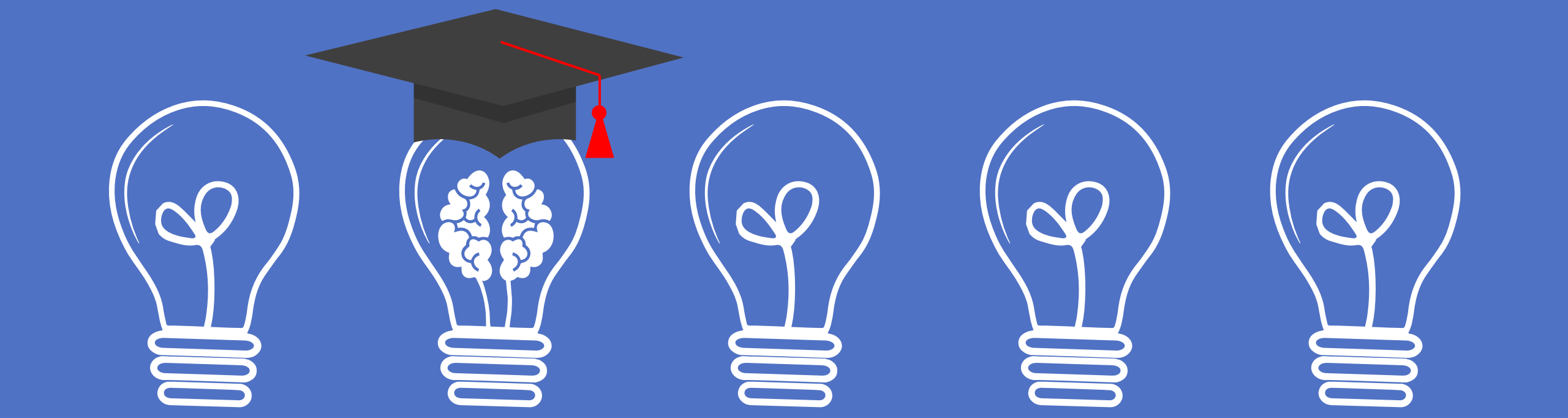
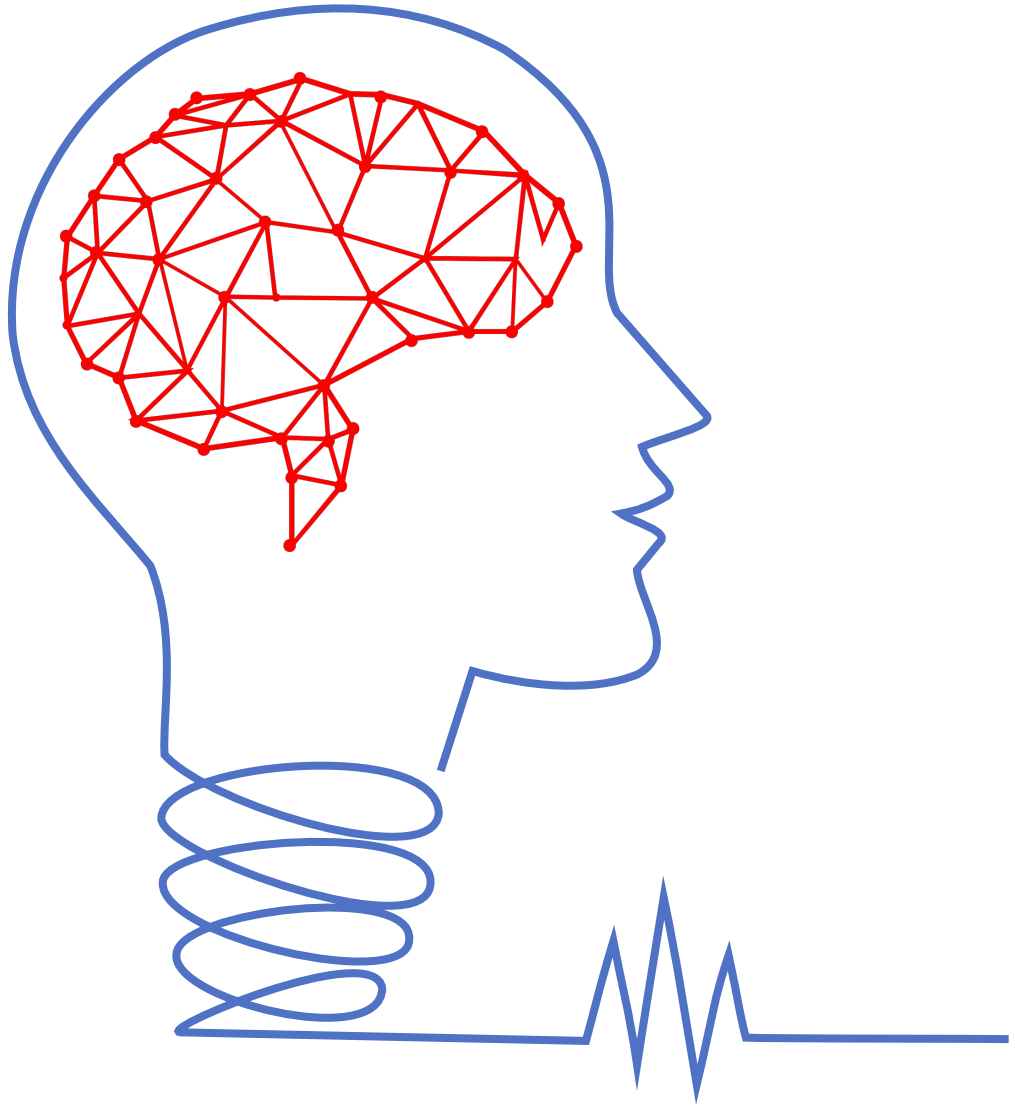


Golden Search Method

MATEMATIKA 2



Penyusun Tugas

1 D4 IT A

01 **Andrian Chaidir**
3120600009

02 **Devafilla Rizqy Santosa**
3120600026

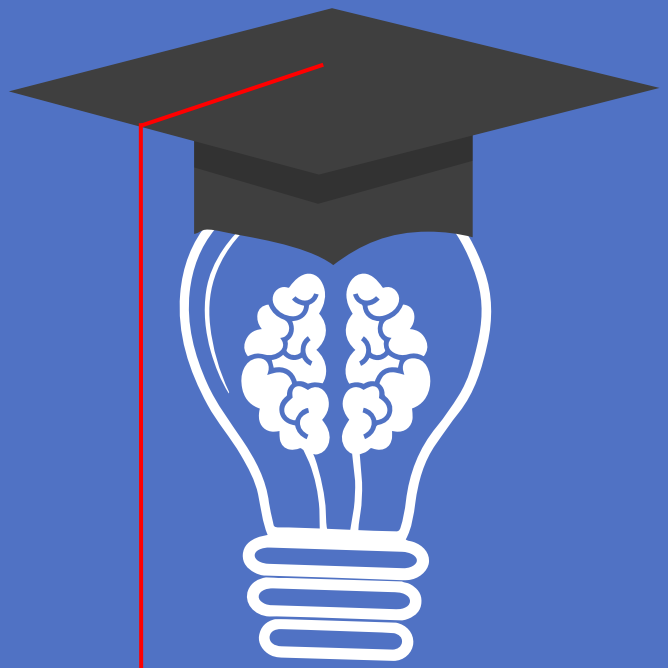
Metode Golden Search

Maksimum Method

1. Tentukan batas bawah, x_l, dan batas atas, x_u, yang mengapit satu nilai local ekstrimum dari $f(x)$
2. Tentukan 2 interior point, x_1 dan x_2, yang dihitung berdasarkan *golden ratio*

   $$x_1 = x_l + d$$
   $$x_2 = x_u - d$$

   Dengan $d = \frac{\sqrt{5}-1}{2}(x_u - x_l)$
3. Lakukan evaluasi :
   - Jika $f(x_1) > f(x_2)$ maka update nilai $x_l = x_2$
   - Jika $f(x_2) > f(x_1)$ maka update nilai $x_u = x_1$
4. Lakukan iterasi selanjutnya dengan nilai x_l dan x_u yang baru



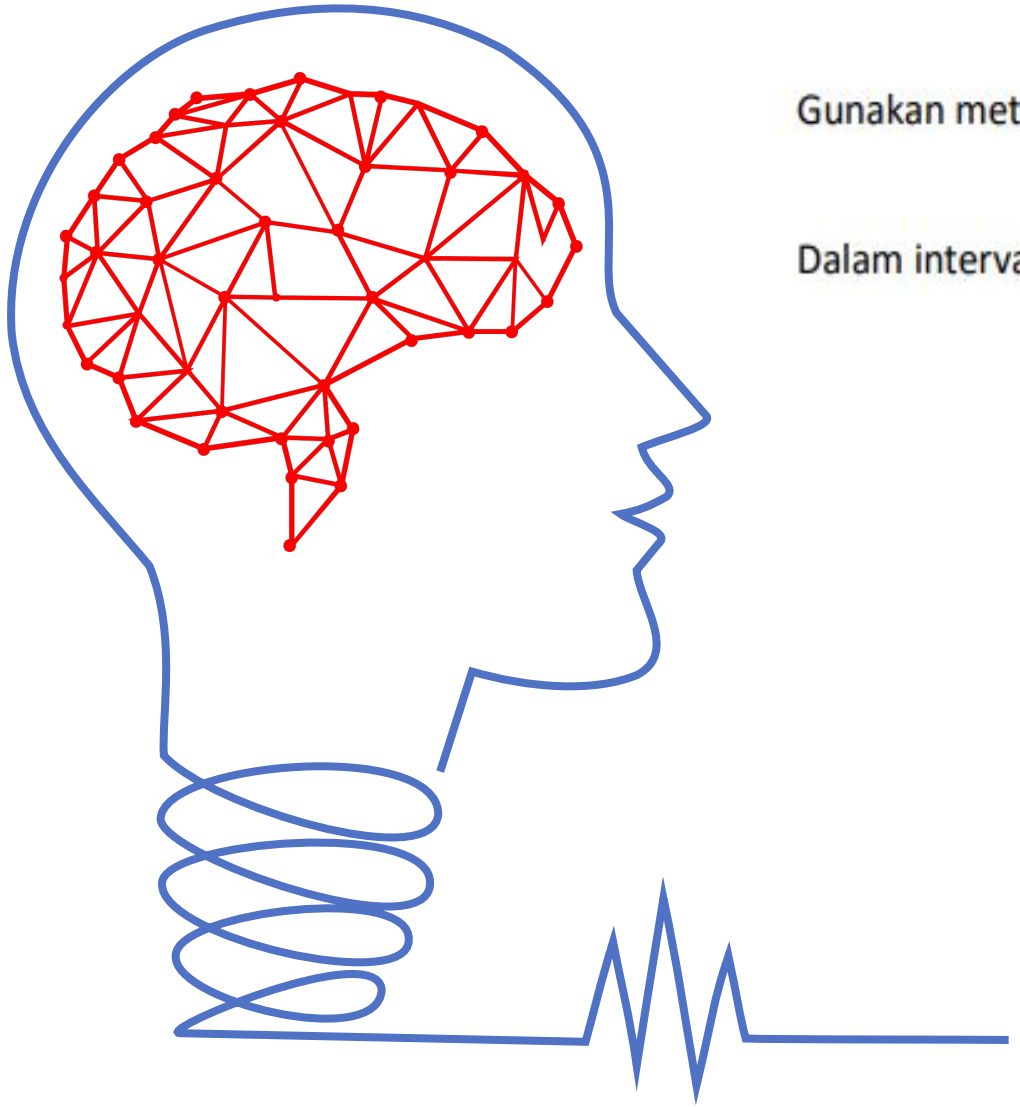
Minimum Point

1. Tentukan batas bawah, x_l, dan batas atas, x_u, yang mengapit satu nilai local ekstrimum dari $f(x)$
2. Tentukan 2 interior point, x_1 dan x_2, yang dihitung berdasarkan *golden ratio*

   $$x_1 = x_l + d$$
   $$x_2 = x_u - d$$

   Dengan $d = \frac{\sqrt{5}-1}{2}(x_u - x_l)$
3. Lakukan evaluasi :
   - Jika $f(x_1) < f(x_2)$ maka update nilai $x_l = x_2$
   - Jika $f(x_2) < f(x_1)$ maka update nilai $x_u = x_1$
4. Lakukan iterasi selanjutnya dengan nilai x_l dan x_u yang baru

Contoh Soal

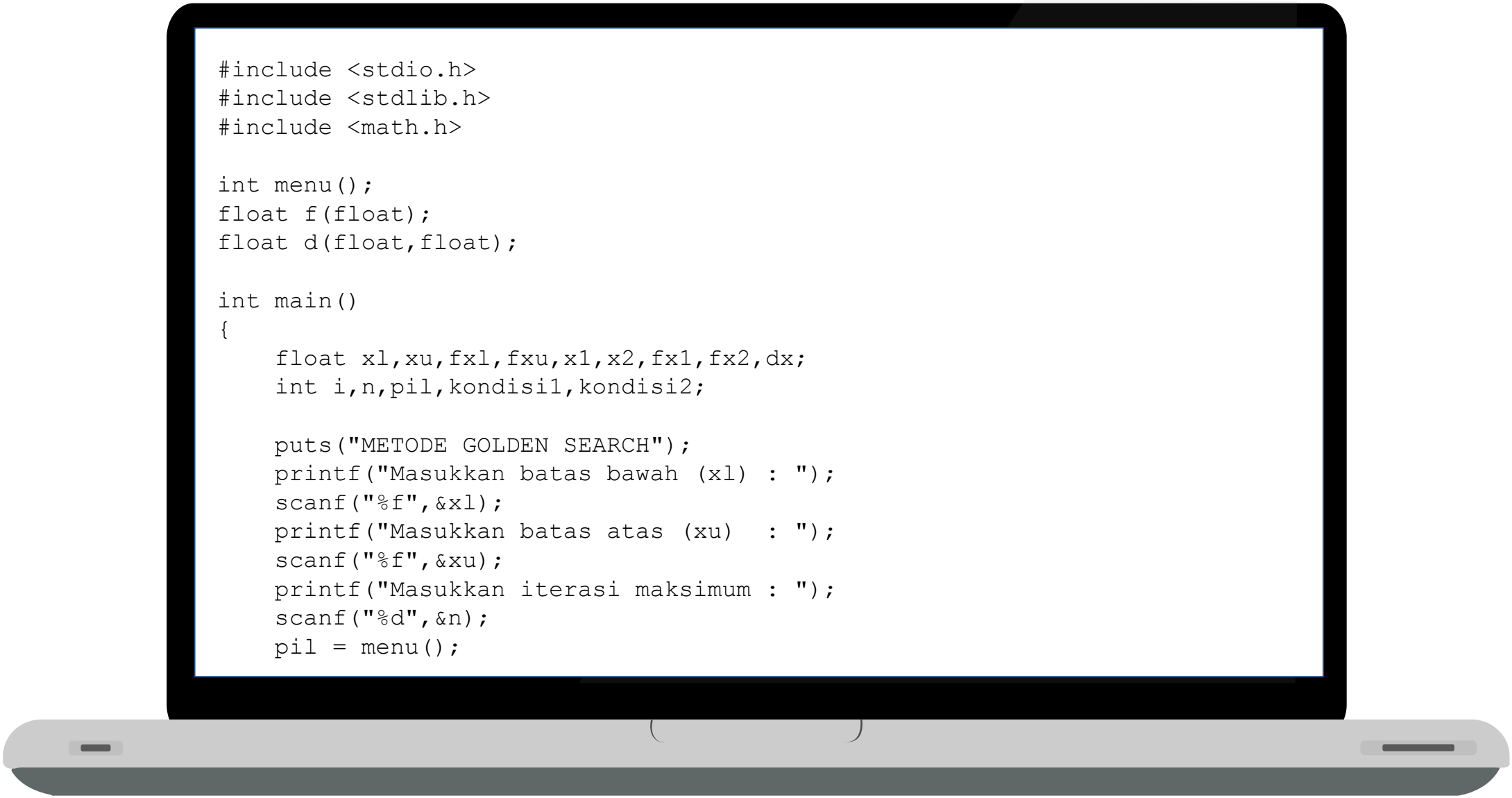


Gunakan metode golden search untuk mendapatkan nilai optimum fungsu berikut :

$$f(x) = 2\sin x - \frac{x^2}{10}$$

Dalam interval $x_l = 0$ dan $x_u = 4$

Source Code



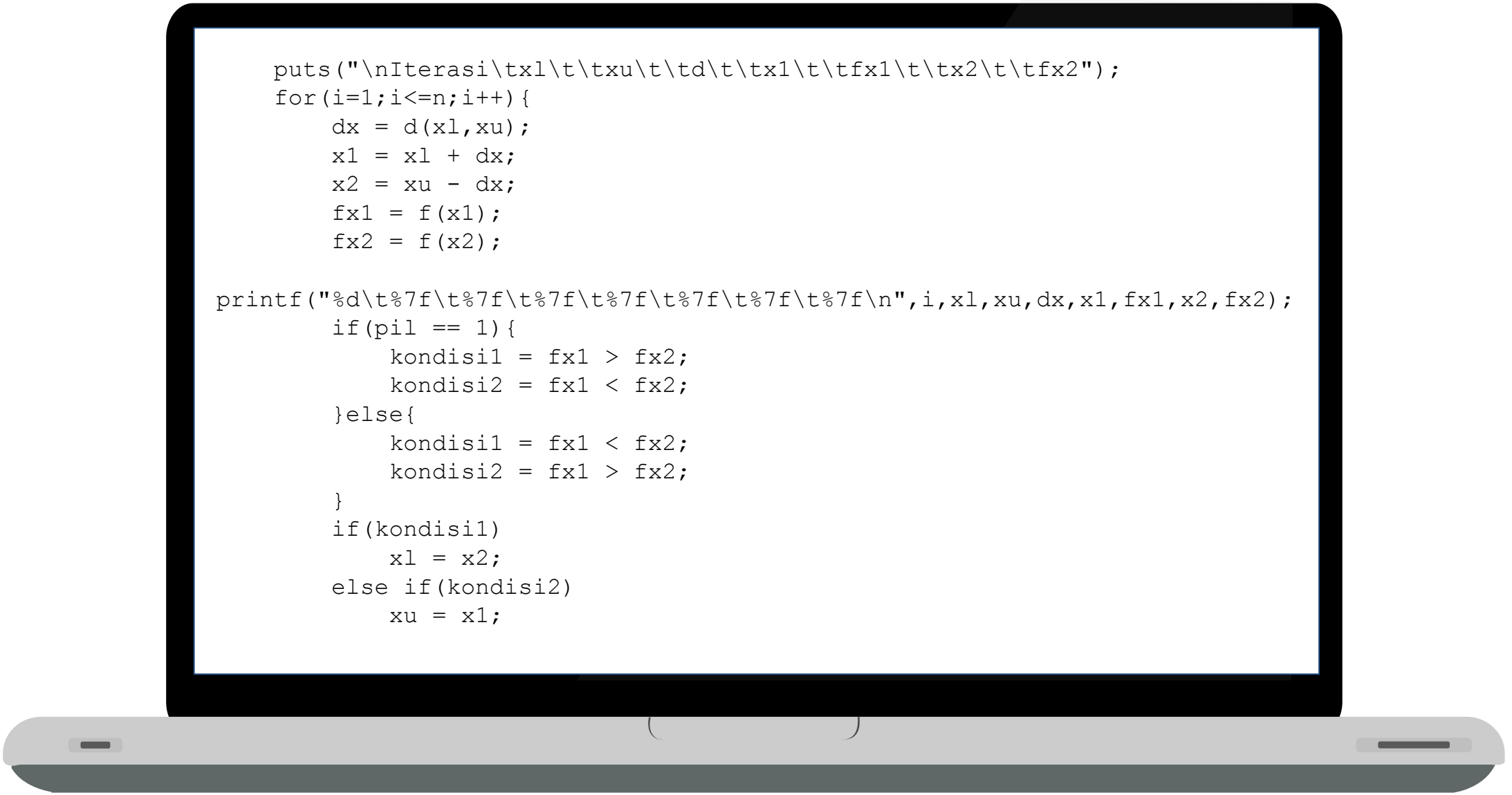
```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int menu();
float f(float);
float d(float,float);

int main()
{
    float xl,xu,fxl,fxu,x1,x2,fx1,fx2,dx;
    int i,n,pil,kondisi1,kondisi2;

    puts("METODE GOLDEN SEARCH");
    printf("Masukkan batas bawah (xl) : ");
    scanf("%f",&xl);
    printf("Masukkan batas atas (xu)  : ");
    scanf("%f",&xu);
    printf("Masukkan iterasi maksimum : ");
    scanf("%d",&n);
    pil = menu();
```

Source Code



```
    puts("\nIterasi\txl\t\txu\t\td\t\tx1\t\tfx1\t\tx2\t\tfx2");
    for(i=1;i<=n;i++){
        dx = d(xl,xu);
        x1 = xl + dx;
        x2 = xu - dx;
        fx1 = f(x1);
        fx2 = f(x2);

printf("%d\t%7f\t%7f\t%7f\t%7f\t%7f\t%7f\t%7f\n",i,xl,xu,dx,x1,fx1,x2,fx2);
        if(pil == 1){
            kondisi1 = fx1 > fx2;
            kondisi2 = fx1 < fx2;
        }else{
            kondisi1 = fx1 < fx2;
            kondisi2 = fx1 > fx2;
        }
        if(kondisi1)
            xl = x2;
        else if(kondisi2)
            xu = x1;
```

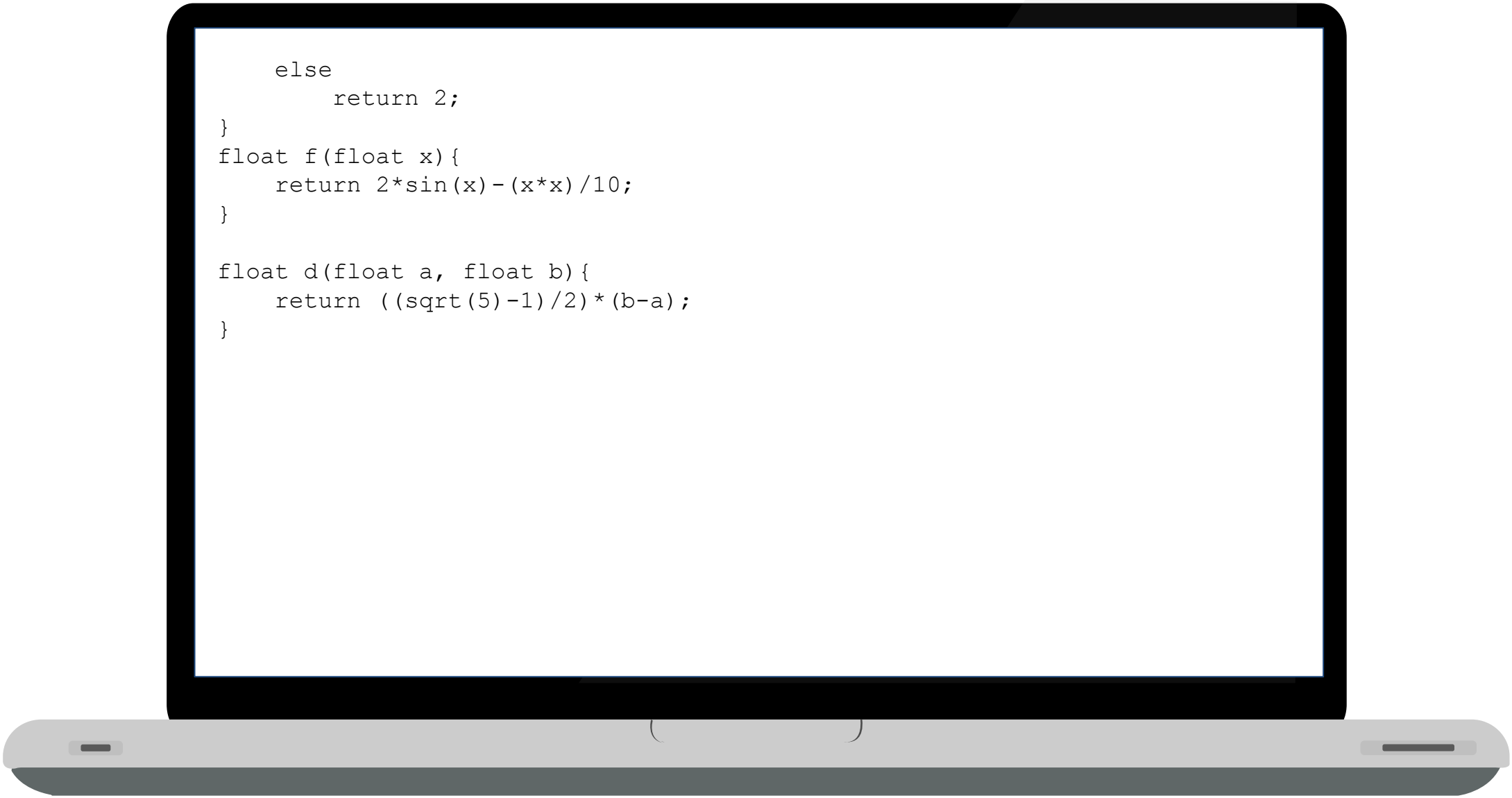
Source Code

```
else{
            puts("\nPenyelesaian ditemukan!");
            printf("Golden Ratio (d) = %7f\n",dx);
            exit(0);
        }
    }
    puts("\nPenyelesaian belum ditemukan tetapi telah mencapai iterasi
maksimum");
    printf("Golden Ratio (d) = %7f\n",dx);
}

int menu(){
    int pil;

    puts("\nMENU GOLDEN SEARCH");
    puts("1. Maksimum Point");
    puts("2. Minimum Point");
    printf("Masukkan pilihan Anda [1/2]: ");
    scanf("%d",&pil);
    if(pil == 1)
        return 1;
```

Source Code



```
    else
        return 2;
}
float f(float x){
    return 2*sin(x)-(x*x)/10;
}

float d(float a, float b){
    return ((sqrt(5)-1)/2)*(b-a);
}
```

Maksimum Point

```
METODE GOLDEN SEARCH
Masukkan batas bawah (xl) : 0
Masukkan batas atas (xu)  : 4
Masukkan iterasi maksimum : 20

MENU GOLDEN SEARCH
1. Maksimum Point
2. Minimum Point
Masukkan pilihan Anda [1/2]: 1

Iterasi xl          xu          d           x1          fx1         x2          fx2
1       0.000000    4.000000    2.472136    2.472136    0.629974    1.527864    1.764720
2       0.000000    2.472136    1.527864    1.527864    1.764720    0.944272    1.530976
3       0.944272    2.472136    0.944272    1.888544    1.543223    1.527864    1.764720
4       0.944272    1.888544    0.583592    1.527864    1.764720    1.304952    1.759452
5       1.304952    1.888544    0.360680    1.665632    1.713580    1.527864    1.764720
6       1.304952    1.665632    0.222912    1.527864    1.764720    1.442719    1.775475
7       1.304952    1.527864    0.137767    1.442719    1.775475    1.390097    1.774199
8       1.390097    1.527864    0.085145    1.475242    1.773242    1.442719    1.775475
9       1.390097    1.475242    0.052623    1.442719    1.775475    1.422619    1.775699
10      1.390097    1.442719    0.032523    1.422619    1.775699    1.410197    1.775398
11      1.410197    1.442719    0.020100    1.430297    1.775717    1.422619    1.775699
12      1.422619    1.442719    0.012422    1.435042    1.775664    1.430297    1.775717
13      1.422619    1.435042    0.007677    1.430297    1.775717    1.427364    1.775726
14      1.422619    1.430297    0.004745    1.427364    1.775726    1.425552    1.775721
15      1.425552    1.430297    0.002933    1.428484    1.775725    1.427364    1.775726
16      1.425552    1.428484    0.001812    1.427364    1.775726    1.426672    1.775725
17      1.426672    1.428484    0.001120    1.427792    1.775726    1.427364    1.775726

Penyelesaian ditemukan!
Golden Ratio (d) = 0.001120

Process returned 0 (0x0)   execution time : 6.989 s
Press any key to continue.
```

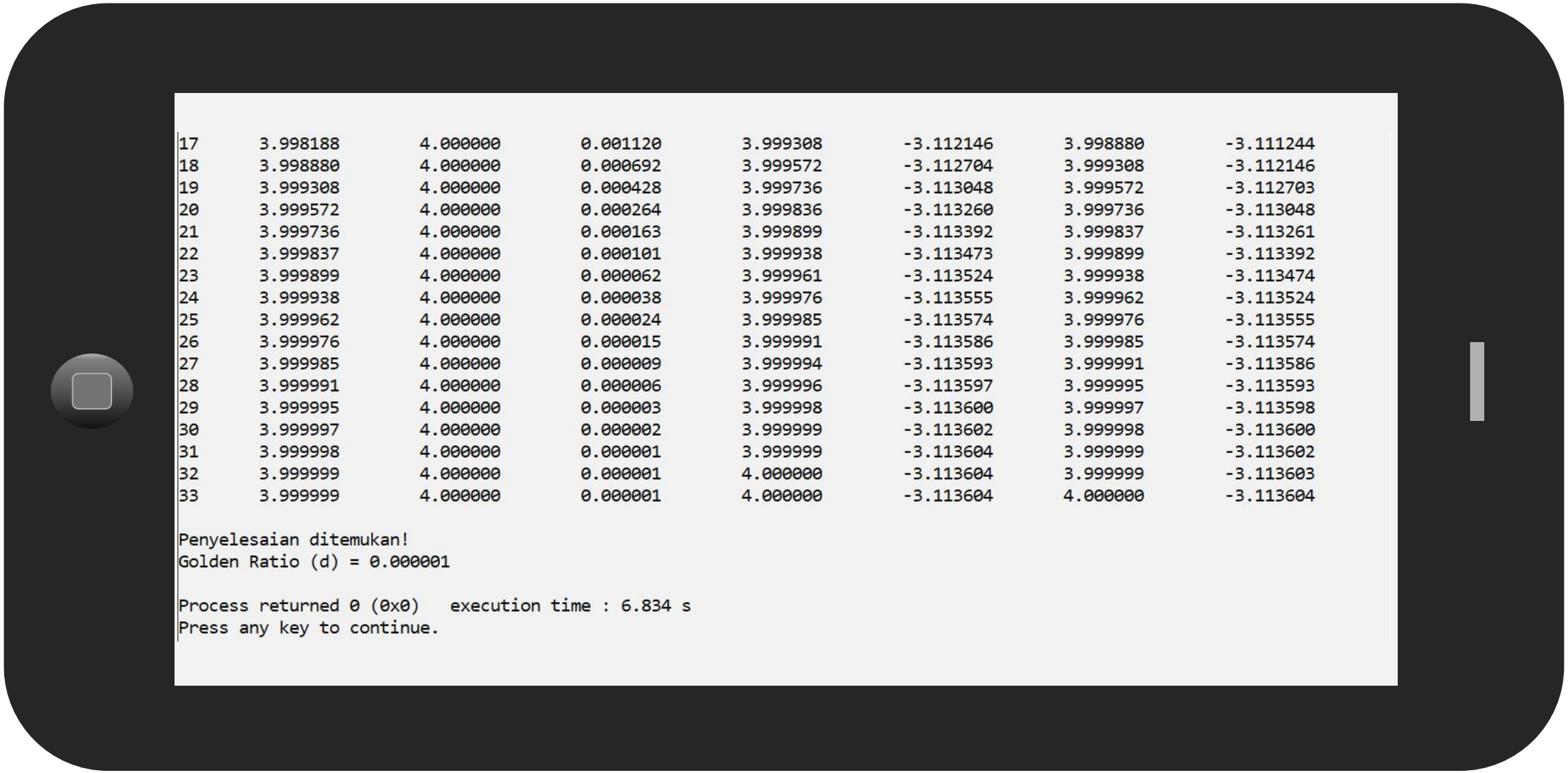
Minimum Point

```
METODE GOLDEN SEARCH
Masukkan batas bawah (xl) : 0
Masukkan batas atas (xu)  : 4
Masukkan iterasi maksimum : 35

MENU GOLDEN SEARCH
1. Maksimum Point
2. Minimum Point
Masukkan pilihan Anda [1/2]: 2
```

Iterasi	xl	xu	d	x1	fx1	x2	fx2
1	0.000000	4.000000	2.472136	2.472136	0.629974	1.527864	1.764720
2	1.527864	4.000000	1.527864	3.055728	-0.762229	2.472136	0.629974
3	2.472136	4.000000	0.944272	3.416408	-1.709922	3.055728	-0.762230
4	3.055728	4.000000	0.583592	3.639320	-2.279326	3.416408	-1.709923
5	3.416408	4.000000	0.360680	3.777088	-2.613791	3.639320	-2.279326
6	3.639320	4.000000	0.222912	3.862233	-2.811416	3.777088	-2.613791
7	3.777088	4.000000	0.137767	3.914855	-2.929556	3.862233	-2.811416
8	3.862233	4.000000	0.085145	3.947378	-3.000928	3.914855	-2.929556
9	3.914855	4.000000	0.052623	3.967478	-3.044384	3.947377	-3.000928
10	3.947377	4.000000	0.032523	3.979900	-3.070985	3.967478	-3.044384
11	3.967478	4.000000	0.020100	3.987577	-3.087326	3.979900	-3.070985
12	3.979900	4.000000	0.012422	3.992322	-3.097388	3.987578	-3.087327
13	3.987578	4.000000	0.007677	3.995255	-3.103591	3.992323	-3.097388
14	3.992323	4.000000	0.004745	3.997067	-3.107420	3.995255	-3.103592
15	3.995255	4.000000	0.002932	3.998188	-3.109783	3.997068	-3.107420
16	3.997068	4.000000	0.001812	3.998880	-3.111244	3.998188	-3.109784

Minimum Point



17	3.998188	4.000000	0.001120	3.999308	-3.112146	3.998880	-3.111244
18	3.998880	4.000000	0.000692	3.999572	-3.112704	3.999308	-3.112146
19	3.999308	4.000000	0.000428	3.999736	-3.113048	3.999572	-3.112703
20	3.999572	4.000000	0.000264	3.999836	-3.113260	3.999736	-3.113048
21	3.999736	4.000000	0.000163	3.999899	-3.113392	3.999837	-3.113261
22	3.999837	4.000000	0.000101	3.999938	-3.113473	3.999899	-3.113392
23	3.999899	4.000000	0.000062	3.999961	-3.113524	3.999938	-3.113474
24	3.999938	4.000000	0.000038	3.999976	-3.113555	3.999962	-3.113524
25	3.999962	4.000000	0.000024	3.999985	-3.113574	3.999976	-3.113555
26	3.999976	4.000000	0.000015	3.999991	-3.113586	3.999985	-3.113574
27	3.999985	4.000000	0.000009	3.999994	-3.113593	3.999991	-3.113586
28	3.999991	4.000000	0.000006	3.999996	-3.113597	3.999995	-3.113593
29	3.999995	4.000000	0.000003	3.999998	-3.113600	3.999997	-3.113598
30	3.999997	4.000000	0.000002	3.999999	-3.113602	3.999998	-3.113600
31	3.999998	4.000000	0.000001	3.999999	-3.113604	3.999999	-3.113602
32	3.999999	4.000000	0.000001	4.000000	-3.113604	3.999999	-3.113603
33	3.999999	4.000000	0.000001	4.000000	-3.113604	4.000000	-3.113604

Penyelesaian ditemukan!
Golden Ratio (d) = 0.000001

Process returned 0 (0x0)   execution time : 6.834 s
Press any key to continue.