

BAB IV

IMPLEMENTASI DAN EVALUASI

Pada bab ini akan dilakukan implementasi dan pembahasan terhadap perangkat lunak program aplikasi yang telah dibuat. Pada tahap implementasi merupakan tahap akhir, sehingga dapat diketahui sampai sejauh mana program aplikasi yang telah dibuat dapat dijalankan sesuai dengan apa yang diharapkan. Dan untuk tahap pembahasan diulas tentang kemampuan dan keterbatasan perangkat lunak yang telah dibuat.

4.1. Implementasi.

Implementasi merupakan suatu tahap yang dilakukan untuk memeriksa sejauh mana program yang telah dibuat telah memenuhi kriteria yang diinginkan. Sebelum proses implementasi dilakukan terlebih dahulu akan membahas spesifikasi perangkat keras dan perangkat lunak yang dibutuhkan untuk menjalankan program aplikasi ini.

4.1.1. Persyaratan software dan hardware

Tabel 4.1. Persyaratan hardware untuk menjalankan program aplikasi

Hardware	
Processor	Pentium II – 200 Mhz
Memory RAM	128 MB
Harddisk	20 GB
Kartu Grafik (VGA)	32 MB
Monitor	14” atau lebih
Keyboard + mouse	

Tabel 4.2. Persyaratan software untuk menjalankan program aplikasi

Software	
Sistem Operasi	Microsoft Windows 98 atau lebih
Database	Microsoft Access 2000

4.1.2. Penerapan metode iterasi Gauss Seidel dengan proses produksi latex

Disini akan dijelaskan langkah demi langkah bagaimana penerapan metode iterasi Gauss Seidel dengan proses produksi latex seperti yang telah digambarkan pada sebelumnya (BAB III).

1. Langkah Pertama (Memilih nomor rencana produksi).

Pada langkah ini sistem langsung menampilkan daftar produk yang akan diproses berdasarkan nomer rencana produksi yang merupakan hasil dari proses sebelumnya. Program aplikasi akan menampilkan Id Produk dan Total Order berdasarkan nomer rencana produksi.

2. Langkah kedua (Penerapan teori iterasi Gauss Seidel)

Penerapan teori iterasi Gauss Seidel diawali dengan membuat suatu persamaan linier dari data yang sudah terproses sebelumnya.

– Membentuk persamaan linier umum

Pembentukan persamaan linier umum dimaksudkan untuk memudahkan dalam pembacaan (pemahaman).

Listing program untuk membentuk persamaan linier umum:

```

procedure TForm1.Button3Click(Sender: TObject);
var i,j:integer; k:real;
begin
with adoquery3 do begin
close;sql.Clear;
sql.Add('select a.*');

```

```

        sql.Add('from komposisi as a');
        sql.Add('where a.id_produk='+Quotedstr(syarat1.text));
        sql.Add('order by a.id_produk, a.id_bahan_baku');
        open;
    end
    jmlpers:=strtoint(pers.text);
    first;
    for j:=1 to varjml do begin
        cbxpers.Text='Bahan Baku' then
            k:=fieldbyname('jumlah').asfloat
            a[j,jmlpers]:=k;
        next;
    end;
end;
k:=0;
with adoquery3 do begin
    cbxpers.Text='Bahan Baku' then begin
        close;
        sql.Clear;
        sql.Add('select b.stock as syarat2');
        sql.Add('from bahan_baku as b');
        sql.Add('where b.id_bahan_baku='+Quotedstr(syarat2.text));
        sql.Add('order by b.id_bahan_baku');
        open;
    end
    k:=fieldbyname('syarat2').asfloat;
    a[varjml+1,jmlpers]:=k;
end;
for j:=1 to varjml+97 do
    StringGrid1.Cells[0,j]:='Persm ke-'+floattostr(j);
    ShowArr(varjml+97,varjml+2,a,StringGrid1);
end;

```

– Menghitung & membandingkan besar diagonal utama dengan jumlah non diagonal utama serta mencari relaksasi.

Setelah persamaan linier terbentuk, maka program aplikasi akan menghitung besar diagonal utama dan akan membandingkannya dengan jumlah non diagonal utama pada tiap-tiap persamaan. Tujuan dilakukan perhitungan ini adalah untuk mencari perbandingan nilai antara nilai diagonal utama dengan nilai non diagonal utama hal ini dilakukan untuk menentukan apakah persamaan yang terbentuk atau yang akan diproses nantinya akan menghasilkan suatu jawaban yang konvergen atau sebaliknya. Apabila hasil perbandingan antara nilai diagonal utama lebih besar dari jumlah nilai non diagonal utama, maka hasil akhir dari proses akan konvergen (akan menghasilkan suatu jawaban), namun sebaliknya apabila hasil perbandingan antara nilai diagonal utama kurang dari jumlah nilai non diagonal

utama, maka hasilnya pasti tidak konvergen dan akan dilakukan sedikit modifikasi dari teori Gauss Seidel yaitu akan dilakukan proses *relaksasi*.

Listing program sebagai berikut :

```

procedure TForm1.Button1Click(Sender: TObject);
var i,j,r,s,m,n,iterasi,jmlketemu:integer;
    k,vlamda,vabs:real;
    arrstr,vkondisi:string;
begin
  with adoquery2 do begin
    close;sql.Clear;
  for i:=1 to varjml+1 do
    x[i]:=d[varjml+1,i];
  m:=1;hasil:=0;vkondisi:='';
  for i:=1 to varjml do begin
    for j:=2 to varjml do begin
      hasil:=hasil+d[j,i];
    end;
    if x[m]<hasil then begin
      vkondisi:='Relaksasi';
    end;
    e[1,i]:=x[m];
    e[2,i]:=hasil;
    inc(m);hasil:=0;
  end;
  StringGrid2.Cells[1,0]:='Diagonal Utama';
  StringGrid2.Cells[2,0]:='Diag Non Utama';
  ShowArr(varjml+97,varjml,e,StringGrid2);
  if vkondisi='Relaksasi' then Application.MessageBox(' Diagonal Utama
< Diagonal Non Utama','Pesan Kesalahan',MB_OK);
  for j:=1 to varjml+97 do
    StringGrid3.Cells[0,j]:='X'+floattostr(j);
  ShowArr(varjml+97,varjml+2,d,StringGrid3);
  Application.MessageBox(' Persamaan Iterasi bisa dilanjutkan ','
Pesan',MB_OK);
  ketemu:=false;
  iterasi:=1;
  for i:=1 to varjml do
    x[i]:=0;
  while not ketemu do begin
    hasil:=0;
    r:=1;
    for j:=1 to varjml do begin
      hasil:=d[1,j];
      for i:=2 to varjml do begin
        if r=j then inc(r);
        hasil:=hasil-(d[i,j]*x[r]);
        inc(r);
      end;
      hasil:=hasil/d[varjml+1,j];
      x[j]:=hasil;
      r:=1;hasil:=0;
    end;
    for i:=1 to varjml do begin
      arrstr:=floattostr(x[i]);
      c[i,iterasi]:=strtofloat(copy(arrstr,1,pos(',',arrstr)+3));
      If (iterasi>1) and (vkondisi='Relaksasi') then
        c[i,iterasi]:=vlamda*x[i]+(1-vlamda)*c[i,iterasi-1];
      arrstr:=floattostr(c[i,iterasi]);
      c[i,iterasi]:=strtofloat(copy(arrstr,1,pos(',',arrstr)+3));
    end;
  end;

```

```

end;
end;

```

– Relaksasi

Relaksasi menyajikan modifikasi yang sedikit dari Metode Gauss-Seidel dan dirancang untuk mempercepat kekonvergenan

Listing program untuk membentuk relaksasi sebagai berikut:

```

for i:=1 to varjml+1 do
  x[i]:=d[varjml+1,i];
  m:=1;hasil:=0;vkondisi:='';
  for i:=1 to varjml do begin
    for j:=2 to varjml do begin
      hasil:=hasil+d[j,i];
    end;
    if x[m]<hasil then begin
      vkondisi:='Relaksasi';
    end;
  end;
end;

```

– Mengubah persamaan linier menjadi bentuk awal teori Iterasi Gauss Seidel.

Yang dilakukan disini adalah mengubah persamaan linier awal yang sudah terbentuk menjadi persamaan dasar dalam menyelesaikan teori iterasi Gauss Seidel, dimana bentuk persamaan tersebut adalah sebagai berikut:

$$\begin{aligned}
 X_1 &= \frac{1}{a_{11}} (b_1 - a_{12}x_2 - a_{13}x_3 - \dots - a_{1n}x_n) \\
 X_2 &= \frac{1}{a_{22}} (b_2 - a_{21}x_1 - a_{23}x_3 - \dots - a_{2n}x_n) \\
 X_3 &= \frac{1}{a_{33}} (b_3 - a_{31}x_1 - a_{32}x_2 - \dots - a_{3n}x_n) \\
 X_n &= \frac{1}{a_{nn}} (b_n - a_{n1}x_1 - a_{n2}x_2 - \dots - a_{n,n-1}x_{n-1})
 \end{aligned}$$

sedangkan listing program untuk mengubah persamaan linier menjadi bentuk awal teori Iterasi Gauss Seidel.

```

procedure TFrmProses.Button1Click(Sender: TObject);
var i,j,r,s,m,n,iterasi:integer;
    k,vlamda,vabs:real;
    arrstr,vkondisi:string;
begin
  with adoquery2 do begin
    close;sql.Clear;
    r:=1;
    for i:=1 to varjml+1 do begin
      if i<>j then inc(r);
      for j:=1 to varjml+1 do begin
        b[1,j]:=a[varjml+1,j];
        if i<>j then begin b[r,j]:=a[i,j]; end
      end;
    end;
  end;
end;

```

– **Proses pembentukan nilai dalam bentuk iterasi.**

Setelah langkah demi langkah penerapan teori iterasi Gauss Seidel terbentuk, maka langkah selanjutnya adalah mengimplementasikannya menurut teori Gauss Seidel yaitu program aplikasi akan memproses seluruh data yang sudah diproses sebelumnya dan akan diterapkan sesuai dengan rumus teori Gauss Seidel yang dihasilkan dengan adanya langkah iterasi.

Listing program untuk menampilkan hasil akhir (nilai) dari penerapan teori Gauss-Seidel kedalam bentuk iterasi adalah sebagai berikut :

```

procedure TFrmProses.Button1Click(Sender: TObject);
var i,j,r,s,m,n,iterasi:integer;
    k,vlamda,vabs:real;
    arrstr,vkondisi:string;
begin
  with adoquery2 do begin
    close;sql.Clear;

    for j:=1 to varjml+96 do
      StringGrid3.Cells[0,j]:='X'+floattostr(j);
      ShowArr(varjml+96,varjml+2,d,StringGrid3);

    Application.MessageBox('Persamaan Iterasi bisa dilanjutkan
    ','Pesan',MB_OK);

    ketemu:=false;
    iterasi:=1;
    for i:=1 to varjml do
      x[i]:=0;
    while not ketemu do begin
      hasil:=0;
      r:=1;
      for j:=1 to varjml do begin
        hasil:=d[1,j];
        for i:=2 to varjml do begin
          if r=j then inc(r);

```

```

        hasil:=hasil-(d[i,j]*x[r]);
        inc(r);
    end;
    hasil:=hasil/d[varjml+1,j];
    x[j]:=hasil;
    r:=1;hasil:=0;
end;

for i:=1 to varjml do begin

    arrstr:=floattostr(x[i]);
    c[i,iterasi]:=strtofloat(copy(arrstr,1,pos(', ',arrstr)+3));

    If (iterasi>1) and (vkondisi='Relaksasi') then
        c[i,iterasi]:=vlamda*x[i]+(1-vlamda)*c[i,iterasi-1];

    arrstr:=floattostr(c[i,iterasi]);
    c[i,iterasi]:=strtofloat(copy(arrstr,1,pos(', ',arrstr)+3));
end;

jmlketemu:=0;
for i:=1 to varjml do begin
    if abs(c[i,iterasi-1]-c[i,iterasi])<=0.002 then inc(jmlketemu);
end;
if (jmlketemu=varjml) then
    ketemu:=true
else
    ketemu:=false;
inc(iterasi);
if iterasi>100 then ketemu:=true;
end;
jmlketemu:=iterasi-1;

ShowArr(varjml+96,varjml+1,c,StringGrid4);
StringGrid4.Cells[0,0]:='Iterasi ke-';
for j:=1 to varjml+96 do
    StringGrid4.Cells[0,j]:=floattostr(j);
j:=1; end;
end;
end;
end;
end;

```

3. Langkah ketiga (Proses Produksi).

Hal terakhir yang harus dilakukan adalah melakukan produksi. Setelah keseluruhan proses (Langkah 1 sampai Langkah 2) terselesaikan, maka akan didapatkan suatu nilai akhir dari penerapan teori iterasi Gauss Seidel dalam bentuk iterasi. Akan tetapi hasil iterasi dari keseluruhan proses diatas ***bukan merupakan hasil maksimal*** yang harus dipaksakan atau dengan kata lain hasil iterasi tersebut bukan merupakan keputusan akhir dari keseluruhan proses produksi guna memenuhi order dari pelanggan. Keputusan akhir dari keseluruhan

proses tergantung dari pihak manajemen perusahaan apakah hasil iterasi tersebut akan langsung dijadikan acuan untuk melakukan produksi ataukah masih perlu dilakukan peninjauan ulang. Jadi keputusan akhir dari penerapan iterasi Gauss Seidel tetap berada ditangan perusahaan itu sendiri dan bisa dikatakan, bahwa nilai akhir tersebut hanya sebagai suatu *alternatif keputusan* yang sifatnya temporer.

Listing program untuk menampung hasil produksi:

```

procedure TFrmProses.BtnProduksiClick(Sender: TObject);
var i,j,r,s,m,n,iterasi:integer;
    k,vlamda,vabs,jmldd:real;
    arrstr,vkondisi:string;
begin
  with adoquery2 do begin
    close;sql.Clear;
    sql.Add('SELECT * FROM HASIL_PRODUKSI WHERE NO_HASIL=:AA');
    Parameters.ParamByName('AA').value:=trim(no_hasil.Text);
    open;
  end;
  if (trim(no_hasil.Text)='') OR (not adoquery2.Eof) then begin
    Application.MessageBox('Isi NO_HASIL dahulu atau NO_HASIL sudah
ada','Peringatan',MB_OK);
    exit;adoquery2.close;
  end;
  try
    vlamda:=strtofloat(jmlldamda.text);
  except on exception do begin
    Application.MessageBox('Isi Lamda dahulu','Pesan
Kesalahan',MB_OK);
    jmlldamda.SetFocus;exit;
  end;
end;
j:=1;
with adoquery2 do begin
  close;sql.Clear;
  sql.Add('SELECT ID_PRODUK, SUM(produksi-jmlorder) AS TOT FROM
HASIL_PRODUKSI WHERE
NO_RENCANA_PRODUKSI='+QUOTEDSTR(adoquery5.Fields[0].AsString)+' GROUP
BY ID_PRODUK');
  open;
  first;
  while not eof do begin
    adoquery3.close;adoquery3.sql.Clear;
    adoquery3.SQL.Add('update PRODUK set SALDO=SALDO - :total where
id_produk=:id_produk');
    if fieldbyname('TOT').asfloat>0 then
      adoquery3.Parameters.ParamByName('total').value:=fieldbyname('TOT').as
float
      else adoquery3.Parameters.ParamByName('total').value:=0;
    adoquery3.Parameters.ParamByName('id_produk').value:=fieldbyname('id_p
roduk').value;
    adoquery3.ExecSQL;next;
  
```



```

end;
close;
adoquery3.Close;
end;
with adoquery2 do begin
close;sql.Clear;
sql.Add('delete from hasil_produksi WHERE
NO_RENCANA_PRODUKSI='+QUOTEDSTR(adoquery5.Fields[0].AsString));
ExecSQL;
close;
sql.Clear;
sql.Add('SELECT ID_PRODUK, SUM(TOTAL) AS TOT_ORDER FROM
RENCANA_PRODUKSI WHERE
NO_RENCANA_PRODUKSI='+QUOTEDSTR(adoquery5.Fields[0].AsString)+' GROUP
BY ID_PRODUK');
open;
first;
while not eof do begin
adoquery3.close;adoquery3.sql.Clear;
adoquery3.SQL.Add('insert into
hasil_produksi(NO_HASIL,id_produk,NO_RENCANA_PRODUKSI,iterasi,jmlorder
,produksi)
values(:NO_HASIL,:id_produk,:NO_RENCANA_PRODUKSI,:iterasi,:order,:prod
uksi)');

adoquery3.Parameters.ParamByName('NO_HASIL').value:=NO_HASIL.TEXT;
adoquery3.Parameters.ParamByName('id_produk').value:=fieldbyname('id_p
roduk').value;
adoquery3.Parameters.ParamByName('NO_RENCANA_PRODUKSI').value:=adoquer
y5.Fields[0].AsString;
adoquery3.Parameters.ParamByName('iterasi').value:=c[j,jmlketemu];
adoquery3.Parameters.ParamByName('order').value:=fieldbyname('tot_orde
r').value;
adoquery3.Parameters.ParamByName('produksi').value:=fieldbyname('tot_o
rder').value*c[j,jmlketemu];
adoquery3.ExecSQL;
next;inc(j);
end;
end;
end;
end;

```

4.2. Evaluasi

Untuk mengetahui kinerja dari sistem ini, telah dilakukan evaluasi dengan menginputkan data-data produksi yang meliputi kebutuhan bahan baku apabila terdapat order.

- Kasus Pertama (Pengisian Data Utama)

Sebagai contoh terdapat suatu kasus sebagai berikut :

- 1) Pada bulan Agustus 1997 perusahaan mendapatkan order dari Customer A untuk memproduksi produk I sebanyak 5 ton, produk II sebanyak 20 ton, produk III sebanyak 10 ton, produk IV sebanyak 10 ton dan produk V sebanyak 15 ton. Untuk Produk I membutuhkan 316.60 Kg raw material A, 35.05 Kg raw material B, 30.10 Kg raw material C, 20.95 Kg raw material D dan 20.75 Kg raw material E. Untuk Produk II membutuhkan 128.20 Kg raw material A, 1206.00 Kg raw material B, 104.20 Kg raw material C, 98.80 Kg raw material D dan 81.60 Kg raw material E. Untuk Produk III membutuhkan 76.80 Kg raw material A, 51.90 Kg raw material B, 613.20 Kg raw material C , 49.40 Kg raw material D dan 33.3.0 Kg raw material E. Untuk Produk IV membutuhkan 72.40 Kg raw material A, 52.20 Kg raw material B, 43.90 Kg raw material C, 593.20 Kg raw material D dan 38.10 Kg raw material E. Untuk Produk V membutuhkan 113.85 Kg raw material A, 94.95 Kg raw material B, 78 Kg raw material C, 62.85 Kg raw material D dan 903.3 Kg raw material E. Sedangkan jumlah persediaan raw material di Gudang Bahan Baku masing-masing adalah 708.00 Kg, 1440.00 Kg , 869.00 Kg, 825.00 Kg dan 1077.00 Kg. Tentukan berapa yang harus diproduksi untuk kelima macam produk tersebut ?.

Hasil output program aplikasi sebagai berikut:

Gambar 4.1. Output Data Entry Rumus Iterasi Gauss-Seidel

Secara tabel dapat digambarkan sebagai berikut:

Tabel 4.3. kebutuhan raw material untuk masing-masing produk

Produk	Material A	Material B	Material C	Material D	Material E
I	316.60	35.05	30.10	20.95	20.75
II	128.20	1206.00	104.20	98.80	81.60
III	76.80	51.90	613.20	49.40	33.30
IV	72.40	52.20	43.90	593.20	38.10
V	113.85	94.95	78.00	62.85	903.30
Persediaan	708.00	1440.00	869.00	825.00	1077.00

Penyelesaian:

a) Persoalan tersebut dapat disusun menjadi suatu persamaan linier, sebagai berikut:

Misalkan:

$$X_1 = \text{Produk I}$$

$$X_2 = \text{Produk II}$$

$$X_3 = \text{Produk III}$$

$$X_4 = \text{Produk IV}$$

$$X_5 = \text{Produk V}$$

Constraint:

$$316.60X_1 + 35.05X_2 + 30.10X_3 + 20.95X_4 + 20.75X_5 \leq 708$$

$$128.20X_1 + 1206X_2 + 104.20X_3 + 98.80X_4 + 81.60X_5 \leq 1440$$

$$76.80X_1 + 51.90 X_2 + 613.20X_3 + 49.40X_4 + 33.3.0X_5 \leq 869$$

$$72.40X_1 + 52.20X_2 + 43.90X_3 + 593.20X_4 + 38.10X_5 \leq 825$$

$$113.85X_1 + 94.95 X_2 + 78.00 X_3 + 62.85 X_4 + 903.30 X_5 \leq 1077$$

b) penyelesaian dengan Metode Gauss-Seidel.

Langkah Pertama : Mengubah bentuk persamaan umum diatas menjadi:

$$X_1 = (708 - 35.05X_2 - 30.10X_3 - 20.95X_4 - 20.75X_5) / 316.60$$

$$X_2 = (1440 - 128.20X_1 - 104.20X_3 - 98.80X_4 - 81.60X_5) / 1206.00$$

$$X_3 = (869 - 76.80X_1 - 51.90 X_2 - 49.40X_4 - 33.3.0X_5) / 613.20$$

$$X_4 = (825 - 72.40X_1 - 52.20X_2 - 43.90X_3 - 38.10X_5) / 593.20$$

$$X_5 = (1077 - 113.85X_1 - 94.95 X_2 - 78.00 X_3 - 62.85 X_4) / 903.30$$

Langkah Kedua : Menganggap $X_2 = X_3 = X_4 = X_5 = 0$, dan dengan substitusi ke persamaan, akan didapat :

$$X_1 = (708 - 35.05X_2 - 30.10X_3 - 20.95X_4 - 20.75X_5) / 316.60$$

$$X_1 = (708 - 35.05*0 - 30.10*0 - 20.95*0 - 20.75*0) / 316.60$$

$$X_1 = (708 - 0 - 0 - 0 - 0) / 316.60$$

$$X_1 = 708 / 316.60$$

$$X_1 = \mathbf{2.236}$$

Hasil dari X_1 tersebut disubstitusikan kembali untuk mencari X_2 pada persamaan,

dimana X_3, X_4 dan X_5 masih sama dengan nol, akan didapat :

$$X_2 = (1440 - 128.20X_1 - 104.20X_3 - 98.80X_4 - 81.60X_5) / 1206.00$$

$$X_2 = (1440 - 128.20*2.236 - 104.20*0 - 98.80*0 - 81.60*0) / 1206.00$$

$$X_2 = (1440 - 286.65 - 0 - 0 - 0) / 1206.00$$

$$X_2 = (1153.35 / 1206.00)$$

$$X_2 = \mathbf{0.956}$$

Hasil dari X_1, X_2 tersebut disubstitusikan kembali untuk mencari X_3 pada persamaan, dimana X_4 dan X_5 masih sama dengan nol, akan didapat :

$$X_3 = (869 - 76.80X_1 - 51.90X_2 - 49.40X_4 - 33.30X_5) / 613.20$$

$$X_3 = (869 - 76.80*2.236 - 51.90*0.956 - 49.40*0 - 33.30*0) / 613.20$$

$$X_3 = (869 - 171.724 - 49.616 - 0 - 0) / 613.20$$

$$X_3 = 647.66 / 613.20$$

$$X_3 = \mathbf{1.056}$$

Hasil dari X_1, X_2 dan X_3 sebelumnya disubstitusikan kembali untuk mencari X_4 sedangkan X_5 masih = 0, akan didapat :

$$X_4 = (825 - 72.40X_1 - 52.20X_2 - 43.90X_3 - 38.10X_5) / 593.20$$

$$X_4 = (825 - 72.40*2.236 - 52.20*0.956 - 43.90*1.056 - 38.10*0) / 593.20$$

$$X_4 = (825 - 161.885 - 49.903 - 46.358 - 0) / 593.20$$

$$X_4 = 566.854 / 593.20$$

$$X_4 = \mathbf{0.955}$$

Hasil dari X_1 , X_2 , X_3 dan X_4 tersebut disubstitusikan kembali untuk mencari X_5 ,

akan didapat :

$$X_5 = (1077 - 113.85X_1 - 94.95 X_2 - 78.00 X_3 - 62.85 X_4) / 903.30$$

$$X_5 = (1077 - 113.85*2.236 - 94.95 *0.956 - 78.00*1.056 - 62.85*0.955) / 903.30$$

$$X_5 = (1077 - 254.568 - 90.772 - 82.368 - 60.021) / 903.30$$

$$X_5 = 589.271 / 903.30$$

$$X_5 = \mathbf{0.652}$$

Setelah nilai X_5 sudah didapatkan, perhitungan untuk putaran (iterasi) pertama berakhir. Untuk iterasi berikutnya (kedua), perkalian diambil dari hasil-hasil pada

iterasi pertama:

untuk mencari X_1 , maka:

$$X_1 = (708 - 35.05X_2 - 30.10X_3 - 20.95X_4 - 20.75X_5) / 316.60$$

$$X_1 = (708 - 35.05*0.956 - 30.10*1.056 - 20.95*0.955 - 2075*0.652) / 316.60$$

$$X_1 = (708 - 33.507 - 31.785 - 20.007 - 13.529) / 316.60$$

$$X_1 = 609.172 / 316.60$$

$$X_1 = \mathbf{1.924}$$

untuk mencari X_2 , maka:

$$X_2 = (1440 - 128.20X_1 - 104.20X_3 - 98.80X_4 - 81.60X_5) / 1206.00$$

$$X_2 = (1440 - 128.20*1.924 - 104.2*1.056 - 98.80*0.955 - 81.60*0.652) / 1206.00$$

$$X_2 = (1440 - 246.565 - 110.035 - 94.354 - 53.203) / 1206.00$$

$$X_2 = 935.843 / 1206.00$$

$$X_2 = \mathbf{0.775}$$

untuk mencari X_3 , maka:

$$X_3 = (869 - 76.80*X_1 - 51.90 X_2 - 49.40X_4 - 33.3.0X_5) / 613.20$$

$$X_3 = (869 - 76.80*1.924 - 51.90*0.775 - 49.40*0.955 - 33.3.0*0.652) / 613.20$$

$$X_3 = (869 - 147.763 - 40.222 - 47.177 - 21.711) / 613.20$$

$$X_3 = 612.127 / 613.20$$

$$X_3 = \mathbf{0.998}$$

untuk mencari X_4 , maka:

$$X_4 = (825 - 72.40X_1 - 52.20X_2 - 43.90X_3 - 38.10X_5) / 593.20$$

$$X_4 = (825 - 72.40*1.924 - 52.20*0.775 - 43.90*0.998 - 38.10*0.652) / 593.20$$

$$X_4 = (825 - 139.297 - 40.455 - 43.812 - 24.841) / 593.20$$

$$X_4 = 576.595 / 593.20$$

$$X_4 = \mathbf{0.971}$$

untuk mencari X_5 , maka :

$$X_5 = (1077 - 113.85X_1 - 94.95 X_2 - 78.00 X_3 - 62.85 X_4) / 903.30$$

$$X_5 = (1077 - 113.85*1.924 - 94.95 *0.775 - 78.00*0.998 - 62.85*0.971) / 903.30$$

$$X_5 = (1077 - 219.047 - 73.586 - 77.844 - 61.027) / 903.30$$

$$X_5 = 645.496 / 903.30$$

$$X_5 = \mathbf{0.714}$$

Setelah nilai X_5 sudah didapatkan, perhitungan untuk putaran (iterasi) kedua berakhir. Untuk iterasi berikutnya (ketiga), perkalian diambil dari hasil-hasil pada iterasi sebelumnya (iterasi kedua), aturan ini berlaku untuk iterasi berikutnya. Dengan melanjutkan persoalan diatas dan mencatat hasil-hasil iterasi sebagai berikut:

The screenshot shows the 'Rumus Gauss Seidel' software interface. It contains several data tables and control elements:

- NO_RENCANA_PRODUKS**: 0000001/RP/W/2004
- ID_PRODUK**: P001, P002, P003, P004, P005
- TOT_ORDER**: 5, 20, 10, 10, 15
- ID_BAHAN_BAKU**: C-108, E-103, E-104, E-109, E-115
- PRIORITAS**: 1, 2, 3, 4, 5
- PROSES**: (empty)
- No. Hasil**: (empty)
- Lambda**: 1
- Persamaan ke-**: 0
- Id Produk**: P001
- Nama Variable Input**: E-115
- Kolum Persamaan ke 1**:

	316,6	35,05	30,1	20,95	20,75
Persm ke-1					
Persm ke-2	128,2	1206	104,2	98,8	81,6
Persm ke-3	76,8	51,9	613,2	49,4	33,3
Stock	708	1440	869	825	1077
- Kolum Perbandingan Diagonal**:

	Diagonal Utama	Diag Non Utari		
Persm ke-1	316,6	106,85	0	0
Persm ke-2	1206	412,8	0	0
Persm ke-3	613,2	211,4	0	0
Persm ke-4	593,2	206,6	0	0
Persm ke-5	903,2	349,65	0	0
- Kolum Hasil Iterasi**:

Iterasi ke-					
1	2,236	0,956	1,056	0,955	0,652
2	1,945	0,775	0,998	0,971	0,713
3	1,944	0,773	0,991	0,966	0,713
4	1,945	0,774	0,991	0,965	0,712
5	0	0	0	0	0

Gambar 4.2. Output Hasil Iterasi Sebelum Update Stock

Karena hasil iterasi ketiga dan keempat hampir sama, maka proses berhenti.

- Kasus kedua (Terjadinya Update)

Dengan melihat hasil akhir dari iterasi diatas, maka hasil *yang paling optimal* menurut teori Gauss-Seidel untuk masing-masing produk yang harus diproduksi untuk memenuhi order dari customer A adalah:

- Produk 1 = $X_1 = 1.945 * 5 = 9.725$ Ton

- Produk 2 = $X_2 = 0.774 * 20 = 15.48$ Ton

- Produk 3 = $X_3 = 0.991 * 10 = 9.910$ Ton
- Produk 4 = $X_4 = 0.965 * 10 = 9.650$ Ton
- Produk 5 = $X_5 = 0.712 * 15 = 10.680$ Ton

Hasil komputerisasi dapat dilihat dibawah ini:

PT. ETERINDO NUSA GRAHA
 Jl. Prof. DR. Moh. Yamin SH
 KIG - Gresik

Tanggal cetak : 27/04/04

No	NO HASIL	ID PRODUK	NO RENCANA PRODUKSI	BESAR ORDER	HASIL ITERASI	JUMLAH PRODUKSI
1	000001/HP/IV/2004	P001	0000001/RP/IV/2004	5	1,945	9,725
2	000001/HP/IV/2004	P002	0000001/RP/IV/2004	20	0,774	15,48
3	000001/HP/IV/2004	P003	0000001/RP/IV/2004	10	0,991	9,91
4	000001/HP/IV/2004	P004	0000001/RP/IV/2004	10	0,965	9,65
5	000001/HP/IV/2004	P005	0000001/RP/IV/2004	15	0,712	10,68
TOTAL				60	5,387	55,445

Page 1 of 1

Gambar 4.3. Laporan hasil produksi sebelum update stock

Hal ini sangatlah tidak mungkin untuk dilakukan proses produksi guna memenuhi order dari customer A, karena ada beberapa produk yang melampaui besar order dan ada yang tidak mungkin terpenuhi (Lihat tabel).

Tabel 4.4. Hasil Produksi Sebelum Update Stock

	X_1	X_2	X_3	X_4	X_5
Jumlah Order(Ton)	5	20	10	10	15
Hasil Iterasi	1.945	0.774	0.991	0.965	0.712
Hasil	9.725	15.48	9.910	9.650	10.680
Selisih	+ 4.725 Ton	- 4.52 Ton	- 0.09 Ton	- 0.35 Ton	- 4.32 Ton

Dengan melihat kenyataan yang ada, maka pihak manajemen akan melakukan pertimbangan atas nilai akhir tersebut, hal ini dikarenakan nilai akhir dari hasil iterasi bukanlah merupakan keputusan akhir dari keseluruhan proses produksi akan tetapi merupakan suatu alternatif keputusan yang sifatnya temporer. Karena bersifat temporer, maka keputusan tersebut bisa berubah tergantung dari pihak manajemen itu sendiri dan perubahan keputusan tersebut dapat dilakukan dengan menambah maupun mengurangi kebutuhan stock.

Misalkan pihak manajemen melakukan update stock, lihat tabel dibawah:

Tabel 4.5. Update Stock Raw Mateial

Id_BB	C-108	E-103	E-104	E-109	E-115
Stock awal	708	1440	859	825	1077
Stock update	430	1700	870	850	1320
Iterasi awal	1.945	0.774	0.991	0.965	0.712
Iterasi update	1	1.052	1.06	1.071	1.058

Hasil proses dari program aplikasi setelah dilakukan update stock sebagai berikut:

The screenshot shows the 'Rumus Gauss Seidel' application interface. It displays various data tables and control panels. The 'ID_PRODUK' table shows product details for P005, including materials C-108, E-103, E-104, E-109, and E-115. The 'Kolom Persamaan ke 1' table shows the initial stock values for each material, with the 'Stock' row highlighted in blue and circled in red. The 'Kolom Hasil Iterasi' table shows the results of the iterative process, with the final iteration (Iterasi ke-5) highlighted in blue and circled in red. The 'Kolom Perbandingan Diagonal' table shows the diagonal elements of the iteration matrix. The 'Kolom Persamaan ke 2' table shows the coefficients of the iteration matrix. The 'No. Hasil' panel shows the iteration number (1) and the equation number (5). The 'Id Produk' and 'Nama Variable Input' panels show the selected product (P005) and material (E-115).

Gambar 4.4. hasil output update stock pada proses iterasi

Sehingga besarnya masing-masing produk yang akan diproduksi setelah dilakukan update untuk memenuhi order dari customer A adalah

- Produk 1 = $X_1 = 1.00 * 5 = 5$ Ton

- Produk 2 = $X_2 = 1.052 * 20 = 21.04$ Ton

- Produk 3 = $X_3 = 1.06 * 10 = 10.6$ Ton

- Produk 4 = $X_4 = 1.071 * 10 = 10.71$ Ton

- Produk 5 = $X_5 = 1.058 * 15 = 15.87$ Ton

hasil produksi dari program aplikasi dapat dilihat sebagai berikut:

PT. ETERINDO NUSA GRAHA
Jl. Prof. DR. Moh. Yamin SH
KIG - Gresik

Tanggal cetak : 27/04/04

No	NO HASIL	ID PRODUK	NO RENCANA PRODUKSI	BESAR ORDER	HASIL ITERASI	JUMLAH PRODUKSI
1	0000001/HP/IV/2004P 001		0000001/RP/IV/2004	5	1	5
2	0000001/HP/IV/2004P 002		0000001/RP/IV/2004	20	1,052	21,04
3	0000001/HP/IV/2004P 003		0000001/RP/IV/2004	10	1,06	10,6
4	0000001/HP/IV/2004P 004		0000001/RP/IV/2004	10	1,071	10,71
5	0000001/HP/IV/2004P 005		0000001/RP/IV/2004	15	1,058	15,87
TOTAL				60	5,241	63,22

Page 1 of 1

Gambar 4.5. Laporan hasil produksi sesudah update stock

- **Kasus ketiga (Terjadinya Relaksasi)**

Kasus berikutnya adalah apabila terjadi relaksasi, dimana relaksasi menyajikan modifikasi yang sedikit dari Metode Gauss-Seidel dan dirancang untuk mempercepat kekonvergenan. dimodifikasi dengan rata-rata berbobot dari hasil-hasil iterasi sebelumnya dan yang sekarang:

$$X_i^{baru} = \lambda x_i^{baru} + (1 - \lambda) x_i^{lama}$$

dimana λ adalah faktor pembobotan yang diberi suatu nilai antara 0 dan 2. Jika $\lambda = 1$, $(1 - \lambda)$ sama dengan nol dan hasil itu tidak dimodifikasi. Namun, jika diberi suatu nilai antara 0 dan 1, maka hasil tersebut merupakan rata-rata berbobot dari hasil-hasil yang sekarang dan sebelumnya. Modifikasi jenis ini disebut *relaksasi bawah* (under relaxation). Pilihan ini secara khas diterapkan untuk membuat suatu sistem yang tidak konvergen menjadi konvergen.

Untuk nilai λ mulai 1 sampai 2, bobot ekstra ditempatkan pada nilai yang sekarang. Dalam hal ini, terdapat asumsi implisit bahwa nilai yang baru bergerak kearah yang benar menuju ke penyelesaian sejati tetapi pada laju yang agak lambat. Jadi, tambahan bobot λ dimaksudkan untuk memperbaiki taksiran dengan mendorongnya lebih mendekati kebenaran. Oleh karena itu, modifikasi jenis ini yang dinamakan *relaksasi atas* (over relaxation) dirancang untuk mempercepat kekonvergenan dari system yang memang sudah konvergen.

Apabila dilakukan relaksasi atas untuk hasil-hasil iterasi sebagai berikut:

Tabel 4.6. Relaksasi Atas

Iterasi	X ₁	X ₂	X ₃	X ₄	X ₅
I	2.236	0.956	1.056	0.955	0.652
II	1.924	0.775	0.998	0.971	0.714
III	1.944	0.773	0.991	0.966	0.713
IV	1.945	0.774	0.991	0.965	0.712

Dimana nilai λ diisi dengan suatu nilai mulai dari 1 sampai 2 (misalkan $\lambda=1.89$), akan tetapi tidak terdapat perubahan, berarti nilai iterasi tersebut adalah nilai iterasi terbaik (terpendek).

Hasil dari proses pada program aplikasi dapat dilihat dibawah ini:

The screenshot shows the 'Rumus Gauss Seidel' application interface. The 'Lambda' field is set to 1.89. The 'Kolon Hasil Iterasi' table shows the results for iteration 4, which are circled in black.

Iterasi ke-	X ₁	X ₂	X ₃	X ₄
3	1.944	0.773	0.991	0.966
4	1.945	0.774	0.991	0.965
5	0	0	0	0
6	0	0	0	0

Gambar 4.6. Output Proses Relaksasi