LISTING PROGRAM C++

METODE KOMPUTASI

2006

DAFTAR ISI

Halaman

PENGANTAR

PENGENALAN KOMPUTASI NUMERIK

Listing Program C++

```
/* Contoh soal Pengantar Metode Komputasi Diambilkan dari Chapra-Canale 2004 */
/* Nama Program : parachute_all.cpp */

#include <iostream.h>
#include <stdlib.h>
#include <conio.h>
#include <stdio.h>
#include <math.h>

#define e 2.718281828
#define g 9.80
#define c 12.5

main()
{
     FILE *analitik, *numerik;
     double    vt, et, cmt, m,
               vti1, vi;
     int i, option;
     double t1, t2, dt;
     char lagi;

     do
     {
          clrscr();

          cout << "\n\nPerhitungan Analitik dan Numerik Falling Parachute";
          cout << "\n================================================\n\n";
          cout << "Menu Pilihan\n";
          cout << "------------\n\n";
          cout << "(1)  Perhitungan Analitik\n\n";
          cout << "(2)  Perhitungan Numerik\n\n";
          cout << "Pilihan Anda ? ";
          cin >> option;

          switch(option)
          {
               case 1 :
               {
                    analitik = fopen("parachute_analitik.txt", "w+");
                    cout << "\n\nPerhitungan Analitik Falling Parachute";
                    cout << "\n====================================\n\n";
                    cout << "Berapa Nilai massa (m)       ? ";
                    cin >> m;
                    cout << "\nMasukkan Waktu Awal (t1)     ? ";
                    cin >> t1;
                    cout << "\nMasukkan Waktu Akhir (t2)    ? ";
                    cin >> t2;
                    cout << "\nMasukkan Interval Waktu (dt) ? ";
                    cin >> dt;
```

```
      printf("\n\t-------------------------------------------------
      --------\n");
      printf("\tIterasi ke-\tt\te(t)\t\tv(t)\n");
      printf("\t-------------------------------------------------
      ------\n\n");

      fprintf(analitik, "\n\t-----------------------------------------
      -------------------\n");
      fprintf(analitik, "\tIterasi ke-\tt\te(t)\t\tv(t)\n");
      fprintf(analitik, "\t-----------------------------------------
      -----------------\n\n");

      for (i = t1; i <= t2; i++)
      {
            //rumus v(t)= (g*m*(1-e^-(c/m)*t))/c
            cmt = (c * t1)/m;
            et = 1 - pow(e,-cmt);
            vt = (g*m*et)/c;

            printf("\t\t%.d\t%.1f\t%.5f\t\t%.5f\n", i+1, t1, et, vt);
            fprintf(analitik,"\t\t%.d\t%.1f\t%.5f\t%.5f\n", i+1, t1,
            et, vt);

            t1 += dt;
      }

      printf("\n\t-------------------------------------------------
      --------\n\n");
      fprintf(analitik,"\n\t-----------------------------------------
      -----------------\n\n");
      fprintf(analitik,"Dimana\n\nMassa       = %.2f\n", m);
      fprintf(analitik,"Koefisien c = %.2f\n", c);
      fprintf(analitik,"Parameter g = %.2f\n", g);
      fclose(analitik);
      break;
}

case 2 :
{
      vi = 0.0;
      t1 = 0;

      numerik = fopen("parachute_numerik.txt", "w+");
      cout << "\n\nPerhitungan Numerik Falling Parachute";
      cout << "\n====================================\n\n";
      cout << "Berapa Nilai massa (m)       ? ";
      cin >> m;
      cout << "\nMasukkan Waktu Awal (t1)     ? ";
      cin >> t1;
      cout << "\nMasukkan Waktu Akhir (t2)    ? ";
      cin >> t2;
      cout << "\nMasukkan Interval Waktu (dt) ? ";
      cin >> dt;

      printf("\n\t-------------------------------------------------
      --------\n");
      printf("\tIterasi ke-\tt\tv(ti)\t\tv(ti+1)\n");
      printf("\t-------------------------------------------------
      ------\n\n");
      fprintf(numerik, "\n\t-----------------------------------------
      -----------------\n");
      fprintf(numerik, "\tIterasi ke-\tt\tv(ti)\t\tv(ti+1)\n");
      fprintf(numerik, "\t-----------------------------------------
      ----------------\n\n");
```

```
                    for (i = t1; i <= t2; i++)
                    {
                         //rumus v(ti+1)= vi + (g - (c*vi)/m)*(ti1+ti)

                         vti1 = vi + (g - (c*vi)/m) * dt;

                         printf("\t\t%.d\t%.1f\t%.4f\t\t%.5f\n", i+1, t1, vi,
                         vti1);
                         fprintf(numerik,"\t\t%.d\t%.1f\t%.4f\t\t%.5f\n", i+1, t1,
                         vi, vti1);

                         vi = vti1;
                         t1 += dt;
                    }

                    printf("\n\t-------------------------------------------------------
                    --------\n\n");
                    fprintf(numerik,"\n\t-----------------------------------------
                    -----------------\n\n");
                    fprintf(numerik,"Dimana\n\nMassa       = %.2f\n", m);
                    fprintf(numerik,"Koefisien c = %.2f\n", c);
                    fprintf(numerik,"Parameter g = %.2f\n", g);
                    fclose(numerik);
                    break;
               }

               default : cout << "\n\nNo Good, man .......... ";
          }

          printf("Dimana\n\nMassa       = %.2f\n", m);
          printf("Koefisien c = %.2f\n", c);
          printf("Parameter g = %.2f\n", g);

          cout << "\n\nCoba lagi dengan data awal yang berbeda (y/t) ? ";
          cin >> lagi;

     }while(lagi != 't');

     return 0;
}
```

KESALAHAN DAN BILANGAN PENDEKATAN

Listing Program C++

```
/* Contoh soal Kesalahan Komputasi Bab I Hal. 6-8 Diktat Metode Komputasi 2004 */
/* Nama Program : error-komputasi.cpp */

#include <iostream.h>
#include <stdlib.h>
#include <conio.h>
#include <stdio.h>
#include <math.h>

#include "faktorial.hpp"

#define e 2.718281828
#define error_limit 0.00000001 /* 8-digit maksimal, lebih overflow */

main()
{
     FILE *stream;
     int j;
     double exact, ex, ex1, x, i;
     double fact;
     double Ee, Ea;
     char lagi;

     do
     {
          clrscr();
          stream = fopen("error-komputasi.txt", "w+");
          cout << "\n\nMencari Kesalahan Komputasi\n\n";
          cout << "==========================\n\n";
          cout << "Persamaan Matematika Approksimasi\n";
          cout << "---------------------------------\n\n";
          cout << "e^x=1+(x/faktorial(x))+......\n\n";

          cout << "\nBerapa Nilai x awal = ";
          cin >> x;

          i = 0.0;
          j = 0;
          fact = 0.0;
          Ea = 0.0;
          /* Penghitungan e^x */
          exact = pow(e,x);
          printf("\nNilai yang didekati adalah %.9f\n", exact);
          cout << "\nTekan Enter untuk melanjutkan .........\n";
          getch();
          ex1 = exact;

          printf("\n\t-------------------------------------------------------------
          \n");
          printf("\tIterasi ke-\tAproksimasi\tEe\t\tEa\n");
          printf("\t-------------------------------------------------------------
          \n\n");
```

```
        fprintf(stream,"\n\t-------------------------------------------------------
        --------\n");
        fprintf(stream,"\tIterasi ke-\tAproksimasi\tEe\t\tEa\n");
        fprintf(stream,"\t-------------------------------------------------------
        ------\n\n");

        do
        {
             j++;
             if (i != 0.0)
             {
                  fact += pow(x,i)/factorial(i);
             }

             ex = 1 + fact;
             Ee = ((exact-ex)/exact)*100;

             if(i != 0.0)
             {
                  Ea = ((ex-ex1)/ex)*100;
                  printf("\t\t%d\t%.9f\t%.9f\t%.9f\n", j, ex, Ee, Ea);
                  fprintf(stream,"\t\t%d\t%.9f\t%.9f\t%.9f\t\n", j, ex, Ee, Ea);
                  ex1 = ex;
             }
             else
             {
                  printf("\t\t%d\t%.9f\t%.8f\t%.5f\n", j, ex, Ee, Ea);
                  fprintf(stream,"\t\t%d\t%.9f\t%.9f\t%.9f\t\n", j, ex, Ee, Ea);
                  ex1 = ex;
             }
             i++;
        } while(fabs(Ee) > error_limit);

        printf("\n\t-------------------------------------------------------------
        \n\n");
        fprintf(stream,"\n\t-------------------------------------------------------
        -------\n\n");
        printf("\nNilai e^%.2f = %.9f didekati dengan x = %.2f  ", x, exact, x);
        printf("dalam %d iterasi", j);
        printf("\n\nDengan Nilai Approksimasi = %.9f", ex);
        printf("\n\nDengan Error Relatif absolut sebesar = %.9f", fabs(Ee));
        fprintf(stream,"\nNilai e^%.2f = %.9f didekati dengan x = %.2f  ", x,
        exact, x);
        fprintf(stream,"dalam %d iterasi", j);
        fprintf(stream,"\n\nDengan Nilai Approksimasi = %.9f", ex);
        fprintf(stream,"\n\nDengan Error Relatif absolut sebesar = %.9f",
        fabs(Ee));

        fclose(stream);

        cout << "\n\n\nCoba lagi dengan Nilai x berbeda (y/t) ? ";
        cin >> lagi;
        clrscr();

    }while(lagi != 't');

    return 0;
}
```

```
/* Header untuk program error_komputasi.cpp */
/* Nama Header : faktorial.hpp */

int factorial(int);

int factorial(int x)
{
  int i, p=1;
  for(i=1; i<=x; i++)
  {
   p*=i;
  }
  return(p);
}
```

```
/* Contoh soal Kesalahan Komputasi Bab I Hal. 6-8 Diktat Metode Komputasi 2004 */
/* Nama Program : error-komputasi_v2.cpp */

#include <iostream.h>
#include <stdlib.h>
#include <conio.h>
#include <stdio.h>
#include <math.h>

#include "faktorial.hpp"

#define e 2.718281828
#define error_limit 0.00000001

main()
{
     FILE *stream;
     int j, k;
     double exact, ex, ex1, x, i;
     double fact;
     double Ee, Ea;
     char lagi;

     do
     {
          clrscr();
          stream = fopen("error-komputasi.txt", "w+");
          cout << "\n\nMencari Kesalahan Komputasi\n\n";
          cout << "==========================\n\n";
          cout << "Persamaan Matematika Approksimasi\n";
          cout << "---------------------------------\n\n";
          cout << "e^x=1+(x/faktorial(x))+......\n\n";

          cout << "\nBerapa Nilai x awal ? ";
          cin >> x;
          cout << "\n\nBerapa suku pendekatan ? ";
          cin >> k;

          cout << "\n\nRumus pendekatan";
          printf("\n\ne^%.2f = %.2f", x, pow(e,0));
          for(j = 1; j < k; j++)
          {
               printf(" + %.2f^%d/%d!", x, j, j);
          }

          i = 0.0;
          j = 0;
          fact = 0.0;
          Ea = 0.0;
          /* Penghitungan e^x */
          exact = pow(e,x);
          printf("\n\n\nNilai yang didekati adalah %.9f\n", exact);
          cout << "\nTekan Enter untuk melanjutkan .........\n";
          getch();
          ex1 = exact;

          printf("\n\t---------------------------------------------------------------
          \n");
          printf("\tIterasi ke-\tAproksimasi\tEe\t\tEa\n");
          printf("\t-------------------------------------------------------------
          \n\n");
          fprintf(stream, "\n\t-----------------------------------------------------
          --------\n");
          fprintf(stream, "\tIterasi ke-\tAproksimasi\tEe\t\tEa\n");
          fprintf(stream, "\t-------------------------------------------------------
          ------\n\n");
```

```
        do
        {
            j++;
            if (i != 0)
            {
                fact += pow(x,i)/factorial(i);
            }

            ex = 1 + fact;
            Ee = ((exact-ex)/exact)*100;

            if(i != 0)
            {
                Ea = ((ex-ex1)/ex)*100;
                printf("\t\t%d\t%.9f\t%.9f\t%.9f\n", j, ex, Ee, Ea);
                fprintf(stream,"\t\t%d\t%.9f\t%.9f\t%.9f\t\n", j, ex, Ee, Ea);
                ex1 = ex;
            }
            else
            {
                printf("\t\t%d\t%.9f\t%.8f\t%.5f\n", j, ex, Ee, Ea);
                fprintf(stream,"\t\t%d\t%.9f\t%.9f\t%.9f\t\n", j, ex, Ee, Ea);
                ex1 = ex;
            }
            i++;
        } while(j!= k);//(fabs(Ee) > error_limit);

        printf("\n\t-----------------------------------------------------------------
        \n\n");
        fprintf(stream,"\n\t-----------------------------------------------------------
        -------\n\n");
        printf("\nNilai e^%.2f = %.9f didekati dengan x = %.2f  ", x, exact, x);
        printf("dalam %d iterasi", j);
        printf("\n\nDengan Nilai Approksimasi = %.9f", ex);
        printf("\n\nDengan Error Relatif absolut sebesar = %.9f", fabs(Ee));
        fprintf(stream,"\nNilai e^%.2f = %.9f didekati dengan x = %.2f  ", x,
        exact, x);
        fprintf(stream,"dalam %d iterasi", j);
        fprintf(stream,"\n\nDengan Nilai Approksimasi = %.9f", ex);
        fprintf(stream,"\n\nDengan Error Relatif absolut sebesar = %.9f",
        fabs(Ee));

        fclose(stream);

        cout << "\n\n\nCoba lagi dengan Nilai x berbeda (y/t) ? ";
        cin >> lagi;
        clrscr();

    }while(lagi != 't');

    return 0;
}
```

PENYELESAIAN AKAR-AKAR PERSAMAAN TIDAK LINIER

Listing Program C++

```
/* Contoh soal Metode Grafik Bab II Hal. 9-10. Diktat Metode Komputasi 2004 */
/* Nama Program : grafik.cpp */

#include <iostream.h>
#include <stdlib.h>
#include <conio.h>
#include <stdio.h>
#include <math.h>

main()
{
     FILE *stream;
     int i, j;
     double x, dx, y1, y2, dy;

     clrscr();
     stream = fopen("grafik.txt", "w+");

     cout << "Metode Grafik\n";
     cout << "-------------\n\n";

     cout << "Maksimum Iterasi = ";
     cin >>j;
     cout<< "Berapa Nilai Awal x = ";
     cin >> x;
     cout<< "Berapa Nilai Interval x = ";
     cin >> dx;

     //persamaan awal x^4 - 3x - 2
     //y1 = x^4, y2 = 3x + 2

     printf("\n\t------------------------------------------------\n");
     printf("\tIterasi ke-\tx\ty1\ty2\tSelisih\n");
     printf("\t------------------------------------------------\n\n");
     fprintf(stream, "\n\t------------------------------------------------\n");
     fprintf(stream, "\tIterasi ke-\tx\ty1\ty2\tSelisih\n");
     fprintf(stream, "\t------------------------------------------------\n\n");

     for(i = 0; i < j; i++)
     {
          y1 = pow(x,4);
          y2 = 3*x + 2;
          dy = y1 - y2;

          printf("\t\t%d\t%.2f\t%.2f\t%.2f\t%.2f\n", i+1, x, y1, y2, dy);
          fprintf(stream,"\t\t%d\t%.2f\t%.2f\t%.2f\t%.2f\n", i+1, x, y1, y2, dy);

          x += dx;
     }
```

```
    printf("\t-----------------------------------------------\n\n");
    fprintf(stream, "\t-----------------------------------------------\n\n");

    fclose(stream);
    getch();
    return 0;
}
```

```
/* Contoh soal Metode Tabulasi Bab II Hal. 10-11 Diktat Metode Komputasi 2004 */
/* Nama Program : tabulasi.cpp */

#include <iostream.h>
#include <stdlib.h>
#include <conio.h>
#include <stdio.h>
#include <math.h>

main()
{
     FILE *stream;
     int i, j;
     double y1, y2, x, dy, dx, y;
     char ulang, lagi1, lagi2, awal;

     do
     {
          clrscr();
          stream = fopen("tabulasi_1a.txt", "w+");
          do
          {
               clrscr();
               cout << "\nMetode Tabulasi\n";
               cout << "Contoh Soal di Diktat\n";
               cout << "=====================\n\n";

               cout << "Persamaan Utama dan Persamaan untuk Titik Potongnya\n\n";
               cout << "f(x) = x^4 - 3*x - 2 = 0\n";
               cout << "y1 = x^4, y2 = 3x + 2\n";
               cout << "\n\nBerapa Nilai Awal x = ";
               cin >> x;
               cout << "\nBerapa Nilai Interval x = ";
               cin >> dx;
               cout << "\nBerapa kali Iterasi = ";
               cin >> j;

               printf("\n\t-------------------------------------------------\n");
               printf("\tIterasi ke-\tx\ty1\ty2\tSelisih\n");
               printf("\t-------------------------------------------------\n\n");
               fprintf(stream, "\n\t-------------------------------------------------
               \n");
               fprintf(stream, "\tIterasi ke-\tx\ty1\ty2\tSelisih\n");
               fprintf(stream, "\t-------------------------------------------------
               \n\n");

               for(i = 0; i < j; i++)
               {
                    y1 = pow(x,3);
                    y2 = -4*x + 6;
                    dy = y1 - y2;

                    printf("\t\t%d\t%.2f\t%.2f\t%.2f\t%.2f\n", i+1, x, y1, y2, dy);
                    fprintf(stream,"\t\t%d\t%.2f\t%.2f\t%.2f\t%.2f\n", i+1, x, y1,
                    y2,dy);

                    x += dx;
               }

               printf("\n\t-------------------------------------------------\n\n");
               fprintf(stream, "\n\t-------------------------------------------------
               \n\n");

               cout << "Apakah x telah sesuai (y/t) ? ";
               cin >> ulang;
```

```
          }while(ulang != 'y');

     do
     {
          cout << "\n\nLanjut ke proses berikutnya ... (y/t) ? ";
          cin >> lagi1;

          if(lagi1 != 't')
          {
               cout << "\n\nSubstitusi Nilai x ke Persamaan f(x)";
               cout << "\n---------------------------------------\n";
               cout << "\nMasukkan Nilai x di bawah Nilai x Aproksimasi = ";
               cin >> x;
               cout << "\nBerapa Nilai Interval x = ";
               cin >> dx;
               printf("\n\n\t--------------------------------\n");
               printf("\tIterasi ke-\tx\ty\n");
               printf("\t--------------------------------\n\n");
               fprintf(stream, "\n\n\t--------------------------------\n");
               fprintf(stream, "\tIterasi ke-\tx\ty\n");
               fprintf(stream, "\t--------------------------------\n\n");

               for(i = 0; i < j+10; i++)
               {
                    y = pow(x,3) + 4*x - 6;
                    printf("\t\t%d\t%.3f\t%.3f\n", i+1, x, y);
                    fprintf(stream, "\t\t%d\t%.3f\t%.3f\n", i+1, x, y);
                    x += dx;
               }

               printf("\t--------------------------------\n\n");
               fprintf(stream, "\t--------------------------------\n\n");
          }
          else
          {
               return 0;
          }

          cout << "\nCoba lagi dengan interval yang lebih akurat (y/t) ? ";
          cin >> lagi2;

     }while(lagi2 != 't');

     cout << "\n\nCoba lagi dari awal (y/t) ? ";
     cin >> awal;
     clrscr();
     fclose(stream);

   }while(awal != 't');

   return 0;
}
```

```
/* Penyelesain Soal Metode Tabulasi Bab II Hal. 20 No. 1.b. Diktat Metode Komputasi
2004 */
/* Nama Program : tabulasi_1b.cpp */

#include <iostream.h>
#include <stdlib.h>
#include <conio.h>
#include <stdio.h>
#include <math.h>

#define e 2.718281828

main()
{
     FILE *stream;
     int i, j;
     double y1, y2, x, dy, dx, y;
     char ulang, lagi1, lagi2, awal;

     do
     {
          clrscr();
          stream = fopen("tabulasi_1b.txt", "w+");

          do
          {
               clrscr();
               cout << "\nMetode Tabulasi\n";
               cout << "Jawaban Soal No. 1.b.\n";
               cout << "=====================\n\n";

               cout << "Persamaan dan Persamaan untuk Titik Potongnya\n\n";
               cout << "f(x) = e^3 - x - 2 = 0\n";
               cout << "y1 = e^3, y2 = x + 2\n";
               cout << "\n\nBerapa Nilai Awal x = ";
               cin >> x;
               cout << "\nBerapa Nilai Interval x = ";
               cin >> dx;
               cout << "\nBerapa kali Iterasi = ";
               cin >> j;

               printf("\n\t-------------------------------------------------\n");
               printf("\tIterasi ke-\tx\ty1\ty2\tSelisih\n");
               printf("\t-------------------------------------------------\n\n");
               fprintf(stream, "\n\t-------------------------------------------------
               \n");
               fprintf(stream, "\tIterasi ke-\tx\ty1\ty2\tSelisih\n");
               fprintf(stream, "\t-------------------------------------------------
               \n\n");

               for(i = 0; i < j; i++)
               {
                    y1 = pow(e,3);
                    y2 = x + 2;
                    dy = y1 - y2;

                    printf("\t\t%d\t%.2f\t%.2f\t%.2f\t%.2f\n", i+1, x, y1, y2, dy);
                    fprintf(stream,"\t\t%d\t%.2f\t%.2f\t%.2f\t%.2f\n", i+1, x, y1,
                    y2,dy);

                    x += dx;
               }

               printf("\n\t-------------------------------------------------\n\n");
               fprintf(stream, "\n\t-------------------------------------------------
               \n\n");
```

```
            cout << "Apakah x telah sesuai (y/t) ? ";
            cin >> ulang;

        }while(ulang != 't');

        do
        {
            cout << "\n\nLanjut ke proses berikutnya ... (y/t) ? ";
            cin >> lagi1;

            if(lagi1 != 't')
            {
                cout << "\n\nSubstitusi Nilai x ke Persamaan f(x)";
                cout << "\n---------------------------------------\n";
                cout << "\nMasukkan Nilai x di bawah Nilai x Aproksimasi = ";
                cin >> x;
                cout << "\nBerapa Nilai Interval x = ";
                cin >> dx;
                printf("\n\n\t--------------------------------\n");
                printf("\tIterasi ke-\tx\ty\n");
                printf("\t--------------------------------\n\n");
                fprintf(stream, "\n\n\t--------------------------------\n");
                fprintf(stream, "\tIterasi ke-\tx\ty\n");
                fprintf(stream, "\t--------------------------------\n\n");

                for(i = 0; i < j+10; i++)
                {
                    y = pow(e,3) - x - 2;
                    printf("\t\t%d\t%.3f\t%.3f\n", i+1, x, y);
                    fprintf(stream, "\t\t%d\t%.3f\t%.3f\n", i+1, x, y);
                    x += dx;
                }

                printf("\t--------------------------------\n\n");
                fprintf(stream, "\t--------------------------------\n\n");
            }
            else
            {
                return 0;
            }

            cout << "\nCoba lagi dengan interval yang lebih akurat (y/t) ? ";
            cin >> lagi2;

        }while(lagi2 != 't');

        cout << "\n\nCoba lagi dari awal (y/t) ? ";
        cin >> awal;
        clrscr();
        fclose(stream);

    }while(awal != 't');

    return 0;
}
```

```
/* Contoh soal Metode Bolzano Bab II Hal. 12-13 Diktat Metode Komputasi 2004 */
/* Nama Program : bolzano.cpp */

#include <iostream.h>
#include <stdlib.h>
#include <conio.h>
#include <stdio.h>
#include <math.h>

main()
{
     FILE *stream;
     int i;
     double fx0, fx1, fx2, fx3, x0, x00, x1, x2, x3;
     char lagi;

     do
     {
          clrscr();
          stream = fopen("bolzano.txt", "w+");

          cout << "Metode Bolzano\n";
          cout << "==============\n\n";
          //Persamaan utama
          cout << "f(x) = x^3 - 9*x +1 = 0\n\n";

          cout << "\nBerapa Nilai batas bawah (x0) ? ";
          cin >> x0;

          x00 = x0;

          do
          {
               cout << "\nBerapa Nilai batas atas (x1) ? ";
               cin >> x1;

               fx0 = pow(x0,3) - 9*x0 + 1;
               fx1 = pow(x1,3) - 9*x1 + 1;

          }while(fx0*fx1 >= 0.0);

          printf("\n\t---------------------------------------------------------------
          \n");
          printf("\tIterasi ke-\tx(i)\t\tfx(i)\t\tinterval x\n");
          printf("\t---------------------------------------------------------------
          \n\n");
          fprintf(stream, "\n\t-----------------------------------------------------------
          --------\n");
          fprintf(stream, "\tIterasi ke-\tx(i)\t\tfx(i)\t\tinterval x\n");
          fprintf(stream, "\t-----------------------------------------------------
          -----------\n\n");

          i = 1;
          //rumus x2 = (x0+x1)/2
          x2 = (x0+x1)/2;
          fx2 = pow(x2,3) - 9*x2 + 1;

          printf("\t   %d\tx%d = %.3f\tfx(%d) = %.8f\t",i, i+1, x2, i+1, fabs(fx2));
          printf("[%.3f,%.3f]\n", x0, x1);
          fprintf(stream,"\t   %d\t\t%.3f\t\t%.8f\t", i, x2, fx2);
          fprintf(stream,"[%.3f,%.3f]\n", x0, x1);
```

```
        if (fx0*fx2 < 0)
        {
             do
             {
                  i++;
                  x3 = (x0+x2)/2;
                  fx3 = pow(x3,3) - 9*x3 + 1;

                  printf("\t   %d\tx%d = %.3f\tfx(%d) = %.8f\t", i, i+1, x3, i+1,
                  fabs(fx3));
                  fprintf(stream,"\t   %d\t\t%.3f\t\t%.8f\t", i, x3, fabs(fx3));

                  if (fx2*fx3 < 0)
                  {
                       printf("[%.3f,%.3f]\n", x0, x3);
                       fprintf(stream,"[%.3f,%.3f]\n", x0, x3);
                       x0 = x3;
                       }
                  else
                  {
                       printf("[%.3f,%.3f]\n", x2, x3);
                       fprintf(stream,"[%.3f,%.3f]\n", x2, x3);
                       x2 = x3;
                  }
             } while(fabs(fx3) > 0.00000001);
        }
        else
        {
             printf("\nAkar-akar Real terletak antara x = %.3f\tx = %.3f", x0, x2);
             getch();
        }

        printf("\n\t------------------------------------------------------------------
        \n\n");
        fprintf(stream,"\n\t------------------------------------------------------------
        ------\n\n");
        printf("\nBatas atas x0 = %.3f dan batas bawah x1 = %.3f\n", x00, x1);
        printf("\nAkar Real adalah x = %.3f\n", x3);
        printf("\nDengan Iterasi sebanyak %d kali", i);
        fprintf(stream,"\nBatas atas x0 = %.3f dan batas bawah x1 = %.3f\n", x00,
        x1);
        fprintf(stream,"\nAkar Real adalah x = %.3f", x3);
        fprintf(stream,"\nDengan Iterasi sebanyak %d kali", i);

        fclose(stream);

        cout << "\n\nCoba lagi (y/t) ? ";
        cin >> lagi;

    }while(lagi != 't');

    return 0;
}
```

```
/* Contoh soal Regula-Falsi Bab II Hal. 14-15 Diktat Metode Komputasi 2004 */
/* Nama Program : regula-falsi.cpp */

#include <iostream.h>
#include <stdlib.h>
#include <conio.h>
#include <stdio.h>
#include <math.h>

#define e 2.718281828
#define error_limit 0.00000000000001 //flexible sesuai kebutuhan

main()
{
     FILE *regula_falsi, *regula_x;
     double     fx1, fx2, fx3,
                x1, x2, x3,
                x10, x20, f1f2,
                x, dx, y1, y2, dy;
     int i, j, k;
     char lagi, belum;

     do
     {
          clrscr();
          j = 0;
          regula_falsi = fopen("regula_falsi.txt", "w+");
          regula_x = fopen("cross_regula.txt", "w+");

          do
          {
               clrscr();
               cout << "Metode Regula-Falsi\n";
               cout << "===================\n\n";
               //Persamaan utama dan persamaan untuk mencari titik potongnya
               cout << "f(x) = 2*e^x - x - 3\n\n";
               cout << "y1 = 2e^x dan y2 = x+3\n\n";

               cout << "Mencari Titik Potong terdekat\n";
               cout << "-----------------------------\n\n";
               cout << "Berapa Nilai awal x = ";
               cin >> x;
               cout << "\nBerapa Nilai Interval x = ";
               cin >> dx;
               cout << "\nBerapa kali Iterasi = ";
               cin >> k;

               printf("\n\t-------------------------------------------------\n");
               printf("\tIterasi ke-\tx\ty1\ty2\tselisih\n");
               printf("\t-------------------------------------------------\n\n");
               fprintf(regula_x,"\n\t-------------------------------------------------
               \n");
               fprintf(regula_x,"\tIterasi ke-\tx\ty1\ty2\tselisih\n");
               fprintf(regula_x,"\t-------------------------------------------------
               \n\n");

               for (i = 0; i < k; i++)
               {
                    y1 = 2*pow(e,x);
                    y2 = x + 3;
                    dy = y1 - y2;

                    printf("\t\t%d\t%.1f\t%.2f\t%.2f\t%.2f\n", i+1, x, y1, y2, dy);
```

```
            fprintf(regula_x, "\t\t%d\t%.1f\t%.2f\t%.2f\t%.2f\n", i+1, x, y1,
            y2,dy);
            x += dx;
      }


      printf("\n\t-------------------------------------------------\n\n");
      fprintf(regula_x, "\t-------------------------------------------------
      \n\n");
      fclose(regula_x);

      cout << "\nApakah Titik Potong telah ditemukan ? ";
      cin >> belum;

}while(belum != 'y');

cout << "\nMencari Titik Potong sebenarnya\n";
cout << "==============================\n\n";
cout << "Masukkan Nilai x1 = ";
cin >> x1;

x10 = x1;
fx1 = 2*pow(e,x1) - x1 - 3;

do
{
      cout << "\nMasukkan Nilai x2 = ";
      cin >> x2;
      fx2 = 2*pow(e,x2) - x2 - 3;
      x20 = x2;

      f1f2 = fx1*fx2;

}while(f1f2 >= 0.0);

printf("\n-------------------------------------------------------------------
----------\n");
printf("Iterasi ke-\tx3\tfx3\t\tfx3 (16 digit) Error Aproksimasi\n");
printf("-------------------------------------------------------------------
--------\n\n");
fprintf(regula_falsi,"\n-----------------------------------------------------
-----------------------\n");
fprintf(regula_falsi,"Iterasi ke-\tx3\tfx3\t\tfx3 (16 digit) Error
Aproksimasi\n");
fprintf(regula_falsi, "-----------------------------------------------------
-----------------------\n\n");

do
{
      j++;
      x3 = (x1*fx2 - x2*fx1)/(fx2 - fx1);
      fx3 = 2*pow(e,x3) - x3 - 3;

      printf("\t%d\t%.3f\t%.6f\t%.16f\n", j, x3, fx3, fabs(fx3));
      fprintf(regula_falsi,"\t%d\t%.3f\t%.6f\t%.16f\n", j, x3, fx3,
      fabs(fx3));

      if(fx1*fx3 < 0)
      {
            x2 = x3;
            fx2 = fx3;
      }
```

```
            else
            {
                  x1 = x3;
                  fx1 = fx3;
            }
      }while(fabs(fx3) > error_limit);


      printf("\n------------------------------------------------------------------
      ----------\n");
      fprintf(regula_falsi,"\n------------------------------------------------------
      ------------------------");
      printf("\n\nDengan Nilai awal x1 = %.2f dan x2 = %.2f", x10, x20);
      printf("\n\nDiperoleh Akar Persamaan x = %.3f", x3);
      printf("\n\nDengan Error sebesar = %.16f", fabs(fx3));
      fprintf(regula_falsi,"\n\nDengan Nilai awal x1 = %.2f dan x2 = %.2f", x10,
      x20);
      fprintf(regula_falsi,"\n\nDiperoleh Akar Persamaan x = %.3f", x3);
      fprintf(regula_falsi,"\n\nDengan Error sebesar = %.16f", fabs(fx3));

      fclose(regula_falsi);
      cout << "\n\nTekan Enter untuk melanjutkan .........";
      getch();
      clrscr();

      cout << "\n\nCoba lagi dengan x dan y berbeda (y/t) ? ";
      cin >> lagi;

}while(lagi != 't');

return 0;
}
```

```
/* Penyelesaian Soal Regula-Falsi Bab II Hal. 20 No. 3.c. Diktat Metode Komputasi 2004 */
/* Nama Program : regula-falsi_1c.cpp */

#include <iostream.h>
#include <stdlib.h>
#include <conio.h>
#include <stdio.h>
#include <math.h>

#define e 2.718281828
#define error_limit 0.000000000000001 //flexible sesuai kebutuhan

main()
{
     FILE *regula_falsi, *regula_x;
     double    fx1, fx2, fx3,
               x1, x2, x3,
               x10, x20, f1f2,
               x, dx, y1, y2, dy;
     int i, j, k;
     char lagi, belum;

     do
     {
          clrscr();
          j = 0;
          regula_falsi = fopen("regula_falsi_3c.txt", "w+");
          regula_x = fopen("cross_regula_3c.txt", "w+");

          do
          {
               clrscr();
               cout << "Metode Regula-Falsi\n";
               cout << "Soal No. 3.c.\n";
               cout << "==================\n\n";
               //Persamaan utama dan persamaan untuk mencari titik potongnya
               cout << "f(x) = 3*x - cos(x) = 0\n\n";
               cout << "y1 = 3*x dan y2 = cos(x)\n\n";

               cout << "Mencari Titik Potong terdekat\n";
               cout << "-----------------------------\n\n";
               cout << "Berapa Nilai awal x = ";
               cin >> x;
               cout << "\nBerapa Nilai Interval x = ";
               cin >> dx;
               cout << "\nMaksimum Iterasi = ";
               cin >> k;

               printf("\n\t-------------------------------------------------\n");
               printf("\tIterasi ke-\tx\ty1\ty2\tselisih\n");
               printf("\t-------------------------------------------------\n\n");
               fprintf(regula_x, "\n\t-------------------------------------------------
               -\n");
               fprintf(regula_x, "\tIterasi ke-\tx\ty1\ty2\tselisih\n");
               fprintf(regula_x, "\t-------------------------------------------------
               \n\n");

               for (i = 0; i < k; i++)
               {
                    y1 = 3*x;
                    y2 = cos(x);
                    dy = y1 - y2;
```

```
            printf("\t\t%d\t%.1f\t%.2f\t%.2f\t%.2f\n", i+1, x, y1, y2, dy);
            fprintf(regula_x, "\t\t%d\t%.1f\t%.2f\t%.2f\t%.2f\n", i+1, x, y1,
            y2, dy);
            x += dx;
      }

      printf("\n\t-------------------------------------------------\n\n");
      fprintf(regula_x, "\t-------------------------------------------------
      \n\n");
      fclose(regula_x);

      cout << "Apakah Titik Potong telah ditemukan ? ";
      cin >> belum;

}while(belum != 'y');

cout << "\n\nMencari Titik Potong sebenarnya\n";
cout << "===============================\n\n";
cout << "Masukkan Nilai x1 = ";
cin >> x1;

x10 = x1;
fx1 = 3*x1 - cos(x1);

do
{
      cout << "\nMasukkan Nilai x2 = ";
      cin >> x2;
      fx2 = 3*x2 - cos(x2);
      x20 = x2;

      f1f2 = fx1*fx2;

}while(f1f2 >= 0.0);

printf("\n-----------------------------------------------------------------
----------\n");
printf("Iterasi ke-\tx3\tfx3\t\tfx3 (16 digit) Error Aproksimasi\n");
printf("-----------------------------------------------------------------
--------\n\n");
fprintf(regula_falsi, "\n-------------------------------------------------
-------------------------\n");
fprintf(regula_falsi, "Iterasi ke-\tx3\tfx3\t\tfx3 (16 digit) Error
Aproksimasi\n");
fprintf(regula_falsi, "-------------------------------------------------
-----------------------\n\n");

do
{
      j++;
      x3 = (x1*fx2 - x2*fx1)/(fx2 - fx1);
      fx3 = 3*x3 - cos(x3);

      printf("\t%d\t%.3f\t%.6f\t%.16f\n", j, x3, fx3, fx3);
      fprintf(regula_falsi,"\t%d\t%.3f\t%.6f\t%.16f\n", j, x3, fx3, fx3);

      if(fx1*fx3 < 0)
      {
            x2 = x3;
            fx2 = fx3;
      }
```

```
            else
            {
                  x1 = x3;
                  fx1 = fx3;
            }

      }while(fabs(fx3) > error_limit);

      printf("\n------------------------------------------------------------------
      ----------\n");
      fprintf(regula_falsi,"\n------------------------------------------------------
      ------------------------");
      printf("\n\nDengan Nilai awal x1 = %.2f dan x2 = %.2f", x10, x20);
      printf("\n\nDiperoleh Akar Persamaan x = %.3f", x3);
      printf("\n\nDengan Error sebesar = %.16f", fabs(fx3));
      fprintf(regula_falsi,"\n\nDengan Nilai awal x1 = %.2f dan x2 = %.2f", x10,
      x20);
      fprintf(regula_falsi,"\n\nDiperoleh Akar Persamaan x = %.3f", x3);
      fprintf(regula_falsi,"\n\nDengan Error sebesar = %.16f", fabs(fx3));

      fclose(regula_falsi);
      cout << "\n\nTekan Enter untuk melanjutkan .........";
      getch();
      clrscr();

      cout << "\n\nCoba lagi dengan x dan y berbeda (y/t) ? ";
      cin >> lagi;
      clrscr();

}while(lagi != 't');

return 0;
}
```

```
/* Contoh Soal Newton-Rhapson Bab II Hal. 16-17 Diktat Metode Komputasi 2004 */
/* Nama Program : newton-rhapson.cpp */

#include <iostream.h>
#include <stdlib.h>
#include <conio.h>
#include <stdio.h>
#include <math.h>

#define error_limit 1E-13      /* flexible sesuai kebutuhan */

main()
{
     FILE *newton;
     int i, j;
     double    x, x0, xuji, dx, y,
               dif1, dif2, dif,
               syarat;
     char lagi;

     do
     {
          clrscr();
          newton = fopen("newton.txt", "w+");
          cout << "Metode Newton-Rhapson\n";
          cout << "=====================\n\n";
          /* Persamaan dan turunannya */
          cout << "y = 4 + 5*x^2 - x^3\n\n";
          cout << "y' = 10*x - 3*x^2\n\n";
          cout << "y'' = 10 - 6x\n\n";

          /* Mencari Nilai x -> 0 */
          cout << "\n\nMencari Nilai x -> 0";
          cout << "\n\nBerapa Nilai x          ? ";
          cin >> xuji;
          cout << "\nBerapa Nilai Interval x ? ";
          cin >> dx;
          cout << "\nMaksimum Iterasi        ? ";
          cin >> j;

          printf("\n\t-------------------------------------\n");
          printf("\tIterasi ke-\tx\ty\n");
          printf("\t-------------------------------------\n\n");
          fprintf(newton,"\n\t-------------------------------------\n");
          fprintf(newton,"\tIterasi ke-\tx\ty\n");
          fprintf(newton,"\t-------------------------------------\n\n");

          for(i = 1; i <= j; i++)
          {
               y = 4 + 5*pow(xuji,2) - pow(xuji,3);

               printf("\t\t%.d\t%.3f\t%.3f\n", i, xuji, y);
               fprintf(newton,"\t\t%.d\t%.3f\t%.3f\n", i, xuji, y);
               xuji += dx;
          }

          printf("\t-------------------------------------\n\n");
          fprintf(newton,"\t-------------------------------------\n\n");

          do
          {
               /* uji nilai x */
               cout << "\n\nUji Nilai x yang dipilih";
               cout << "\n\nBerapa Nilai x -> 0 ? ";
               cin >> x;
```

```
            y = 4 + 5*pow(x,2) - pow(x,3);
            dif1 = 10*x - 3*pow(x,2);
            dif2 = 10 - 6*x;

            /* Syarat agar x awal terpenuhi */
            syarat = abs((y*dif2)/(dif1*dif1));

        }while(syarat >= 1.00);

        x0 = x;

        printf("\n\t---------------------------------------------------------------\n");
        printf("\tIterasi ke-\tx\ty\tdy\t\ty/dy\n");
        printf("\t---------------------------------------------------------\n\n");
        fprintf(newton, "\n\t-------------------------------------------------------
        ----\n");
        fprintf(newton, "\tIterasi ke-\tx\ty\tdy\t\ty/dy\n");
        fprintf(newton, "\t-------------------------------------------------------
        --\n\n");

        /* Perhitungan Newton-Rhapson */
        i = 1;

        do
        {
            /* rumus x(i+1)=x-(y/y') */
            y = 4 + 5*pow(x,2) - pow(x,3);
            dif1 = (10*x) - 3*pow(x,2);
            dif = y/dif1;
            x = x - dif;
            printf("\t\t%.d\t%.3f\t%.3f\t%.3f\t\t\t%.4f\n", i, x, y, dif2, dif);

            fprintf(newton,"\t\t%.d\t%.3f\t%.3f\t%.3f\t\t\t%.4f\n", i, x, y, dif2,
            dif);
            i++;

        }while(fabs(y) > error_limit);

        printf("\t---------------------------------------------------------\n\n");
        fprintf(newton,"\t-------------------------------------------------------
        -\n\n");
        printf("\nDengan tebakan awal x = %.3f", x0);
        printf("\n\nDiperoleh Akar Persamaan x = %.8f", x);
        printf("\n\nDengan Error sebesar = %.8f", y);
        fprintf(newton,"\nDengan tebakan awal x = %.3f", x0);
        fprintf(newton,"\n\nDiperoleh Akar Persamaan x = %.8f", x);
        fprintf(newton,"\n\nDengan Error sebesar = %.8f", y);

        fclose(newton);

        cout << "\n\nCoba dengan x awal yang berbeda (y/t) ? ";
        cin >> lagi;

    }while(lagi != 't');

    return 0;
}
```

```
/* Penyelesaian Soal Newton-Rhapson Bab II Hal. 20 No. 4.a. Diktat Metode Komputasi
2004 */
/* Nama Program : newton-rhapson_4a.cpp */

#include <iostream.h>
#include <stdlib.h>
#include <conio.h>
#include <stdio.h>
#include <math.h>

main()
{
     FILE *newton;
     int i, j;
     double    x0, x, dx, y,
               dif1, dif2, dif,
               syarat;
     char ya, lagi;

     do
     {
          newton = fopen("newton_4a.txt", "w+");

          do
          {
               clrscr();
               cout << "Metode Newton-Rhapson\n";
               cout << "Jawaban Soal No. 4.a.\n";
               cout << "====================\n\n";
               //Persamaan dan turunannya
               cout << "y = 3*x - cos(x)\n\n";
               cout << "y' = 3 + sin(x)\n\n";
               cout << "y'' = cos(x)\n\n";

               //Mencari Nilai x -> 0
               cout << "Berapa Nilai x awal = ";
               cin >> x;

               y = 3*x - cos(x);
               dif1 = 3 + sin(x);
               dif2 = cos(x);

               //Syarat agar x awal terpenuhi

               syarat = abs((y*dif2)/(dif1*dif1));

               cout << "\nNilai " << x << " menghasilkan " << syarat;
               cout << "\n\nApakah x memenuhi syarat ? ";
               cin >> ya;

          }while(ya != 'y');

          x0 = x;

          cout << "\nBerapa Nilai Interval x = ";
          cin >> dx;
          cout << "\nMaksimum Iterasi = ";
          cin >> j;

          printf("\n\t-------------------------------------\n");
          printf("\tIterasi ke-\tx\ty\n");
          printf("\t-------------------------------------\n\n");
          fprintf(newton, "\n\t-------------------------------------\n");
          fprintf(newton, "\tIterasi ke-\tx\ty\n");
          fprintf(newton, "\t-------------------------------------\n\n");
```

```
        for(i = 1; i <= j; i++)
        {
             y = 3*x - cos(x);

             printf("\t\t%.d\t%.3f\t%.3f\n", i, x, y);
             fprintf(newton,"\t\t%.d\t%.3f\t%.3f\n", i, x, y);

             x += dx;
        }

        printf("\t-------------------------------------\n\n");
        fprintf(newton,"\t-------------------------------------\n\n");

        //Perhitungan Newton-Rhapson
        cout << "Perhatikan Nilai x yang menghasilkan fungsi y paling mendekati
        0\n";
        cout << "Pilih Nilai x tersebut untuk perhitungan berikutnya\n\n";
        cout << "\nBerapa Nilai x (y -> 0) = ";
        cin >> x;
        cout << "\nMaksimum Iterasi = ";
        cin >> j;

        printf("\n\t-----------------------------------------------------------\n");
        printf("\tIterasi ke-\tx\ty\tdy\t\ty/dy\n");
        printf("\t-----------------------------------------------------------\n\n");
        fprintf(newton, "\n\t-----------------------------------------------------
        ---\n");
        fprintf(newton, "\tIterasi ke-\tx\ty\tdy\t\ty/dy\n");
        fprintf(newton, "\t-----------------------------------------------------
        --\n\n");

        for(i = 1; i <= j; i++)
        {
             //rumus x(i+1)=x-(y/y')
             y = 3*x - cos(x);
             dif1 = 3 + sin(x);
             dif = y/dif1;
             x = x - dif;

             printf("\t\t%.d\t%.3f\t%.3f\t%.3f\t\t\t%.4f\n", i, x, y, dif1, dif);
             fprintf(newton,"\t\t%.d\t%.3f\t%.3f\t%.3f\t\t\t%.4f\n", i, x, y, dif1,
             dif);
        }

        printf("\t-----------------------------------------------------------\n\n");
        fprintf(newton,"\t-----------------------------------------------------
        -\n\n");
        printf("\nDengan tebakan awal x = %.3f", x0);
        printf("\n\nDiperoleh Akar Persamaan x = %.3f", x);
        printf("\n\nDengan Error sebesar = %.8f", y);
        fprintf(newton,"\nDengan tebakan awal x = %.3f", x0);
        fprintf(newton,"\n\nDiperoleh Akar Persamaan x = %.3f", x);
        fprintf(newton,"\n\nDengan Error sebesar = %.8f", y);

        fclose(newton);

        cout << "\n\nCoba dengan x awal yang berbeda (y/t) ? ";
        cin >> lagi;

    }while(lagi != 't');

    return 0;
}
```

```
/* Contoh Soal Metode Iterasi Bentuk x = g(x) Bab II Hal. 18-19 Diktat Metode
Komputasi 2004 */
/* Nama Program : x_gx.cpp */

#include <iostream.h>
#include <stdlib.h>
#include <conio.h>
#include <stdio.h>
#include <math.h>

#define error_limit 0.00000001

main()
{
     FILE *x_gx;
     int i;
     double fx, g1x, x, xi, x0;
     char lagi;

     do
     {
          clrscr();
          x_gx = fopen("x_gx.txt", "w+");

          do
          {
               clrscr();
               cout << "Metode Iterasi x = g(x)\n";
               cout << "======================\n\n";

               //Persamaan, konversinya dan turunannya
               cout << "f(x) = x^3 - 9*x^2 + 18x - 6 = 0\n\n";
               cout << "g(x) = -(x^3/18) + x^2/2 + 1/3\n\n";
               cout << "g'(x) = -(x^2/6) + x\n\n";

               cout << "Bila berulang berarti Nilai g'(x) > 1\n\n";
               cout << "Masukkan Nilai x Asumsi = ";
               cin >> x;
               g1x = (-pow(x,2)/6)+ x;
               printf("\nx = %.3f\tg'(x) = %.6f\n\n", x, abs(g1x));

               cout << "\nTekan Enter .........";
               getch();

          } while(abs(g1x) >= 1);

          x0 = x;
          i = 0;

          printf("\n\t-------------------------------------------\n");
          printf("\tIterasi ke-\tx = g(x)\tf(x)\n");
          printf("\t-------------------------------------------\n\n");
          fprintf(x_gx, "\n\t-------------------------------------------\n");
          fprintf(x_gx, "\tIterasi ke-\tx = g(x)\tf(x)\n");
          fprintf(x_gx, "\t-------------------------------------------\n\n");

          do
          {
               /* rumus xi = g(x) = -(x^3)/18 + x^2/2 +(1/3) */
               xi = -pow(x,3)/18 + pow(x,2)/2 + 0.3333333333;
               fx = pow(xi,3) - 9*pow(xi,2) + 18*xi - 6;

               printf("\t\t%d\t%.8f\t%.8f\n", i+1, xi, fx);
               fprintf(x_gx,"\t\t%d\t%.8f\t%.8f\n", i+1, xi, fx);
```

```
            x = xi;
            i++;

        }while(fabs(fx) > error_limit);

        printf("\n\t-----------------------------------------\n\n");
        fprintf(x_gx, "\n\t-----------------------------------------\n\n");
        printf("\nDengan tebakan awal x = %.3f", x0);
        printf(" diperoleh\n");
        printf("\nAproksimasi Akar Persamaan adalah x = %.10f\n", x);
        printf("\nError Hasil Aproksimasi adalah f(x) = %.14f\n", fabs(fx));
        fprintf(x_gx,"\nDengan tebakan awal x = %.3f", x0);
        fprintf(x_gx," diperoleh\n");
        fprintf(x_gx,"\nAproksimasi Akar Persamaan adalah x = %.10f\n", x);
        fprintf(x_gx,"\nError Hasil Aproksimasi adalah f(x) = %.10f\n", fabs(fx));

        fclose(x_gx);

        cout << "\n\nCoba lagi dengan x awal yang berbeda (y/t) ? ";
        cin >> lagi;

    }while(lagi != 't');

    return 0;
}
```

```
/* Penyelesaian Soal Metode Iterasi Bentuk x = g(x)Bab II Hal. 20 No. 5.d. Diktat
Metode Komputasi 2004 */
/* Nama Program : x_gx_5d.cpp */

#include <iostream.h>
#include <stdlib.h>
#include <conio.h>
#include <stdio.h>
#include <math.h>

#define e 2.718281828

main()
{
     FILE *x_gx;
     int i, j;
     double fx, g1x, x, xi, x0;
     char lagi;

     do
     {
          clrscr();
          x_gx = fopen("x_gx_5d.txt", "w+");

          do
          {
               clrscr();
               cout << "Metode Iterasi x = g(x)\n";
               cout << "Jawaban Soal No. 5.d.  \n";
               cout << "======================\n\n";

               //Persamaan, konversinya dan turunannya
               cout << "f(x) = e^x - 3*x = 0\n\n";
               cout << "g(x) = e^x/3\n\n";
               cout << "g'(x) = e^x/3\n\n";

               cout << "Bila berulang berarti Nilai g'(x) > 1\n\n";
               cout << "Masukkan Nilai x Asumsi = ";
               cin >> x;
               g1x = pow(e,x)/3;
               printf("\nx = %.3f\tg'(x) = %.6f\n\n", x, fabs(g1x));
               cout << "\nTekan Enter .........";
               getch();

          } while(abs(g1x) >= 1);

          x0 = x;

          cout << "\n\nBerapa Kali Iterasi (min. 20) = ";
          cin >> j;

          clrscr();
          printf("\n\t-------------------------------------------\n");
          printf("\tIterasi ke-\tx = g(x)\tf(x)\n");
          printf("\t-------------------------------------------\n\n");
          fprintf(x_gx, "\n\t-------------------------------------------\n");
          fprintf(x_gx, "\tIterasi ke-\tx = g(x)\tf(x)\n");
          fprintf(x_gx, "\t-------------------------------------------\n\n");
```

```
        for(i = 0; i < j; i++)
        {
            //rumus xi = g(x) = e^x/3
            xi = pow(e,x)/3;
            fx = pow(e,x) - 3*x;

            printf("\t\t%d\t%.6f\t%.8f\n", i+1, xi, fx);
            fprintf(x_gx,"\t\t%d\t%.6f\t%.8f\n", i+1, xi, fx);

            x = xi;
        }

        printf("\n\t-----------------------------------------\n\n");
        fprintf(x_gx, "\n\t-----------------------------------------\n\n");
        printf("\nDengan tebakan awal x = %.3f", x0);
        printf(" diperoleh\n");
        printf("\nAproksimasi Akar Persamaan adalah x = %.6f\n", x);
        printf("\nError Hasil Aproksimasi adalah f(x) = %.8f\n", fabs(fx));
        fprintf(x_gx,"\nDengan tebakan awal x = %.3f", x0);
        fprintf(x_gx," diperoleh\n");
        fprintf(x_gx,"\nAproksimasi Akar Persamaan adalah x = %.6f\n", x);
        fprintf(x_gx,"\nError Hasil Aproksimasi adalah f(x) = %.8f\n", fabs(fx));

        fclose(x_gx);

        cout << "\n\nCoba lagi dengan x awal yang berbeda ? ";
        cin >> lagi;

    }while(lagi != 'n');

    return 0;
}
```

METODE FAKTORISASI PERSAMAAN POLINOMIAL

Listing Program C++

```
/* Contoh soal Faktorisasi P3(x)=P(1,2) Bab II Hal. 23 Diktat Metode Komputasi 2004 */
/* Nama Program : faktorp3.cpp */

#include <iostream.h>
#include <stdlib.h>
#include <conio.h>
#include <stdio.h>
#include <math.h>

#include "akarp345.hpp"

main()
{
     FILE *stream;
     int j, i;
     double A0, A1, A2;
     double b01, b0, a1, a0, tempb0;
     char jawab, ganda, filedata[20];

     do
     {
          clrscr();

          i = 1;
          b0 = 0.0;

          cout << "\nMetode Faktorisasi P3(x) = P(1,2)\n";
          cout << "================================\n\n";
          cout << "\nMasukkan Nama File Penyimpan Data ? ";
          cin >> filedata;
          stream = fopen(filedata, "w+");
          cout << "\nMasukkan Nilai A2 = ";
          cin >> A2;
          cout << "\nMasukkan Nilai A1 = ";
          cin >> A1;
          cout << "\nMasukkan Nilai A0 = ";
          cin >> A0;
          cout <<"\nBerapa kali Iterasi = ";
          cin >> j;

          printf("\n\n\t--------------------------------\n");
          printf("\titerasi\t  b0\t  a1\t  a0\n");
          printf("\t--------------------------------\n");
          fprintf(stream,"\n\n\t--------------------------------\n");
          fprintf(stream,"\titerasi\t  b0\t  a1\t  a0\n");
          fprintf(stream,"\t--------------------------------\n");
```

```
do
{
     if(b0 != 0)
     {
          a1 = A2 - b0;
          a0 = A1 - (a1*b0);
          b01 = A0/a0;

          if (i != j)
          {
               printf("\t   %d\t%.3f\t%.3f\t%.3f\n", i, b01, a1, a0);
               fprintf(stream,"\t   %d\t%.3f\t%.3f\t%.3f\t\n", i, b01, a1,
               a0);
          }
          else
          {
               break;
          }
     }
     else
     {
          a1 = A2 - b0;
          a0 = A1 - (a1*b0);

          printf("\t   %d\t%.3f\t%.3f\t%.3f\n", i, b0, a1, a0);
          fprintf(stream,"\t   %d\t%.3f\t%.3f\t%.3f\t\n", i, b0, a1, a0);

          if (a0 != 0)
          {
               b01 = A0/a0;
          }
          else
          {
               break;
          }
     }
     //swap nilai b0
     tempb0 = b0;
     b0 = b01;
     b01 = tempb0;

     i++;

}while( i <= j);

printf("\t---------------------------------\n");
fprintf(stream,"\t---------------------------------\n");

if (a0 != 0)
{
     do
     {
          akarp345();
          cout << "\n\nParameter ganda (y/t)? ";
          cin >> ganda;
     } while(ganda != 't');
}
else
{
     printf("\n\na0 = %.2f not valid (error floating point)\n\n", a0);
}

cout << "\nCoba lagi atau persamaan lainnya (y/t) ? ";
cin >> jawab;
```

```
    } while(jawab != 't');

    fclose(stream);
    return 0;
}
```

```
/* Contoh soal Faktorisasi P4(x)=P(2,2) Bab II Hal. 23 Diktat Metode Komputasi 2004 */
/* Nama Program : faktorp4.cpp */

#include <iostream.h>
#include <stdlib.h>
#include <conio.h>
#include <stdio.h>
#include <math.h>

#include "akarp345.hpp"

main()
{
     FILE *stream;
     int j, i;
     double A3, A2, A1, A0;
     double b1, b0, a1, a0;
     double b11, b01, temp1, temp0;
     char lagi, filedata[20];

     do
     {
          clrscr();

          i = 1;
          b1 = 0.0;
          b0 = 0.0;
          a0 = 0.0;

          cout << "\nMetode Faktorisasi P4 = P(2,2)\n";
          cout << "==============================\n\n";
          cout << "\nMasukkan Nama File Penyimpan Data ? ";
          cin >> filedata;
          stream = fopen(filedata, "w+");
          cout << "\nBerapa Nilai A3 = ";
          cin >> A3;
          cout << "\nBerapa Nilai A2 = ";
          cin >> A2;
          cout << "\nBerapa Nilai A1 = ";
          cin >> A1;
          cout << "\nBerapa Nilai A0 = ";
          cin >> A0;
          cout << "\nBerapa kali Iterasi = ";
          cin >> j;

          printf("\n\n\t---------------------------------------\n");
          printf("\titerasi\t  b0\t  b1\t  a1\t  a0\n");
          printf("\t---------------------------------------\n");
          fprintf(stream,"\n\n\t---------------------------------------\n");
          fprintf(stream,"\titerasi\t  b0\t  b1\t  a1\t  a0\n");
          fprintf(stream,"\t---------------------------------------\n");

          do
          {
               if(b0 != 0 && b1 != 0)
               {
                    //rumus b1 = (A1-a1*b0)/a0, b0=A0/a0; a1=A3-b1, a0=A2-b0-a1*b1
                    b11 = (A1 - (a1*b0))/a0;
                    b01 = A0/a0;
                    a1 = A3 - b11;
                    a0 = A2 - b0 - (a1*b11);
```

```
                    if (i != j)
                    {
                         printf("\t   %d\t%.3f\t%.3f\t%.3f\t%.3f\n", i, b01, b11, a1,
                         a0);
                         fprintf(stream,"\t   %d\t%.3f\t%.3f\t%.3f\t%.3f\n", i, b01,
                         b11, a1, a0);
                    }
                    else
                    {
                         break;
                    }
               }
               else
               {
                    //rumus b1 = (A1-a1*b0)/a0, b0=A0/a0; a1=A3-b1, a0=A2-b0-a1*b1
                    a1 = A3 - b1;
                    a0 = A2 - b0 - (a1*b1);

                    printf("\t   %d\t%.3f\t%.3f\t%.3f\t%.3f\n", i, b0, b1, a1, a0);
                    fprintf(stream,"\t   %d\t%.3f\t%.3f\t%.3f\t%.3f\n", i, b0, b1, a1,
                    a0);

                    if (a0 != 0) b01 = A0/a0;
                    else
                    {
                         printf("\n\na0 = %.2f not valid (error floating point)\n\n",
                         a0);
                         break;
                    }
               }

               temp1 = b1; temp0 = b0;
               b1 = b11; b0 = b01;
               b11 = temp1; b01 = temp0;

               i++;

          } while (i <= j);

          printf("\t----------------------------------------\n");
          fprintf(stream,"\t----------------------------------------\n");

          fclose(stream);

          if (a0 != 0) akarp345();
          else
          {
               printf("\n\na0 = %.2f not valid (error floating point)\n\n", a0);
          }

          cout << "\n\nTekan Enter ........ untuk lanjut .. ";
          getch();
          clrscr();
          cout << "\n\nCoba lagi (y/t) ? ";
          cin >> lagi;

     }while(lagi != 't');

     return 0;
}
```

```
/* Contoh soal Faktorisasi P5(x)=P(1,2,2) Bab II Hal. 23 Diktat Metode Komputasi 2004 */
/* Nama Program : faktorp5.cpp */

#include <iostream.h>
#include <stdlib.h>
#include <conio.h>
#include <stdio.h>
#include <math.h>

#include "akarp345.hpp"

main()
{
     FILE *stream;
     int j, i;
     double A4, A3, A2, A1, A0;
     double c1, c0, b1, b0, a0;
     double b11, b01, a01;
     double tempa0, tempb1, tempb0;
     char lagi, filedata[20];

     do
     {
          clrscr();

          i = 1;
          b1 = 0.0;
          b0 = 0.0;
          a0 = 0.0;
          c0 = 0.0;
          c1 = 0.0;

          cout << "\nMetode Faktorisasi P5 = P(1,2,2)\n";
          cout << "================================\n\n";
          cout << "\nMasukkan Nama File Penyimpan Data ? ";
          cin >> filedata;
          stream = fopen(filedata, "w+");
          cout << "\nBerapa Nilai A4 = ";
          cin >> A4;
          cout << "\nBerapa Nilai A3 = ";
          cin >> A3;
          cout << "\nBerapa Nilai A2 = ";
          cin >> A2;
          cout << "\nBerapa Nilai A1 = ";
          cin >> A1;
          cout << "\nBerapa Nilai A0 = ";
          cin >> A0;
          cout << "\nBerapa kali Iterasi = ";
          cin >> j;

          printf("\n\n\t-------------------------------------------------\n");
          printf("\titerasi\t  b0\t  b1\t  a0\t  c1\t  c0\n");
          printf("\t-------------------------------------------------\n");
          fprintf(stream,"\n\n\t-------------------------------------------------\n");
          fprintf(stream,"\titerasi\t  b0\t  b1\t  a0\t  c1\t  c0\n");
          fprintf(stream,"\t-------------------------------------------------\n");

          do
          {
               if(b0 != 0 && b1 != 0 && a0 != 0)
               {
                    b11 = (A2 - (a0*A3) + (pow(a0,2)*A4) - pow(a0,3) - (c1*b0))/c0;
                    b01 = (A1 - (a0*A2) + (pow(a0,2)*A3) - (pow(a0,3)*A4) +
                    pow(a0,4))/c0;
                    a01 = A0 /(b01*c0);
                    c1 = A4 - a01 - b11;
```

```
            c0 = A3 - (a01*A4) + pow(a01,2) - b01 - (c1*b11);

            if (i != j)
            {
                 printf("\t   %d\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\n", i, b01,
                 b11, a01, c1, c0);
                 fprintf(stream,"\t   %d\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\n", i,
                 b01, b11, a01, c1, c0);
            }
            else
                 break;
       }
       else
       {
            c1 = A4 - a0 - b1;
            c0 = A3 - (a0*A4) + pow(a0,2) - b0 - (c1*b1);

            printf("\t   %d\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\n", i, b0, b1, a0,
            c1, c0);
            fprintf(stream,"\t   %d\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\n", i, b0,
            b1, a0, c1, c0);

            if (c0 != 0)
            {
                 b11 = (A2 - (a0*A3) + (pow(a0,2)*A4) - pow(a0,3) -
                 (c1*b0))/c0;
                 b01 = (A1 - (a0*A2) + (pow(a0,2)*A3) - (pow(a0,3)*A4) +
                 pow(a0,4))/c0;
            }
            else
            {
                 printf("\n\na0 = %.2f not valid (error floating point)\n\n",
                 c0);
                 getch();
                 break;
            }
       }

       tempa0 = a0;
       a0 = a01;
       a01 = tempa0;

       tempb1 = b1;
       b1 = b11;
       b11 = tempb1;

       tempb0 = b0;
       b0 = b01;
       b01 = tempb0;

       i++;

}while(i <= j);

printf("\t-------------------------------------------------\n");
fprintf(stream,"\t-------------------------------------------------\n");

fclose(stream);

if (c0 != 0)
{
     akarp345();
}
```

```
            else
            {
                  printf("\n\nc0 = %.2f not valid (error floating point)", c0);
            }

            cout << "\n\nTekan Enter ........ untuk lanjut .. ";
            getch();
            clrscr();
            cout << "\n\nCoba lagi (y/t) ? ";
            cin >> lagi;

      }while(lagi != 't');

      return 0;
}
```

```
/* Contoh soal Faktorisasi Bairstow Bab II Hal. 23 Diktat Metode Komputasi 2004 */
/* Nama Program : bairstow.cpp */

#include <iostream.h>
#include <stdlib.h>
#include <conio.h>
#include <stdio.h>

#include "discriminant.hpp"

#define MaksPoly 20

main()
{
     FILE *bairstow;
     double    a[MaksPoly],
               b[MaksPoly],
               c[MaksPoly];
     double    r, s,
               R1, R2,
               S1, S2,
               dR, dS;
     int n, i, j, k;
     char lagi;

     do
     {
          clrscr();
          cout << "\nMetode Faktorisasi Bairstow\n";
          cout << "===========================\n\n";

          bairstow = fopen("bairstow.txt", "w+"); //sesuaikan dengan no. soal

          cout << "\nMasukkan Pangkat Polinomial terbesar + 1 ? ";
          cin >> n;
          cout << "\nMasukkan Nilai r yang telah ditentukan (-1) ? ";
          cin >> r;
          cout << "\nMasukkan Nilai s yang telah ditentukan (-1) ? ";
          cin >> s;
          cout << "\nBerapa kali Iterasi ? ";
          cin >> k;

          //Input Koefisien a1, a2, a3, a4, a5, ........
          cout << "\n\nMasukkan Koefisien a1, a2, a3, a4, a5, ........";
          cout << "\n-----------------------------------------------\n";

          for (i = 1; i <= n; i++)
          {
               printf("\na%d = ", i);
               cin >> a[i];
          }
          j = 0;

          printf("\n\t---------------------------------------\n");
          printf("\tb1\tb2\tb3\tb4\tb5\n");
          printf("\t---------------------------------------\n");
          printf("\tc1\tc2\tc3\tc4\tc5\n");
          printf("\t---------------------------------------\n");
          printf("\ti\tdR\tdS\tr\ts\n");
          printf("\t---------------------------------------\n\n");
          fprintf(bairstow,"\n\t---------------------------------------\n");
          fprintf(bairstow,"\ti\tdR\tdS\tr\ts\n");
          fprintf(bairstow,"\t---------------------------------------\n\n");
```

```
        do
        {
             j++;
             //Hitung Koefisien b1, b2, b3, b4, b5, ........ , bn
             for (i = 1; i <= n; i++)
             {
                  b[i] = a[i] + r*b[i-1] + s*b[i-2];
                  printf("\t%.2f", b[i]);
             }
             //Hitung Koefisien c1, c2, c3, c4, c5, ........ , cn
             printf("\n");
             for (i = 1; i <= n; i++)
             {
                  c[i] = b[i] + r*c[i-1] + s*c[i-2];
                  printf("\t%.2f", c[i]);
             }

             R1 = (-b[n-1]*c[n-2])-(-b[n]*c[n-3]);
             R2 = (c[n-2]*c[n-2])-(c[n-1]*c[n-3]);
             dR = R1/R2;
             r += dR;

             S1 = (c[n-2]*-b[n])-(c[n-1]*-b[n-1]);
             S2 = (c[n-2]*c[n-2])-(c[n-1]*c[n-3]);
             dS = S1/S2;
             s += dS;

             printf("\n\t%d\t%.3f\t%.3f\t%.3f\t%.3f\n\n", j, dR, dS, r, s);
             fprintf(bairstow,"\t %d\t%.3f\t%.3f\t%.3f\t%.3f\n", j, dR, dS, r, s);

        } while(j != k);

        printf("\n\t----------------------------------------\n\n");
        fprintf(bairstow,"\n\t----------------------------------------\n\n");

        for(i = 1; i <= n; i++)
        {
             printf("\n\tb%d = %.3f", i, b[i]);
        }
        printf("\n\n\tdR = %.8f", fabs(dR));
        printf("\tr = %.3f", r);
        printf("\n\tdS = %.8f", fabs(dS));
        printf("\ts = %.3f", s);

        for(i = 1; i <= n; i++)
        {
             fprintf(bairstow,"\n\tb%d = %.3f", i, b[i]);
        }
        fprintf(bairstow,"\n\n\tdR = %.8f", fabs(dR));
        fprintf(bairstow,"\tr = %.3f", r);
        fprintf(bairstow,"\n\tdS = %.8f", fabs(dS));
        fprintf(bairstow,"\ts = %.3f", s);

        fclose(bairstow);

        discbairstow();

        cout << "\n\nCoba lagi (y/t) ? ";
        cin >> lagi;

    } while(lagi != 't');

    return 0;
}
```

```
/* Header untuk program Faktorisasi Polinomial Bairstow */
/* Nama Program : bairstow.hpp */

#include <iostream.h>
#include <stdlib.h>
#include <conio.h>
#include <stdio.h>
#include <math.h>
#include <complex>

void discbairstow();

using namespace std;

//Diskriminan utk Metode Bairstow
void discbairstow()
{
     FILE *stream;
     complex<double> b1, b2, b3, r, s;
     complex<double> D1, x1, x2, x3, x4, D2;
     char akarbairstow[20];

     cout << "\n\n\nMencari Akar-akar Persamaan Polinomial" << endl;
     cout << "--------------------------------------" << endl;
     cout << "\n\nMasukkan Nama File Penyimpan Data ? ";
     cin >> akarbairstow;
     stream = fopen(akarbairstow, "w+");
     cout << "\n\nMasukkan Parameter-parameter r, s, b1, b2, b3\n" << endl;
     cout << "\nr = ";
     cin >> r;
     cout << "\ns = ";
     cin >> s;
     cout << "\nb1 = ";
     cin >> b1;
     cout << "\nb2 = ";
     cin >> b2;
     cout << "\nb3 = ";
     cin >> b3;

     D1 = sqrt(pow(r,2) + 4.0*s);
     D2 = sqrt(pow(b2,2) - 4.0*b1*b3);
     x1 = (r+D1) / 2.0;
     x2 = (r-D1) / 2.0;
     x3 = (-b2+D2) / 2.0;
     x4 = (-b2-D2) / 2.0;

     printf("\n\nD1 = %.3lfi %.8lfj", real(D1), imag(D1));
     printf("\n\nx1 = %.3lfi %.8lfj", real(x1), imag(x1));
     printf("\tx2 = %.3lfi %.8lfj", real(x2), imag(x2));

     printf("\n\nD2 = %.3lfi %.8lfj", real(D2), imag(D2));
     printf("\n\nx3 = %.3lfi %.8lfj", real(x3), imag(x3));
     printf("\tx4 = %.3lfi %.8lfj", real(x4), imag(x4));

     fprintf(stream, "x1 = %.3lfi %.8lfj\tx2 = %.3lfi %.8lfj\n"
     "x3 = %.3lfi %.8lfj\tx4 = %.3lfi %.8lfj\n", real(x1), imag(x1), real(x2), imag(x2),
     real(x3), imag(x3), real(x4), imag(x4));

     fclose(stream);
}
```

```
/* Header untuk program Faktorisasi Polinomial P3, P4 dan P5 */
/* Nama Program : akar345.hpp */

#include <iostream.h>
#include <stdlib.h>
#include <conio.h>
#include <stdio.h>
#include <math.h>
#include <complex>

void akarp345();

using namespace std;

void akarp345()
{
     FILE *akarp3, *akarp4, *akarp5;
     complex<double> a1, a0, b1, b0, c1, c0;
     complex<double> D1, x1, x2, x3, x4, D2, x5;
     int i;
     char fakarp3[20], fakarp4[20], fakarp5[20];

     cout << "\n\n\nMencari Akar-akar Persamaan Polinomial" << endl;
     cout << "--------------------------------------" << endl;
     cout << "\nPilih (1) P3 = P(1,2); (2) P4 = P(2,2); (3) P5 = P(1,2,2)" << endl;
     cout << "\nPilihan : ";
     cin >> i;

     switch (i)
     {
          case 1    :
          {
               cout << "\n\nMasukkan Nama File Penyimpan Data ? ";
               cin >> fakarp3;
               akarp3 = fopen(fakarp3, "w+");
               cout << "\n\nMasukkan Parameter-parameter b0, a1, a0\n" << endl;
               cout << "\nb0 = ";
               cin >> b0;
               cout << "\na1 = ";
               cin >> a1;
               cout << "\na0 = ";
               cin >> a0;

               D1 = sqrt(pow(a1,2)- 4.0*a0);
               x1 = -b0;
               x2 = (-a1+D1) / 2.0;
               x3 = (-a1-D1) / 2.0;

               cout << "\n\nD = " << D1 << endl;
               cout << "\nx1 = " << x1;
               cout << "\tx2 = " << x2;
               cout << "\tx3 = " << x3;

               fprintf(akarp3,"\n\nD1 = %.3lfi %.8lfj", real(D1), imag(D1));
               fprintf(akarp3,"\n\nx1 = %.3lfi %.8lfj", real(x1), imag(x1));
               fprintf(akarp3,"\tx2 = %.3lfi %.8lfj", real(x2), imag(x2));
               fprintf(akarp3,"\tx3 = %.3lfi %.8lfj", real(x3), imag(x3));
               fclose(akarp3);
               break;
          }
```

```
case 2    :
     {
          cout << "\n\nMasukkan Nama File Penyimpan Data ? ";
          cin >> fakarp4;
          akarp4 = fopen(fakarp4, "w+");
          cout << "\n\nMasukkan Parameter-parameter b0, b1, a1, a0\n" << endl;
          cout << "\nb0 = ";
          cin >> b0;
          cout << "\nb1 = ";
          cin >> b1;
          cout << "\na1 = ";
          cin >> a1;
          cout << "\na0 = ";
          cin >> a0;

          D1 = sqrt(pow(b1,2)- 4.0*b0);
          D2 = sqrt(pow(a1,2)- 4.0*a0);
          x1 = (-b1+D1) / 2.0;
          x2 = (-b1-D1) / 2.0;
          x3 = (-a1+D2) / 2.0;
          x4 = (-a1-D2) / 2.0;

          printf("\n\nD1 = %.3lfi %.8lfj", real(D1), imag(D1));
          printf("\n\nx1 = %.3lfi %.8lfj", real(x1), imag(x1));
          printf("\tx2 = %.3lfi %.8lfj", real(x2), imag(x2));
          printf("\n\nD2 = %.3lfi %.8lfj", real(D2), imag(D2));
          printf("\n\nx3 = %.3lfi %.8lfj", real(x3), imag(x3));
          printf("\tx4 = %.3lfi %.8lfj", real(x4), imag(x4));

          fprintf(akarp4,"\n\nD1 = %.3lfi %.8lfj", real(D1), imag(D1));
          fprintf(akarp4,"\n\nx1 = %.3lfi %.8lfj", real(x1), imag(x1));
          fprintf(akarp4,"\tx2 = %.3lfi %.8lfj", real(x2), imag(x2));
          fprintf(akarp4,"\n\nD2 = %.3lfi %.8lfj", real(D2), imag(D2));
          fprintf(akarp4,"\n\nx3 = %.3lfi %.8lfj", real(x3), imag(x3));
          fprintf(akarp4,"\tx4 = %.3lfi %.8lfj", real(x4), imag(x4));
          fclose(akarp4);
          break;
     }

case 3    :
     {
          cout << "\n\nMasukkan Nama File Penyimpan Data ? ";
          cin >> fakarp5;
          akarp5 = fopen(fakarp5, "w+");
          cout << "\n\nMasukkan Parameter-parameter b0, b1, a0, c1, c0\n" << endl;
          cout << "\nb0 = ";
          cin >> b0;
          cout << "\nb1 = ";
          cin >> b1;
          cout << "\na0 = ";
          cin >> a0;
          cout << "\nc1 = ";
          cin >> c1;
          cout << "\nc0 = ";
          cin >> c0;

          x1 = -a0;
          D1 = sqrt(pow(b1,2)- 4.0*b0);
          D2 = sqrt(pow(c1,2)- 4.0*c0);
          x2 = (-b1+D1) / 2.0;
          x3 = (-b1-D1) / 2.0;
          x4 = (-c1+D2) / 2.0;
          x5 = (-c1-D2) / 2.0;
```

```
                cout << "\nx1 = " << x1 << endl;
                cout << "\n\nD1 = " << D1 << endl;
                cout << "\nx2 = " << x2;
                cout << "\tx3 = " << x3 << endl;
                cout << "\nD2 = " << D2 << endl;
                cout << "\nx4 = " << x4;
                cout << "\tx5 = " << x5 << endl;

                fprintf(akarp5,"\nx1 = %.3lfi %.8lfj", real(x1), imag(x1));
                fprintf(akarp5,"\n\nD1 = %.3lfi %.8lfj", real(D1), imag(D1));
                fprintf(akarp5,"\nx2 = %.3lfi %.8lfj", real(x2), imag(x2));
                fprintf(akarp5,"\tx3 = %.3lfi %.8lfj", real(x3), imag(x3));
                fprintf(akarp5,"\n\nD2 = %.3lfi %.8lfj", real(D2), imag(D2));
                fprintf(akarp5,"\nx4 = %.3lfi %.8lfj", real(x4), imag(x4));
                fprintf(akarp5,"\tx5 = %.3lfi %.8lfj", real(x5), imag(x5));
                fclose(akarp5);
                break;
            }
            default : cout << "\nNo good ..... " << endl;
        }
}
```

PENYELESAIAN PERSAMAAN LINIER SERENTAK

Listing Program C++

```
/* Contoh soal Penyelesaian Persamaan Linier Serentak (PLS) */
/* menggunakan Metode Invers dan Determinan Matriks Bab IV  */
/* Hal. 35-36 Diktat Metode Komputasi 2004                  */
/* Nama Program : det_invers.cpp */

#include <iostream.h>
#include <stdlib.h>
#include <conio.h>
#include <stdio.h>

#define MaksData 3

main()
{
     FILE *det_invers;
     double    x[MaksData][MaksData],
               A[MaksData][MaksData],
               H[MaksData][MaksData],
               adjA[MaksData][MaksData],
               inversA[MaksData][MaksData],
               Atrans[MaksData][MaksData],
               cramer1, cramer2, cramer3,
               detA,
               adjA1, adjA2, adjA3, adjA4, adjA5,
               adjA6, adjA7, adjA8, adjA9;
     int  m, n,
          p, q,
          i, j, k;
     char lagi;

     do
     {
          clrscr();
          det_invers = fopen("det_invers.txt", "w+");

          cout << "\n\nMetode Invers-Determinan Matriks untuk PLS\n";
          cout << "==========================================\n\n";

          /*************************************\
          \*         A l g o r i t m a         */
          /*         -----------------         *\
          \*                                   */
          /*   1.   Masukkan Matriks A         *\
          \*   2.   Masukkan Matriks H         */
          /*   3.   Cari Determinan Matriks A  *\
          \*   4.   Cari Adjoint Matriks A     */
          /*   5.   Cari Invers Matriks A      *\
          \*   6.   Cari Transpose Matriks A   */
          /*   7.   Hitung Solusi Matriks x    *\
          \*                                   */
          /*************************************/
```

```
/* Pemasukan Data Matriks A  (umumnya (3 x 3) */
cout << "\nMasukkan Nilai Matriks A (hanya (3 x 3))\n";
cout << "---------------------------------------\n";
cout << "\nMasukkan Jumlah Baris ? ";
cin >> p;
cout << "\nMasukkan Jumlah Kolom ? ";
cin >> q;

/*******************************************************/
/* Matriks A (3 x 3)                                   */
/*******************************************************/

printf("\n");
for (i = 0; i < p; i++)
{
     for (j = 0; j < q; j++)
     {
          printf("A[%d][%d] = ", i+1, j+1);
          cin >> A[i][j];
     }
     printf("\n");
}
/* Tampilkan Matriks A */
printf("\nA = ");
fprintf(det_invers,"\nA = ");
for (i = 0; i < p; i++)
{
     printf("\n\t");
     fprintf(det_invers,"\n\t");
     for (j = 0; j < q; j++)
     {
          printf(" %.2f", A[i][j]);
          fprintf(det_invers," %.2f", A[i][j]);
     }
}

cout << "\n\n\nTekan Enter untuk lanjutkan proses berikutnya .... (Matriks
H)\n\n";
getch();

/* Pemasukan Data Matriks H (umumnya (3 x 1) */
cout << "\nMasukkan Nilai Matriks H (hanya (3 x 1))\n";
cout << "---------------------------------------\n";
cout << "\nMasukkan Jumlah Baris ? ";
cin >> m;
cout << "\nMasukkan Jumlah Kolom ? ";
cin >> n;

/*******************************************************/
/* Matriks H (3 x 1)                                   */
/*******************************************************/

printf("\n");
for (i = 0; i < m; i++)
{
     for (j = 0; j < n; j++)
     {
          printf("H[%d][%d] = ", i+1, j+1);
          cin >> H[i][j];
     }
     printf("\n");
}
```

```
/* Tampilkan Matriks H */
printf("\nH = ");
fprintf(det_invers,"\n\nH = ");
for (i = 0; i < m; i++)
{
     printf("\n\t");
     fprintf(det_invers,"\n\t");

     for (j = 0; j < n; j++)
     {
          printf("%.2f", H[i][j]);
          fprintf(det_invers,"%.2f", H[i][j]);
     }
}

/*********************************************************/
/* Menyiapkan Matriks x (3 x 1) (Inisialisasi dengan 0)  */
/*********************************************************/

for (i = 0; i < m; i++)
{
     for (j = 0; j < n; j++)
     {
          x[i][j] = 0;
     }
}

cout << "\n\n\nTekan Enter untuk lanjutkan proses berikutnya ....
Determinan A)\n\n";
getch();

/*********************************************************/
/* Determinan Matriks A (3 x 3)                          */
/*********************************************************/

cout << "\nDeterminan Matriks A (3 x 3)";

for (i = 0; i < p; i++)
{
     for (j = 0; j < q; j++)
     {
          cramer1 = A[i-2][j-2]*(A[i-1][j-1]*A[i][j] - A[i-1][j]*
          A[i][j-1]);
          cramer2 = A[i-1][j-2]*(A[i-2][j-1]*A[i][j] - A[i-2][j]*
          A[i][j-1]);
          cramer3 = A[i][j-2]*(A[i-2][j-1]*A[i-1][j] - A[i-2][j]*
          A[i-1][j-1]);
     }
     detA =  cramer1 - cramer2 + cramer3;
}

printf("\n\nDet A = %.2f\n\n", detA);
fprintf(det_invers,"\n\nDet A = %.2f\n\n", detA);

cout << "\n\nTekan Enter untuk lanjutkan proses berikutnya ....
Adjoint A)\n\n";
getch();
```

```
/*******************************************************/
/* Adjoint Matriks A (3 x 3)                           */
/*******************************************************/

cout << "\nAdjoint Matriks A (3 x 3)\n";

for (i = 0; i < p; i++)
{
     for (j = 0; j < q; j++)
     {
          adjA1 = A[i-1][j-1]*A[i][j] - A[i][j-1]*A[i-1][j];
          adjA2 = -(A[i-1][j-2]*A[i][j] - A[i-1][j]*A[i][j-2]);
          adjA3 = A[i-1][j-2]*A[i][j-1] - A[i-1][j-1]*A[i][j-2];
          adjA4 = -(A[i-2][j-1]*A[i][j] - A[i-2][j]*A[i][j-1]);
          adjA5 = A[i-2][j-2]*A[i][j] - A[i-2][j]*A[i][j-2];
          adjA6 = -(A[i-2][j-2]*A[i][j-1] - A[i-2][j-1]*A[i][j-2]);

          adjA7 = A[i-2][j-1]*A[i-1][j] - A[i-2][j]*A[i-1][j-1];
          adjA8 = -(A[i-2][j-2]*A[i-1][j] - A[i-2][j]*A[i-1][j-2]);
          adjA9 = A[i-2][j-2]*A[i-1][j-1] - A[i-2][j-1]*A[i-1][j-2];
     }
}

printf("\nadjA[1][1] = %.1f\t", adjA1);
printf("adjA[1][2] = %.1f\t", adjA2);
printf("adjA[1][3] = %.1f\n", adjA3);
printf("adjA[2][1] = %.1f\t", adjA4);
printf("adjA[2][2] = %.1f\t", adjA5);
printf("adjA[2][3] = %.1f\n", adjA6);
printf("adjA[3][1] = %.1f\t", adjA7);
printf("adjA[3][2] = %.1f\t", adjA8);
printf("adjA[3][3] = %.1f\n", adjA9);

/* Pemasukan Data Matriks Adjoint A */

cout << "\n\nMasukkan Data di atas untuk bentuk Matriks Adj A\n\n";
for (i = 0; i < p; i++)
{
     for (j = 0; j < q; j++)
     {
          printf("Adj A[%d][%d] = ", i+1, j+1);
          cin >> adjA[i][j];
     }
     cout << "\n";
}

/* Cetak Adjoint Matriks A (3 x 3) */

cout << "\nAdj A = \n";
fprintf(det_invers,"\nAdj A = \n");
for (i = 0; i < p; i++)
{
     for (j = 0; j < q; j++)
     {
          printf("\t%.2f", adjA[i][j]);
          fprintf(det_invers,"\t%.2f", adjA[i][j]);
     }
     printf("\n");
     fprintf(det_invers,"\n");
}

cout << "\n\nTekan Enter untuk lanjutkan proses berikutnya ....
(Invers A)\n\n";
getch();
```

```
/*********************************************************/
/* Invers Matriks A (3 x 3)                              */
/*********************************************************/

/* Penghitungan untuk mendapatkan Invers Matriks A (A^-1) */
cout << "\nPenghitungan Invers Matriks A \n\n";
cout << "\nInvers A = \n\n";
fprintf(det_invers,"\nInvers A = \n\n");
for (i = 0; i < p; i++)
{
     for (j = 0; j < q; j++)
     {
          inversA[i][j] = adjA[i][j]/detA;
          printf("\t%.2f", inversA[i][j]);
          fprintf(det_invers,"\t%.2f", inversA[i][j]);
     }

     printf("\n");
     fprintf(det_invers,"\n");
}

cout << "\n\nTekan Enter untuk lanjutkan proses berikutnya ....
(Transpose A)\n\n";
getch();

/*********************************************************/
/* Tranpose Matriks A (3 x 3)                            */
/*********************************************************/

/* Proses Transpose Invers Matriks A */
cout << "\nProses Transpose Invers Matriks A \n\n";
cout << "\nA Transpose = \n\n";
fprintf(det_invers,"\nA Transpose = \n\n");
for (i = 0; i < p; i++)
{
     for (j = 0; j < q; j++)
     {
          Atrans[i][j] = inversA[j][i];
          printf("\t%.2f", Atrans[i][j]);
          fprintf(det_invers,"\t%.2f", Atrans[i][j]);
     }
     printf("\n");
     fprintf(det_invers,"\n");
}

cout << "\n\nTekan Enter untuk lanjutkan proses berikutnya .... (
Solusi x[i])\n\n";
getch();

/*********************************************************/
/* Hitung Solusi Matriks x (3 x 1)                       */
/*********************************************************/

/* Perkalian Matriks A^T dan H */
cout << "\nSolusi Matriks x\n";
printf("\nx = ");
fprintf(det_invers,"\nSolusi Matriks x\n");
fprintf(det_invers,"\nx = ");
for (i = 0; i < p; i++)
{
     for (j = 0; j < n; j++)
     {
          for (k = 0; k < q; k++)
          {
               x[i][j] = x[i][j] + Atrans[i][k] * H[k][j];
          }
```

```
                    printf("\t%.2f", x[i][j]);
                    fprintf(det_invers,"\t%.2f", x[i][j]);
               }
               printf("\n");
               fprintf(det_invers,"\n");
          }
          /* Solusi x[i] */
          cout << "\n\n\nMaka ";
          fprintf(det_invers,"\n\n\nMaka ");
          for (i = 0; i < m; i++)
          {
               for (j = 0; j < n; j++)
               {
                    printf("\tx[%d] = %.2f", i+1, x[i][j]);
                    fprintf(det_invers,"\tx[%d] = %.2f", i+1, x[i][j]);
               }
          }

          cout << "\n\n\nCoba lagi dengan data yang berbeda (y/t) ? ";
          cin >> lagi;

          fclose(det_invers);

     }while(lagi != 't');

     return 0;
}
```

```
/* Contoh soal Penyelesaian Persamaan Linier Serentak (PLS) menggunakan Metode Matriks
Augmented Bab IV  Hal. 37 Diktat Metode Komputasi 2004 */
/* Nama Program : m_augmented.cpp */

#include<iostream.h>
#include<stdlib.h>
#include<conio.h>
#include<stdio.h>

#define baris 10
#define kolom 10

main()
{
     FILE *m_augmented;
     double    A[baris][kolom],        /* Matriks Persamaan (A) */
               AH[baris][kolom],       /* Matriks Augmented (AH) */
               x[baris],               /* Matriks Solusi (x) */
               H[baris],               /* Matriks Persamaan (H) */
               m, temp;
     int i, j, k, n, p, q, r;
     char lagi;

     do
     {
          clrscr();
          m_augmented = fopen("m_augmented.txt", "w+");

          cout << "\n\nMetode Invers-Determinan Matriks untuk PLS\n";
          cout << "==========================================\n\n";

          /*************************************\
          \*         A l g o r i t m a         */
          /*         -----------------         *\
          \*                                   */
          /*   1.   Masukkan Matriks A         *\
          \*   2.   Masukkan Matriks H         */
          /*   3.   Ubah ke Bentuk Matriks AH  *\
          \*   4.   Buat Matriks Segitiga Bawah*/
          /*   5.   Hitung Solusi Matriks x    *\
          \*                                   */
          /*************************************/

          /*********************************************************/
          /* Pengisian Matriks A (3 x 3) dan H (3 x 1)             */
          /*********************************************************/
          cout << "\nMasukkan Nilai Matriks A (umumnya (3 x 3))\n";
          cout << "-------------------------------------\n";
          cout << "\nMasukkan Jumlah Baris ? ";
          cin >> p;
          cout << "\nMasukkan Jumlah Kolom ? ";
          cin >> q;

          /*********************************************************/
          /* Matriks A (3 x 3)                                     */
          /*********************************************************/
          printf("\n");
          for (i = 0; i < p; i++)
          {
               for (j = 0; j < q; j++)
               {
                    printf("A[%d][%d] = ", i+1, j+1);
                    cin >> A[i][j];
               }
               printf("\n");
          }
```

```
/* Pengisian Data Matriks H (umumnya (3 x 1) */
cout << "\nMasukkan Nilai Matriks H (umumnya (3 x 1))\n";
cout << "-----------------------------------------\n";
cout << "\nMasukkan Jumlah Baris ? ";
cin >> r;

/*********************************************************/
/* Matriks H (3 x 1)                                     */
/*********************************************************/
printf("\n");
for (i = 0; i < r; i++)
{
     printf("H[%d] = ", i+1);
     cin >> H[i];
}

/*********************************************************/
/* Tampilkan Matriks A (3 x 3) dan Matriks H (3 x 1)     */
/*********************************************************/
/* Cetak Matriks A */
printf("\nA = ");
fprintf(m_augmented,"\n\nA = ");
for (i = 0; i < p; i++)
{
     printf("\n");
     fprintf(m_augmented,"\n");
     for (j = 0; j < q; j++)
     {
          printf("\t%.2f", A[i][j]);
          fprintf(m_augmented,"\t%.3f", A[i][j]);
     }
}
/* Cetak Matriks H */
printf("\n\nH = ");
fprintf(m_augmented,"\n\nH = \n");
for (i = 0; i < r; i++)
{
     printf("\n\t%.2f", H[i]);
fprintf(m_augmented,"\t%.3f\n", H[i]);
}

/*********************************************************/
/* Inisialisasi Matriks x (3 x 1)                        */
/*********************************************************/
/* Inisialisasi x[i] = 0 */
for (i = 0; i < r; i++)
{
     x[i] = 0;
}

/*********************************************************/
/* Proses membentuk Matriks Augmented AH                 */
/*********************************************************/
cout << "\n\n\nMatriks AH sebelum proses Segitiga Bawah\n";
cout << "-------------------------------------";
/* Augmented Matrix */
printf("\n\nAH = ");
fprintf(m_augmented,"\n\nAH = \n");
for (i = 0; i < p; i++)
{
     printf("\n");
     fprintf(m_augmented,"\n");
```

```
      for (j = 0; j < q+1; j++)
      {
            if(j != q)
            {
                  AH[i][j] = A[i][j];
                  printf("\t%.2f", AH[i][j]);
                  fprintf(m_augmented,"\t%.3f", AH[i][j]);
            }
            else
            {
                  AH[i][j] = H[i];
                  printf("\t%.2f", AH[i][j]);
                  fprintf(m_augmented,"\t%.3f", AH[i][j]);
            }
      }
}

/*********************************************************/
/* Proses Membuat Matriks Segitiga Bawah AH              */
/*********************************************************/
for (k = 0; k < p-1; k++)
{
      for (i = k+1; i < q+1; i++)
      {
            m = AH[i][k]/AH[k][k];
            for (j = k; j < q+1; j++)
            {
                  AH[i][j] = AH[i][j] - m * AH[k][j];
            }
      }
}

cout << "\n\n\nMatriks AH setelah proses Segitiga Bawah\n";
cout << "----------------------------------------";
/* Cetak Matriks AH */
printf("\n\nAH = ");
fprintf(m_augmented,"\n\nAH = \n");
for (i = 0; i < p; i++)
{
      printf("\n");
      fprintf(m_augmented,"\n");
      for (j = 0; j < q+1; j++)
      {
            if(j != q)
            {
                  A[i][j] = AH[i][j];
                  printf("\t%.2f", A[i][j]);
                  fprintf(m_augmented,"\t%.3f", A[i][j]);
            }
            else
            {
                  H[i] = AH[i][j];
                  fprintf(m_augmented,"\t%.3f", H[i]);
            }
      }
}
/* Cetak Matriks H */
printf("\n\nH = ");
fprintf(m_augmented,"\n\nH = \n");
for (i = 0; i < r; i++)
{
      printf("\n\t%.2f", H[i]);
fprintf(m_augmented,"\n\t%.3f", H[i]);
}
```

```
        /*********************************************************/
        /* Matriks x (3 x 1) Hasil Penghitungan                  */
        /*********************************************************/
        /* Solusi x[i] */
        for(k = p-1; k >= 0; k--)
        {
             temp = 0.0;
             for(j = k+1; j < q; j++)
             {
                  temp = temp + A[k][j] * x[j];
             }
             x[k] = (H[k] - temp)/A[k][k];
        }
        /* Cetak Matriks x hasil Perhitungan */
        cout << "\n\nSolusi x\n";
        cout << "--------";
        printf("\n\nx = \n");
        fprintf(m_augmented,"\n\nSolusi x\n");
        for (n = r-1; n >= 0; n--)
        {
             printf("\tx[%d] = %.3f\n", n+1, x[n]);
        fprintf(m_augmented,"\tx[%d] = %.3f\n", n+1, x[n]);
        }

        fclose(m_augmented);

        cout << "\n\n\nCoba lagi dengan data yang berbeda (y/t) ? ";
        cin >> lagi;

    }while(lagi != 't');

    return 0;
}
```

```
/* Contoh soal Metode Eliminasi Gauss untuk penyelesaian Persamaan Linier Serentak
(PLS). Bab IV Hal. 38 Diktat Metode Komputasi 2004 */
/* Nama Program : eliminasi_gauss.cpp */
/* Untuk penyelesaian Matriks n x n */

#include<iostream.h>
#include<stdlib.h>
#include<conio.h>
#include<stdio.h>

#define Data 10

main()
{
     FILE *eliminasi_gauss;
     double    A[Data][Data],          /* Matriks A */
               H[Data],                /* Matriks H */
               x[Data],                /* Matriks x */
               m,                      /* Faktor Pengali */
               temp;
     int i, j, k, n, p, q, r;
     har lagi;

     do
     {
          clrscr();
          eliminasi_gauss = fopen("eliminasi_gauss.txt", "w+");

          cout << "\n\nMetode Eliminasi Gauss Matriks untuk PLS\n";
          cout << "========================================\n\n";

          /*********************************************************/
          /* Pengisian Matriks A (3 x 3) dan H (3 x 1)             */
          /*********************************************************/

          /* Pengisian Data Matriks A  (umumnya (3 x 3) */
          cout << "\nMasukkan Nilai Matriks A (umumnya (3 x 3))\n";
          cout << "------------------------------------------\n";
          cout << "\nMasukkan Jumlah Baris ? ";
          cin >> p;
          cout << "\nMasukkan Jumlah Kolom ? ";
          cin >> q;
          printf("\n");

          /* Matriks A */
          for (i = 0; i < p; i++)
          {
               for (j = 0; j < q; j++)
               {
                    printf("A[%d][%d] = ", i+1, j+1);
                    cin >> A[i][j];
               }
               printf("\n");
          }

          /* Pengisian Data Matriks H (umumnya (3 x 1) */
          cout << "\nMasukkan Nilai Matriks H (umumnya (3 x 1))\n";
          cout << "------------------------------------------\n";
          cout << "\nMasukkan Jumlah Baris ? ";
          cin >> r;
          printf("\n");
```

```
/* Matriks H */
for (n = 0; n < r; n++)
{
     printf("H[%d] = ", n+1);
     cin >> H[n];
}

/*********************************************************/
/* Mencetak Data Matriks A (3 x 3) dan H (3 x 1)         */
/*********************************************************/
/* Cetak Matriks A */
cout << "\n\n\nMatriks A dan H sebelum proses Eliminasi Gauss\n";
cout << "-----------------------------------------";
printf("\n\nA = ");
fprintf(eliminasi_gauss,"\n\nA = ");
for (i = 0; i < p; i++)
{
     printf("\n");
     fprintf(eliminasi_gauss,"\n");
     for (j = 0; j < q; j++)
     {
          printf("\t%.3f", A[i][j]);
          fprintf(eliminasi_gauss,"\t%.3f", A[i][j]);
     }
}
/* Cetak Matriks H */
printf("\n\nH = \n");
fprintf(eliminasi_gauss,"\n\nH = \n");
for (n = 0; n < r; n++)
{
     printf("\t%.3f\n", H[n]);
     fprintf(eliminasi_gauss,"\t%.3f\n", H[n]);
}

/*********************************************************/
/* Eliminasi Gauss Matriks A (3 x 3) dan H (3 x 1)       */
/*********************************************************/
/* Proses Eliminasi Gauss */
for (k = 0; k < p-1; k++)
{
     for (i = k+1; i < q; i++)
     {
          m = A[i][k]/A[k][k];
          for (j = k; j < q; j++)
          {
               A[i][j] = A[i][j] - m * A[k][j];
          }
          H[i] = H[i] - m * H[k];
     }
}

/*********************************************************/
/* Cetak Matriks A (3 x 3) dan H (3 x 1) Hasil Eliminasi */
/*********************************************************/
/* Cetak Matriks A hasil Eliminasi Gauss */
cout << "\n\nMatriks A dan H setelah proses Eliminasi Gauss\n";
cout << "-----------------------------------------";

printf("\n\nA = ");
fprintf(eliminasi_gauss,"\n\nA = ");
for (i = 0; i < p; i++)
{
     printf("\n");
     fprintf(eliminasi_gauss,"\n");
```

```
                for (j = 0; j < q; j++)
                {
                     printf("\t%.3f", A[i][j]);
                     fprintf(eliminasi_gauss,"\t%.3f", A[i][j]);
                }
           }

           /* Cetak Matriks H hasil Eliminasi Gauss */
           printf("\n\nH = \n");
           fprintf(eliminasi_gauss,"\n\nH = \n");
           for (n = 0; n < r; n++)
           {
                printf("\t%.3f\n", H[n]);
                fprintf(eliminasi_gauss,"\t%.3f\n", H[n]);
           }

           /*********************************************************/
           /* Matriks x (3 x 1) Hasil Penghitungan                  */
           /*********************************************************/
           /* Solusi x[i] */
           for(k = p-1; k >= 0; k--)
           {
                temp = 0.0;
                for(j = k+1; j < q; j++)
                {
                     temp = temp + A[k][j] * x[j];
                }
                x[k] = (H[k] - temp)/A[k][k];
           }

           /* Cetak Matriks x hasil Perhitungan */
           cout << "\n\nSolusi x\n";
           cout << "--------";
           printf("\n\nx = \n");
           fprintf(eliminasi_gauss,"\n\nSolusi x\n");
           for (n = r-1; n >= 0; n--)
           {
                printf("\tx[%d] = %.3f\n", n+1, x[n]);
                fprintf(eliminasi_gauss,"\tx[%d] = %.3f\n", n+1, x[n]);
           }

           fclose(eliminasi_gauss);

           cout << "\n\n\nCoba lagi dengan data yang berbeda (y/t) ? ";
           cin >> lagi;

     }while(lagi != 't');

     return 0;
}
```

```
/* Contoh soal Metode Gauss-Seidel untuk penyelesaian Persamaan Linier Serentak (PLS)
Bab IV Hal. 39-40 Diktat Metode Komputasi 2004 */
/* Nama Program : gauss-seidel.cpp */

#include <iostream.h>
#include <stdlib.h>
#include <conio.h>
#include <stdio.h>
#include <math.h>

#define MaksData 100
#define MaksIter 100

main()
{
     FILE *gauss_seidel;
     double    x[MaksData],
               y[MaksData],
               z[MaksData],
               x1,
               y0, y1, tempy,
               z0, z1, tempz;
     int i;
      char lagi;

     do
     {
          clrscr();
          gauss_seidel = fopen("gauss_seidel.txt", "w+");

          cout << "\n\nMetode Gauss-Seidel untuk PLS\n";
          cout << "=============================\n\n";
          cout << "Persamaan-persamaan sbb.\n\n";
          cout << "27*x + 6*y - z = 85\n\n";
          cout << "6*x + 15*y + 2*z = 72\n\n";
          cout << "x + y + 54*z = 110\n\n";
          cout << "Konversi Persamaan-persamaan sbb.\n\n";
          cout << "x = (85 - 6*y + z)/27\n\n";
          cout << "y = (72 - 6*x - 2*z)/15\n\n";
          cout << "z = (110 - x - y)/54\n\n";

          printf("\n\n\t-----------------------------------------\n");
          printf("\titerasi ke-\tx[i]\ty[i]\tz[i]\n");
          printf("\t-----------------------------------------\n\n");
          fprintf(gauss_seidel,"\n\t-----------------------------------------\n");
          fprintf(gauss_seidel,"\titerasi ke-\tx[i]\ty[i]\tz[i]\n");
          fprintf(gauss_seidel,"\t-----------------------------------------\n\n");

          /* inisialisasi awal */
          y0 = 0.0;
          z0 = 0.0;
          i = 0;

          do
          {
               /* Perhitungan Gauss-Seidel */
               x[i] = (85 - 6*y0 + z0)/27;
               y[i] = (72 - 6*x[i] - 2*z0)/15;
               z[i] = (110 - x[i] - y[i])/54;

               printf("\t\t%d\t%.5f\t%.5f\t%.5f\n", i+1, x[i], y[i], z[i]);
               fprintf(gauss_seidel, "\t\t%d\t%.5f\t%.5f\t%.5f\n", i+1, x[i], y[i],
               z[i]);

               x1 = x[i];
```

```
            tempy = y[i];
            y1 = y0;
            y0 = tempy;

            tempz = z[i];
            z1 = z0;
            z0 = tempz;

            i++;
        }while((y1 != y0) && (z1 != z0));

        printf("\n\t-----------------------------------------\n\n");
        fprintf(gauss_seidel,"\n\t-----------------------------------------\n\n");

        printf("\nMaka x = %.5f, dengan akurasi 3 desimal  x = %.3f", x1, x1);
        printf("\n     y = %.5f, dengan akurasi 3 desimal  y = %.3f", y0, y0);
        printf("\n     z = %.5f, dengan akurasi 3 desimal  z = %.3f\n", z0, z0);
        printf("\nDengan iterasi sebanyak = %.d", i);
        fprintf(gauss_seidel,"\nMaka x = %.5f, dengan akurasi 3 desimal  x = %.3f",
        x1, x1);
        fprintf(gauss_seidel,"\n     y = %.5f, dengan akurasi 3 desimal  y = %.3f",
        y0, y0);
        fprintf(gauss_seidel,"\n     z = %.5f, dengan akurasi 3 desimal  z =
        %.3f\n", z0, z0);
        fprintf(gauss_seidel,"\nDengan iterasi sebanyak = %.d", i);

        fclose(gauss_seidel);

        cout << "\n\nCoba lagi dengan data awal atau rumus yang berbeda (y/t) ? ";
        cin >> lagi;

    }while(lagi != 't');

    return 0;
}
```

```
/* Penyelesaian Soal Metode Gauss-Seidel untuk penyelesaian Persamaan Linier Serentak
(PLS) Bab IV Hal. 41 Diktat Metode Komputasi 2004 */
/* Nama Program : gauss-seidel_3a.cpp */

#include <iostream.h>
#include <stdlib.h>
#include <conio.h>
#include <stdio.h>
#include <math.h>

#define MaksData 100
#define MaksIter 100

main()
{
     FILE *gauss_seidel_3a;
     double    x1[MaksData],
               x2[MaksData],
               x3[MaksData],
               x11,
               x20, x21, tempx2,
               x30, x31, tempx3;
     int i;
     char lagi;

     do
     {
          clrscr();
          gauss_seidel_3a = fopen("gauss_seidel_3a.txt", "w+");

          cout << "\n\nMetode Gauss-Seidel untuk PLS\n";
          cout << "=============================\n\n";
          cout << "Persamaan-persamaan sbb.\n\n";
          cout << "3.5*x1 + 2.8*x2 + 6.2x3 = 9.8999\n\n";
          cout << "2.7*x1 + 8*x2 + 3*x3 = -6.1744\n\n";
          cout << "-4*x1 - 3.6*x2 - 2.8*x3 = 5.6512\n\n\n";
          cout << "Konversi Persamaan-persamaan sbb.\n\n";
          cout << "x1 = (9.8999 - 2.8*x2 - 6.2*x3)/3.5\n\n";
          cout << "x2 = (-6.1744 - 2.7*x1 - 3*x3)/8\n\n";
          cout << "x3 = (5.6512 + 4*x1 + 3.6*x2)/(-2.8)\n\n";

          printf("\n\n\t----------------------------------------\n");
          printf("\titerasi ke-\tx1[i]\tx2[i]\tx3[i]\n");
          printf("\t----------------------------------------\n\n");
          fprintf(gauss_seidel_3a,"\n\t----------------------------------------\n");
          fprintf(gauss_seidel_3a,"\titerasi ke-\tx1[i]\tx2[i]\tx3[i]\n");
          fprintf(gauss_seidel_3a,"\t----------------------------------------\n\n");

          /* inisialisasi awal */
          x20 = 0.0;
          x30 = 0.0;
          i = 0;

          do
          {
               /* Perhitungan Gauss-Seidel */
               x1[i] = (9.8999 - 2.8*x20 - 6.2*x30)/3.5;
               x2[i] = (-6.1744 - 2.7*x1[i] - 3*x30)/8;
               x3[i] = (5.6512 + 4*x1[i] + 3.6*x2[i])/(-2.8);

               printf("\t    %d\t\t%.5f\t%.5f\t%.5f\n", i+1, x1[i], x2[i], x3[i]);
               fprintf(gauss_seidel_3a, "\t    %d\t\t%.5f\t%.5f\t%.5f\n", i+1, x1[i],
               x2[i], x3[i]);

               x11 = x1[i];
```

```
            tempx2 = x2[i];
            x21 = x20;
            x20 = tempx2;

            tempx3 = x3[i];
            x31 = x30;
            x30 = tempx3;

            i++;
            getch();
        }while((x21 != x20) && (x31 != x30));

        printf("\n\t----------------------------------------\n\n");
        fprintf(gauss_seidel_3a,"\n\t----------------------------------------
        \n\n");

        printf("\nMaka x1 = %.5f, dengan akurasi 3 desimal  x1 = %.3f", x11, x11);
        printf("\n     x2 = %.5f, dengan akurasi 3 desimal  x2 = %.3f", x20, x20);
        printf("\n     x3 = %.5f, dengan akurasi 3 desimal  x3 = %.3f\n", x30, x30);
        printf("\nDengan iterasi sebanx2ak = %.d", i);
        fprintf(gauss_seidel_3a,"\nMaka x1 = %.5f, dengan akurasi 3 desimal  x1 =
        %.3f", x11, x11);
        fprintf(gauss_seidel_3a,"\n     x2 = %.5f, dengan akurasi 3 desimal  x2 =
        %.3f", x20, x20);
        fprintf(gauss_seidel_3a,"\n     x3 = %.5f, dengan akurasi 3 desimal  x3 =
        %.3f\n", x30, x30);
        fprintf(gauss_seidel_3a,"\nDengan iterasi sebanyak = %.d", i);

        fclose(gauss_seidel_3a);

        cout << "\n\nCoba lagi dengan data awal atau rumus yang berbeda (y/t) ? ";
        cin >> lagi;

    }while(lagi != 't');

    return 0;
}
```

```
/* Contoh soal Metode Gauss-Seidel untuk penyelesaian Persamaan Linier Serentak (PLS)
Bab IV Hal. 39-40 Diktat Metode Komputasi 2004 */
/* Nama Program : gauss-seidel_ex.cpp */

#include <iostream.h>
#include <stdlib.h>
#include <conio.h>
#include <stdio.h>
#include <math.h>

#define MaksData 100
#define MaksIter 100

main()
{
     FILE *gauss_seidel;
     double    x[MaksData],
               y[MaksData],
               z[MaksData],
               x1,
               y0, y1, tempy,
               z0, z1, tempz,
           pers1, pers2, pers3;
     int i;
     char lagi;

     do
     {
          clrscr();
          gauss_seidel = fopen("gauss_seidel_ex.txt", "w+");

          cout << "\n\nMetode Gauss-Seidel untuk PLS\n";
          cout << "=============================\n\n";
          cout << "Persamaan-persamaan sbb.\n\n";
          cout << "27*x + 6*y - z = 85\n\n";
          cout << "6*x + 15*y + 2*z = 72\n\n";
          cout << "x + y + 54*z = 110\n\n";
          cout << "Konversi Persamaan-persamaan sbb.\n\n";
          cout << "x = (85 - 6*y + z)/27\n\n";
          cout << "y = (72 - 6*x - 2*z)/15\n\n";
          cout << "z = (110 - x - y)/54\n\n";

          printf("\n\n\t----------------------------------------\n");
          printf("\titerasi ke-\tx[i]\ty[i]\tz[i]\n");
          printf("\t----------------------------------------\n\n");
          fprintf(gauss_seidel,"\n\t----------------------------------------\n");
          fprintf(gauss_seidel,"\titerasi ke-\tx[i]\ty[i]\tz[i]\n");
          fprintf(gauss_seidel,"\t----------------------------------------\n\n");

          /* inisialisasi awal */
          y0 = 0.0;
          z0 = 0.0;
          i = 0;

          do
          {
               /* Perhitungan Gauss-Seidel */
               x[i] = (85 - 6*y0 + z0)/27;
               y[i] = (72 - 6*x[i] - 2*z0)/15;
               z[i] = (110 - x[i] - y[i])/54;

               printf("\t    %d\t\t%.5f\t%.5f\t%.5f\n", i+1, x[i], y[i], z[i]);
               fprintf(gauss_seidel, "\t    %d\t\t%.5f\t%.5f\t%.5f\n", i+1, x[i],
               y[i], z[i]);

               x1 = x[i];
```

```
            tempy = y[i];
            y1 = y0;
            y0 = tempy;

            tempz = z[i];
            z1 = z0;
            z0 = tempz;

            i++;
        }while((y1 != y0) && (z1 != z0));

        printf("\n\t----------------------------------------\n\n");
        fprintf(gauss_seidel,"\n\t----------------------------------------\n\n");

        printf("\nMaka x = %.5f, dengan akurasi 3 desimal  x = %.3f", x1, x1);
        printf("\n     y = %.5f, dengan akurasi 3 desimal  y = %.3f", y0, y0);
        printf("\n     z = %.5f, dengan akurasi 3 desimal  z = %.3f\n", z0, z0);
        printf("\nDengan iterasi sebanyak = %.d", i);
        fprintf(gauss_seidel,"\nMaka x = %.5f, dengan akurasi 3 desimal  x = %.3f",
        x1, x1);
        fprintf(gauss_seidel,"\n     y = %.5f, dengan akurasi 3 desimal  y = %.3f",
        y0, y0);
        fprintf(gauss_seidel,"\n     z = %.5f, dengan akurasi 3 desimal  z =
        %.3f\n", z0, z0);
        fprintf(gauss_seidel,"\nDengan iterasi sebanyak = %.d", i);

        /* Uji hasil penghitungan Gauss-Seidel */
        cout << "\n\n\nUji Hasil penghitungan Gauss-Seidel\n";
        cout << "-----------------------------------\n";

        pers1 = 27*x1 + 6*y0 - z0;
        pers2 = 6*x1 + 15*y0 + 2*z0;
        pers3 = x1 + y0 + 54*z0;

        printf("\n\n27x + 6y - z = 85 dengan x = %.3f, y = %.3f, z = %.3f diperoleh
        %.2f\n", x1, y0, z0, pers1);
        printf("\n\n6x + 15y + 2z = 72 dengan x = %.3f, y = %.3f, z = %.3f diperoleh
        %.2f\n", x1, y0, z0, pers2);
        printf("\n\nx + y + 54z = 110 dengan x = %.3f, y = %.3f, z = %.3f diperoleh
        %.2f\n", x1, y0, z0, pers3);

        fclose(gauss_seidel);

        cout << "\n\n\nCoba lagi dengan data awal atau rumus yang berbeda (y/t) ? ";
        cin >> lagi;

    }while(lagi != 't');

    return 0;
}
```

PENYELESAIAN PERSAMAAN TIDAK LINIER SERENTAK

Listing Program C++

```
/* Contoh soal Metode Newton-Rhapson untuk penyelesaian Persamaan Tidak Linier
Serentak (PLTS) Bab V Hal. 44-45 Diktat Metode Komputasi 2004 */
/* Nama Program : plts_nr.cpp */

#include <iostream.h>
#include <stdlib.h>
#include <conio.h>
#include <stdio.h>
#include <math.h>

#define e 2.718281828
#define error_limit 1E-16      /* flexible sesuai kebutuhan */

main()
{
     FILE *plts_nr;
     double    x1, x10, tempx1,
               x2, x20, tempx2,
               x100, x200,
               h, k,
               Fx, dFx1, dFx2,
               Gx, dGx1, dGx2;
     int i;
     char lagi;

     do
     {
          clrscr();
          i = 0;
          plts_nr = fopen("plts_nr.txt", "w+");

          cout << "\nMetode Newton-Rhapson untuk PLTS\n";
          cout << "\nContoh Soal PLTS Metode Newton-Rhapson\n";
          cout << "=====================================\n\n";

          cout << "\nCAUTION !";
          cout << "\n---------\n";
          cout << "\nKekurang cermatan pemilihan tebakan awal x1 dan x2";
          cout << "\nakan berakibat pada kegagalan proses komputasi";
          cout << "\n-------------------------------------------------\n";

          cout << "\nMasukkan Nilai x10 (tebakan awal) : ";
          cin >> x10;
          cout << "\nMasukkan Nilai x20 (tebakan awal) : ";
          cin >> x20;

          x100 = x10;
          x200 = x20;
```

```
printf("\n\n\t-----------------------------------------------------------------
------\n");
printf("\ti ke-\tx1\t\tx2\t\tFx\t\tGx\n");
printf("\t-----------------------------------------------------------------
--\n");
printf("\t\tkoreksi h\t\t\tkoreksi k\n");
printf("\t-----------------------------------------------------------------
--\n\n");


fprintf(plts_nr,"\n\t-----------------------------------------------------------
------------\n");
fprintf(plts_nr,"\ti ke-\tx1\t\tx2\t\tFx\t\tGx\n");
fprintf(plts_nr,"\t-----------------------------------------------------------
-----------\n");
fprintf(plts_nr,"\t\tkoreksi h\t\t\tkoreksi k\n");
fprintf(plts_nr,"\t-----------------------------------------------------------
-----------\n\n");

do
{
     i++;

     Fx = x10*pow(e,-x20) - x20*x10 + 12.6;
     Gx = 4*log(x20) + pow(x10,2) + 0.3 - 3*x10*sqrt(x20);

     dFx1 = pow(e,-x20) - x20;
     dFx2 = -(x10*pow(e,-x20)) - x10;
     dGx1 = 2*x10 - 3*sqrt(x20);
     dGx2 = 4/x20 - (3*x10)/(2*sqrt(x20));

     h = ((-Fx*dGx2) - (-Gx*dFx2))/((dFx1*dGx2) - (dFx2*dGx1));
     k = ((-Gx*dFx1) - (-Fx*dGx1))/((dFx1*dGx2) - (dFx2*dGx1));

     printf("\t  %d\t%.9f\t%.9f\t%.9f\t%.9f\n", i, x10, x20, Fx, Gx);
     printf("\t\t%.16f\t\t%.16f\n\n", h, k);
     fprintf(plts_nr,"\t  %d\t%.9f\t%.9f\t%.9f\t%.9f\n", i, x10, x20, Fx,
     Gx);
     fprintf(plts_nr,"\t\t%.16f\t\t%.16f\n\n", h, k);

     x1 = x10 + h;
     x2 = x20 + k;

     tempx1 = x1;
     x1 = x10;
     x10 = tempx1;

     tempx2 = x2;
     x2 = x20;
     x20 = tempx2;

} while(fabs(h) > error_limit && fabs(k) > error_limit);

printf("\t-----------------------------------------------------------------
--\n\n");
fprintf(plts_nr,"\t-----------------------------------------------------------
-----------\n\n");
printf("\nTebakan awal x1 = %.3f, x2 = %.3f menghasilkan", x100, x200);
printf("\n\nx1 = %.8f dan x2 = %.8f\n\n", x1, x2);
printf("\nDengan koreksi h = %.16f", h);
printf("\n               k = %.16f", k);
fprintf(plts_nr,"\nTebakan awal x1 = %.3f, x2 = %.3f menghasilkan", x100,
x200);
fprintf(plts_nr,"\n\nx1 = %.8f dan x2 = %.8f\n\n", x1, x2);
fprintf(plts_nr,"\nDengan koreksi h = %.16f", h);
fprintf(plts_nr,"\n               k = %.16f", k);
```

```
        /* Pembuktian */
        cout << "\n\nSubstitusi Nilai x dan y ke dalam persamaan menghasilkan :";
        Fx = x1*pow(e,-x2) - x2*x1 + 12.6;
        Gx = 4*log(x2) + pow(x1,2) + 0.3 - 3*x1*sqrt(x2);
        printf("\n\nF(x) = %.9f dan G(x) = %.9f\n\n", Fx, Gx);


        fclose(plts_nr);


        cout << "\n\nCoba lagi dengan x dan y berbeda (y/t) ? ";
        cin >> lagi;

    }while(lagi != 't');

    return 0;
}
```

```
/* Penyelesaian Soal Metode Newton-Rhapson untuk penyelesaian Persamaan Tidak Linier
Serentak (PLTS) Bab V Hal. 48 No. 1.c. Diktat Metode Komputasi 2004 */
/* Nama Program : plts_nr_1c.cpp */

#include <iostream.h>
#include <stdlib.h>
#include <conio.h>
#include <stdio.h>
#include <math.h>

#define e 2.718281828
#define radian 0.01745
#define error_limit 1E-16      /* flexible sesuai kebutuhan */

main()
{
     FILE *plts_nr;
     double    x1, x, tempx1,
               y1, y, tempy1,
               x0, y0,
               h, k,
               Fx, dFx, dFy,
          Gx, dGx, dGy;
     int i;
     char lagi;

     do
     {
          clrscr();
          cout << "Metode Newton-Rhapson untuk PLTS\n";
          cout << "\nJawaban Soal PLTS Bab V No. 1.c.\n";
          cout << "================================\n\n";

          cout << "\nCAUTION !";
          cout << "\n---------\n";
          cout << "\nKekurang cermatan pemilihan tebakan awal x dan y";
          cout << "\nakan berakibat pada kegagalan proses komputasi";
          cout << "\n-------------------------------------------------\n";

          plts_nr = fopen("plts_nr_1c.txt", "w+");
          cout << "\nMasukkan Nilai x (tebakan awal) : ";
          cin >> x;
          cout << "\nMasukkan Nilai y (tebakan awal) : ";
          cin >> y;

          i = 0;
          x0 = x;
          y0 = y;

          printf("\n\n\t-------------------------------------------------------------------
          ------\n");
          printf("\ti ke-\tx\t\ty\t\tFx\t\tGx\n");
          printf("\t-------------------------------------------------------------------
          --\n");
          printf("\t\tkoreksi h\t\t\tkoreksi k\n");
          printf("\t-------------------------------------------------------------------
          --\n\n");
          fprintf(plts_nr,"\n\t-----------------------------------------------------------
          -------------\n");
          fprintf(plts_nr,"\ti ke-\tx\t\ty\t\tFx\t\tGx\n");
          fprintf(plts_nr,"\t-----------------------------------------------------------
          -----------\n");
          fprintf(plts_nr,"\t\tkoreksi h\t\t\tkoreksi k\n");
          fprintf(plts_nr,"\t-----------------------------------------------------------
          -----------\n\n");
```

```
        do
        {
             i++;

             Fx = pow(x,2) + x - pow(y,2) - 1;
             Gx = y - pow(pow(sin(x),2),2);

             dFx = 2*x + 1;
             dFy = -2*y;
             dGx = -2*sin(x * radian);
             dGy = 1;

             h = ((-Fx*dGy) - (-Gx*dFy))/((dFx*dGy) - (dFy*dGx));
             k = ((-Gx*dFx) - (-Fx*dGx))/((dFx*dGy) - (dFy*dGx));

             printf("\t  %d\t%.9f\t%.9f\t%.9f\t%.9f\n", i, x, y, Fx, Gx);
             printf("\t\t%.16f\t\t%.16f\n\n", h, k);
             fprintf(plts_nr,"\t  %d\t%.9f\t%.9f\t%.9f\t%.9f\n", i, x, y, Fx, Gx);
             fprintf(plts_nr,"\t\t%.16f\t\t%.16f\n\n", h, k);

             x1 = x + h;
             y1 = y + k;

             tempx1 = x1;
             x1 = x;
             x = tempx1;

             tempy1 = y1;
             y1 = y;
             y = tempy1;

        }while(fabs(h) > error_limit && fabs(k) > error_limit);

        printf("\t-------------------------------------------------------------------
        --\n\n");
        fprintf(plts_nr,"\t-------------------------------------------------------------
        -----------\n\n");

        printf("\nTebakan awal x = %.3f, y = %.3f menghasilkan", x0, y0);
        printf("\n\nx = %.9f dan y = %.9f\n\n", x1, y1);
        printf("\nDengan koreksi h = %.16f", h);
        printf("\n               k = %.16f", k);
        fprintf(plts_nr,"\nTebakan awal x = %.3f, y = %.3f menghasilkan", x0, y0);
        fprintf(plts_nr,"\n\nx = %.9f dan y = %.9f\n", x1, y1);
        fprintf(plts_nr,"\nDengan koreksi h = %.16f", h);
        fprintf(plts_nr,"\n               k = %.16f", k);

        /* Pembuktian */
        cout << "\n\nSubstitusi Nilai x dan y ke dalam persamaan menghasilkan :";
        Fx = pow(x1,2) + x1 - pow(y1,2) - 1;
        Gx = y1 - pow(pow(sin(x1),2),2);
        printf("\n\nF(x) = %.9f dan G(x) = %.9f\n\n", Fx, Gx);

        fclose(plts_nr);

        cout << "\n\nCoba lagi dengan x dan y berbeda (y/t) ? ";
        cin >> lagi;
        clrscr();

    } while(lagi != 't');

    return 0;
}
```

```
/* Contoh soal Metode Iterasi x = F(x,y,z, .....)untuk penyelesaian Persamaan Tidak
Linier Serentak (PLTS)Bab V Hal. 46-47 Diktat Metode Komputasi 2004 */
/* Nama Program : x_fxyz.cpp */


#include <iostream.h>
#include <stdlib.h>
#include <conio.h>
#include <stdio.h>
#include <math.h>

#define e 2.718281828

main()
{
     FILE *x_fxyz;
     double    x1, x10, x1awal, tempx1,
               x2, x20, x2awal, tempx2,
               dF1, dF1x1, dF1x2,
               dF2, dF2x1, dF2x2;
     int i;
     char lagi;

     do
     {
          clrscr();
          i = 0;
          x_fxyz = fopen("x_fxyz.txt", "w+");

          cout << "\n\nMetode Iterasi x = f(x,y,z,...)\n";
          cout << "================================\n\n\n";

          //Persamaan dan turunannya
          cout << "x1= -(4 - x2^2)^1/2\n\n";
          cout << "x2 = 1 - e^x1\n\n";
          cout << "dF1/dx1 = 0\n\n";
          cout << "dF1/dx2 = -(x2 / (4 - x2^2)^1/2))\n\n";
          cout << "dF1/dx1 = -e^x1\n\n";
          cout << "dF1/dx1 = 0\n\n";

          do
          {
               cout << "\nMasukkan Nilai x10 (tebakan awal) ? ";
               cin >> x10;
               cout << "\nMasukkan Nilai x20 (tebakan awal) ? ";
               cin >> x20;

               x1awal = x10;
               x2awal = x20;

               dF1x1 = 0;
               dF1x2 = -(x20/(sqrt(4-pow(x20,2))));
               dF2x1 = -pow(e,x10);
               dF2x2 = 0;

               dF1 = dF1x1 + dF1x2;
               dF2 = dF2x1 + dF2x2;

               printf("\n\tdF1 = %.6f", dF1);
               printf("\n\tdF2 = %.6f", dF2);
               getch();
               clrscr();

          }while(fabs(dF1) >= 1 && fabs(dF2) >= 1);
```

```
        printf("\n\n\n\t-----------------------------------------\n");
        printf("\titerasi ke-\tx1\t\tx2\n");
        printf("\t-----------------------------------------\n\n");
        fprintf(x_fxyz,"\n\n\t-----------------------------------------\n");
        fprintf(x_fxyz,"\titerasi ke-\tx1\t\tx2\n");
        fprintf(x_fxyz,"\t-----------------------------------------\n\n");

        do
        {
             i++;

             x1 = -(sqrt(4-pow(x20,2)));
             x2 = 1 - pow(e,x10);

             printf("\t   %d\t\t%.8f\t%.8f\n", i, x1, x2);
             fprintf(x_fxyz,"\t   %d\t\t%.8f\t%.8f\n", i, x1, x2);

             tempx1 = x1;
             x1 = x10;
             x10 = tempx1;

             tempx2 = x2;
             x2 = x20;
             x20 = tempx2;

        }while(x10 != x1 && x20 != x2);

        printf("\n\t-----------------------------------------\n\n");
        fprintf(x_fxyz,"\n\t-----------------------------------------\n\n");

        printf("\nBila x1[0] = %.3f dan x2[0] = %.3f\n\n", x1awal, x2awal);
        printf("\nDiperoleh x1[%d] = %.8f dan x2[%d] = %.8f\n\n", i, x1, i, x2);
        printf("\nDengan iterasi sebanyak %d kali\n", i);
        fprintf(x_fxyz,"\nBila x1[0] = %.3f dan x2[0] = %.3f\n\n", x1awal, x2awal);
        fprintf(x_fxyz,"\nDiperoleh x1[%d] = %.8f dan x2[%d] = %.8f\n\n", i, x1, i,
        x2);
        fprintf(x_fxyz,"\nDengan iterasi sebanyak %d kali\n", i);

        fclose(x_fxyz);
        cout << "\n\nCoba lagi dengan x awal yang berbeda (y/t) ? ";
        cin >> lagi;
        clrscr();

    }while(lagi != 't');

    return 0;
}
```

INTEGRASI NUMERIK

Listing Program C++

```
/* Contoh soal Metode Trapezoidal untuk penyelesaian Integrasi Numerik Bab VI Hal. 50-
51 Diktat Metode Komputasi 2004 */
/* Nama Program : trapezoidal.cpp */

#include <iostream.h>
#include <stdlib.h>
#include <conio.h>
#include <stdio.h>
#include <math.h>

#define MaksU 100

main()
{
     FILE *trapezoidal;
     double    u[MaksU], x[MaksU],
               sum, sumx, x1, x10, x2, h,
               L;
     int i, j, opsi;
     char lagi;

     do
     {
          clrscr();
          trapezoidal = fopen("trapezoidal.txt", "w+");

          cout << "\n\nMetode Trapezoidal untuk Integrasi Numerik\n";
          cout << "==========================================\n\n";

          /*************************************************/
          /* Setiap perubahan persamaan matematika, YAKINKAN  */
          /* bahwa Anda telah merubah persamaan matematika    */
          /* di dalam program ini ------------------------- */
          /*************************************************/

          cout << "\nMasukkan Batas Bawah (x1)   ? ";
          cin >> x1;
          cout << "\nMasukkan Batas Atas (x2)    ? ";
          cin >> x2;
          cout << "\nMasukkan Besar Interval (h) ? ";
          cin >> h;

          cout << "\n\nCara pemasukan data : \n\n";
          cout << "1. Rumus Integral (rubah dulu kode program ssi rumus)\n";
          cout << "Contoh dalam program ini adalah : integral(0-10) x^2 dx\n\n";
          cout << "2. Tabel Data\n\n";
          cout <<"Masukkan pilihan : ";
          cin >> opsi;

          switch(opsi)
          {
               case 1 :
               {
                    cout << "\n\nPersamaan Integral : integral(0-10) x^2 dx\n";
                    cout << "-------------------------------------\n\n";
                    cout << "\nJumlah Data yang akan diproses (lihat tabel) ? ";
                    cin >> j;
```

```
        sumx = 0.0; /* Inisialisasi Penyimpan Sementara Penjumlahan */
        x10 = x1;   /* No urut x sesuai interval + h */

        printf("\t\n\nh = %.2f", h);

        printf("\n\n\t-----------------------------------------\n");
        printf("\tdata ke-\tx\tU(x)\tSum U(x)\n");
        printf("\t-----------------------------------------\n\n");
        fprintf(trapezoidal,"\n\t---------------------------------------
        ---\n");
        fprintf(trapezoidal,"\tdata ke-\tx\tU(x)\tSum U(x)\n");
        fprintf(trapezoidal,"\t---------------------------------------
        -\n\n");

        for(i = 0; i < j; i++)
        {
             u[i] = pow(i,2);
             sumx += u[i];
             sum = sumx - (u[j-j] + u[j-1]);
             x[i] = x1;

             printf("\t   %d\t\t%.1f\t%.2f\t%.2f\n", i, x[i], u[i],
             sumx);
             fprintf(trapezoidal, "\t   %d\t\t%.1f\t%.2f\t%.2f\n", i,
             x[i], u[i], sumx);
             x1 += h;

             /* Rumus Integrasi Trapezoidal */
             L = h*(u[j-j] + 2*sum + u[j-1])/2;
        }

        printf("\n\t-----------------------------------------\n\n");
        fprintf(trapezoidal,"\n\t---------------------------------------
        ---\n\n");

        printf("\nDengan batas bawah = %.2f, atas = %.2f dan interval
        %.2f\n", x10, x2, h);
        fprintf(trapezoidal,"\nDengan batas bawah = %.2f, atas = %.2f dan
        interval %.2f\n", x10, x2, h);
        printf("\nDiperoleh Luas daerah kurva L adalah %.2f", L);
        fprintf(trapezoidal,"\nDiperoleh Luas daerah kurva L adalah
        %.2f", L);

        fclose(trapezoidal);
        break;
   }

   case 2 :
   {
        cout << "\n\nPemasukan Data Berdasarkan Tabel Data\n";
        cout << "-------------------------------------\n";
        cout << "\nBerapa Jumlah Data yang akan diproses (lihat tabel) ?
        ";
        cin >> j;

        for(i = 0; i < j; i++)
        {
             printf("\nu[%d] = ", i);
             cin >> u[i];
        }

        sumx = 0.0; /* Inisialisasi Penyimpan Sementara Penjumlahan */
        x10 = x1;   /* No urut x sesuai interval + h */

        printf("\n\n\th = %.2f", h);

        printf("\n\n\t-----------------------------------------\n");
        printf("\tdata ke-\tx\tU(x)\tSum U(x)\n");
        printf("\t-----------------------------------------\n\n");
        fprintf(trapezoidal,"\n\t---------------------------------------
        ---\n");
        fprintf(trapezoidal,"\tdata ke-\tx\tU(x)\tSum U(x)\n");
        fprintf(trapezoidal,"\t---------------------------------------
        -\n\n");
```

```
                    for(i = 0; i < j; i++)
                    {
                         sumx += u[i];
                         sum = sumx - (u[j-j] + u[j-1]);
                         x[i] = x1;


                         printf("\t   %d\t\t%.1f\t%.4f\t%.4f\n", i, x[i], u[i],
                         sumx);
                         fprintf(trapezoidal, "\t   %d\t\t%.1f\t%.4f\t%.4f\n", i,
                         x[i], u[i], sumx);
                         x1 += h;

                         /* Rumus Integrasi Trapezoidal */
                         L = h*(u[j-j] + 2*sum + u[j-1])/2;
                    }

                    printf("\n\t----------------------------------------\n\n");
                    fprintf(trapezoidal,"\n\t----------------------------------------
                    ---\n\n");

                    printf("\nDengan batas bawah = %.2f, atas = %.2f dan interval
                    %.2f\n", x10, x2, h);
                    fprintf(trapezoidal,"\nDengan batas bawah = %.2f, atas = %.2f dan
                    interval %.2f\n", x10, x2, h);
                    printf("\nDiperoleh Luas daerah kurva L adalah %.4f", L);
                    fprintf(trapezoidal,"\nDiperoleh Luas daerah kurva L adalah
                    %.4f", L);

                    fclose(trapezoidal);
                    break;
               }
          default :  cout << "\nNo Good ......... Try Again, please ..\n";
          }

          cout << "\n\nCoba lagi dengan data yang berbeda (y/t) ? ";
          cin >> lagi;

     }while(lagi != 't');

     return 0;
}
```

```
/* Contoh soal Metode Simpson 1/3 untuk penyelesaian Integrasi Numerik. Bab VI Hal.
52-53 Diktat Metode Komputasi 2004 */
/* Nama Program : simpson1_3.cpp */

#include <iostream.h>
#include <stdlib.h>
#include <conio.h>
#include <stdio.h>
#include <math.h>

#define MaksU 100

main()
{
     FILE *simpson1_3;
     double    u[MaksU], x[MaksU],
               h, x1, x2,
               L, L1, L2;
     int i, j, k, opsi;
     char lagi;

     do
     {
          clrscr();
          simpson1_3 = fopen("simpson1_3.txt", "w+");

          cout << "\n\nMetode Simpson 1/3 untuk Integrasi Numerik\n";
          cout << "=========================================\n";

          /************************************************/
          /* Setiap perubahan persamaan matematika, YAKINKAN */
          /* bahwa Anda telah merubah persamaan matematika   */
          /* di dalam program ini ------------------------- */
          /************************************************/

          cout << "\nMasukkan Batas Bawah (x1)   ? ";
          cin >> x1;
          cout << "\nMasukkan Batas Atas (x2)    ? ";
          cin >> x2;
          cout << "\nMasukkan Besar Interval (h) ? ";
          cin >> h;

          cout << "\n\nCara pemasukan data : \n\n";
          cout << "1. Rumus Integral (rubah dulu kode program ssi rumus)\n";
          cout << "Contoh dalam program ini adalah : integral(0-10) x^2 dx\n\n";
          cout << "2. Tabel Data\n\n";
          cout <<"Masukkan pilihan : ";
          cin >> opsi;

          switch(opsi)
          {
               case 1 :
               {
                    cout << "\n\nPersamaan Integral : integral(0-10) x^2 dx\n\n";

                    j = 0;

                    while(x1 != x2+h)
                    {
                         u[j] = pow(x1,2);
                         x[j] = x1;
                         x1 += h;
                         printf("\nu[%d] = %.2f", j, u[j]);
                         j++;
                    }


                    break;
               }
```

```
      case 2 :
      {
            cout << "\n\nPemasukan Data Berdasarkan Tabel Data\n";
            cout << "-------------------------------------\n";

            j = 0;

            while(x1 != x2+h)
            {
                  printf("\nx[%.1f] = ", x1);
                  cin >> u[j];
                  x[j] = x1;
                  x1 += h;
                  j++;
            }
            break;
      }
default :  cout << "\nTry Again .....\n\n";
}

printf("\n\nu[%d] = %.2f", j-j, u[j-j]);
printf("\tu[%d] = %.2f", j-1, u[j-1]);
printf("\th = %.2f", h);

printf("\n\n\t-------------------------------\n");
printf("\tdata ke-\tx\tU(x)\n");
printf("\t-------------------------------\n\n");
fprintf(simpson1_3,"\n\t-------------------------------\n");
fprintf(simpson1_3,"\tdata ke-\tx\tU(x)\n");
fprintf(simpson1_3,"\t-------------------------------\n\n");

//Tampilkan data x dan U(x)
for(i = 0; i < j; i++)
{
      printf("\t   %d\t\t%.1f\t%.2f\n", i, x[i], u[i]);
      fprintf(simpson1_3,"\t   %d\t\t%.1f\t%.2f\n", i, x[i], u[i]);
}

printf("\n\t-------------------------------\n\n");
fprintf(simpson1_3,"\n\t-------------------------------\n\n");
cout << "\nTekan Enter untuk melihat hasil perhitungan ......\n";
getch();

L1 = 0.0;
L2 = 0.0;

printf("\n\n\t---------------------------------------\n");
printf("\titerasi ke-\tU(x)\tL1(x)\tL2(x)\n");
printf("\t---------------------------------------\n\n");
fprintf(simpson1_3,"\n\t---------------------------------------\n");
fprintf(simpson1_3,"\titerasi ke-\tU(x)\tL1(x)\tL2(x)\n");
fprintf(simpson1_3,"\t---------------------------------------\n\n");

for(i = 1; i <= j-2; i++)
{
      k = fmod(i,2);
      if(k != 0)
      {
            L1 += u[i];
            printf("\t\t%d\t%.2f\t%.2f\t%.2f\n", i, u[i], L1, L2);
            fprintf(simpson1_3,"\t\t%d\t%.2f\t%.2f\t%.2f\n", i, u[i], L1,L2);
      }


      else
      {
            L2 += u[i];
            printf("\t\t%d\t%.2f\t%.2f\t%.2f\n", i, u[i], L1, L2);
            fprintf(simpson1_3,"\t\t%d\t%.2f\t%.2f\t%.2f\n", i, u[i], L1,L2);
      }

      /* Rumus Integrasi Simpson 1/3 */
      L = h*(u[j-j] + 4*L1 + 2*L2 + u[j-1])/3;
}
```

```
            printf("\n\t---------------------------------------\n\n");
            fprintf(simpson1_3,"\n\t---------------------------------------\n\n");

            printf("\nMaka Luas daerah kurva L adalah %.2f", L);
            fprintf(simpson1_3,"\nMaka Luas daerah kurva L adalah %.2f", L);
            fclose(simpson1_3);

            cout << "\n\nCoba lagi dengan data awal yang berbeda (y/t) ? ";
            cin >> lagi;

      }while(lagi != 't');

      return 0;
}
```

```
/* Contoh soal Metode Simpson 1/3 untuk penyelesaian Integrasi Numerik. Bab VI Hal. 58
Diktat Metode Komputasi 2004 */
/* Nama Program : simpson1_3_2c.cpp */

#include <iostream.h>
#include <stdlib.h>
#include <conio.h>
#include <stdio.h>
#include <math.h>

#define MaksU 100

main()
{
     FILE *simpson1_3;
     double    u[MaksU], x[MaksU],
               h, x1, x2,
               L, L1, L2;
     int i, j, k, opsi;
     char lagi;

     do
     {
          clrscr();
          simpson1_3 = fopen("simpson1_3_2c.txt", "w+");

          cout << "\n\nMetode Simpson 1/3 untuk Integrasi Numerik\n";
          cout << "==========================================\n";

          /************************************************/
          /* Setiap perubahan persamaan matematika, YAKINKAN */
          /* bahwa Anda telah merubah persamaan matematika   */
          /* di dalam program ini ------------------------- */
          /************************************************/

          cout << "\nMasukkan Batas Bawah (x1)           ? ";
          cin >> x1;
          cout << "\nMasukkan Batas Atas (x2)            ? ";
          cin >> x2;
          cout << "\nMasukkan Besar Interval (h) min. 0.5 ? ";
          cin >> h;

          cout << "\n\nCara pemasukan data : \n";
          cout << "\n1. Rumus Integral (rubah dulu kode program ssi rumus)";
          cout << "\n2. Tabel Data\n\n";
          cout <<"Masukkan pilihan : ";
          cin >> opsi;

          switch(opsi)
          {
               case 1 :
               {
                    cout << "\n\nPersamaan Integral : integral(1-5) x^3/(1 + x^2)
                    dx";
                    cout << "\n  ---------------------------------------------------
                    \n\n";

                    j = 0;

                    while(x1 != x2+h)
                    {
                         u[j] = pow(x1,3)/(1 + pow(x1,2));
                         x[j] = x1;
                         x1 += h;
                         printf("\nu[%d] = %.4f", j, u[j]);
                         j++;
                    }
                    break;
               }
```

```
      case 2 :
      {
            cout << "\n\nMasukkan data ....\n";

            j = 0;

            while(x1 != x2+h)
            {
                  printf("\nx[%.1f] = ", x1);
                  cin >> u[j];
                  x[j] = x1;
                  x1 += h;
                  j++;
            }
            break;
      }
default :  cout << "\nTry Again .....\n\n";
}

printf("\n\nu[%d] = %.2f", j-j, u[j-j]);
printf("\tu[%d] = %.2f", j-1, u[j-1]);
printf("\th = %.2f", h);

printf("\n\n\t-------------------------------\n");
printf("\tdata ke-\tx\tU(x)\n");
printf("\t-------------------------------\n\n");
fprintf(simpson1_3,"\n\t-------------------------------\n");
fprintf(simpson1_3,"\tdata ke-\tx\tU(x)\n");
fprintf(simpson1_3,"\t-------------------------------\n\n");

//Tampilkan data x dan U(x)
for(i = 0; i < j; i++)
{
      printf("\t   %d\t\t%.1f\t%.4f\n", i, x[i], u[i]);
      fprintf(simpson1_3,"\t   %d\t\t%.1f\t%.4f\n", i, x[i], u[i]);
}

printf("\n\t-------------------------------\n\n");
fprintf(simpson1_3,"\n\t-------------------------------\n\n");
cout << "\nTekan Enter untuk melihat hasil perhitungan ......\n";
getch();

L1 = 0.0;
L2 = 0.0;

printf("\n\n\t---------------------------------------\n");
printf("\titerasi ke-\tU(x)\tL1(x)\tL2(x)\n");
printf("\t---------------------------------------\n\n");
fprintf(simpson1_3,"\n\t---------------------------------------\n");
fprintf(simpson1_3,"\titerasi ke-\tU(x)\tL1(x)\tL2(x)\n");
fprintf(simpson1_3,"\t---------------------------------------\n\n");

for(i = 1; i <= j-2; i++)
{
      k = fmod(i,2);
      if(k != 0)
      {
            L1 += u[i];
            printf("\t\t%d\t%.2f\t%.2f\t%.2f\n", i, u[i], L1, L2);
            fprintf(simpson1_3,"\t\t%d\t%.2f\t%.2f\t%.2f\n", i, u[i], L1,L2);
      }
      Else


      {
            L2 += u[i];
            printf("\t\t%d\t%.2f\t%.2f\t%.2f\n", i, u[i], L1, L2);
            fprintf(simpson1_3,"\t\t%d\t%.2f\t%.2f\t%.2f\n", i, u[i], L1,L2);
      }

      /* Rumus Integrasi Simpson 1/3 */
      L = h*(u[j-j] + 4*L1 + 2*L2 + u[j-1])/3;
}
```

```
            printf("\n\t---------------------------------------\n\n");
            fprintf(simpson1_3,"\n\t---------------------------------------\n\n");


            printf("\nMaka Luas daerah kurva L adalah %.4f", L);
            fprintf(simpson1_3,"\nMaka Luas daerah kurva L adalah %.4f", L);
            fclose(simpson1_3);

            cout << "\n\nCoba lagi dengan data awal yang berbeda (y/t) ? ";
            cin >> lagi;

      }while(lagi != 't');

      return 0;
}
```

```
/* Contoh soal Metode Simpson 3/8 untuk penyelesaian Integrasi Numerik Bab VI Hal. 54-
55 Diktat Metode Komputasi 2004 */
/* Nama Program : simpson3_8.cpp */

#include <iostream.h>
#include <stdlib.h>
#include <conio.h>
#include <stdio.h>
#include <math.h>

#define MaksU 100

main()
{
     FILE *simpson3_8;
     double    u[MaksU], x[MaksU],
               h, x1, x2,
               L, L1, L2;
     int i, j, k, n, opsi;
     char lagi;

     do
     {
          clrscr();
          simpson3_8 = fopen("simpson3_8.txt", "w+");

          cout << "\n\nMetode Simpson 3/8 untuk Integrasi Numerik\n";
          cout << "=========================================\n\n";

          /************************************************/
          /* Setiap perubahan persamaan matematika, YAKINKAN */
          /* bahwa Anda telah merubah persamaan matematika   */
          /* di dalam program ini ------------------------- */
          /************************************************/

          cout << "\nMasukkan Batas Bawah (x1)   ? ";
          cin >> x1;
          cout << "\nMasukkan Batas Atas (x2)    ? ";
          cin >> x2;
          cout << "\nMasukkan Besar Interval (h) ? ";
          cin >> h;

          cout << "\n\nCara pemasukan data : \n\n";
          cout << "1. Rumus Integral (rubah dulu kode program ssi rumus)\n";
          cout << "Contoh dalam program ini adalah : integral(0-10) x^2 dx\n\n";
          cout << "2. Tabel Data\n\n";
          cout <<"Masukkan pilihan : ";
          cin >> opsi;

          switch(opsi)
          {
               case 1 :
               {
                    cout << "\n\nPersamaan Integral : integral(0-10) x^2 dx";
                    cout << "\n  ----------------------------------------\n\n";

                    j = 0;

                    while(x1 != x2+h)
                    {
                         u[j] = pow(x1,2);
                         x[j] = x1;
                         x1 += h;
                         printf("\nu[%d] = %.2f", j, u[j]);
                         j++;
                    }

                    break;
               }
```

```
     case 2 :
     {
          cout << "\n\nPemasukan Data Berdasarkan Tabel Data\n";
          cout << "--------------------------------------\n";
          cout << "\nBerapa Jumlah Data yang akan diproses (lihat tabel) ?
          ";
          cin >> n;

          for(i = 0, j = 0; i < n; i++, j++)
          {
               printf("\nx[%.2f] = ", x1);
               cin >> u[j];
               x[j] = x1;
               x1 += h;
          }
          break;
     }
     default :  cout << "\nNo Good ....... Try Again, please ...\n\n";
}

/* Cetak Data Awal */
printf("\n\nu[%d] = %.4f", j-j, u[j-j]);
printf("\tu[%d] = %.4f", j-1, u[j-1]);
printf("\th = %.2f", h);

cout << "\n\nTekan Enter untuk melihat hasil perhitungan ..... \n";
getch();

printf("\n\n\t-------------------------------\n");
printf("\tdata ke-\tx\tU(x)\n");
printf("\t-------------------------------\n\n");
fprintf(simpson3_8,"\n\t-------------------------------\n");
fprintf(simpson3_8,"\tdata ke-\tx\tU(x)\n");
fprintf(simpson3_8,"\t-------------------------------\n\n");

//Tampilkan data x dan U(x)
for(i = 0; i < j; i++)
{
     printf("\t   %d\t\t%.2f\t%.4f\n", i, x[i], u[i]);
     fprintf(simpson3_8,"\t   %d\t\t%.2f\t%.4f\n", i, x[i], u[i]);
}

printf("\n\t-------------------------------\n\n");
fprintf(simpson3_8,"\n\t-------------------------------\n\n");
cout << "\nTekan Enter untuk melihat hasil perhitungan ......\n";
getch();
printf("\n\n\t---------------------------------------\n");
printf("\titerasi ke-\tU(x)\tL1(x)\tL2(x)\n");
printf("\t---------------------------------------\n\n");
fprintf(simpson3_8,"\n\t---------------------------------------\n");
fprintf(simpson3_8,"\titerasi ke-\tU(x)\tL1(x)\tL2(x)\n");
fprintf(simpson3_8,"\t---------------------------------------\n\n");

L1 = 0.0;
L2 = 0.0;

//Perhitungan Simpson 3/8
for(i = 1; i <= j-2; i++)
{
     k = fmod(i,3);
     if(k != 0)
     {
          L1 += u[i];
          printf("\t\t%d\t%.4f\t%.4f\t%.4f\n", i, u[i], L1, L2);
          fprintf(simpson3_8,"\t\t%d\t%.4f\t%.4f\t%.4f\n", i, u[i], L1,
          L2);
     }
     else
     {
          L2 += u[i];
          printf("\t\t%d\t%.4f\t%.4f\t%.4f\n", i, u[i], L1, L2);
          fprintf(simpson3_8,"\t\t%d\t%.4f\t%.4f\t%.4f\n", i, u[i], L1,
          L2);
     }
```

```
                /* Rumus Integrasi Simpson 3/8 */
                L = 3*h*(u[j-j] + 3*L1 + 2*L2 + u[j-1])/8;
           }

           printf("\n\t---------------------------------------\n\n");
           fprintf(simpson3_8,"\n\t---------------------------------------\n\n");

           printf("\nMaka Luas daerah kurva L adalah %.4f", L);
           fprintf(simpson3_8,"\nMaka Luas daerah kurva L adalah %.4f", L);

           fclose(simpson3_8);

           cout << "\n\nCoba lagi dengan data awal yang berbeda (y/t) ? ";
           cin >> lagi;

     }while(lagi != 't');

     return 0;
}
```

```
/* Contoh soal Metode Weddle untuk penyelesaian Integrasi Numerik Bab VI Hal. 55-57
Diktat Metode Komputasi 2004 */
/* Nama Program : weddle.cpp */

#include <iostream.h>
#include <stdlib.h>
#include <conio.h>
#include <stdio.h>
#include <math.h>

#define MaksU 100

main()
{
     FILE *weddle;
     double    u[MaksU], x[MaksU],
               x1, x2, h, L;
     int i, j, k;

     char lagi;

     do
     {
          clrscr();
          weddle = fopen("weddle.txt", "w+");

          cout << "\n\nMetode Weddle untuk Integrasi Numerik\n";
          cout << "=====================================\n\n";
          cout << "Contoh pada Diktat Metode Komputasi\n\n";
          cout << "Persamaan Integral : integral(4-5,2) ln(x) dx\n\n";

          /*************************************************/
          /* Setiap perubahan persamaan matematika, YAKINKAN */
          /* bahwa Anda telah merubah persamaan matematika   */
          /* di dalam program ini -------------------------- */
          /*************************************************/

          cout << "\nMasukkan Batas Bawah (x1) ? ";
          cin >> x1;
          cout << "\nMasukkan Batas Atas (x2)  ? ";
          cin >> x2;
          cout << "\nBerapa kali iterasi (j)   ? ";
          cin >> k;

          j = 6;              /* Aturan Weddle adalah kelipatan 6 */

          h = (x2 - x1)/j;    /* Interval */

          printf("\t\n\nh = %.2f", h);

          //Cetak x dan U(x)
          printf("\n\n\t------------------------\n");
          printf("\t U(x)\tx\tf(x)\n");
          printf("\t------------------------\n");
          fprintf(weddle,"\n\t------------------------\n");
          fprintf(weddle,"\t U(x)\tx\tf(x)\n");
          fprintf(weddle,"\t------------------------\n");

          /* Pemasukan dan Penghitungan Data */
          for(i = 0; i < k; i++)
          {
               u[i] = log(x1);
               x[i] = x1;
               printf("\t   %d\t%.1f\t%.3f\n", i, x[i], u[i]);
               fprintf(weddle,"\t   %d\t%.1f\t%.3f\n", i, x[i], u[i]);


               x1 += h;

               /* Rumus Weddle */
               L = 3*h*(u[i-6] + 5*u[i-5] + u[i-4] + 6*u[i-3] + u[i-2] + 5*u[i-1] +
               u[i])/10;
          }
```

```
            printf("\t------------------------\n");
            fprintf(weddle,"\t------------------------\n");

            printf("\nMaka Luas daerah kurva L adalah %.5f", L);
            fprintf(weddle,"\nMaka Luas daerah kurva L adalah %.5f", L);
            fclose(weddle);

            cout << "\n\nCoba lagi dengan data awal yang berbeda (y/t) ? ";
            cin >> lagi;

  }while(lagi != 't');

  return 0;
}
```

```
/* Contoh soal Metode Weddle untuk penyelesaian Integrasi Numerik Bab VI Hal. 58
Diktat Metode Komputasi 2004 */
/* Nama Program : weddle_4b.cpp */

#include <iostream.h>
#include <stdlib.h>
#include <conio.h>
#include <stdio.h>
#include <math.h>

#define MaksU 100

main()
{
     FILE *weddle;
     double    u[MaksU], x[MaksU],
               x1, x2, h, L;
     int i, j, k;

     char lagi;

     do
     {
          clrscr();
          weddle = fopen("weddle_4b.txt", "w+");

          cout << "\n\nMetode Weddle untuk Integrasi Numerik\n";
          cout << "=====================================\n\n";
          cout << "Soal No.4.b dalam Diktat Metode Komputasi\n\n";
          cout << "Persamaan Integral : integral(1-7) ln(x)/2*x^3 dx\n\n";

          /*************************************************/
          /* Setiap perubahan persamaan matematika, YAKINKAN */
          /* bahwa Anda telah merubah persamaan matematika   */
          /* di dalam program ini -------------------------- */
          /*************************************************/

          cout << "\nMasukkan Batas Bawah (x1)         ? ";
          cin >> x1;
          cout << "\nMasukkan Batas Atas (x2)          ? ";
          cin >> x2;
          cout << "\nBerapa kali iterasi (k)(max. 7)   ? ";
          cin >> k;

          j = 6;              /* Aturan Weddle adalah kelipatan 6 */

          h = (x2 - x1)/j;    /* Interval */

          printf("\t\n\nh = %.2f", h);

          //Cetak x dan U(x)
          printf("\n\n\t------------------------\n");
          printf("\t U(x)\tx\tf(x)\n");
          printf("\t------------------------\n");
          fprintf(weddle,"\n\t------------------------\n");
          fprintf(weddle,"\t U(x)\tx\tf(x)\n");
          fprintf(weddle,"\t------------------------\n");

          /* Pemasukan dan Penghitungan Data */
          for(i = 0; i < k; i++)
          {
               u[i] = log(x1)/(2*pow(x1,3));
               x[i] = x1;
               printf("\t   %d\t%.1f\t%.3f\n", i, x[i], u[i]);
               fprintf(weddle,"\t   %d\t%.1f\t%.3f\n", i, x[i], u[i]);


               x1 += h;

               /* Rumus Weddle */
               L = 3*h*(u[i-6] + 5*u[i-5] + u[i-4] + 6*u[i-3] + u[i-2] + 5*u[i-1] +
               u[i])/10;
          }
```

```
            printf("\t------------------------\n");
            fprintf(weddle,"\t------------------------\n");

            printf("\nMaka Luas daerah kurva L adalah %.5f", L);
            fprintf(weddle,"\nMaka Luas daerah kurva L adalah %.5f", L);
            fclose(weddle);

            cout << "\n\nCoba lagi dengan data awal yang berbeda (y/t) ? ";
            cin >> lagi;

      }while(lagi != 't');

      return 0;
}
```

PENYELESAIAN PERSAMAAN DIFERENSIAL BIASA

Listing Program C++

```
/* Contoh soal Metode Taylor untuk penyelesaian Diferensiasi Numerik Bab VII Hal. 60-
61 Diktat Metode Komputasi 2004 */
/* Nama Program : taylor.cpp */

#include <iostream.h>
#include <stdlib.h>
#include <conio.h>
#include <stdio.h>
#include <math.h>

#include "faktorial.hpp"

#define MaksData 100

main()
{
     FILE *taylor;
     double    y[MaksData],
               xm, x0, yn1, ytemp,
               h;
     int i, m;
     char lagi;

     do
     {
          clrscr();
          taylor = fopen("taylor.txt", "w+");

          cout << "\n\nMetode Taylor untuk Diferensiasi Numerik\n";
          cout << "========================================\n\n";
          cout << "Contoh pada Diktat Metode Komputasi\n\n";
          cout << "Persamaan Diferensial : f(x,y) = dy/dx = x + y\n";
          cout << "----------------------------------------------\n\n";

          /************************************************/
          /* Setiap perubahan persamaan matematika, YAKINKAN */
          /* bahwa Anda telah merubah persamaan matematika   */
          /* di dalam program ini ------------------------- */
          /************************************************/

          cout << "\nTitik x yang dicari (xm)         ? ";
          cin >> xm;
          cout << "\nTitik x awal (x0)                ? ";
          cin >> x0;
          cout << "\nJumlah Deret yang diperlukan (m) ? ";
          cin >> m;

          ytemp = 0.0;
          h = xm - x0;                              /* Interval */

          printf("\n\nh = %.2f\n\n", h);

          /* Pemasukan Nilai Deret hasil Diferensiasi */
          for(i = 0; i < m; i++)
          {
               printf("\ny(%d) = ", i+1);
               cin >> y[i];                      /* Pemasukan Data */
          }
```

```
          /* Cetak y[i] serta penghitungan Taylor */
          printf("\n\n\t----------------------\n");
          printf("\tdata ke-\ty[i]\n");
          printf("\t----------------------\n");
          fprintf(taylor,"\n\t----------------------\n");
          fprintf(taylor,"\tData ke-\ty[i]\n");
          fprintf(taylor,"\t----------------------\n");


          /* Penghitungan Data */
          for(i = 0; i < m; i++)
          {
               /* Rumus Diferensiasi Taylor */
               ytemp += (pow(h,i+1)*y[i])/factorial(i+1);
               yn1 = y[i-i] + ytemp;

               printf("\t   %d\t\t%.5f\n", i+1, y[i]);
               fprintf(taylor,"\t   %d\t\t%.5f\n", i+1, y[i]);
          }

          printf("\t----------------------\n");
          fprintf(taylor,"\t----------------------\n");
          printf("\n\nMaka y[%.1f] = %.7f", xm, yn1);
          fprintf(taylor,"\nMaka y[%.1f] = %.7f", xm, yn1);
          fclose(taylor);

          cout << "\n\n\nCoba lagi dengan data awal yang berbeda (y/t) ? ";
          cin >> lagi;

     }while(lagi != 't');

     return 0;
}
```

```
/* Header untuk program taylor.cpp */
/* Nama Header : faktorial.hpp */

int factorial(int);

int factorial(int x)
{
  int i, p=1;
  for(i=1; i<=x; i++)
  {
   p*=i;
  }
  return(p);
}
```

```
/* Contoh soal Metode Euler untuk penyelesaian Diferensiasi Numerik Bab VII Hal. 62-63
Diktat Metode Komputasi 2004 */
/* Nama Program : euler.cpp */

#include <iostream.h>
#include <stdlib.h>
#include <conio.h>
#include <stdio.h>
#include <math.h>

#define MaksData 100

main()
{
     FILE *euler;
     double    x[MaksData], y[MaksData],
               dx, dy[MaksData],
               xn, x0, xtemp,
               y0, y00, ytemp;
     int i, n;
     char lagi;

     do
     {
          clrscr();
          euler = fopen("euler.txt", "w+");

          cout << "\n\nMetode Euler untuk Diferensiasi Numerik\n";
          cout << "=======================================\n\n";
          cout << "Contoh pada Diktat Metode Komputasi\n\n";
          cout << "Persamaan Diferensial : f(x,y) = dy/dx = (y - x)/(y + x)\n";
          cout << "-----------------------------------------------------\n\n";

          /************************************************/
          /* Setiap perubahan persamaan matematika, YAKINKAN */
          /* bahwa Anda telah merubah persamaan matematika   */
          /* di dalam program ini ------------------------- */
          /************************************************/

          /* Pemasukan Data yang diberikan */
          cout << "\nMasukkan x[0] ? ";
          cin >> x0;
          cout << "\nMasukkan y[0] ? ";
          cin >> y0;
          cout << "\nNilai x yang dicari ? ";
          cin >> xn;
          cout << "\nBerapa kali Iterasi ? ";
          cin >> n;

          dx = (xn - x0)/n;

          printf("\n\n\tx[0] = %.2f", x0);
          printf("\n\n\ty[0] = %.2f", y0);
          printf("\n\n\tYang dicari adalah y[%.2f]", xn);
          printf("\n\n\th = %.2f", dx);
          printf("\n\n\tIterasi (n) = %d  kali", n);

          /* Cetak x[i], y[i], dx[i] dan dy[i] serta penghitungan Euler */
          printf("\n\n\t-------------------------------------------------------\n");
          printf("\tIterasi\t\tx[i]\tdy[i]\t\tdx[i]\ty[i]\n");
          printf("\t-------------------------------------------------------\n");
          fprintf(euler,"\n\t-------------------------------------------------------
          \n");
          fprintf(euler,"\tIterasi\t\tx[i]\tdy[i]\t\tdx[i]\ty[i]\n");
          fprintf(euler,"\t-------------------------------------------------------
          \n");

          /* Pemasukan dan Penghitungan Data */
          for(i = 1; i <= n; i++)
          {
               /* Rumus Diferensiasi Euler */
               dy[i] = ((y0 - x0)/(y0 + x0)) * dx;
               y[i] = y0 + dy[i];
               x[i] = x0 + dx;
```

```
            printf("\t   %d\t\t%.2f\t%.5f\t\t%.2f\t%.5f\n", i, x[i-1], dy[i], dx,
            y[i]);
            fprintf(euler,"\t   %d\t\t%.2f\t%.5f\t\t%.2f\t%.5f\n", i, x[i-1],
            dy[i], dx, y[i]);

            ytemp = y[i];
            y0 = ytemp;

            xtemp = x[i];
            x0 = xtemp;
        }

        printf("\t-------------------------------------------------------\n");
        fprintf(euler,"\t-------------------------------------------------------
        \n");
        fprintf(euler,"\n\nx[%d] = %.2f", i-n-1, x[i-n-1]);
        fprintf(euler,"\n\nYang dicari adalah y[%.2f]", xn);
        fprintf(euler,"\n\nh = %.2f", dx);
        fprintf(euler,"\n\nIterasi (n) = %d  kali", n);
        printf("\n\nMaka y[%.2f] = %.5f", xn, y0);
        fprintf(euler,"\n\nMaka y[%.2f] = %.5f", xn, y0);
        fclose(euler);

        cout << "\n\nCoba lagi dengan data awal atau rumus yang berbeda (y/t) ? ";
        cin >> lagi;

    }while(lagi != 't');

    return 0;
}
```

```
/* Contoh soal Metode Modified Euler untuk penyelesaian Diferensiasi Numerik Bab VII
Hal. 64-65 Diktat Metode Komputasi 2004 */
/* Nama Program : modified_euler.cpp */

#include <iostream.h>
#include <stdlib.h>
#include <conio.h>
#include <stdio.h>
#include <math.h>

#define MaksData 100

main()
{
     FILE *modified_euler;
     double    y[MaksData], x[MaksData],
               xn, x0,
               yn1, y0, ytemp,
               h;
     int  i,
          j = 0,  /* inisialisasi counter loop xn */
          n = 1;  /* inisialisasi counter loop y[n] dan x[n] */
     char lagi;

     do
     {
          clrscr();
          modified_euler = fopen("modified_euler.txt", "w+");

          cout << "\n\nMetode Modified Euler untuk Diferensiasi Numerik\n";
          cout << "================================================\n\n";
          cout << "Contoh pada Diktat Metode Komputasi\n\n";
          cout << "Persamaan Diferensial : f(x,y) = dy/dx = y + x\n";
          cout << "------------------------------------------------\n\n";

          /*************************************************/
          /* Setiap perubahan persamaan matematika, YAKINKAN */
          /* bahwa Anda telah merubah persamaan matematika   */
          /* di dalam program ini ------------------------- */
          /*************************************************/

          /* Pemasukan Data yang diberikan */
          cout << "\nNilai x yang dicari   ? ";
          cin >> xn;
          cout << "\nMasukkan x[" << j << "] = ";
          cin >> x0;
          cout << "\nMasukkan y[" << j << "] = ";
          cin >> y0;
          cout << "\nMasukkan Interval (h) ? ";
          cin >> h;

          do
          {
               printf("\n\n\tx[%d] = %.2f", j, x0);
               printf("\n\n\ty[%d] = %.10f", j, y0);
               printf("\n\n\tYang dicari adalah y[%.2f]", xn);
               printf("\n\n\th = %.2f", h);

               /* Cetak x[i] dan y[i] serta penghitungan Modified Euler */
               fprintf(modified_euler,"\n\nProses ke-%d\n\n", n);
               printf("\n\n\t-----------------------------------\n");
               printf("\tIterasi\t\tx[i]\ty[i]\n");
               printf("\t-----------------------------------\n");
               fprintf(modified_euler,"\n\t-----------------------------------\n");
               fprintf(modified_euler,"\tIterasi\t\tx[i]\ty[i]\n");
               fprintf(modified_euler,"\t-----------------------------------\n");

               y[n] = y0 + h*(y0 + x0);
               x[n] = x0 + h;
```

```
                i = 0;    /* inisialisasi counter loop Euler */

                /* Pemasukan dan Penghitungan Data */
                do
                {
                     /* Rumus Diferensiasi Modified Euler */
                     yn1 = y0 + h*((y0 + x0) + (y[n] + x[n]))/2;

                     printf("\t   %d\t\t%.2f\t%.6f\n", i+1, x[n], yn1);
                     fprintf(modified_euler,"\t   %d\t\t%.2f\t%.6f\n", i+1, x[n],
                     yn1);

                     ytemp = yn1;
                     yn1 = y[n];
                     y[n] = ytemp;

                     i++;

                }while(y[n] != yn1);

                printf("\t-----------------------------------\n");
                fprintf(modified_euler,"\t-----------------------------------\n");

                printf("\nx(%d) = %.6f\n", n, x[n]);
                printf("\n\ny(%d) = %.6f\n", n, y[n]);
                printf("\n\nMaka y[%.2f] = %.6f\n\n", x[n], yn1);
                fprintf(modified_euler,"\nx(%d) = %.6f\n", n, x[n]);
                fprintf(modified_euler,"\n\ny(%d) = %.6f\n", n, y[n]);
                fprintf(modified_euler,"\n\nMaka y[%.2f] = %.6f\n\n", x[n], yn1);

                y0 = yn1;
                x0 = x[n];

                j++;
                n++;

                cout << "\n\nTekan Enter untuk lanjutkan iterasi nilai berikutnya ....
                ";
                getch();

           }while(x0 != xn);

           cout << "\n\n\nCoba lagi dengan data atau rumus berbeda (y/t) ? ";
           cin >> lagi;

           j = 0;
           n = 0;

           fclose(modified_euler);

      }while(lagi != 't');

      return 0;
}
```

```
/* Contoh soal Metode Runge-Kutta orde 4 untuk penyelesaian Diferensiasi Numerik Bab
VII Hal. 67-68 Diktat Metode Komputasi 2004 */
/* Nama Program : runge-kutta4.cpp */

#include <iostream.h>
#include <stdlib.h>
#include <conio.h>
#include <stdio.h>
#include <math.h>

#define MaksData 100

main()
{
     FILE *runge_kutta4;
     double    k1[MaksData], k2[MaksData],
               k3[MaksData], k4[MaksData],
               x, x0, y0, yn1, ytemp,
               h;
     int i;
     char lagi;

     do
     {
          clrscr();
          runge_kutta4 = fopen("runge_kutta4.txt", "w+");

          cout << "\n\nMetode Runge-Kutta Orde 4 untuk Diferensiasi Numerik\n";
          cout << "==================================================\n\n";
          cout << "Contoh pada Diktat Metode Komputasi\n\n";
          cout << "Persamaan Diferensial : f(x,y) = dy/dx = y + x\n";
          cout << "-------------------------------------------\n\n";

          /************************************************/
          /* Setiap perubahan persamaan matematika, YAKINKAN */
          /* bahwa Anda telah merubah persamaan matematika   */
          /* di dalam program ini ------------------------- */
          /************************************************/

          /* Pemasukan Data yang diberikan */
          cout << "\nMasukkan x[0] ? ";
          cin >> x0;
          cout << "\nMasukkan y[0] ? ";
          cin >> y0;
          cout << "\nNilai titik x yang dicari ? ";
          cin >> x;
          cout << "\nInterval h  ? ";
          cin >> h;

          printf("\n\n\tx[0] = %.2f", x0);
          printf("\n\n\ty[0] = %.2f", y0);
          printf("\n\n\tYang dicari adalah y[%.2f]", x);
          printf("\n\n\th = %.2f", h);

          /* Cetak x[i], y[i] dan f[i] serta penghitungan Runge-Kutta Orde 4 */
          printf("\n\n\t-------------------------------------------------------\n");
          printf("\tIterasi\t\tk1\tk2\tk3\tk4\ty[i]\n");
          printf("\t-------------------------------------------------------\n");
          fprintf(runge_kutta4,"\n\t-------------------------------------------------
          -----\n");
          fprintf(runge_kutta4,"\tIterasi\t\tk1\tk2\tk3\tk4\ty[i]\n");
          fprintf(runge_kutta4,"\t-------------------------------------------------
          ---\n");

          i = 0;

          /* Pemasukan dan Penghitungan Data */
          while(x0 != x)//for(i = 0; i < n; i++)
          {
               k1[i] = y0 + x0;
               k2[i] = (y0 + h*k1[i]/2) + (x0 + h/2);
               k3[i] = (y0 + h*k2[i]/2) + (x0 + h/2);
               k4[i] = (y0 + h*k3[i]) + (x0 + h);
```

```
            /* Rumus Diferensiasi Runge-Kutta Orde 4 */
            yn1 = y0 + h*(k1[i] + 2*k2[i] + 2*k3[i] + k4[i])/6;
            x0 += h;

            printf("\t   %d\t\t%.5f\t%.5f\t%.5f\t%.5f\t%.5f\n", i+1, k1[i], k2[i],
            k3[i], k4[i], yn1);
            fprintf(runge_kutta4,"\t   %d\t\t%.5f\t%.5f\t%.5f\t%.5f\t%.5f\n", i+1,
            k1[i], k2[i], k3[i], k4[i], yn1);

            ytemp = yn1;
            y0 = ytemp;

            i++;           /* counter */
        }

        printf("\t-------------------------------------------------------\n");
        fprintf(runge_kutta4,"\t-------------------------------------------------------
        ---\n");
        fprintf(runge_kutta4,"\n\nx[0] = %.2f", x0);
        fprintf(runge_kutta4,"\n\ny[0] = %.2f", y0);
        fprintf(runge_kutta4,"\n\nYang dicari adalah y[%.2f]", x);
        fprintf(runge_kutta4,"\n\nh = %.2f", h);
        printf("\n\nMaka y[%.2f] = %.5f", x, y0);
        fprintf(runge_kutta4,"\n\nMaka y[%.2f] = %.5f", x, y0);
        fclose(runge_kutta4);

        cout << "\n\nCoba lagi dengan data awal atau rumus yang berbeda (y/t) ? ";
        cin >> lagi;

    }while(lagi != 't');

    return 0;
}
```

```
/* Contoh soal Metode Adam untuk penyelesaian Diferensiasi Numerik Bab VII Hal. 69-70
Diktat Metode Komputasi 2004 */
/* Nama Program : adam.cpp */

#include <iostream.h>
#include <stdlib.h>
#include <conio.h>
#include <stdio.h>
#include <math.h>

#define MaksU 100

main()
{
     FILE *adam;
     double    x[MaksU],
               y[MaksU],
               f[MaksU],
               xn1, yn1, fn1,
               h;
     int i, n;
     char lagi;

     do
     {
          clrscr();
          adam = fopen("adam.txt", "w+");

          cout << "\n\nMetode Adam untuk Diferensiasi Numerik\n";
          cout << "======================================\n\n";
          cout << "Contoh pada Diktat Metode Komputasi\n\n";
          cout << "Persamaan Diferensial : f(x,y) = dy/dx = x + y\n";
          cout << "----------------------------------------------\n\n";

          /************************************************/
          /* Setiap perubahan persamaan matematika, YAKINKAN */
          /* bahwa Anda telah merubah persamaan matematika   */
          /* di dalam program ini ------------------------- */
          /************************************************/

          cout << "\nTitik x dimana nilai fungsi f(x,y) = y yang dicari ? ";
          cin >> xn1;
          cout << "\nBerapa Jumlah Data yang akan diproses             ? ";
          cin >> n;

          /* Pemasukan Data sesuai Tabel yang diberikan */
          for(i = 0; i < n; i++)
          {
               printf("\n(x[%d], y[%d]) = ", i+1, i+1);
               cin >> x[i] >> y[i];                  /* Pemasukan Data */
               f[i] = x[i] + y[i];                   /* Nilai f(x,y) */
               h = x[i] - x[i-1];                    /* Interval */
          }

          printf("\n\n\th = %.2f", h);

          /* Cetak x[i], y[i] dan f[i] serta penghitungan Adam */
          printf("\n\n\t--------------------------------------\n");
          printf("\tdata ke-\tx[i]\ty[i]\tf[i]\n");
          printf("\t--------------------------------------\n");
          fprintf(adam,"\n\t--------------------------------------\n");
          fprintf(adam,"\tData ke-\tx[i]\ty[i]\tf[i]\n");
          fprintf(adam,"\t--------------------------------------\n");

          /* Pemasukan dan Penghitungan Data */
          for(i = 0; i < n; i++)
          {
               printf("\t   %d\t\t%.1f\t%.5f\t%.5f\n", i+1, x[i], y[i], f[i]);
               fprintf(adam,"\t   %d\t\t%.1f\t%.5f\t%.5f\n", i+1, x[i], y[i], f[i]);

               /* Rumus Diferensiasi Adam */
               yn1 = y[i] + h*(55*f[i] - 59*f[i-1] + 37*f[i-2] - 9*f[i-3])/24;
               fn1 = xn1 + yn1;
          }
```

```
            printf("\t---------------------------------------\n");
            fprintf(adam,"\t---------------------------------------\n");
            printf("\n\nf[%.1f] = %.5f", xn1, fn1);
            printf("\n\nMaka y[%.1f] = %.5f", xn1, yn1);
            fprintf(adam,"\nMaka y[%.1f] = %.5f", xn1, yn1);
            fclose(adam);

            cout << "\n\nCoba lagi dengan data awal yang berbeda (y/t) ? ";
            cin >> lagi;

      }while(lagi != 't');

      return 0;
}
```

```
/* Contoh soal Metode Adam-Moulton untuk penyelesaian Diferensiasi Numerik Bab VII
Hal. 72-73 Diktat Metode Komputasi 2004 */
/* Nama Program : adam-moulton.cpp */

#include <iostream.h>
#include <stdlib.h>
#include <conio.h>
#include <stdio.h>
#include <math.h>

#define MaksData 100

main()
{
     FILE *adam_moulton;
     double    x[MaksData],
               y[MaksData],
               f[MaksData],
               h, xn1, yn1, fn1;
     int i, n;
     char lagi;

     do
     {
          clrscr();
          adam_moulton = fopen("adam_moulton.txt", "w+");

          cout << "\n\nMetode Adam-Moulton untuk Diferensiasi Numerik\n";
          cout << "==============================================\n\n";
          cout << "Contoh pada Diktat Metode Komputasi\n\n";
          cout << "Persamaan Diferensial : f(x,y) = dy/dx = x + y\n\n";

          /************************************************/
          /* Setiap perubahan persamaan matematika, YAKINKAN */
          /* bahwa Anda telah merubah persamaan matematika   */
          /* di dalam program ini ------------------------- */
          /************************************************/

          cout << "\nBerapa Jumlah Data yang akan diproses   ? ";
          cin >> n;
          cout << "\nData yang akan diketahui x(n+1)         ? ";
          cin >> xn1;

          /* Pemasukan Data x[i] dan y[i] serta penghitungan h dan f[i] */
          for(i = 0; i < n; i++)
          {
               printf("\nx[%d] dan y[%d] = ", i+1, i+1);
               cin >> x[i] >> y[i];
               f[i] = x[i] + y[i];
               h = x[i] - x[i-1];
          }
          printf("\n\nh = %.1f", h);

          /* Cetak x[i], y[i] dan f[i] */
          printf("\n\n\t----------------------------------------\n");
          printf("\tData ke-\tx[i]\ty[i]\tf[i]\n");
          printf("\t----------------------------------------\n");
          fprintf(adam_moulton,"\n\t----------------------------------------\n");
          fprintf(adam_moulton,"\tData ke-\tx[i]\ty[i]\tf[i]\n");
          fprintf(adam_moulton,"\t----------------------------------------\n");

          /* Cetak semua parameter dan lakukan perhitungan diferensial y[n+1] */
          for(i = 0; i < n; i++)
          {
               printf("\t   %d\t\t%.1f\t%.5f\t%.5f\n", i+1, x[i], y[i], f[i]);
               fprintf(adam_moulton,"\t   %d\t\t%.1f\t%.5f\t%.5f\n", i+1, x[i], y[i],
               f[i]);

               /* Rumus Prediktif Adam-Moulton */
               yn1 = y[i] + h*(55*f[i] - 59*f[i-1] + 37*f[i-2] - 9*f[i-3])/24;
               fn1 = yn1 + xn1;

               /* Rumus Korektif Adam-Moulton */
               yn1 = y[i] + h*(9*fn1 + 19*f[i] - 5*f[i-1] + f[i-2])/24;
          }
```

```
            printf("\t----------------------------------------\n");
            fprintf(adam_moulton,"\t----------------------------------------\n");
            printf("\n\nMaka y(%.1f) = %.5f", xn1, yn1);
            fprintf(adam_moulton,"\n\nMaka y(%.1f) = %.5f", xn1, yn1);
            fclose(adam_moulton);

            cout << "\n\nCoba lagi dengan data awal yang berbeda (y/t) ? ";
            cin >> lagi;

      }while(lagi != 't');

      return 0;
}
```

```
/* Contoh soal Metode Milne untuk penyelesaian Diferensiasi Numerik Bab VII Hal. 70-71
Diktat Metode Komputasi 2004 */
/* Nama Program : milne.cpp */

#include <iostream.h>
#include <stdlib.h>
#include <conio.h>
#include <stdio.h>
#include <math.h>

#define MaksU 100

main()
{
     FILE *milne;
     double    x[MaksU],
               y[MaksU],
               f[MaksU],
               xn1, yn1, fn1, yn1a,
               h;
     int i, n;
     char lagi;

     do
     {
          clrscr();
          milne = fopen("milne.txt", "w+");

          cout << "\n\nMetode Milne untuk Diferensiasi Numerik\n";
          cout << "=======================================\n\n";
          cout << "Contoh pada Diktat Metode Komputasi\n\n";
          cout << "Persamaan Diferensial : f(x,y) = dy/dx = x + y\n";
          cout << "----------------------------------------------\n\n";

          /************************************************/
          /* Setiap perubahan persamaan matematika, YAKINKAN */
          /* bahwa Anda telah merubah persamaan matematika   */
          /* di dalam program ini ------------------------- */
          /************************************************/

          cout << "\nTitik x dimana nilai fungsi f(x,y) = y yang dicari ? ";
          cin >> xn1;
          cout << "\nBerapa Jumlah Data yang akan diproses             ? ";
          cin >> n;

          /* Pemasukan Data sesuai Tabel yang diberikan */
          for(i = 0; i < n; i++)
          {
               printf("\n(x[%d], y[%d]) = ", i+1, i+1);
               cin >> x[i] >> y[i];                  /* Pemasukan Data */
               f[i] = x[i] + y[i];                   /* Nilai f(x,y) */
               h = x[i] - x[i-1];                    /* Interval */

          }

          printf("\n\n\th = %.2f", h);

          /* Cetak x[i], y[i] dan f[i] serta penghitungan Milne */
          printf("\n\n\t---------------------------------------\n");
          printf("\tdata ke-\tx[i]\ty[i]\tf[i]\n");
          printf("\t---------------------------------------\n");
          fprintf(milne,"\n\t---------------------------------------\n");
          fprintf(milne,"\tData ke-\tx[i]\ty[i]\tf[i]\n");
          fprintf(milne,"\t---------------------------------------\n");

          /* Pemasukan dan Penghitungan Data */
          for(i = 0; i < n; i++)
          {
               printf("\t   %d\t\t%.1f\t%.5f\t%.5f\n", i+1, x[i], y[i], f[i]);
               fprintf(milne,"\t   %d\t\t%.1f\t%.5f\t%.5f\n", i+1, x[i], y[i], f[i]);
```

```
            /* Rumus Diferensiasi Predikif Milne */
            yn1 = y[i-3] + 4*h*(2*f[i-2] - f[i-1] + 2*f[i])/3;
            yn1a = yn1;
            fn1 = xn1 + yn1;

            /* Rumus Diferensiasi Korektif Milne */
            yn1 = y[i-1] + h*(f[i-1] + 4*f[i] + fn1)/3;
        }

        printf("\t---------------------------------------\n");
        fprintf(milne,"\t---------------------------------------\n");
        printf("\ny[%.1f] prediktif = %.5f", xn1, yn1a);
        printf("\n\nf[%.1f] = %.5f", xn1, fn1);
        printf("\n\nMaka y[%.1f] = %.5f", xn1, yn1);
        fprintf(milne,"\nMaka y[%.1f] = %.5f", xn1, yn1);
        fclose(milne);

        cout << "\n\nCoba lagi dengan data awal yang berbeda (y/t) ? ";
        cin >> lagi;

    }while(lagi != 't');

    return 0;
}
```

DAFTAR PUSTAKA

1. _____, "**Metode Komputasi**", Diktat Kuliah Karbol AAU, Skep Gubernur AAU No.Skep/250/XII/1994 tanggal 23 Desember 2004, AAU, Yogyakarta, 2004.

2. Atkinson, Kendall E., "**An Introduction to Numerical Analysis**", 2nd Ed., John Wiley & Sons Inc., USA, 1989.

3. Basuki, Drs. A. dan Nana R., S.Kom, "**Metode Numerik dan Algoritma Komputasi**", Andi Offset, Yogyakarta, 2005

4. Chapra, Steven C. dan Raymond P. Canale, "**Numerical Methods for Engineer**", 2nd Ed., McGraw-Hill, USA, 1988.

5. Croft, Anthony; Robert Davison dan Martin Hargreaves, "**Engineering Mathematics: A Modern Foundation for Electronic, Electrical and Control Engineers**", Addison-Wesley, USA, 1992.

6. Djojodihardjo, Harijono, "**Metode Numerik**", Gramedia, Jakarta, 2000.

7. Kreyszig, Erwin, "**Advanced Engineering Mathematics**", 7th Ed., Wiley & Sons Inc., USA, 1993.

8. Munir, Rinaldi, "**Metode Numerik**", Informatika, Bandung, 2003.

9. Oualline, Steve, "**Practical C Programming**", O'Reilly & Associates, Inc., USA, 1993.

10. Rice, John H., "**Numerical Methods, Software and Analysis**", McGraw-Hil, USA, 1983.

11. Perry, Greg dan Marcus Johnson, "**Turbo C++ By Example**", Que, USA, 1992