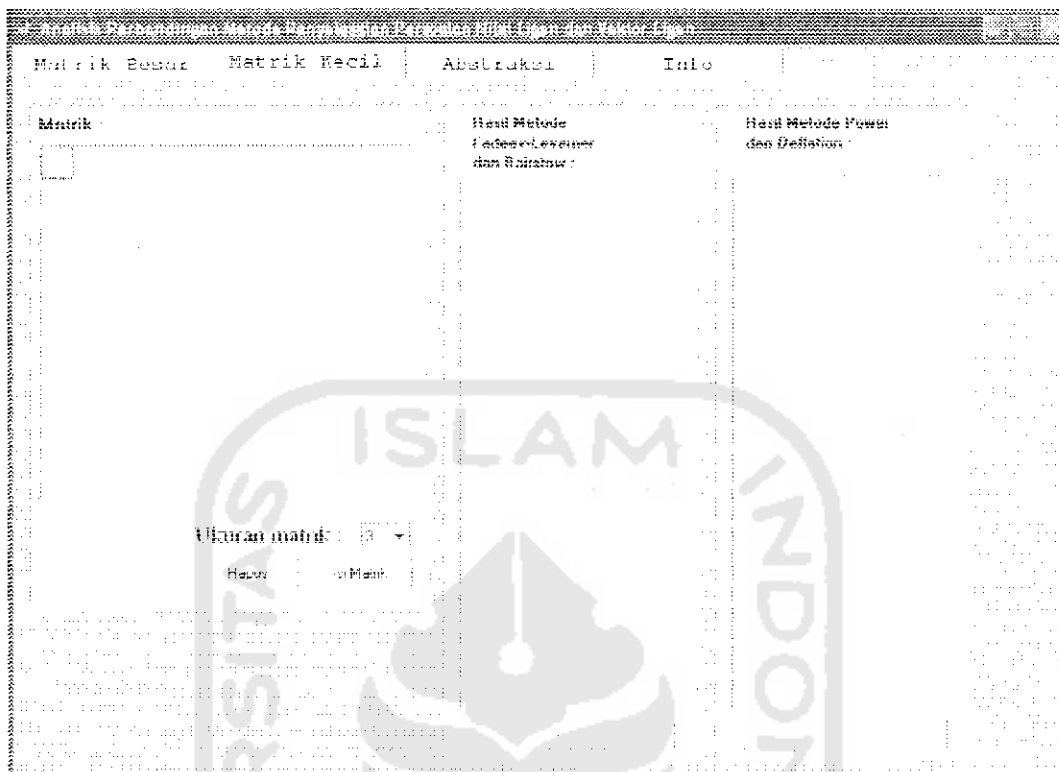


Gambar 5.1 Tampilan Menu Matrik Besar

5.3.2. Menu Matrik Kecil

Tampilan menu matrik kecil berisi fasilitas yang hampir sama dengan menu matrik besar. Perbedaannya, pengguna dapat menginputkan sendiri data yang diinginkan untuk mengetahui hasil dari masing-masing metode. Tetapi ukuran matriknya dibatasi, yaitu hanya dari ordo 3 sampai ordo 10 saja.

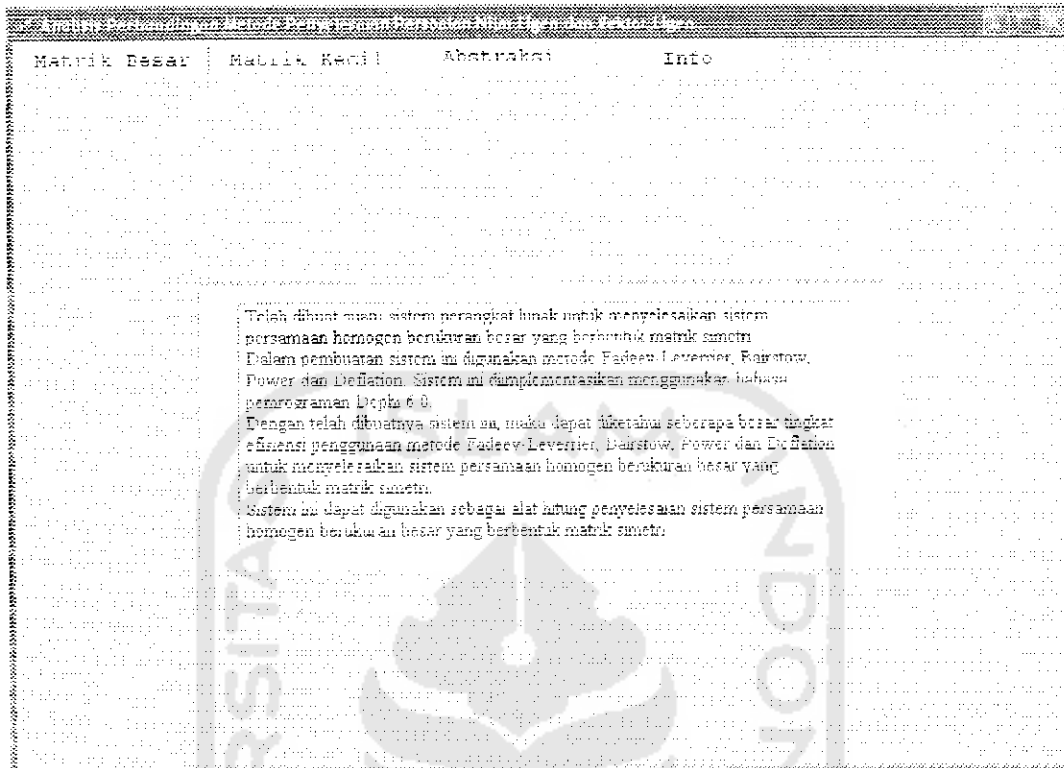
Kemudian pengguna dapat melihat dengan jelas hasil perhitungan untuk masing-masing metode dengan tingkat ketelitian sampai lima dibelakang koma.



Gambar 5.2 Tampilan Menu Matrik Kecil

5.3.3. Menu Abstraksi

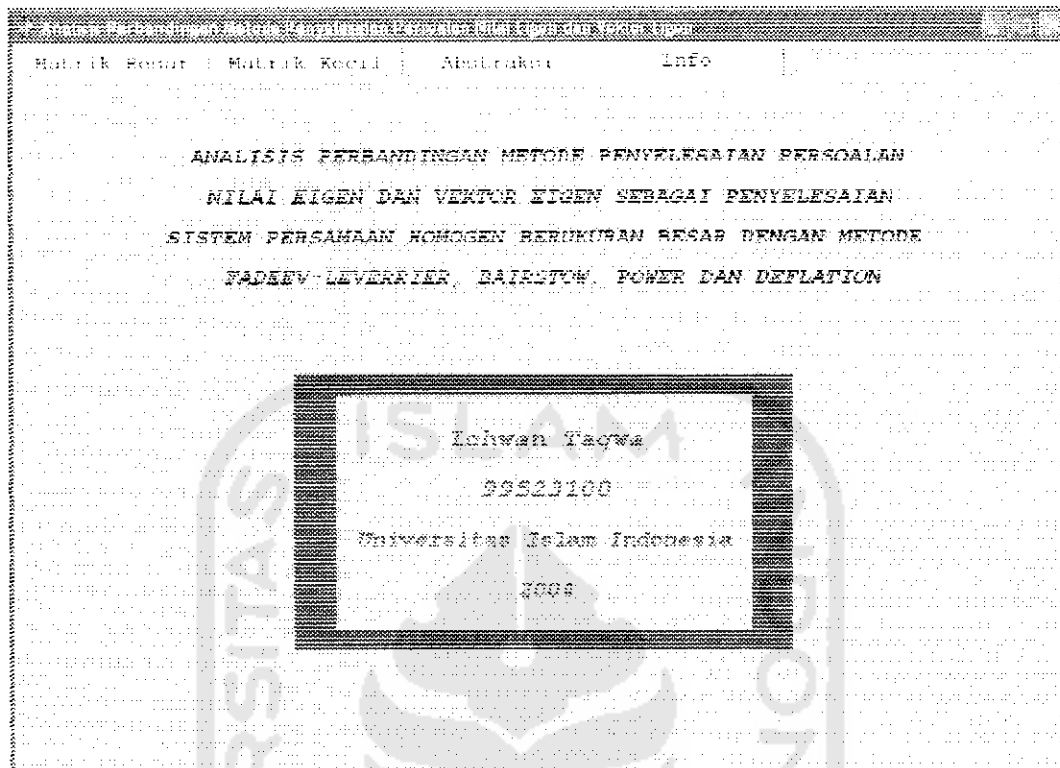
Tampilan menu abstraksi ini memberikan informasi dari abstraksi yang dibuat berdasarkan pembuatan tugas akhir ini. Dengan tujuan pengguna dapat mengetahui abstraksi atau gambaran singkat tentang mengapa sistem ini dibuat, sehingga pengguna dapat mengetahui manfaatnya.



Gambar 5.3 Tampilan Menu Abstraksi

5.3.4. Menu Info

Tampilan menu info ini memberikan informasi tentang pembuat sistem analisis perbandingan persoalan nilai eigen dan vektor eigen sebagai penyelesaian sistem persamaan homogen berukuran besar.



Gambar 5.4 Tampilan Menu Info

5.4. Struktur Data

Struktur data yang digunakan di dalam perangkat lunak ini adalah *array*. Digunakan *array* pada perangkat lunak ini dikarenakan banyaknya perhitungan yang melibatkan matrik dan vektor. Oleh karena itulah digunakan *array* sebagai struktur data.

Sebagai contoh, suatu matrik didefinisikan sebagai *array* yang berdimensi dua dengan perintah :

```
matriks : array [min..max, min..max] of integer;
```

5.5. Prosedur dan Algoritma

5.5.1. Teknik Pemrograman

Di dalam pembuatan perangkat lunak ini *source code* dibedakan menjadi tiga macam, yaitu :

1. File dengan ekstensi *.dfm*

File ini berisi informasi mengenai tampilan form, ukuran, lokasi, maupun sebagai tempat meletakkan *object*.

2. File dengan ekstensi *.pas*

File ini digunakan untuk menyimpan program yang mengontrol *form* dan penanganan kejadian (*event driven*), serta sebagai tempat penulisan kode program.

3. File dengan ekstensi *.dpr*

File ini digunakan untuk mengatur unit-unit yang digunakan sehingga dapat dijalankan menjadi sebuah aplikasi.

5.5.2. Prosedur – Prosedur dalam Perangkat Lunak

Pemrograman di dalam perangkat lunak ini terdiri atas prosedur-prosedur dari metode-metode yang digunakan. Berikut ini kilasan prosedur-prosedur dari keempat metode yang digunakan.

5.5.2.1. Prosedur Faddeev-Leverrier dan Bairstow

Sesuai namanya, prosedur ini digunakan untuk menjalankan metode Faddeev-Leverrier. Berikut ini dapat dilihat penerapan dari prosedur Faddeev-Leverrier dan Bairstow sebagai berikut :

```

Trace:=0;
for i:=1 to n do trace:=trace+matB[i,i];
P[n-1]:=Trace;
for k:=(n-2) downto 0 do
  begin
    for i:=1 to n do matB[i,i]:=matB[i,i]-P[n-1];
    for i:=1 to n do
      for j:=1 to n do
        matC[i,j]:=0;
      for i:=1 to n do
        for j:=1 to n do
          for k:=1 to n do
            matC[i,j]:=-matC[i,j]+matA[i,k]*matB[k,j];
          for i:=1 to n do
            for j:=1 to n do
              matB[i,i]:=matC[i,i];
            Trace:=0;
            for i:=1 to n do trace:=trace+matB[i,i];
            P[k]:=Trace/(n-k);
          end;
          for i:=0 to n do imajiner[i]:=0;
          for i:=0 to n do A[n-i]:=P[i];
          {Inisialisasi}
          p1:=0.0; q:=0.0; k:=0.0000; e:=0.0001; ind:=1;
          {memulai bagian pencarian faktor}
100:  if n<=2 then goto 500;
      j:=0;
200:  if j>k then goto 1000;
      j:=j+1;
      B[0]:=0.0; B[1]:=0.0; C[0]:=0.0; C[1]:=0.0;
      for i:=2 to n+2 do
        begin
          B[i]:=A[i-2]-p1*B[i-1]-q*B[i-2];
          C[i]:=-B[i-1]-p1*C[i-1]-q*C[i-2];
        end;
      {dp=a1 dq=b1}
      x2:=B[n+1]; y:=B[n+2]; z:=C[n];
      t:=C[n+1]; u:=C[n+2];
      d:=t*t-x2*(u*x2);
      if d=0 then
        begin
          p1:=p1+1;
          q:=q+1;
          j:=0;
          goto 200;
        end;
      a1:=(x*y-x2*t)/d;
      b1:=(-x2*(q*z+p1*t)-y*t)/d;
      p1:=p1+a1; q:=q+b1;

```



```

1: (ABS(p1)+ABS(p2))/(ABS(p1)+ABS(p2));
if f>e then goto 200;
d:=p1*p1-4*q;
if d<0 then
begin
  d:=sqrt(-d)/2.0;
  eigen[ind]:= p1/2.0;
  imajiner[ind]:=d;
  inc(ind,1);
  eigen[ind]:= p1/2.0;
  imajiner[ind]:=-d;
  inc(ind,1);
  k03:=k03+(2*j);
end
else
begin
  d:=SQRT(d);
  eigen[ind]:=(-p1+d)/2.0;
  inc(ind,1);
  eigen[ind]:=(-p1-d)/2.0;
  inc(ind,1);
  k03:=k03+(2*j);
end;
n:=n-2;
for i:=0 to n do A[i]:= B[i+2];
goto 100;
300: if n=2 then goto 350;
eigen[ind]:=-A[1]/A[0]; y:=0.0;
inc(ind,1);
k03:=k03+j;
goto 1in;
550: p1:=A[1]/A[0]; q:=A[2]/A[0];
d:=p1*p1-4*q;
if d<0 then
begin
  d:=SQRT(-d)/2.0;
  eigen[ind]:= p1/2.0;
  imajiner[ind]:=d;
  inc(ind,1);
  eigen[ind]:=-p1/2.0;
  imajiner[ind]:=-d;
  inc(ind,1);
  k03:=k03+(2*j);
end
else
begin
  d:=SQRT(d);
  eigen[ind]:=(-p1+d)/2.0;
  inc(ind,1);
  eigen[ind]:=(-p1-d)/2.0;
  inc(ind,1);
  k03:=k03+(2*j);
end;
goto fin;
1000: ListBox3.Items.Add('Proses tidak konvergen!!!');
fin

```

5.5.2.2. Prosedur Power dan Deflation

Sesuai namanya, prosedur ini digunakan untuk menjalankan metode Power dan Deflation. Berikut ini dapat dilihat penerapan metode Power dan Deflation sebagai berikut :

```

for r:=1 to n do X0[r]:=1.0/SQRT(I);
itr:=-1; l:=1; dta:=0.0001; eps:=0.0001; m:=20000;
while ((l = 1) and (l <= m)) do
begin
  gamma:=0.0;
  for r:=1 to n do
  begin
    x1[r]:= 0.0;
    for j:=1 to n do X1[r]:=X1[r]+A[r, j]*X0[j];
    if ABS(X1[r])>ABS(gamma) then gamma:=X1[r];
  end;
  if ABS(gamma) < eps then itr:=0
  else
  begin
    for r:=1 to n do X1[r]:=X1[r]/gamma;
    phi:= 0.0;
    for r:=1 to n do
    begin
      l:= ABS(X1[r]-X0[r]);
      if l>phi then phi:=l;
    end;
    if phi<dta then l:=0
    else
    begin
      X0:=X1;
      Inc(l);
    end
  end
end
end;
for r:= 1 to n do vektor2[r]:= X1[r];
Eigen[i]:=gamma;
k2:=k2+1;
{akhir power}
for i:= 2 to n do
begin
  sum:=0;
  for r:=1 to n do
    for s:=1 to n do C[r, s]:=0;
  for r:=1 to n do
    for s:=1 to n do B[r, s]:=0;
  for s:=1 to n do
    for r:=1 to n do B[r, s]:=0;
  for i:= 1 to n do sum:=sum+vektor2[i];
  for r:=1 to n do
    for s:=1 to n do C[r, s]:=vektor2[r];
  for r:= 1 to n do
    for s:=1 to n do D[r, s]:=C[r, s]*Eigen[i-1]/sum;
  for s:=1 to n do
    for r:=1 to n do B[r, s]:= A[r, s]-D[r, s];

```

```

[initial power]
for r:=1 to n do X0[r]:= 1.0/SQRT(I);
it:=-1; it:=1; dba:=0.01; eps:=0.0001; m:=1000;
While ((it=-1) and (it=m)) do
begin
  gamma:= 0.0;
  for r:=1 to n do
  begin
    X1[r]:= 0.0;
    for j:= 1 to n do X1[r]:=X1[r]+B[r,j]*X0[j];
    if ABS(X1[r])>ABS(gamma) then gamma:=X1[r]
  end;
  if ABS(gamma) < eps then it:=0
  else
  begin
    for r:=1 to n do X1[r]:=X1[r]/gamma;
    phi:= 0.0;
    for r:=1 to n do
    begin
      t:=ABS(X1[r]-X0[r]);
      if t>phi then phi:=t
    end;
    if phi<dba then it:=1
    else
    begin
      X0:=X1;
      Inc(I)
    end
  end
end;
for r:=1 to n do vektor2[r]:=X1[r];
Eigen[i]:=gamma;
k2:=k2+1;
[akhir power]
for r:=1 to n do
  for s:=1 to n do A[r,s]:= B[r,s];
end;

```