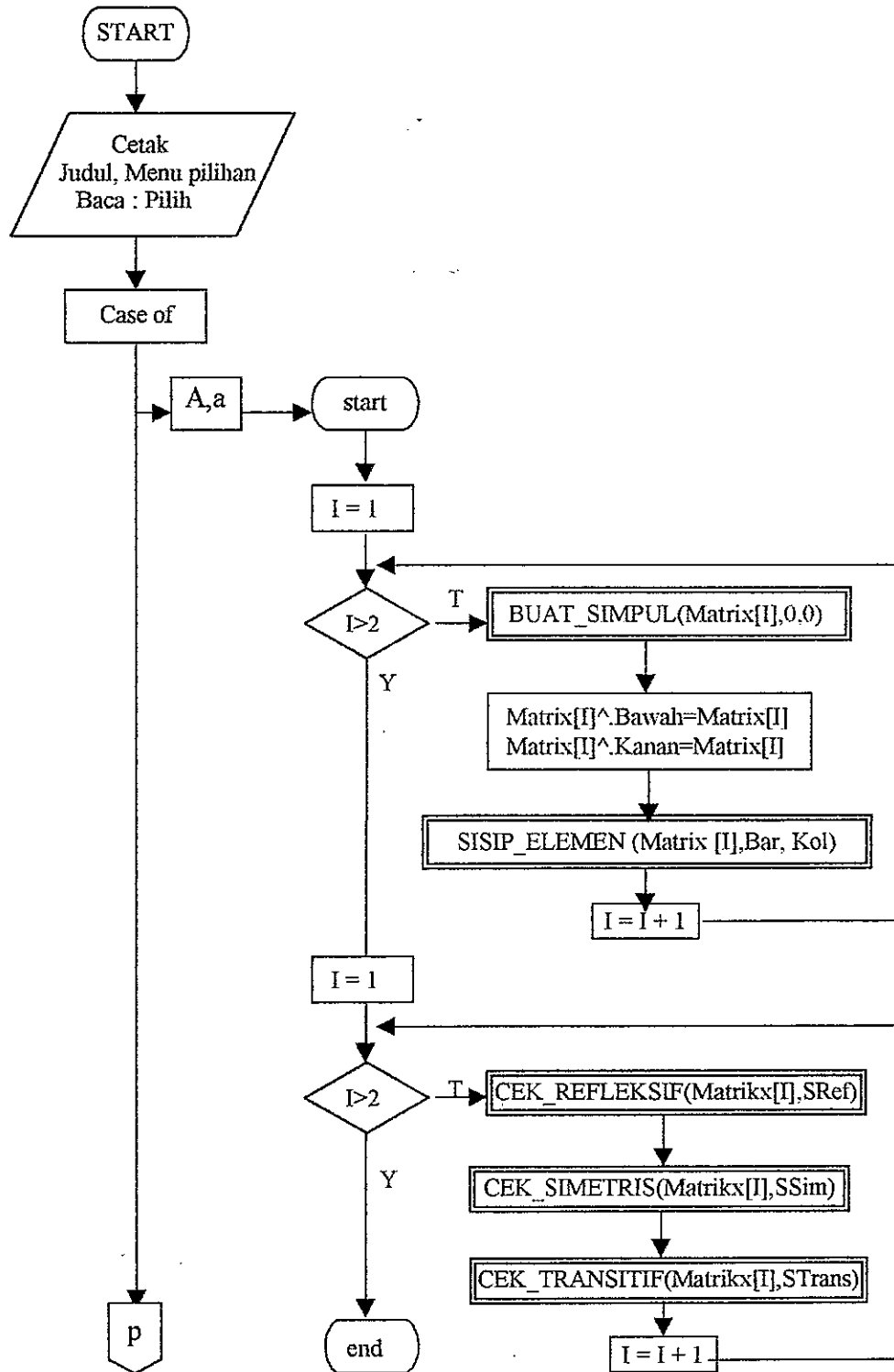
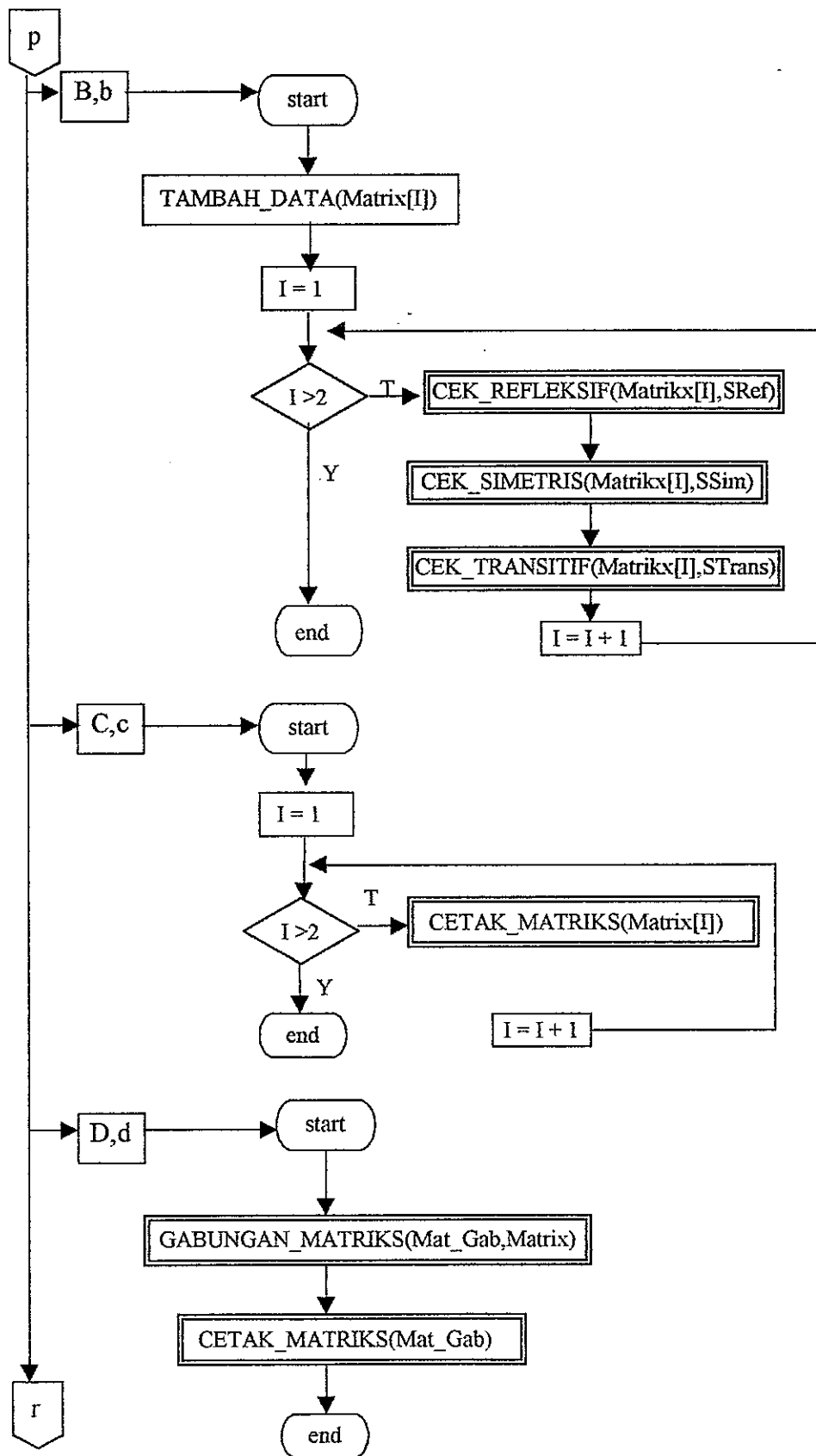


Lampiran 1. Flowchart Program Transitif Klosur dari Gabungan Dua Relasi Ekuivalensi pada Suatu Himpunan

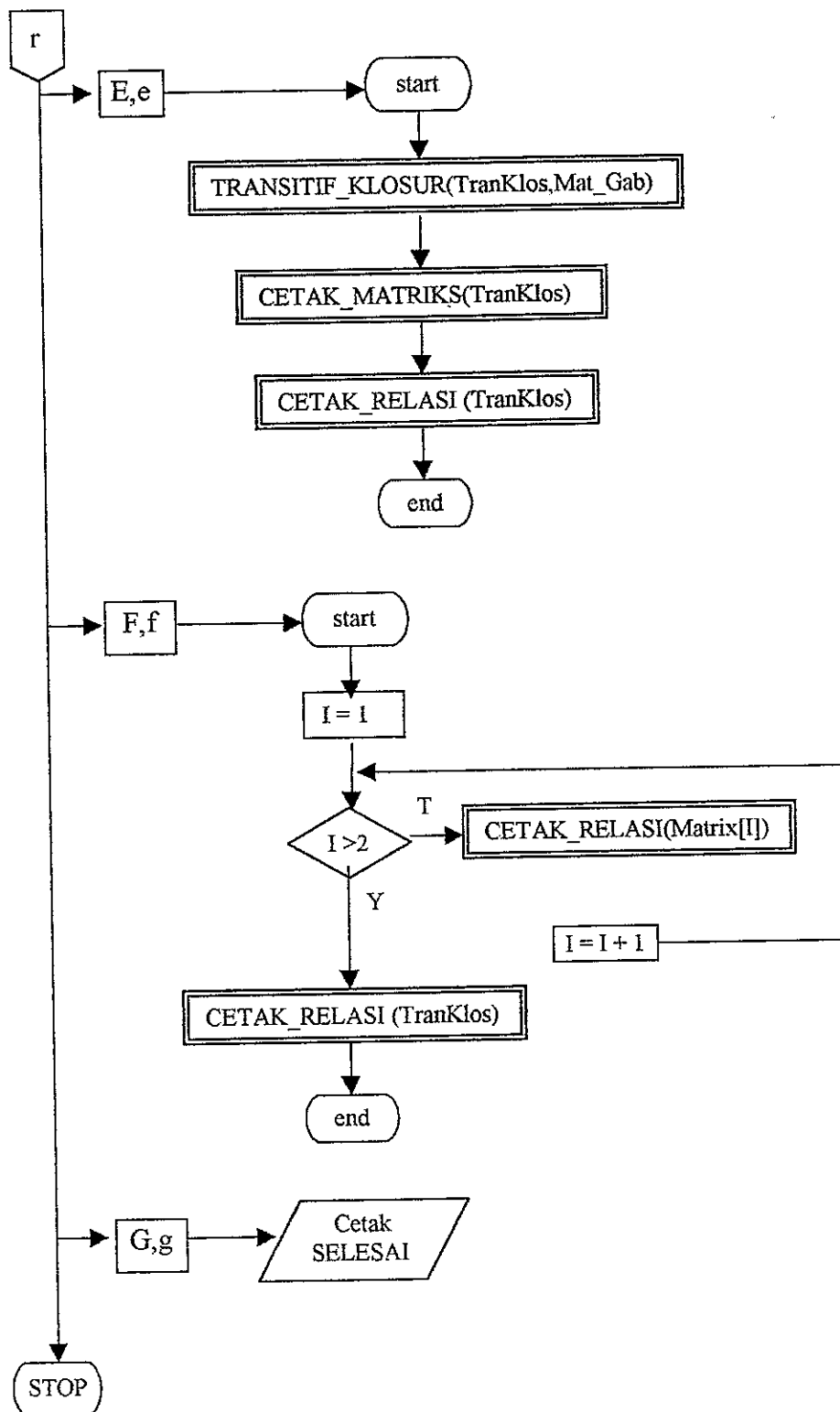
A. Garis Besar Flowchart Program Utama



Lampiran 1. Flowchart program transitif klosur dari gabungan dua relasi ekuivalensi pada suatu himpunan



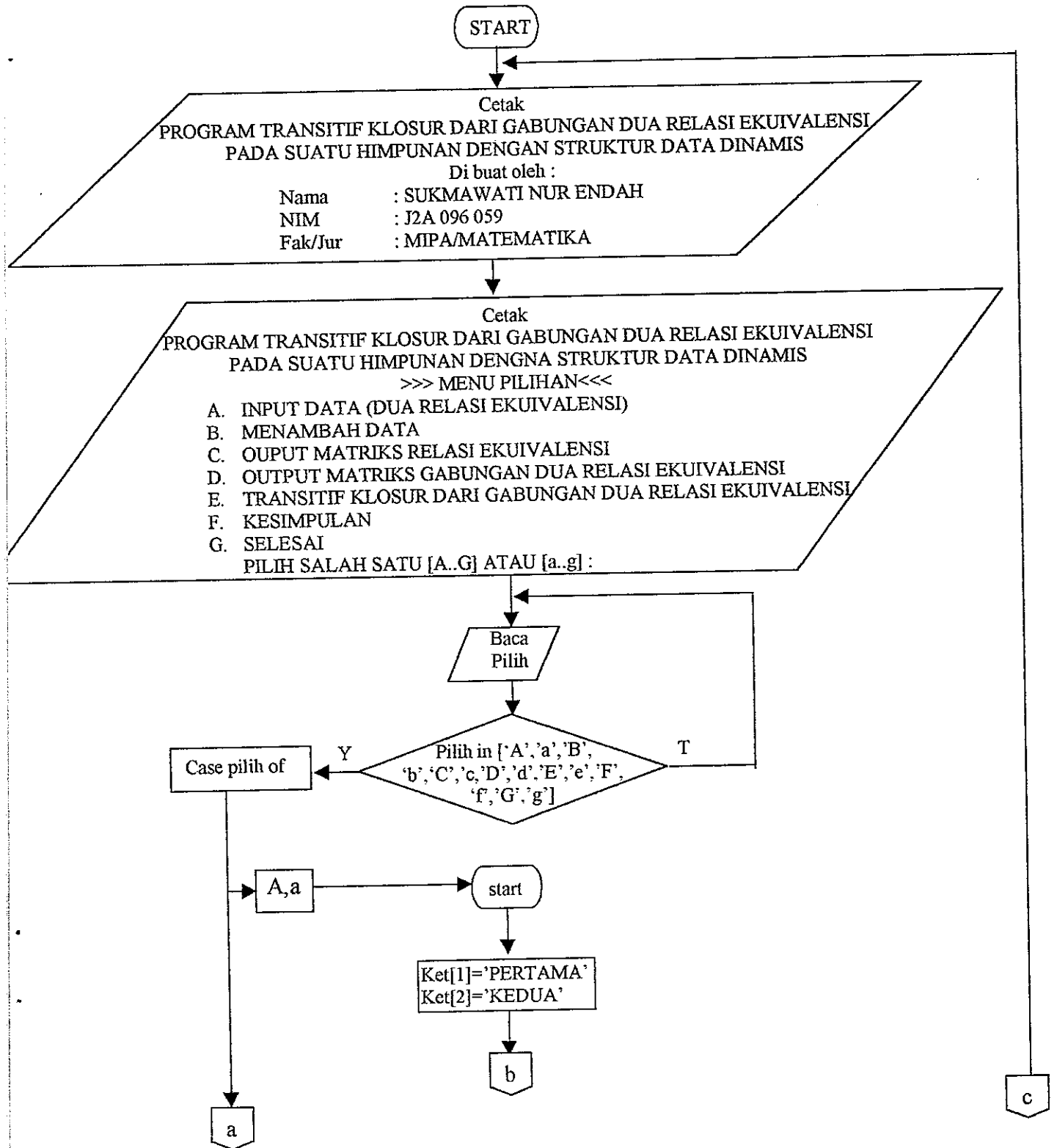
Lampiran 1. Flowchart program transitif klosur dari gabungan dua relasi ekuivalensi pada suatu himpunan



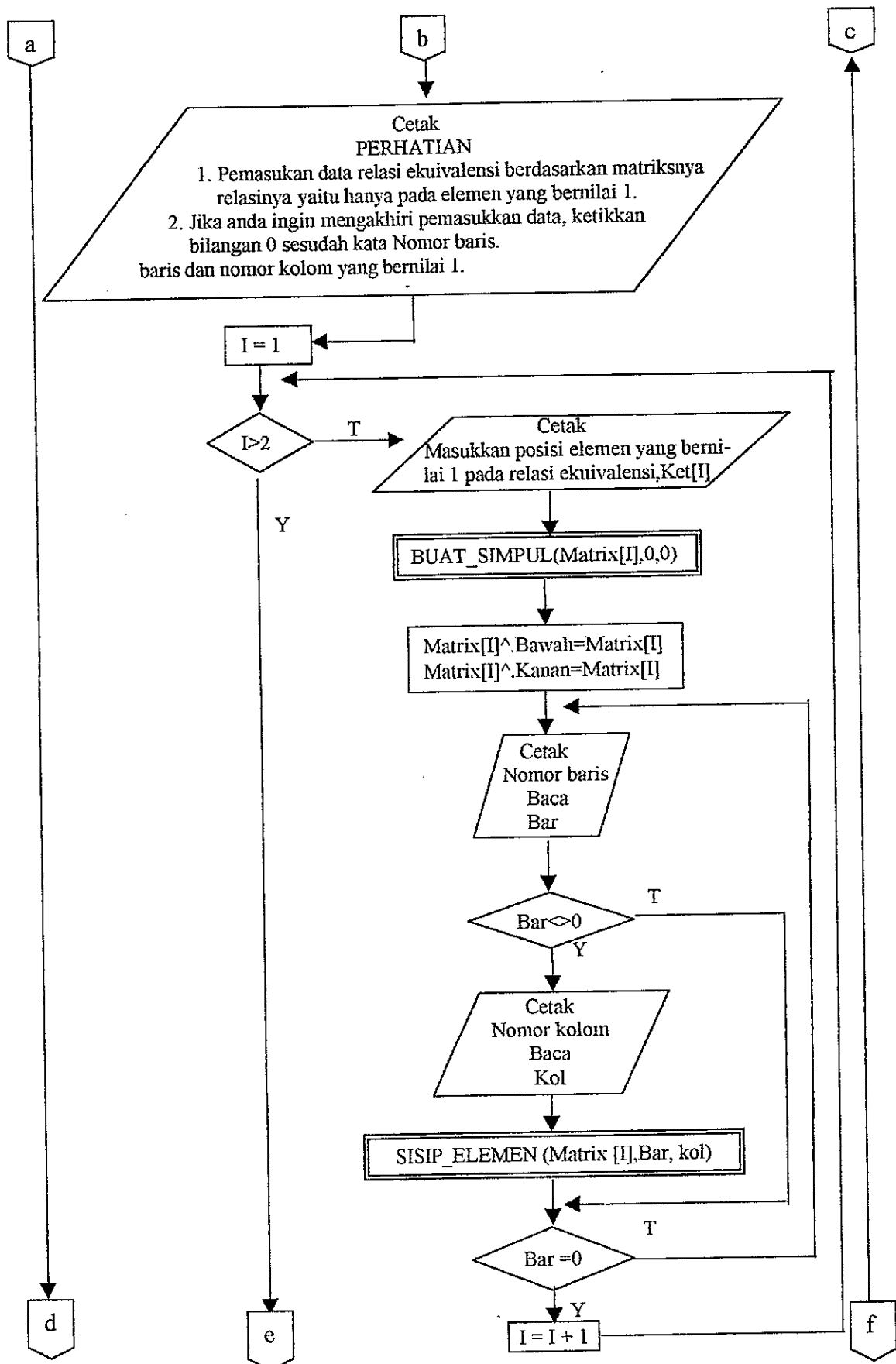
Empiran 1. Flowchart program transitif klosur dari gabungan dua relasi ekuivalensi pada suatu himpunan

B. Flowchart Program Transitif Klossur dari Gabungan Dua Relasi

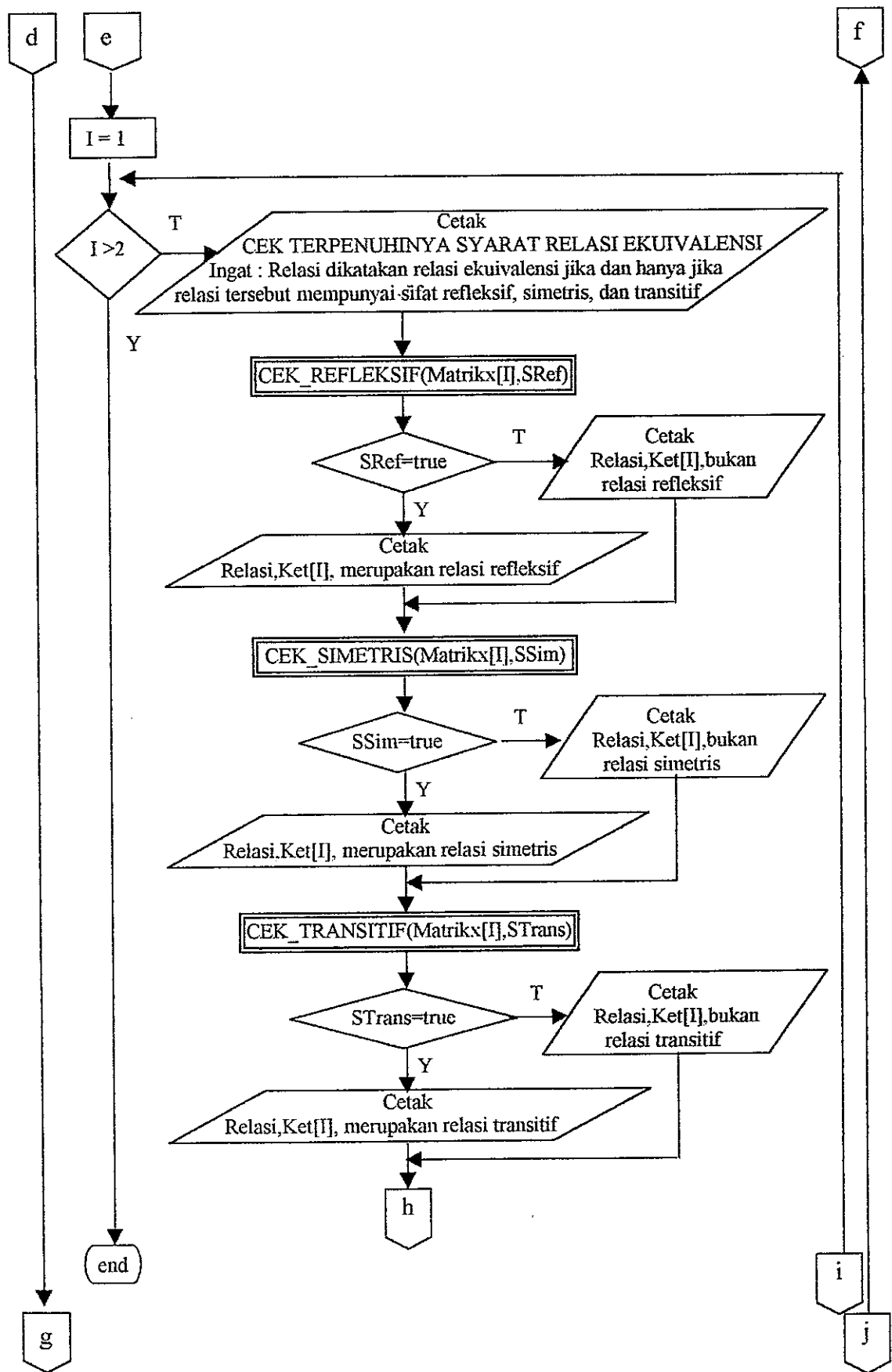
Ekuivalensi pada Suatu Himpunan



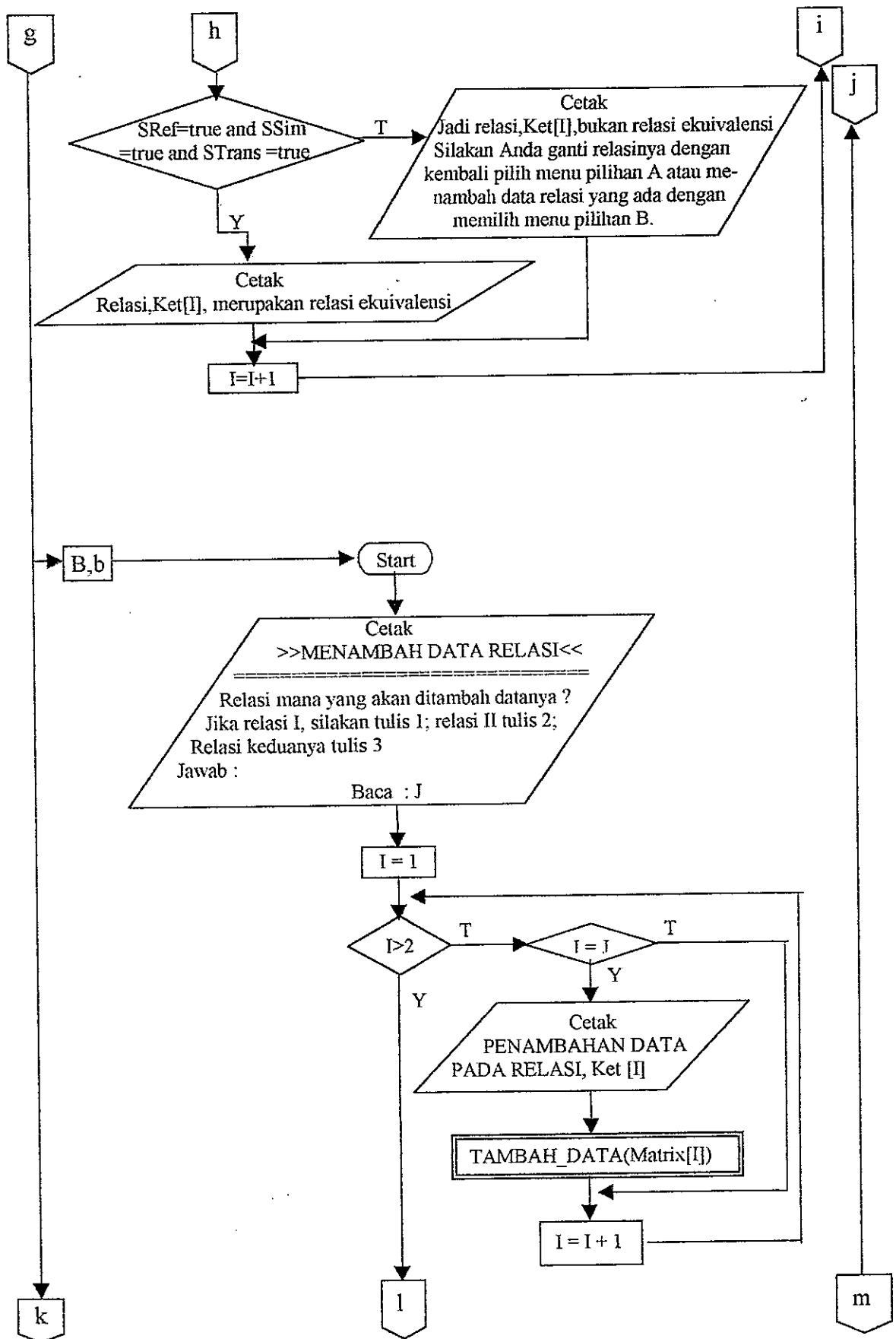
Lampiran 1. Flowchart program transitif klossur dari gabungan dua relasi ekuivalensi pada suatu himpunan



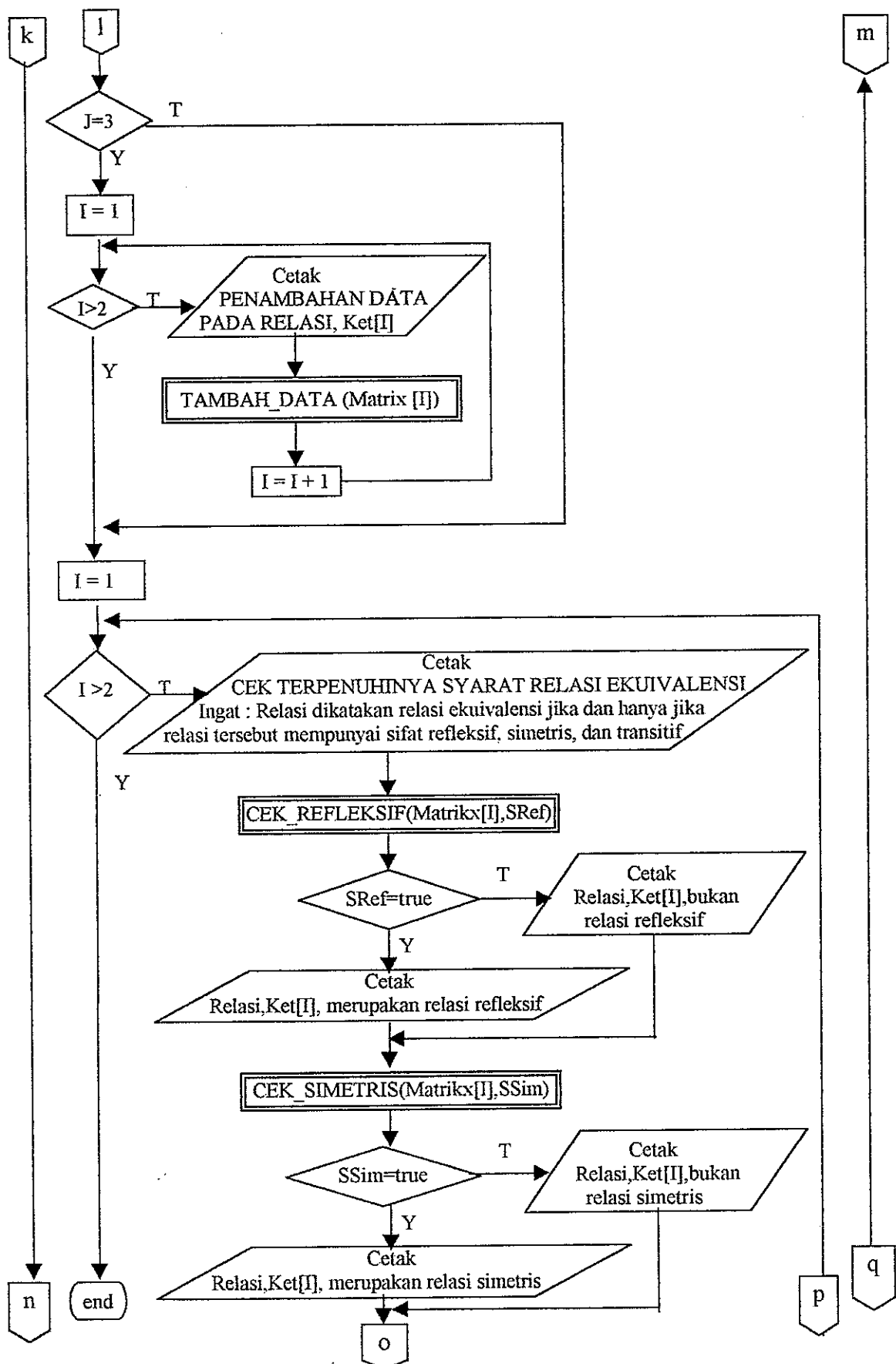
Lampiran 1. Flowchart program transitif klosur dari gabungan dua relasi ekuivalensi pada suatu himpunan



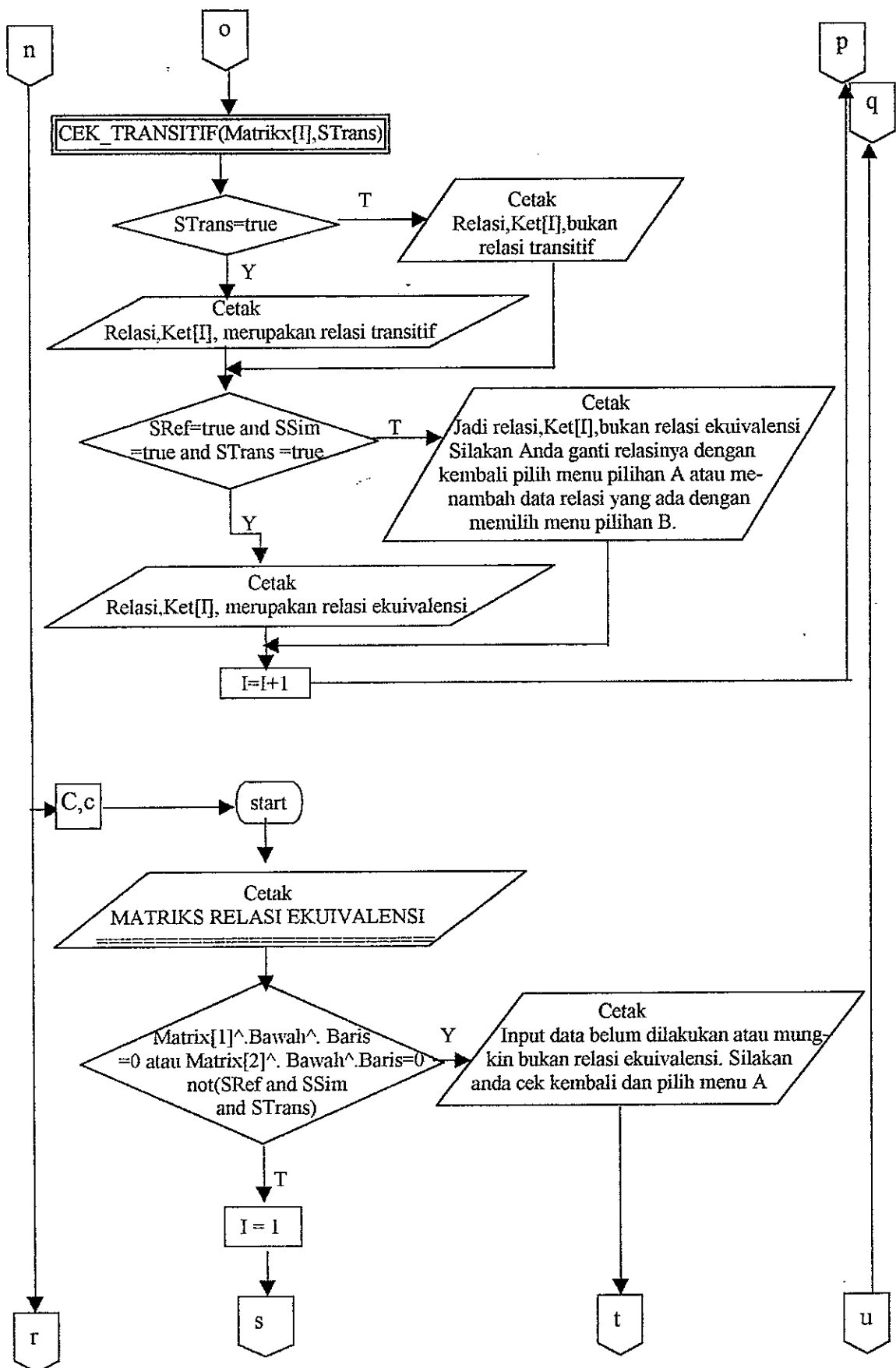
Lampiran 1. Flowchart program transitif klosur dari gabungan dua relasi ekuivalensi pada suatu himpunan



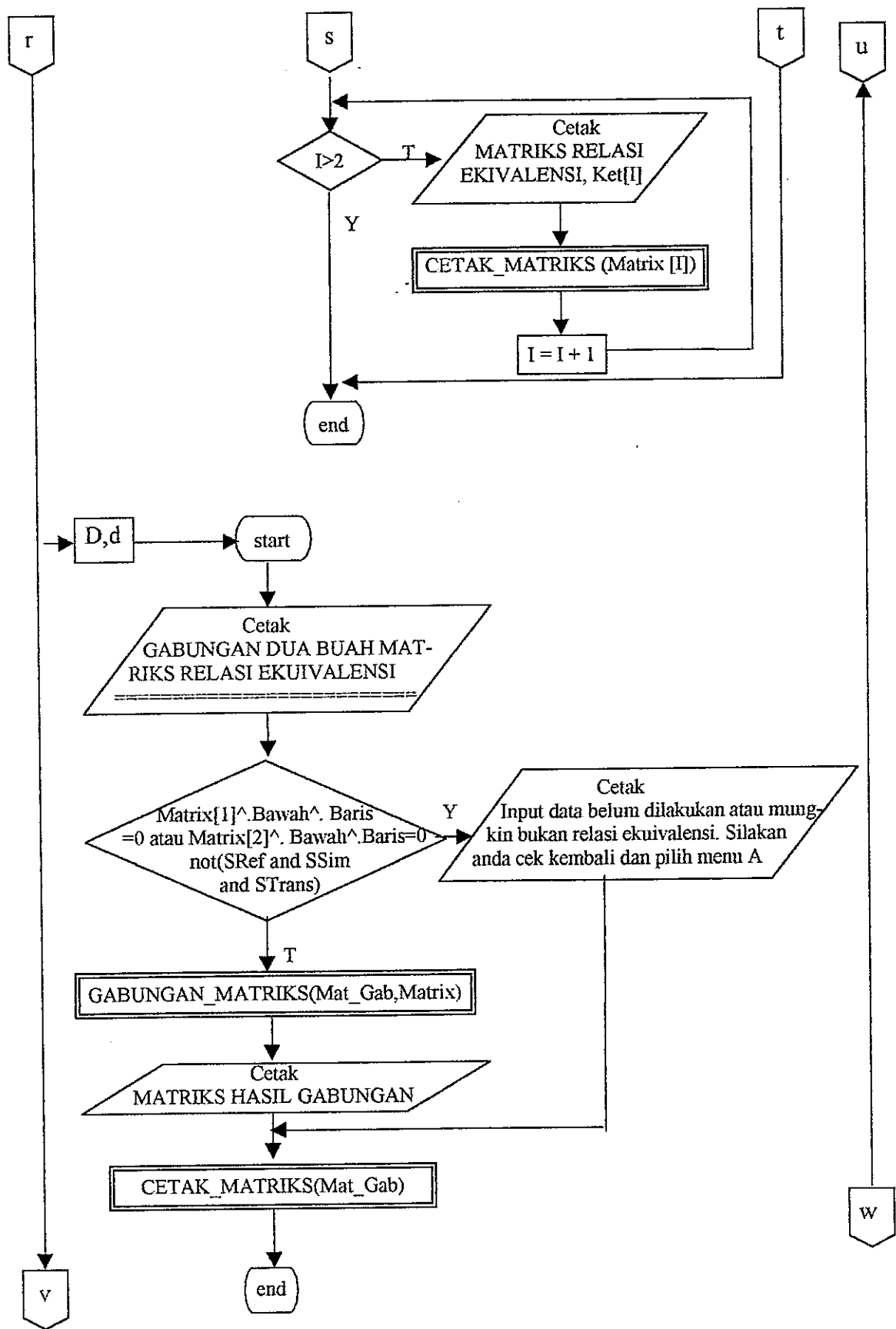
Lampiran 1. Flowchart program transitif klosur dari gabungan dua relasi ekuivalensi pada suatu himpunan



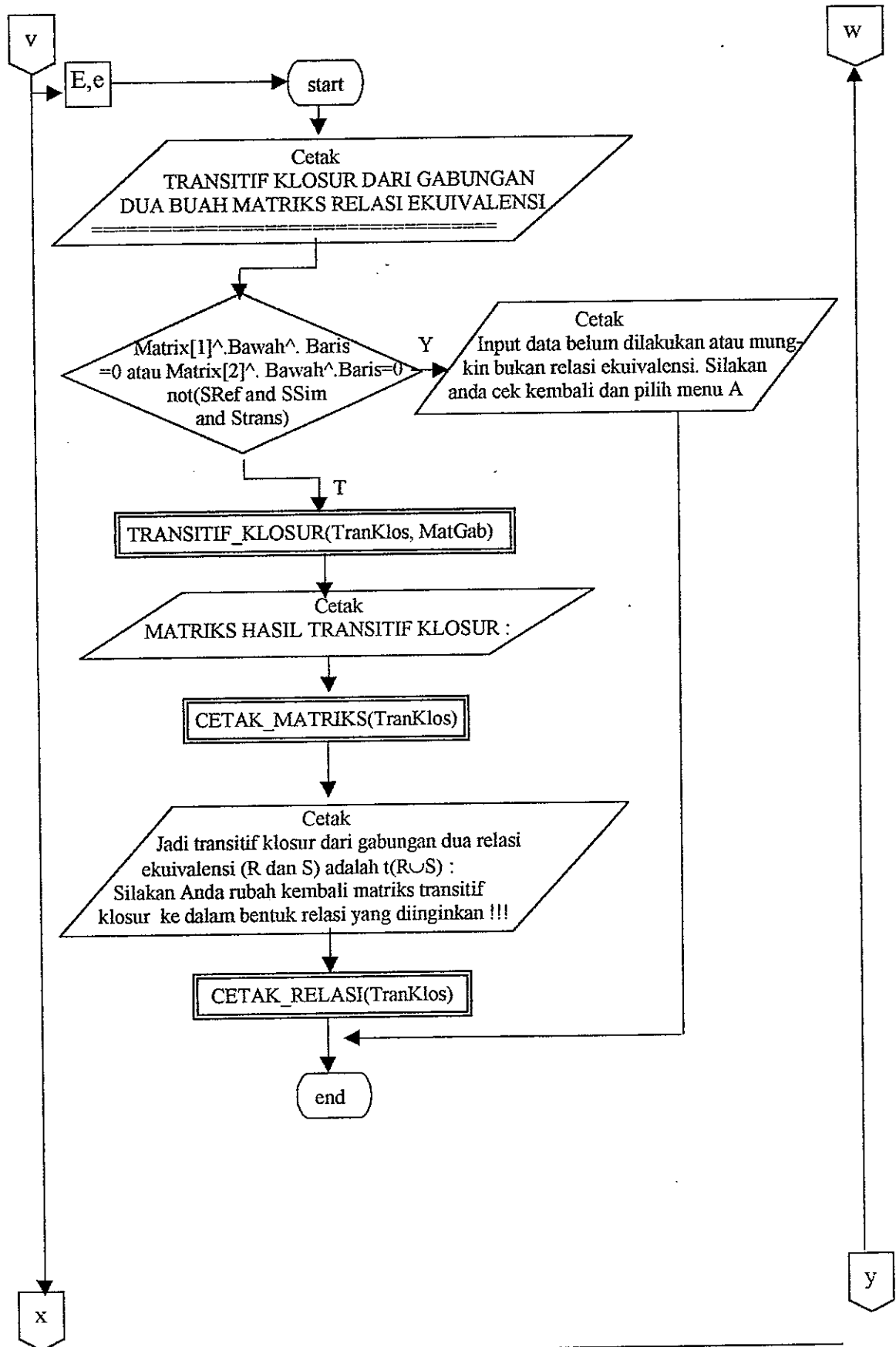
Campiran 1. Flowchart program transitif klosur dari gabungan dua relasi ekuivalensi pada suatu himpunan



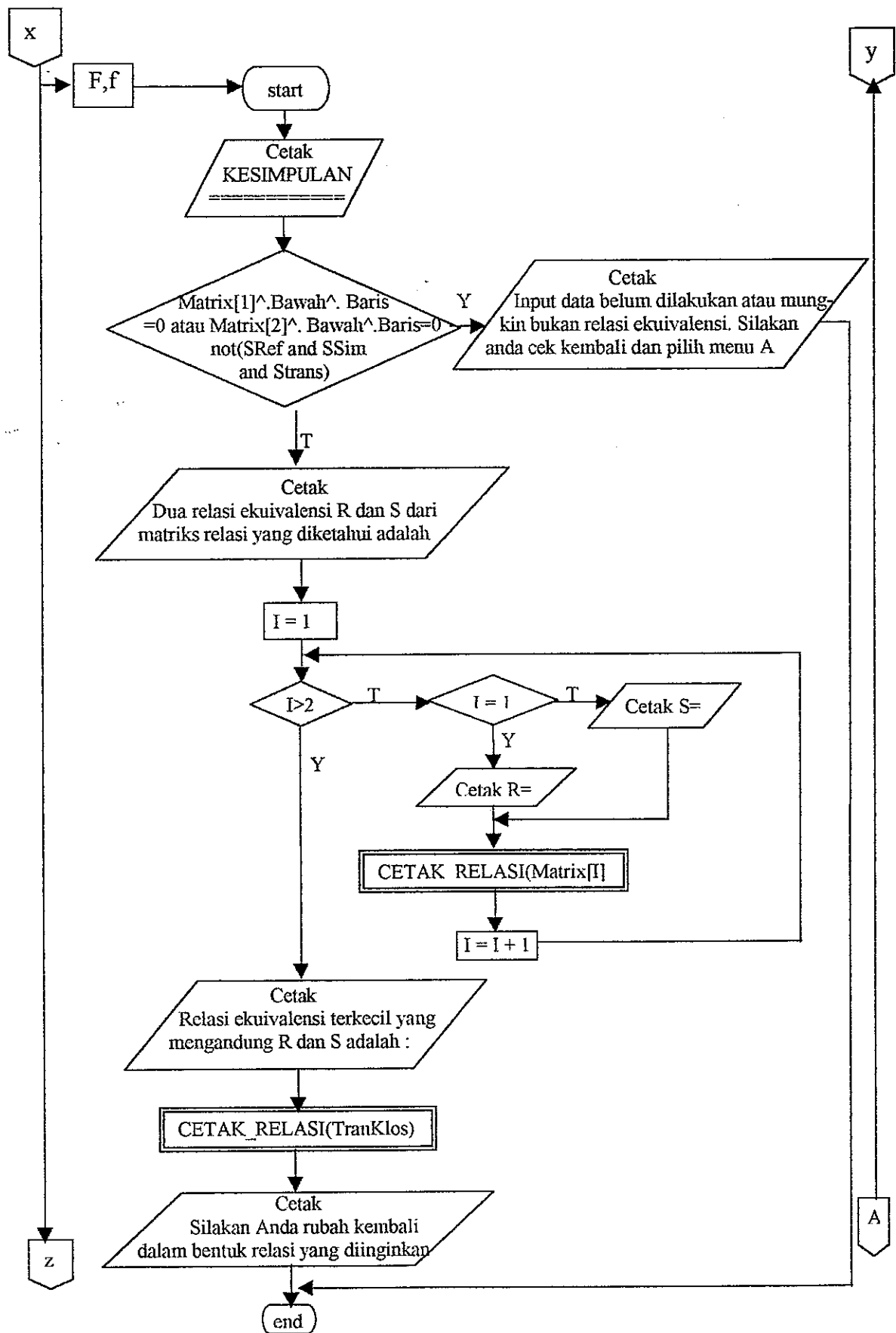
Lampiran 1. Flowchart program transitif klosur dari gabungan dua relasi ekuivalensi pada suatu himpunan



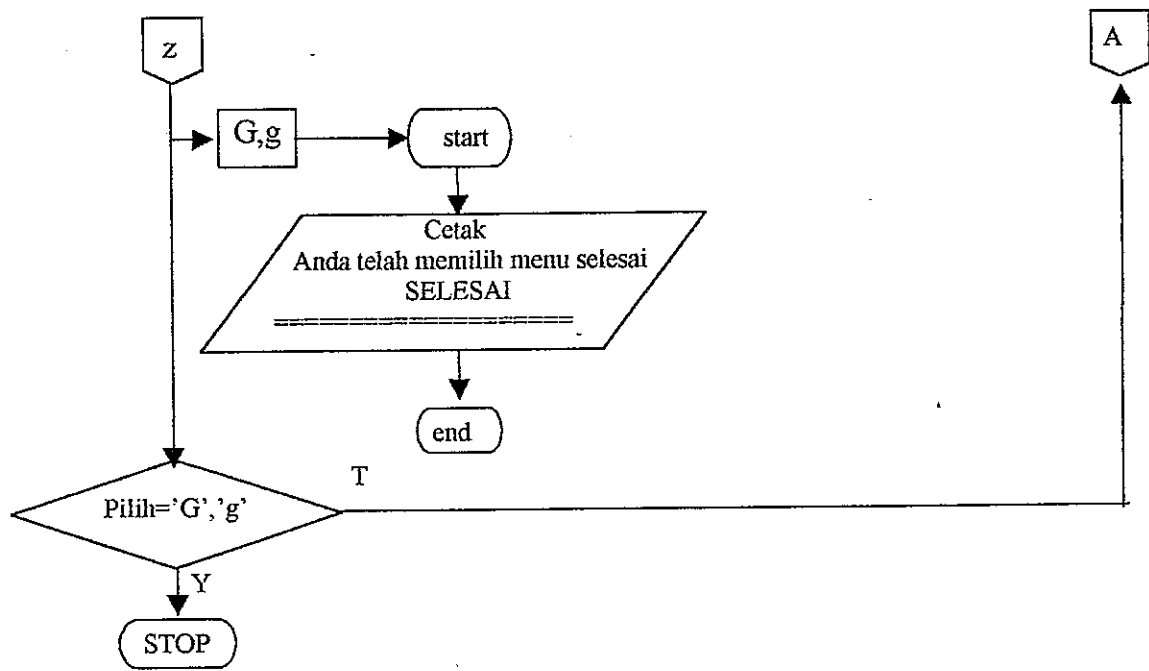
Contoh 1. Flowchart program transitif klosur dari gabungan dua relasi ekuivalensi pada suatu himpunan



Lampiran 1. Flowchart program transitif klosur dari gabungan dua relasi ekuivalensi pada suatu himpunan



Campiran 1. Flowchart program transitif klosur dari gabungan dua relasi ekuivalensi pada suatu himpunan



Lampiran 2. Program Transitif Klosur dari Gabungan Dua Relasi Ekuivalensi pada Suatu Himpunan

```
PROGRAM TRANSITIF_KLOSUR_DARI_GABUNGAN_DUA_RELASI_EKUIVALENSI ;
uses crt;
type Mat          = ^Elemen;
   Elemen         = record
       Baris, Kolom : integer;
       Kanan, Bawah : Mat
   end;
   Tabel          = array [1..2] of Mat;

var Matrix        : Tabel;
   Mat_Gab, TranKlos : Mat;
   Bar, Kol, I, J   : integer;
   Lagi, Pilih     : char;
   SRef, SSim, STrans : boolean;
   Ket             : array [1..2] of string[7];

{Prosedur untuk membuat simpul baru}
procedure BUAT_SIMPUL (var Baru : Mat; B, K : integer);
begin
   new(Baru);
   with Baru^ do
   begin
      Baris := B;
      Kolom := K;
      Kanan := nil;
      Bawah := nil
   end;
end;

{Prosedur mencek nomor baris}
procedure ADA_BARIS (var Nb : Mat; var Ada : boolean; Awal : Mat; B
: integer);
begin
   Ada := false;
   Nb := Awal^.Bawah;
   repeat
      if Nb^.Baris = B then
         Ada := true {Nomor baris ketemu}
      else
         Nb := Nb^.Bawah; {Belum ketemu}
   until Ada or (Nb = Awal) or (Nb^.Baris > B)
end;

procedure ADA_KOLOM (var Nk : Mat; var Ada : boolean; Awal : Mat; K
: integer);
begin
   Ada := false;
   Nk := Awal^.Kanan;
   repeat
      if Nk^.Kolom = K then
         Ada := true {Nomor kolom ketemu}

```

```

    else
        Nk := Nk^.Kanan; {Belum ketemu}
    until Ada or (Nk = Awal) or (Nk^.kolom > K)
end;

{Prosedur untuk menyisipkan elemen}
procedure SISIP_ELEMEN (var Awal : Mat; B,K : integer);
var Baru, Sb, Sk, N_Baris,N_Kolom, Bb : Mat;
    Ada : Boolean;
begin
    BUAT_SIMPUL(Baru, B, K);
    {Mentest nomor baris}
    ADA_BARIS(N_Baris, Ada, Awal, B);
    if not (ADA) then
        {Ini adalah baris baru, menentukan letak simpul kepala untuk
        nomor baris}
        begin
            BUAT_SIMPUL(Sb,B,0);
            {Membentuk senarai berputar}
            Sb^.Kanan := Sb;
            Sb^.Bawah := Sb;
            Bb := Awal;
            while (Bb^.Bawah^.Baris < B) and (Bb^.Bawah <> Awal) do
                Bb := Bb^.Bawah;
            if Bb^.Bawah = Awal then
                Sb^.Bawah := Awal
            else
                Sb^.Bawah := Bb^.Bawah;
                Bb^.Bawah := Sb;
            N_Baris :=Sb;
        end;

        {Mentest nomor kolom}
        ADA_KOLOM(N_Kolom, Ada, Awal, K);
        if not (ADA) then
            {Ini adalah kolom baru, menentukan letak simpul kepala untuk
            nomor kolom}
            begin
                BUAT_SIMPUL(Sk,0,K);
                {Membentuk senarai berputar}
                Sk^.Kanan := Sk;
                Sk^.Bawah := Sk;
                Bb := Awal;
                while (Bb^.Kanan^.Kolom < K) and (Bb^.Kanan <> Awal) do
                    Bb := Bb^.Kanan;
                if Bb^.Kanan = Awal then
                    Sk^.Kanan := Awal
                else
                    Sk^.kanan := Bb^.Kanan;
                    Bb^.Kanan := Sk;
                N_Kolom :=Sk;
            end;

            {menentukan letak elemen baru pada lokasinya yang tepat}
            Sb := N_Baris;
            {Mencari lokasi baris}

```

```

while(Sb^.Kanan^.Kolom < K) and (Sb^.Kanan ^.Kolom <> 0) do
  Sb := Sb^. Kanan;
  if Sb^.Kanan ^.Kolom = 0 then
    Baru ^.Kanan := N_Baris
  else
    Baru ^.Kanan := Sb^.Kanan;
  Sb^.Kanan :=Baru;
  Sk := N_Kolom;

  {Mencari lokasi kolom }
  while (Sk^.Bawah^.Baris < B) and ( Sk^.Bawah^.Baris <>0) do
    Sk := Sk^.Bawah;
  if Sk^.Bawah^.Baris = 0 then
    Baru^. Bawah := N_Kolom
  else
    Baru^.Bawah :=Sk^.Bawah;
    Sk^.Bawah := Baru;
end;

{Prosedur untuk menambah data}
procedure TAMBAH_DATA (var MT : Mat);
var Bar, Kol, Jwb, i : integer;
begin
  write('Berapa jumlah data yang ditambah? ');
  readln(Jwb);
  writeln;
  for i := 1 to Jwb do
    begin
      write('Masukkan data ke ',i,', nomor baris : ');
      readln(Bar);
      gotoxy(37,whereY-1);
      write('nomor kolom : ');
      readln(Kol);
      SISIP_ELEMEN(MT,Bar,Kol);
    end;
  end;

  {Prosedur untuk menggabungkan matriks}
  procedure GABUNGAN_MATRIKS (var Gab : Mat; M : Tabel);
  var N, S : Tabel;
      I,J,K : integer;
  begin
    {Inisialisasi matriks hasil gabungan dan membentuk senarai
    berputar}
    BUAT_SIMPUL(Gab,0,0);
    Gab^.Kanan := Gab;
    Gab^.Bawah := Gab;
    N[1] := M[1]^Bawah;
    N[2] := M[2]^Bawah;
    {proses penggabungan terjadi di sini}
    repeat
      if N[1]^Baris <> N[2]^Baris then
        {nomor baris berbeda}
        begin
          if N[1]^Baris > N[2]^Baris then
            {nomor baris pada matriks kedua lebih kecil}
            K:=2

```



```

else
  {nomor baris pada matriks pertama lebih kecil}
  K := 1;
  S[K] := N[K]^Kanan;
  {Memindah elemen dari baris yang lebih kecil}
  repeat
    SISIP_ELEMEN(Gab, S[K]^Baris, S[K]^Kolom);
    S[K] := S[K]^Kanan;
  until S[K] = N[K];
  N[K] := N[K]^Bawah
end
else
begin
  S[1] := N[1]^Kanan;
  S[2] := N[2]^Kanan;
  repeat
    if S[1]^Kolom <> S[2]^Kolom then
      {nomor kolom berbeda}
      begin
        if S[1]^Kolom > S[2]^Kolom then
          {nomor kolom pada matriks kedua lebih kecil}
          I:=2
        else
          {nomor kolom pada matriks pertama lebih
          kecil}
          I := 1;
          SISIP_ELEMEN(Gab, S[I]^Baris, S[I]^Kolom);
          S[I] := S[I]^Kanan;
        end
      end
    else
      {nomor kolom sama}
      begin
        SISIP_ELEMEN(Gab, S[1]^Baris, S[1]^Kolom);
        S[1] := S[1]^Kanan;
        S[2] := S[2]^Kanan;
      end;
    until (S[1] = N[1]) or (S[2] = N[2]);
    {semua kolom telah diproses}
    I := 0;
    if S[1] <> N[1] then
      {kolom pada matriks pertama belum semuanya di proses}
      I:= 1;
    if S[2] <> N[2] then
      {kolom pada matriks kedua belum semuanya di proses}
      I:= 2;
    if I <> 0 then
      {Ada kolom yang belum di proses}
      repeat
        SISIP_ELEMEN(Gab, S[I]^Baris, S[I]^Kolom);
        S[I] := S[I]^Kanan;
      until S[I] = N[I];
      N[1] := N[1]^Bawah;
      N[2] := N[2]^Bawah;
    end;
  until (N[1] = M[1]) or (N[2] = M[2]);
  {Semua matriks telah diproses}
  I:= 0;

```

```

if N[1] <> M[1] then
  { matriks pertama belum di proses semua}
  I := 1;
if N[2] <> M[2] then
  { matriks kedua belum semuanya di proses}
  I:= 2;
if I <> 0 then
  {Ada matriks yang belum habis}
  repeat
    S[I] := N[I]^Kanan;
    repeat
      SISIP_ELEMEN(Gab,S[I]^Baris,S[I]^Kolom);
      S[I] := S[I]^Kanan;
    until S[I] = N[I];
    N[I] := N[I]^Bawah;
  until N[I] = M[I]
end;

{Prosedur untuk mencetak matriks}
procedure CETAK_MATRIKS (Awal : Mat);
var Bb, Bbl          : Mat;
    Jk, Jb, I,J,N_Bar,N_Kol,Uk : integer;
begin
  {menentukan ukuran matriks}
  Bb := Awal^Kanan;

  {Mencari ukuran kolom maksimum}
  while Bb^kanan^Kolom <> 0 do
    Bb := Bb^Kanan;
  Jk := Bb^Kolom;
  Bb := Awal^Bawah;

  {Mencari ukuran baris maksimum}
  while Bb^Bawah^Baris <> 0 do
    Bb := Bb^Bawah;
  Jb := Bb^Baris;
  Uk := Jb;

  {Memilih ukuran maksimum}
  if (Jb <> Jk) then
    if Jb < Jk then
      Uk := Jk
    else
      Uk := Jb;
  Bb := Awal^Bawah; N_bar := 0;

  {Proses mencetak}
  repeat
    if (Bb^Baris - N_bar) <> 1 then
      {Elemen tak nol pertama tidak pada baris pertama}
      for I := N_bar + 1 to Bb^Baris - 1 do
        begin
          for J:= 1 to Uk do
            write (0:8);
          writeln;
        end;
  end;

```

```

Bb1 := Bb^.Kanan;
if (Bb1^.Kolom - Bb^.Kolom) <> 1 then
  {Elemen tak nol bukan pada kolom pertama}
  for I := 1 to Bb1^.Kolom - 1 do
    write (0:8);
  repeat
    write (1:8);
    N_Kol := Bb1^.Kolom; {elemen tak nol}
    Bb1 := Bb1^.Kanan;
    if Bb1^.Kolom <> 0 then
      if (Bb1^.Kolom - N_Kol) <> 1 then
        {ada elemen nol lainnya}
        for I := 1 to Bb1^.Kolom - N_Kol - 1 do
          write (0:8);
    until Bb1^.Kolom = 0;
    if N_Kol < Uk then
      {Menambah sejumlah '0'}
      for I := 1 to Uk - N_Kol do
        write (0:8);
    writeln;
    N_Bar := Bb^.Baris;
    Bb := Bb^.Bawah;
  until Bb^.Baris = 0;

if Jb < Jk then
  {Menambah sejumlah '0' lainnya}
  for I := 1 to Jk - Jb do
    begin
      for J := 1 to Jk do
        write (0:8);
      writeln;
    end;
end;

end;

procedure CETAK_RELASI(R:Mat);
var R1 : Mat;
begin
  write('(');
  while R^.Bawah^.Baris <> 0 do
    begin
      R := R^.Bawah;
      R1 := R^.Kanan;
      while R1^.Kolom <> 0 do
        begin
          write('(', R^.Baris, ', ', R1^.Kolom, ')');
          if (R^.Bawah^.Baris <> 0) or (R1^.Kanan^.Kolom <> 0) then
            write(', ');
          else
            writeln(')');
          R1 := R1^.Kanan;
        end;
      end;
    end;
end;
end;

```

```

{Prosedur untuk mencari transitif klosur}
Procedure TRANSITIF_KLOSUR (var TranKlos : Mat; MGab : Mat);
var Nb, Nok, Nob, Bantu, Bantul, Bantu2 : Mat;
    Ada, Adal, Ada2, AdaBar, AdaKol      : boolean;
begin
    BUAT_SIMPUL(TranKlos,0,0);
    {Membentuk senarai berputar}
    TranKlos^.Kanan := TranKlos;
    TranKlos^.Bawah := TranKlos;
    TranKlos := MGab;
    Bantu := MGab^.Kanan;
    {Mencari baris}
    while Bantu^.Kolom <> 0 do
        begin
            Bantul :=Bantu^.Bawah;
            {Mencari kolom}
            while Bantul^.Baris <> 0 do
                begin
                    {Mengecek baris}
                    ADA_BARIS (Nb,Ada,MGab,Bantul^.Kolom);
                    {Ditemukan baris}
                    if Ada then
                        begin
                            Bantu2 := Nb;
                            while Bantu2^.Kanan^.Kolom <> 0 do
                                begin
                                    Bantu2 := Bantu2^.Kanan;
                                    ADA_BARIS (Nob,Adal,TranKlos,Bantul^.
                                        Baris);
                                    ADA_KOLOM(Nok,Ada2,TranKlos,Bantu2^.
                                        Kolom);
                                    if Adal and Ada2 then
                                        {Nomor baris dan nomor kolom sudah ada}
                                        begin
                                            AdaBar := false;
                                            AdaKol := false;
                                            Nob := Nob^.Kanan;
                                            Nok := Nok^.Bawah;
                                            {Mencari elemen kolom}
                                            repeat
                                                if Nob^.Kolom=Bantu2^.Kolom then
                                                    AdaBar := true
                                                else
                                                    Nob := Nob^.Kanan;
                                            until AdaBar or (Nob^.Kolom>Bantu2
                                                ^.Kolom) or (Nob^.Kolom=0);
                                            {Mencari elemen baris}
                                            repeat
                                                if Nok^.Baris=Bantul^.Baris then
                                                    AdaKol := true
                                                else
                                                    Nok := Nok^.Bawah;
                                            until AdaKol or (Nok^.Baris>Bantul
                                                ^.Baris) or (Nok^.Baris=0);
                                            {Tidak ada elemen di baris dan
                                                kolom yang sudah ada}
                                            if not(AdaBar and AdaKol) then

```

```

                SISIP_ELEMEN(TranKlos,Bantul^.
                Baris,Bantu2^.Kolom);
            end
        else
            SISIP_ELEMEN(TranKlos,Bantul^.Baris,
            Bantu2^.Kolom)
        end;
    end;
    Bantul := Bantul^.Bawah;
end;
writeln('Matriks W(',Bantul^.Kolom,')');
writeln;
CETAK_MATRIKS(TranKlos); writeln;
if (Bantul^.Kolom mod 2) = 0 then
    begin
        write('TEKAN ENTER UNTUK MELANJUTKAN !!!');
        readkey; clrscr;
    end;
    Bantu :=Bantul^.Kanan;
end;
end;

procedure CEK_REFLEKSIF(MRel : Mat; var SRef : Boolean);
var Bantul, Bantu2, Nb, Nk : Mat;
    Ref1, Ref2, Adal, Ada2 : Boolean;
begin
    Bantul := MRel^.Bawah;
    repeat
        ADA_KOLOM (Nk, Adal, MRel, Bantul^.Baris);
        if Adal then
            begin
                Nk := Nk^.Bawah;
                Ref1 := false;
                repeat
                    if Nk^.Baris = Bantul^.Baris then
                        Ref1 := true
                    else
                        Nk := Nk^.Bawah;
                    until (Nk^.Baris=0) or Ref1 or (Nk^.Baris>Bantul^.Baris);
                end
            end
        else
            Ref1 := false;
            Bantul := Bantul^.Bawah;
        until (Bantul^.Baris = 0) or not(Ref1);
        Bantu2 := MRel^.Kanan;
        repeat
            ADA_BARIS (Nb, Ada2, MRel, Bantu2^.Kolom);
            if Ada2 then
                begin
                    Nb := Nb^.Kanan;
                    Ref2 := false;
                    repeat
                        if Nb^.Kolom = Bantu2^.Kolom then
                            Ref2 := true
                        else
                            Nb := Nb^.Kanan;
                        until (Nb^.Baris=0) or Ref2 or (Nb^.Kolom>Bantu2^.Kolom);
                    end
                end
            end
        until (Nb^.Baris=0) or Ref2 or (Nb^.Kolom>Bantu2^.Kolom);
    end
end;

```

```

        end
    else
        Ref2 := false;
        Bantu2 := Bantu2^.Kanan;
        until (Bantu2^.Kolom = 0) or not(Ref2);
        if Ref1 and Ref2 then
            SRef := true;
        else
            SRef := false;
        end;
end;

procedure CEK_SIMETRIS(MRel : Mat; var SSim : boolean);
var Bantul, Bantu2, Nb      : Mat;
    Ada                      : boolean;
begin
    Bantul := MRel^.Bawah;
    repeat
        Bantu2 := Bantul^.Kanan;
        repeat
            ADA_BARIS(Nb,Ada,MRel,Bantu2^.Kolom);
            if Ada then
                begin
                    Nb := Nb^.Kanan;
                    SSim := false;
                    repeat
                        if Nb^.Kolom = Bantul^.Baris then
                            SSim := true
                        else
                            Nb := Nb^.Kanan;
                        until SSim or (Nb^.Kolom > Bantul^.Baris) or (Nb^.Kolom = 0);
                    end
                end
            else
                SSim := false;
                Bantu2 := Bantu2^.Kanan;
                until (Bantu2^.Kolom = 0) or not(SSim);
                Bantul := Bantul^.Bawah;
            until (Bantul^.Baris = 0) or not(SSim);
        end;
end;

procedure CEK_TRANSITIF(MRel : Mat; var STrans : boolean);
var Bantul, Bantu2, Bantu3, Nb : Mat;
    Ada                      : boolean;
begin
    STrans := false;
    Bantul := MRel^.Bawah;
    repeat
        Bantu2 := Bantul^.Kanan;
        repeat
            ADA_BARIS(Nb,Ada,MRel,Bantu2^.Kolom);
            if Ada then
                begin
                    Bantu3 := Bantul;
                    repeat
                        Nb := Nb^.Kanan;
                        STrans := false;
                        repeat
                            if Bantu3^.Kolom = Nb^.Kolom then

```

```

        STrans := true
      else
        Bantu3 :=Bantu3^.Kanan;
        until STrans or (Bantu3^.Kolom>Nb^.Kolom) or (Bantu3^.
          Kolom=0);
        until (Nb^.Kanan^.Kolom = 0) or not(STrans)
      end;
      Bantu2 := Bantu2^.Kanan;
      until (Bantu2^.Kolom = 0) or not(STrans);
      Bantul := Bantul^.Bawah;
      until (Bantul^.Baris = 0) or not(STrans);
    end;

{Program utama}
begin
  clrscr;
  gotoxy(10,6);
  writeln('PROGRAM TRANSITIF KLOSUR DARI GABUNGAN DUA RELASI
    EKUIVALENSI');
  gotoxy(16,7);
  writeln('DALAM SUATU HIMPUNAN DENGAN STRUKTUR DATA DINAMIS');
  gotoxy(33,12); writeln('Di buat oleh :');
  gotoxy(25,14); writeln('Nama      : SUKMAWATI NUR ENDAH');
  gotoxy(25,15); writeln('NIM      : J2A 096 059');
  gotoxy(25,16); writeln('Jurusan  : MATEMATIKA');
  gotoxy(18,20);
  writeln('FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM');
  gotoxy(28,21);
  writeln('UNIVERSITAS DIPONEGORO');
  gotoxy(37,22); writeln('2001');
  readkey;
  repeat
    clrscr;
    gotoxy(10,5);
    writeln('PROGRAM TRANSITIF KLOSUR DARI GABUNGAN DUA RELASI
      EKUIVALENSI');
    gotoxy(16,6);
    writeln('DALAM SUATU HIMPUNAN DENGAN STRUKTUR DATA DINAMIS');
    gotoxy(30,9); writeln('>>> MENU PILIHAN <<<');
    gotoxy(17,11);writeln('A. INPUT DATA (DUA RELASI
      EKUIVALENSI)');
    gotoxy(17,12);writeln('B. MENAMBAH DATA');
    gotoxy(17,13);writeln('C. OUTPUT MATRIKS RELASI EKUIVALENSI');
    gotoxy(17,14);writeln('D. OUTPUT MATRIKS GABUNGAN DUA RELASI
      EKUIVALENSI');
    gotoxy(17,15);writeln('E. TRANSITIF KLOSUR DARI GABUNGAN DUA
      RELASI EKUIVALENSI');
    gotoxy(17,16);writeln('F. KESIMPULAN');
    gotoxy(17,17);writeln('G. SELESAI');
    gotoxy(20,19);write('PILIH SALAH SATU (A..G atau a..g) : ');
    repeat
      gotoxy (58,19); write (' ');
      gotoxy (58,19); readln(Pilih);
    until Pilih in ['A','a','B','b','C','c','D','d','E','e','F',
      'f','G','g'];
    clrscr;
  end;
end;

```

```

case Pilih of
'A','a':begin
    Ket[1] := 'PERTAMA';Ket[2] := 'KEDUA';
    gotoxy(10,5);
    writeln('«=====
=====»');
    gotoxy(10,6);
    writeln('|                PERHATIAN                |');
    gotoxy(10,7);
    writeln('|                |');
    gotoxy(10,8);
    writeln('| 1. Pemasukan data relasi ekuivalensi
berdasarkan matriks |');
    gotoxy(10,9);
    writeln('|                relasinya yaitu hanya pada elemen
yang bernilai 1.    |');
    gotoxy(10,10);
    writeln('| 2. Jika Anda ingin mengakhiri pemasukan
data, ketikkan    |');
    gotoxy(10,11);
    writeln('|                bilangan 0 sesudah kata Nomor baris.
|');
    gotoxy(10,12);
    writeln('|                |');
    gotoxy(10,13);
    writeln('«=====
=====»');
    readln;
    clrscr;
    gotoxy(10,2);
    writeln('PROGRAM TRANSITIF KLOSUR DARI GABUNGAN DUA
RELASI EKUIVALENSI');
    gotoxy(16,3);
    writeln('DALAM SUATU HIMPUNAN DENGAN STRUKTUR DATA
DINAMIS');
    gotoxy(1,5);
    for I:= 1 to 2 do
        begin
            writeln('MASUKKAN POSISI ELEMEN YANG BERNILAI 1
PADA MATRIKS RELASI EKUIVALENSI',Ket[I],':');
            writeln;
            {inisialisasi matrik}
            BUAT_SIMPUL(Matrix[I],0,0);
            Matrix[I]^Bawah := Matrix[I];
            Matrix[I]^Kanan := Matrix[I];
            {Memasukkan data}
            repeat
                write ('Nomor baris : ');
                readln(Bar);
                if Bar <> 0 then
                    begin
                        gotoxy(24,whereY-1);
                        write ('Nomor kolom : ');
                        readln(Kol);
                        {menyusun matriknya}
                        SISIP_ELEMEN(Matrix[I],Bar,Kol);
                    end;
            until Bar = 0;
        end;
end;

```



```

        until Bar = 0;writeln;
    end;
    clrscr;
    writeln(' >> CEK TERPENUHINYA SYARAT RELASI
    EKUIVALENSI << ');
    writeln;
    writeln('Ingat : Relasi dikatakan relasi
    ekuivalensi jika dan hanya jika relasi ');
    writeln('      tersebut mempunyai sifat
    refleksif, simetris, dan transitif');
    writeln;
    for I := 1 to 2 do
        begin
            CEK_REFLEKSIF(Matrix[I],SRef);
            if SRef then
                writeln('Relasi ',Ket[I],' merupakan
                relasi refleksif')
            else
                writeln('Relasi ',Ket[I],' bukan relasi
                refleksif');
            CEK_SIMETRIS(Matrix[I],SSim);
            if SSim then
                writeln('Relasi ',Ket[I],' merupakan
                relasi simetris')
            else
                writeln('Relasi ',Ket[I],' bukan relasi
                simetris');
            CEK_TRANSITIF(Matrix[I],STrans);
            if STrans then
                writeln('Relasi ',Ket[I],' merupakan
                relasi transitif')
            else
                writeln('Relasi ',Ket[I],' bukan relasi
                transitif');
            writeln;
            if SRef and SSim and STrans then
                writeln('Relasi ',Ket[I],' merupakan
                relasi ekuivalensi')
            else
                begin
                    writeln('Jadi relasi ',Ket[I],' bukan
                    relasi ekuivalensi');
                    writeln('Silakan Anda ganti relasinya
                    dengan kembali pilih menu pilihan A
                    atau');
                    writeln('menambah data relasi yang
                    telah ada dengan memilih menu pilihan
                    B. ');
                end;
            writeln;
        end;
    end;
    readln;
end;
'B','b':begin
    gotoxy(10,2);
    writeln('PROGRAM TRANSITIF KLOSUR DARI GABUNGAN DUA
    RELASI EKUIVALENSI');

```

```

gotoxy(16,3);
writeln('DALAM SUATU HIMPUNAN DENGAN STRUKTUR DATA
DINAMIS');
gotoxy(28,5); writeln('>> MENAMBAH DATA RELASI
<<');
gotoxy(31,6); writeln('=====');
writeln;
writeln('Relasi mana yang akan ditambah datanya ?
(Jika relasi I, silakan tulis 1;');
writeln('relasi II, tulis 2; relasi keduanya, tulis
3)');
writeln;
write('          Jawab : ');
readln(J);
writeln;
{Menambah data di relasi pertama atau kedua}
for I:=1 to 2 do
  if I = J then
    begin
      writeln('PENAMBAHAN DATA PADA RELASI
',KET[I]);
      writeln;
      TAMBAH_DATA(Matrix[I]);
    end;
{Menambah data di relasi pertama dan kedua}
if J = 3 then
  for I :=1 to 2 do
    begin
      writeln('PENAMBAHAN DATA PADA RELASI
',KET[I]);
      writeln;
      TAMBAH_DATA(Matrix[I]);
      writeln;
    end;
  clrscr;
  writeln(' >> CEK TERPENUHINYA SYARAT RELASI
EKUIVALENSI << ');
  writeln;
  writeln('Ingat : Relasi dikatakan relasi
ekuivalensi jika dan hanya jika relasi ');
  writeln('          tersebut mempunyai sifat
refleksif, simetris, dan transitif');
  writeln;
  for I := 1 to 2 do
    begin
      CEK_REFLEKSIF(Matrix[I],SRef);
      if SRef then
        writeln('Relasi ',Ket[I],' merupakan
relasi refleksif')
      else
        writeln('Relasi ',Ket[I],' bukan relasi
refleksif');
      CEK_SIMETRIS(Matrix[I],SSim);
      if SSim then
        writeln('Relasi ',Ket[I],' merupakan
relasi simetris')
      else

```

```

        writeln('Relasi ',Ket[I],' bukan relasi
        simetris');
    .CEK_TRANSITIF(Matrix[I],STrans);
    if STrans then
        writeln('Relasi ',Ket[I],' merupakan
        relasi transitif')
    else
        writeln('Relasi ',Ket[I],' bukan relasi
        transitif');
    writeln;
    if SRef and SSim and STrans then
        writeln('Relasi ',Ket[I],' merupakan
        relasi ekuivalensi')
    else
        begin
            writeln('Jadi relasi ',Ket[I],' bukan
            relasi ekuivalensi');
            writeln('Silakan Anda ganti relasinya
            dengan kembali pilih menu pilihan A
            atau');
            writeln('menambah data relasi yang telah
            ada dengan memilih menu pilihan B.');
```

```

            end;
        writeln;
        end;
        readln;
    end;
    'C','c':begin
        gotoxy(10,2);
        writeln('PROGRAM TRANSITIF KLOSUR DARI GABUNGAN DUA
        RELASI EKUIVALENSI');
        gotoxy(16,3);
        writeln('DALAM SUATU HIMPUNAN DENGAN STRUKTUR DATA
        DINAMIS');
        gotoxy(23,5);    writeln('>>          MATRIKS          RELASI
        EKUIVALENSI <<');
        gotoxy(26,6);
        writeln('=====');
        writeln;writeln;
        if (Matrix[1]^Bawah^.Baris=0) or (Matrix[2]^
        Bawah^.Baris=0) or not(SRef and SSim and
        STrans)then
            begin
                writeln('Input data belum dilakukan atau
                mungkin relasi yang dimasukkan bukan');
                writeln('relasi ekuivalensi. Silakan Anda cek
                kembali dan pilih menu A');
            end
        else
            for I:= 1 to 2 do
                begin
                    writeln('MATRIKS RELASI EKUIVALENSI
                    ',Ket[I],':');writeln;
                    CETAK_MATRIKS(Matrix[I]); writeln;
                end;
            readln;
        end;
    end;

```

```

'D','d':begin
    gotoxy(10,2);
    writeln('PROGRAM TRANSITIF KLOSUR DARI GABUNGAN DUA
    RELASI EKUIVALENSI');
    gotoxy(16,3);
    writeln('DALAM SUATU HIMPUNAN DENGAN STRUKTUR DATA
    DINAMIS');
    gotoxy(15,5);
    writeln('>> GABUNGAN DUA BUAH MATRIKS RELASI
    EKUIVALENSI <<');
    gotoxy(18,6);
    writeln('=====');
    writeln;
    if (Matrix[1]^Bawah^.Baris=0) or (Matrix[2]^
    Bawah^.Baris=0) or not(SRef and SSim and
    STrans)then
        begin
            writeln('Input data belum dilakukan atau
            mungkin relasi yang dimasukkan bukan');
            writeln('relasi ekuivalensi. Silakan Anda cek
            kembali dan pilih menu A');
        end
    else
        begin
            GABUNGAN_MATRIKS(Mat_Gab,Matrix);
            writeln('MATRIKS HASIL GABUNGAN: ');
            writeln;
            CETAK_MATRIKS(Mat_Gab);
        end;
        readln;
    end;
'E','e':begin
    gotoxy(10,2);
    writeln('PROGRAM TRANSITIF KLOSUR DARI GABUNGAN DUA
    RELASI EKUIVALENSI');
    gotoxy(16,3);
    writeln('DALAM SUATU HIMPUNAN DENGAN STRUKTUR DATA
    DINAMIS');
    gotoxy(4,5);
    writeln('>> TRANSITIF KLOSUR DARI GABUNGAN DUA BUAH
    MATRIKS RELASI EKUIVALENSI <<');
    gotoxy(7,6);
    writeln('=====');
    GABUNGAN_MATRIKS(Mat_Gab,Matrix);
    writeln;
    if (Matrix[1]^Bawah^.Baris=0) or (Matrix[2]^
    Bawah^.Baris=0) or not(SRef and SSim and STrans)
    then
        begin
            writeln('Input data belum dilakukan atau
            mungkin relasi yang dimasukkan bukan');
            writeln('relasi ekuivalensi. Silakan Anda cek
            kembali dan pilih menu A');
        end
    else
        begin
            TRANSITIF_KLOSUR(TranKlos,Mat_Gab);

```

```

        writeln('Matriks Hasil Transitif Klossur : ');
        writeln;
        CETAK_Matriks(TranKloss);
        writeln;
        writeln('Jadi transitif klossur dari gabungan
        dua relasi ekuivalensi (R dan S) adalah');
        write('t(R U S) = ');
        CETAK_RELASI(TranKloss);
        writeln;
        writeln('Silakan Anda rubah kembali matriks
        transitif klossur ke dalam bentuk relasi yang
        diinginkan !!');
    end;
    readln;
end;
'F','f':begin
    gotoxy(10,2);
    writeln('PROGRAM TRANSITIF KLOSSUR DARI GABUNGAN DUA
    RELASI EKUIVALENSI');
    gotoxy(16,3);
    writeln('DALAM SUATU HIMPUNAN DENGAN STRUKTUR DATA
    DINAMIS');
    gotoxy(30,6);writeln('>> KESIMPULAN <<');
    gotoxy(33,7);writeln('=====');
    writeln;
    if (Matrix[1]^Bawah^.Baris=0) or (Matrix[2]^
    Bawah^.Baris=0) or not(SRef and SSim and STRans)
    then
        begin
            writeln('Input data belum dilakukan atau
            mungkin relasi yang dimasukkan bukan');
            writeln('relasi ekuivalensi. Silakan Anda cek
            kembali dan pilih menu A');
        end
    else
        begin
            writeln('Dua relasi ekuivalensi R dan S dari
            matriks relasi yang diketahui adalah :');
            for I:= 1 to 2 do
                begin
                    if I = 1 then
                        begin
                            gotoxy(5,10);
                            write('R = ');
                        end
                    else
                        begin
                            gotoxy(5,11);
                            write('S = ');
                        end;
                    CETAK_RELASI(MATRIX[I]);
                end;
            writeln;
            writeln('Relasi ekuivalensi terkecil yang
            mengandung R dan S adalah');
            write(' ');
            CETAK_RELASI(TranKloss);
        end;
    end;
end;

```

```

        writeln;
        writeln('Silakan anda rubah kembali dalam
        bentuk relasi yang diinginkan');
    end;
    readln;
end;
'G','g':begin
    gotoxy(20,10);writeln('Anda telah memilih menu
    selesai !!!');
    writeln;
    gotoxy(33,11);writeln('SELESAI');
    gotoxy(20,12);
    writeln('=====');
end;
end;
until Pilih in ['G','g'];
readln;
end.

```

Lampiran 3. Output Program

1. Tampilan halaman depan output

PROGRAM TRANSITIF KLOSUR DARI GABUNGAN DUA RELASI EKUIVALENSI
DALAM SUATU HIMPUNAN DENGAN STRUKTUR DATA DINAMIS

Di buat oleh :

Nama : SUKMAWATI NUR ENDAH
NIM : J2A 096 059
Jurusan : MATEMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS DIPONEGORO
2001

2. Tampilan menu pilihan

PROGRAM TRANSITIF KLOSUR DARI GABUNGAN DUA RELASI EKUIVALENSI
DALAM SUATU HIMPUNAN DENGAN STRUKTUR DATA DINAMIS

>>> MENU PILIHAN <<<

- A. INPUT DATA (DUA RELASI EKUIVALENSI)
- B. MENAMBAH DATA
- C. OUTPUT MATRIKS RELASI EKUIVALENSI
- D. OUTPUT MATRIKS GABUNGAN DUA RELASI EKUIVALENSI
- E. TRANSITIF KLOSUR DARI GABUNGAN DUA RELASI EKUIVALENSI
- F. KESIMPULAN
- G. SELESAI

PILIH SALAH SATU (A..G atau a..g) :

3. Tampilan menu pilihan A (Input Data)

```
-----+-----
                          PERHATIAN
1. Pemasukan data relasi ekuivalensi berdasarkan matriks
   relasinya yaitu hanya pada elemen yang bernilai 1.
2. Jika Anda ingin mengakhiri pemasukan data, ketikkan
   bilangan 0 sesudah kata Nomor baris.
-----+-----
```

4. Tampilan contoh pemasukan data relasi ekuivalensi (contoh 3.1.3.1.)

PROGRAM TRANSITIF KLOSUR DARI GABUNGAN DUA RELASI EKUIVALENSI
DALAM SUATU HIMPUNAN DENGAN STRUKTUR DATA DINAMIS

MASUKKAN POSISI ELEMEN YANG BERNILAI 1 PADA MATRIKS RELASI
EKUIVALENSI PERTAMA:

```
Nomor baris : 1      Nomor kolom : 1
Nomor baris : 1      Nomor kolom : 2
Nomor baris : 2      Nomor kolom : 1
Nomor baris : 2      Nomor kolom : 2
Nomor baris : 3      Nomor kolom : 3
Nomor baris : 3      Nomor kolom : 4
Nomor baris : 4      Nomor kolom : 3
Nomor baris : 4      Nomor kolom : 4
Nomor baris : 5      Nomor kolom : 5
Nomor baris : 0
```

MASUKKAN POSISI ELEMEN YANG BERNILAI 1 PADA MATRIKS RELASI
EKUIVALENSI KEDUA:

```
Nomor baris : 1      Nomor kolom : 1
Nomor baris : 2      Nomor kolom : 2
Nomor baris : 3      Nomor kolom : 3
Nomor baris : 4      Nomor kolom : 4
Nomor baris : 4      Nomor kolom : 5
Nomor baris : 5      Nomor kolom : 4
Nomor baris : 5      Nomor kolom : 5
Nomor baris : 0
```

>> CEK TERPENUHINYA SYARAT RELASI EKUIVALENSI <<

Ingat : Relasi dikatakan relasi ekuivalensi jika dan hanya jika
relasi tersebut mempunyai sifat refleksif, simetris, dan
transitif

Relasi PERTAMA merupakan relasi refleksif
Relasi PERTAMA merupakan relasi simetris
Relasi PERTAMA merupakan relasi transitif

Relasi PERTAMA merupakan relasi ekuivalensi

Relasi KEDUA merupakan relasi refleksif
Relasi KEDUA merupakan relasi simetris
Relasi KEDUA merupakan relasi transitif

Relasi KEDUA merupakan relasi ekuivalensi

5. Tampilan menu pilihan C (Matriks Relasi Ekuivalensi)

PROGRAM TRANSITIF KLOSUR DARI GABUNGAN DUA RELASI EKUIVALENSI
DALAM SUATU HIMPUNAN DENGAN STRUKTUR DATA DINAMIS

>> MATRIKS RELASI EKUIVALENSI <<
=====

MATRIKS RELASI EKUIVALENSI PERTAMA:

1	1	0	0	0
1	1	0	0	0
0	0	1	1	0
0	0	1	1	0
0	0	0	0	1

MATRIKS RELASI EKUIVALENSI KEDUA:

1	0	0	0	0
0	1	0	0	0
0	0	1	0	0
0	0	0	1	1
0	0	0	1	1

6. Tampilan menu pilihan D (Matriks Gabungan Dua Relasi Ekuivalensi)

PROGRAM TRANSITIF KLOSUR DARI GABUNGAN DUA RELASI EKUIVALENSI
DALAM SUATU HIMPUNAN DENGAN STRUKTUR DATA DINAMIS

>> GABUNGAN DUA BUAH MATRIKS RELASI EKUIVALENSI <<
=====

MATRIKS HASIL GABUNGAN:

1	1	0	0	0
1	1	0	0	0
0	0	1	1	0
0	0	1	1	1
0	0	0	1	1

7. Tampilan menu pilihan E (Transitif Klossur dari Gabungan Dua Relasi

Ekuivalensi)

PROGRAM TRANSITIF KLOSSUR DARI GABUNGAN DUA RELASI EKUIVALENSI
DALAM SUATU HIMPUNAN DENGAN STRUKTUR DATA DINAMIS

>> TRANSITIF KLOSSUR DARI GABUNGAN DUA BUAH MATRIKS RELASI EKUIVALENSI <<

=====

Matriks W(1):

1	1	0	0	0
1	1	0	0	0
0	0	1	1	0
0	0	1	1	1
0	0	0	1	1

Matriks W(2):

1	1	0	0	0
1	1	0	0	0
0	0	1	1	0
0	0	1	1	1
0	0	0	1	1

TEKAN ENTER UNTUK MELANJUTKAN !!

Matriks W(3):

1	1	0	0	0
1	1	0	0	0
0	0	1	1	0
0	0	1	1	1
0	0	0	1	1

Matriks W(4):

1	1	0	0	0
1	1	0	0	0
0	0	1	1	1
0	0	1	1	1
0	0	1	1	1

TEKAN ENTER UNTUK MELANJUTKAN !!

Matriks W(5):

1	1	0	0	0
1	1	0	0	0
0	0	1	1	1
0	0	1	1	1
0	0	1	1	1

MATRIKS HASIL TRANSITIF KLOSUR :

1	1	0	0	0
1	1	0	0	0
0	0	1	1	1
0	0	1	1	1
0	0	1	1	1

Jadi transitif klosur dari gabungan dua relasi ekuivalensi (R dan S) adalah

$t(R \cup S) = \{(1,1), (1,2), (2,1), (2,2), (3,3), (3,4), (3,5), (4,3), (4,4), (4,5), (5,3), (5,4), (5,5)\}$

Silakan Anda rubah kembali matriks transitif klosur ke dalam bentuk relasi yang diinginkan !!

8. Tampilan menu pilihan F (Kesimpulan)

PROGRAM TRANSITIF KLOSUR DARI GABUNGAN DUA RELASI EKVIVALENSI
DALAM SUATU HIMPUNAN DENGAN STRUKTUR DATA DINAMIS

>> KESIMPULAN <<
=====

Dua relasi ekuivalensi R dan S dari matriks relasi yang diketahui adalah :

$R = \{(1,1), (1,2), (2,1), (2,2), (3,3), (3,4), (4,3), (4,4), (5,5)\}$
 $S = \{(1,1), (2,2), (3,3), (4,4), (4,5), (5,4), (5,5)\}$

Relasi ekuivalensi terkecil yang mengandung R dan S adalah

$\{(1,1), (1,2), (2,1), (2,2), (3,3), (3,4), (3,5), (4,3), (4,4), (4,5), (5,3), (5,4), (5,5)\}$

Silakan anda rubah kembali dalam bentuk relasi yang diinginkan

9. Tampilan menu pilihan G (Selesai)

Anda telah memilih menu selesai !!!
SELESAI

10. Tampilan menu pilihan A untuk data relasi yang bukan ekuivalensi

PROGRAM TRANSITIF KLOSUR DARI GABUNGAN DUA RELASI EKUIVALENSI
DALAM SUATU HIMPUNAN DENGAN STRUKTUR DATA DINAMIS

MASUKKAN POSISI ELEMEN YANG BERNILAI 1 PADA MATRIKS RELASI
EKUIVALENSI PERTAMA:

Nomor baris : 1	Nomor kolom : 1
Nomor baris : 2	Nomor kolom : 2
Nomor baris : 1	Nomor kolom : 2
Nomor baris : 2	Nomor kolom : 3
Nomor baris : 3	Nomor kolom : 4
Nomor baris : 4	Nomor kolom : 4
Nomor baris : 0	

MASUKKAN POSISI ELEMEN YANG BERNILAI 1 PAA MATRIKS RELASI
EKUIVALENSI KEDUA:

Nomor baris : 1	Nomor kolom : 1
Nomor baris : 2	Nomor kolom : 2
Nomor baris : 3	Nomor kolom : 3
Nomor baris : 4	Nomor kolom : 4
Nomor baris : 1	Nomor kolom : 2
Nomor baris : 3	Nomor kolom : 1
Nomor baris : 0	

>> CEK TERPENUHINYA SYARAT RELASI EKUIVALENSI <<

Ingat : Relasi dikatakan relasi ekuivalensi jika dan hanya jika relasi tersebut mempunyai sifat refleksif, simetris, dan transitif

Relasi PERTAMA bukan relasi refleksif
Relasi PERTAMA bukan relasi simetris
Relasi PERTAMA merupakan relasi transitif

Relasi PERTAMA bukan relasi ekuivalensi
Silakan Anda ganti relasinya dengan kembali pilih menu pilihan A atau menambah data relasi yang telah ada dengan memilih menu pilihan B.

Relasi KEDUA merupakan relasi refleksif
Relasi KEDUA bukan relasi simetris
Relasi KEDUA merupakan relasi transitif

Relasi KEDUA bukan relasi ekuivalensi
Silakan Anda ganti relasinya dengan kembali pilih menu pilihan A atau menambah data relasi yang telah ada dengan memilih menu pilihan B.

11. Tampilan menu pilihan E untuk data relasi yang bukan ekuivalensi

PROGRAM TRANSITIF KLOSUR DARI GABUNGAN DUA RELASI EKUIVALENSI
DALAM SUATU HIMPUNAN DENGAN STRUKTUR DATA DINAMIS

>> TRANSITIF KLOSUR DARI GABUNGAN DUA BUAH MATRIKS RELASI EKUIVALENSI <<
=====

Input data belum dilakukan atau mungkin relasi yang dimasukkan bukan relasi ekuivalensi. Silakan Anda cek kembali dan pilih menu A

12. Tampilan menu pilihan C untuk data relasi yang bukan ekuivalensi

PROGRAM TRANSITIF KLOSUR DARI GABUNGAN DUA RELASI EKUIVALENSI
DALAM SUATU HIMPUNAN DENGAN STRUKTUR DATA DINAMIS

>> MATRIKS RELASI EKUIVALENSI <<
=====

Input data belum dilakukan atau mungkin relasi yang dimasukkan bukan relasi ekuivalensi. Silakan Anda cek kembali dan pilih menu A

13. Tampilan menu pilihan B (Menambah Data)

PROGRAM TRANSITIF KLOSUR DARI GABUNGAN DUA RELASI EKUIVALENSI
DALAM SUATU HIMPUNAN DENGAN STRUKTUR DATA DINAMIS

>> MENAMBAH DATA RELASI <<
=====

Relasi mana yang akan ditambah datanya ? (Jika relasi I, silakan tulis 1; relasi II, tulis 2; relasi keduanya, tulis 3)

Jawab : 3

PENAMBAHAN DATA PADA RELASI PERTAMA

Berapa jumlah data yang ditambah? 1

Masukkan data ke 1, nomor baris : 2 nomor kolom : 3

PENAMBAHAN DATA PADA RELASI KEDUA

Berapa jumlah data yang ditambah? 2

Masukkan data ke 1, nomor baris : 1 nomor kolom : 3

Masukkan data ke 2, nomor baris : 4 nomor kolom : 2

>> CEK TERPENUHINYA SYARAT RELASI EKUIVALENSI <<

Ingat : Relasi dikatakan relasi ekuivalensi jika dan hanya jika relasi tersebut mempunyai sifat refleksif, simetris, dan transitif

Relasi PERTAMA merupakan relasi refleksif

Relasi PERTAMA bukan relasi simetris

Relasi PERTAMA bukan relasi transitif

Jadi relasi PERTAMA bukan relasi ekuivalensi

Silakan Anda ganti relasinya dengan kembali pilih menu pilihan A atau menambah data relasi yang telah ada dengan memilih menu pilihan B.

Relasi KEDUA merupakan relasi refleksif

Relasi KEDUA bukan relasi simetris

Relasi KEDUA bukan relasi transitif

Jadi relasi KEDUA bukan relasi ekuivalensi

Silakan Anda ganti relasinya dengan kembali pilih menu pilihan A atau menambah data relasi yang telah ada dengan memilih menu pilihan B.