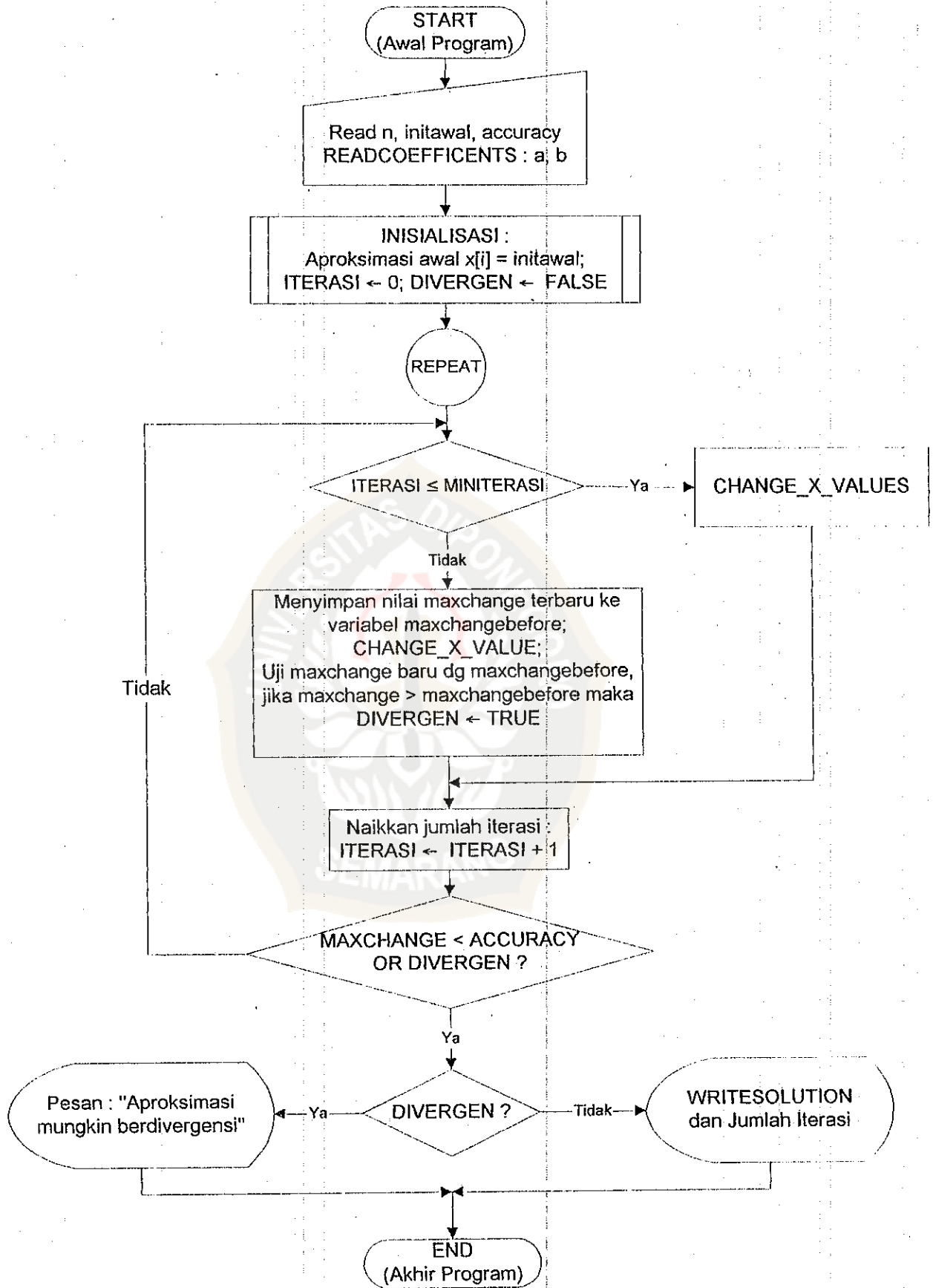
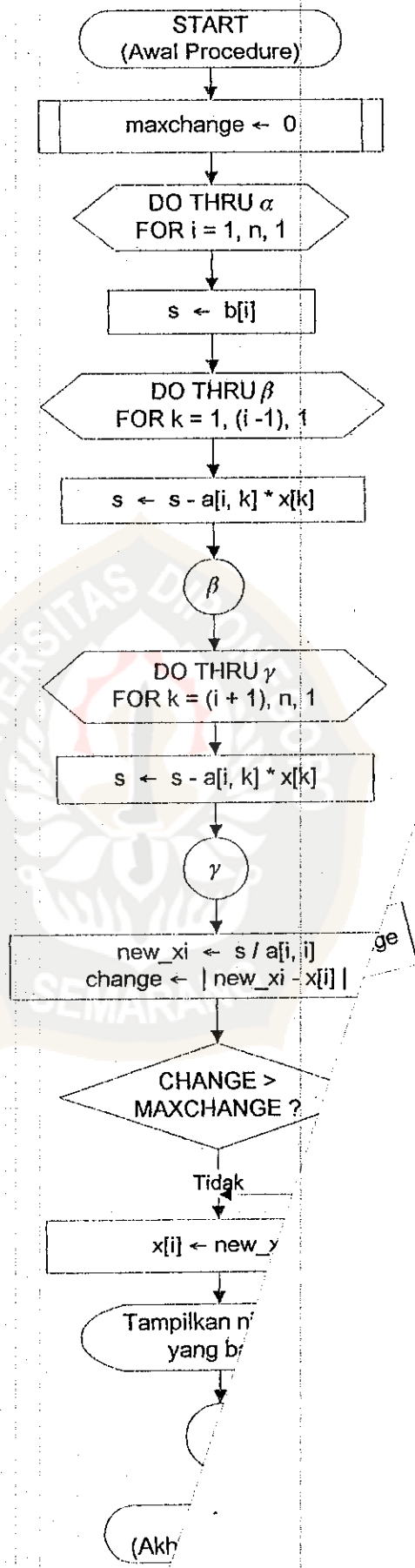


LAMPIRAN



FLOW CHART GAUSS SEIDEL





```

PROGRAM GaussSeidel;

USES crt;

CONST
  miniterasi = 40;
  {jumlah minimum iterasi untuk memulai pengecekan
  divergensi}

TYPE
  element = 1..100; {Maksimal 100 persamaan}
  coeffmatrix = array [element,element] of real;
  vector = array [element] of real;

VAR
  a      : coeffmatrix;
  b, x  : vector;
  n      : element; {Jumlah persamaan, maksimal 100 persamaan}
  iterasi, {Jumlah seluruh iterasi dalam perhitungan}
  fp     : integer; {Jumlah digit dibelakang titik desimal
  yang ditampilkan}

  initawal, {aproksimasi awal}
  accuracy, {ketelitian sebagai kondisi berhenti iterasi}
  maxchange, {perubahan terbesar dalam satu set nilai_x}
  maxchangebefore : real; {maxchange sebelumnya}

  divergen : boolean; {tipe boolean untuk pengujian diver-
  gensi}
  i : element; {untuk traversal for ..do}

{*****}
PROCEDURE readcoefficients;
{Procedure untuk membaca matriks koefisien.
Diasumsikan a[i,i] bukan 0.}
VAR
  i, j : element; {untuk traversal for ..do}

BEGIN
  For i := 1 to n do
  begin
    Write('Pers (' , i, ') : ');
    For j := 1 to n do
    begin
      read(a[i,j]);
    end;
    readln(b[i]);
  end;
END;

{*****}
PROCEDURE change_x_values;
{Procedure untuk menghitung satu set nilai x hasil aproksi
masi serta mencatat maxchange, yaitu perubahan/perbedaan
terbesar dalam satu set nilai x.}

```

```

VAR
  i, k : element; {untuk traversal for..do}
  s,
  change, {perubahan antara xi dan new_xi}
  new_xi : real; {aproksimasi xi yang baru}

BEGIN
  maxchange := 0; {inisialisasi maxchange}
  for i := 1 to n do
    begin
      {Menghitung aproksimasi dari suatu nilai xi baru,new_xi}
      s := b[i];
      for k := 1 to (i - 1) do
        s := s - a[i,k] * x[k];
      for k := (i + 1) to n do
        s := s - a[i,k] * x[k];
      new_xi := s / a[i,i];

      {Mencatat maxchange}
      change := abs(new_xi-x[i]);
      if change > maxchange then maxchange:= change;

      x[i] := new_xi; {x[i] diisi dengan aproksimasi xi baru}
      WriteLn('x[' , i, ']', x[i]:16:fp); {Menampilkan nilai x[i]
      yang baru}

    end;
  readln;
END;

{*****}
PROCEDURE writesolution;
VAR
  i : element;
BEGIN
  for i := 1 to n do
    writeln('X', i, ' = ', x[i]:16:fp);
  END;
{*****}

{***** PROGRAM UTAMA *****}
BEGIN
  clrscr;
  write('Jumlah persamaan = '); readln(n);
  write('Aproksimasi awal = '); readln(initawal);
  write('Ketelitian/accuracy = '); readln(accuracy);
  write('Digit dibelakang titik desimal yang ditampilkan = ');
  readln (fp);

  readcoefficients;

  {aproksimasi awal dengan nilai initawal}
  for i := 1 to n do x[i] := initawal;

  {Inisialisasi jumlah iterasi dan divergensi}
  iterasi:=0;

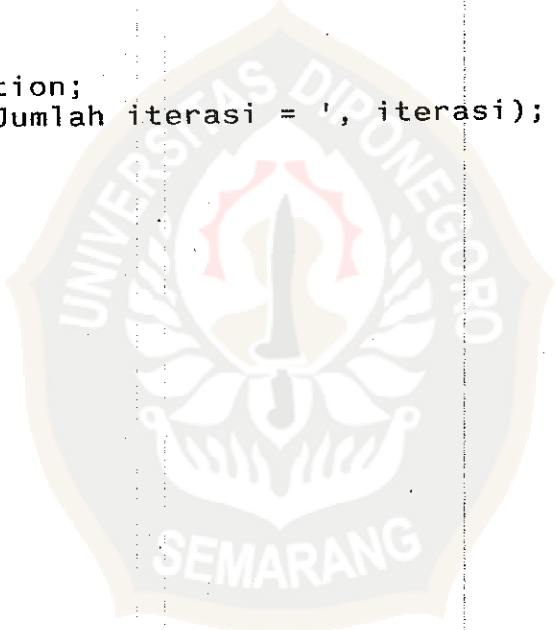
```

```

divergen := false;

{Mulai Iterasi}
REPEAT
  if iterasi<=miniterasi then
    change_x_values
  else {if iterasi>miniterasi}
    begin
      maxchangebefore := maxchange;
      change_x_values;
      if maxchange > maxchangebefore then divergen:=true;
    end;
  iterasi:=iterasi+1;
UNTIL (maxchange < accuracy) or divergen;

If divergen then
begin
writeln('Setelah',iterasi,'iterasi,maxchange terus bertambah');
writeln('Aproksimasi mungkin berdivergensi.');
```



```

end
else
begin
  writesolution;
  writeln('Jumlah iterasi = ', iterasi);
end;
readln;
END.

```