

LAPORAN PRAKTIKUM IV
BAHASA PEMROGRAMAN C
STATEMENT PENENTU KEPUTUSAN

Disusun Untuk Memenuhi Tugas

Mata kuliah : Algoritma dan Pemrograman

Tanggal praktikum : 29 Oktober 2024

Dosen Pengampu : Ida Hastuti,S.ST.,M.T



Disusun Oleh :

Ilma Hasanah (C030324021)

KEMENTRIAN, PENDIDIKAN, KEBUDAYAAN,
RISET DAN TEKNOLOGI
PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK ELEKTRO
POLITEKKNIK NEGRI BANJARMASIN
TAHUN 2024

PENDAHULUAN

1. Latar Belakang

Dalam computer terdapat berbagai jenis bahasa yang digunakan dan biasa disebut bahasa pemrograman. Beberapa contoh dari bahasa pemrograman adalah bahasa pemrograman pascal, pemrograman C++, pemrograman C, pemrograman VB, dan masih banyak sekali bahasa pemrograman lainnya yang dapat digunakan. Dalam laporan kali ini, saya akan membahas mengenai Bahasa pemrograman C.

Walaupun bahasa pemrograman C termasuk bahasa yang cukup tua, tapi jika untuk tujuan mengasah keterampilan coding, Bahasa pemrograman C sangat cocok. Praktikum kali ini akan membahas tentang penggunaan statement penentu keputusan yaitu statement if-else. Memanfaatkan statement if-else untuk membuat proses perulangan yaitu dengan dikombinasikan dengan statement goto-label. Mempelajari statement if-else bersarang (nested if). Mempelajari bentuk lain dari statement if-else yaitu statement else-if. Mempelajari penggunaan dari statement switch-case sebagai alternatif pengganti dari statement else-if.

2. Tujuan Pembelajaran

Mempelajari dan mengamati penggunaan statement penentu keputusan yaitu statement if-else. Memanfaatkan statement if-else untuk membuat proses perulangan yaitu dengan dikombinasikan dengan statement goto-label. Mempelajari statement if-else bersarang (nested if). Mempelajari bentuk lain dari statement if-else yaitu statement else-if. Mempelajari penggunaan dari statement switch-case sebagai alternatif pengganti dari statement else-if.

DASAR TEORI

1. Bahasa Pemrograman C

Bahasa C adalah evolusi dari bahasa B yang dikembangkan oleh Dennis Ritchie, merupakan general-purpose language, yaitu bahasa pemrograman yang dapat digunakan untuk tujuan apa saja. Bahasa C mempunyai kemampuan lebih dari bahasa pemrograman lain. Banyak sekali aplikasi-aplikasi yang dibangun dengan bahasa C, mulai dari pemrograman sistem, aplikasi cerdas (artificial intelligence), sistem pakar, utility, driver, database, browser, network programming, sistem operasi, game, virus, dan lainnya, bahkan Software Development Kit untuk Windows juga ditulis dalam bahasa C. Karena sifat bahasa pemrogramannya yang portable, yaitu dengan sedikit atau tanpa perubahan, suatu program yang ditulis dengan bahasa C pada suatu komputer dapat dijalankan pada komputer lain.

Sebagai bahasa yang digolongkan dalam middle level language, bahasa C mempunyai kemudahan dalam mengakses perangkat keras, juga kecepatan prosesnya yang mendekati low level language seperti Assembly, tetapi memberikan kemudahan yang tidak ditawarkan Assembly. Disamping itu, bahasa C jauh lebih mudah untuk dipelajari jika dibandingkan dengan bahasa low level karena mendekati frase-frase dalam bahasa manusia, yaitu bahasa Inggris. Bahasa C juga mempunyai banyak keuntungan dibanding bahasa pemrograman lain. Dikarenakan kokoh dan memberikan keleluasaan kepada penggunanya, pada tahun 80-an, penggunaan bahasa C di dunia industri semakin luas, sehingga distandarisasi oleh ANSI dan kemudian diadopsi oleh ISO, lalu diadopsi ulang oleh ANSI. Official name bahasa C adalah ISO/IEC 9899-1990.

Bahasa C adalah bahasa pemrograman. Bahasa C termasuk dalam bahasa pemrograman tingkat menengah, artinya bahasa C lebih mudah dipelajari karena mudah dipahami namun memiliki kemampuan yang tinggi. Bahasa pemrograman C sangat cocok untuk memperkenalkan konsep pemrograman kepada pemula. Dengan kata lain, pemula yang belum pernah mencoba coding. Usia bahasa pemrograman C memang sudah tua, namun masih banyak orang yang menggunakannya hingga saat ini.

Bahasa C atau dilafalkan “si” merupakan bahasa pemrograman tingkat tinggi dan bertujuan umum yang digunakan dalam kehidupan sehari-hari. Yang dimaksud dengan tujuan umum adalah dapat digunakan untuk membuat program apa saja. Bahasa C dikembangkan oleh Dennis Ritchie dan Ken Thompson. Bahasa C telah menginspirasi banyak bahasa pemrograman lain seperti C++, C#, Java, PHP, Javascript, dll. Bahasa-bahasa tersebut mempunyai sintaksis yang mirip dengan C. Bahasa C juga dijuluki sebagai “Bahasa Pemrograman Tuhan”, artinya bahasa C adalah bapak dari semua bahasa pemrograman lainnya.

Bahas C mempunyai beberapa kelebihan dibandingkan bahasa lain seperti assembly, diantaranya: Bahasa C tersedia hampir di semua jenis komputer, kode Bahasa C bersifat portable, berbagai struktur data dan pengendalian proses disediakan dalam Bahasa C sehingga pembuatan program lebih terstruktur, mudah dipahami tanpa harus mengetahui mesin secara detail, memungkinkan manipulasi data dalam bentuk bit maupun byte. Namun ada pula beberapa kelemahan Bahasa C khususnya bagi pemula, kebanyakan dikarenakan banyaknya operator dan fleksibilitas penulisan program kadang-kadang membingungkan, dan umumnya pemrogram Bahasa C tingkat pemula belum pernah mengenal pointer dan tidak terbiasa menggunakannya.

2. *Visual Studio Code*

Visual Studio Code adalah editor source code yang dikembangkan oleh Microsoft untuk Windows, Linux dan MacOS. Ini termasuk dukungan untuk debugging, GIT Control yang disematkan, penyorotan sintaks, penyelesaian kode cerdas, cuplikan, dan kode refactoring. Hal ini juga dapat disesuaikan, sehingga pengguna dapat mengubah tema editor, shortcut keyboard, dan preferensi.

Visual Studio Code gratis dan open-source, meskipun unduhan resmi berada di bawah lisensi proprietary. Kode Visual Studio didasarkan pada Elektron, kerangka kerja yang digunakan untuk menyebarkan aplikasi Node.js untuk desktop yang berjalan pada Blinklayout. Meskipun menggunakan kerangka Elektron, Visual Studio Code tidak menggunakan Atom dan menggunakan komponen editor yang sama (diberi kode nama "Monaco") yang digunakan dalam Visual Studio Team Services yang sebelumnya disebut Visual Studio Online (Lardinois, 2015) .

3. *MinGW-64*

Pada tahun 2005, Mingw-w64 diciptakan oleh OneVision Software di bawah prinsip rekayasa perangkat lunak ruang bersih, karena proyek MinGW asli tidak cepat memperbarui basis kodenya, termasuk penyertaan beberapa API baru yang penting dan juga dukungan 64-bit yang sangat dibutuhkan. Pada tahun 2008.

OneVision kemudian menyumbangkan kode tersebut kepada Kai Tietz, salah satu pengembang utamanya, dengan syarat kode tersebut tetap menjadi sumber terbuka. Kode tersebut pertama kali diserahkan ke proyek MinGW asli, tetapi ditolak karena dicurigai menggunakan informasi yang tidak umum atau hak milik. Karena berbagai alasan, pengembang utama dan salah satu pendiri proyek MinGW-w64, Kai Tietz, memutuskan untuk tidak mencoba kerja sama lebih lanjut dengan MinGW.

METOLOGI PERCOBAAN

1. Alat dan bahan

- 1) Laptop
- 2) *Visual Studio Code*
- 3) *MinGW-64*

2. Gambar Alat

- 1) Laptop



- 2) *Visual Studio Code*



- 3) *MinGW-64*



3. Program Percobaan

- 1) Mengamati penggunaan statement penentu keputusan sederhana, yaitu menentukan apakah bilangan yang dimasukkan lebih kecil dari nol.

Input :

```
percobaan.c X
> Users > ADVAN > Desktop > C percobaan.c > main()
1  #include <stdio.h>
2  int main(){
3      int bil, absolut;
4      printf("Masukkan bilangan bulat = ");
5      scanf("%d", &bil);
6      if(bil < 0) absolut = -bil;
7      printf("Nilai absolut dari bil adalah = %d", bil, absolut);
8      return 0;
9  }
```

Ouput :

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\ADVAN> cd "c:\Users\ADVAN\Desktop\" ; if ($?) { gcc percobaan.c -o percobaan } ; if ($?) { .\percobaan }
Masukkan bilangan bulat = -5
Nilai absolut dari bil adalah = -5
PS C:\Users\ADVAN\Desktop>
```

- 2) Mengamati penggunaan statement penentu keputusan sederhana, yaitu menentukan apakah bilangan yang dimasukkan lebih kecil dari nol.

Input :

```
C: > Users > ADVAN > Desktop > C percobaan2.c > main()
1  #include <stdio.h>
2  int main()
3  {
4      char c;
5      int pembilang = 10, penyebut = 5;
6
7      printf("Masukkan suatu karakter :");
8      scanf("%c", &c);
9
10     if (c >= 'a' && c <= 'z') {
11         printf("%c tidak habis dibagi %d\n", c, pembilang, penyebut);
12     }
13     else{
14         printf("%c habis dibagi %d\n", c, pembilang, penyebut);
15     }
16     return 0;
17 }
```

Output :

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\ADVAN> cd "c:\Users\ADVAN\Desktop\" ; if ($?) { gcc percobaan2.c -o percobaan2 } ; if ($?) { .\percobaan2 }
Masukkan suatu karakter :d
d tidak habis dibagi 10
PS C:\Users\ADVAN\Desktop> cd "c:\Users\ADVAN\Desktop\" ; if ($?) { gcc percobaan2.c -o percobaan2 } ; if ($?) { .\percobaan2 }
Masukkan suatu karakter :2
2 habis dibagi 10
PS C:\Users\ADVAN\Desktop>
```

- 3) Menentukan keputusan bahwa suatu nilai termasuk dalam suatu batas jangkauan suatu nilai tertentu.

Input :

C: > Users > ADVAN > Desktop > C: percobaan3.c > main()

```
1  #include <stdio.h>
2  int main()
3  {
4      char c;
5
6      printf("Masukkan suatu karakter = ");
7
8      scanf("%c", &c);
9
10     if(c>='a' && c<='z'){
11         printf("\n Karakter tersebut termasuk huruf kecil\n");
12     }
13     else{
14         printf("\n Karakter tersebut bukan huruf kecil\n");
15     }
16     return 0;
17 }
```

Output :

```
PS C:\Users\ADVAN> cd "c:\Users\ADVAN\Desktop\" ; if ($?) { gcc percobaan3.c -o percobaan3 } ; if ($?) { .\percobaan3 }
Masukkan suatu karakter = A

Karakter tersebut bukan huruf kecil
PS C:\Users\ADVAN\Desktop> █
```

- 4) Suatu proses adakalanya dilakukan berulang, untuk mendapatkan nilai perhitungan yang lebih teliti (iterasi) atau untuk mengolah suatu data nilai secara table seperti pada daftar nilai siswa.

Input :

: > Users > ADVAN > Desktop > C: percobaan4.C > main()

```
1  #include <stdio.h>
2  int main()
3  {
4      int I=0, hasil=1;
5      ulang:
6          I++;
7          hasil*=2;
8          printf("Iterasi ke %2d : %5d\n", I, hasil);
9
10         if(I<10) goto ulang;
11
12     return 0;
13 }
```

Output :

```
PS C:\Users\ADVAN> cd "c:\Users\ADVAN\Desktop\" ; if ($?) { g++ percobaan4.C -o percobaan4 } ; if ($?) { .\percobaan4 }
Iterasi ke 1 :      2
Iterasi ke 2 :      4
Iterasi ke 3 :      8
Iterasi ke 4 :     16
Iterasi ke 5 :     32
Iterasi ke 6 :     64
Iterasi ke 7 :    128
Iterasi ke 8 :    256
Iterasi ke 9 :    512
Iterasi ke 10 : 1024
PS C:\Users\ADVAN\Desktop> █
```


5) if-else penggunaan Compound Statement.

Input :

```
C:\Users\ADVAN > Desktop > C perobaan5.c > main()
1  #include <stdio.h>
2  int main()
3  {
4      char tag;
5      printf("Masukkan tipe tag (2/4) =");
6      scanf("%c", &tag);
7
8      if(tag=='2'){
9          printf("Merek sepeda motor bertipe 2 tag\n\n");
10
11          printf("1. YA MAHA");
12          printf("2. YA SUZUKI");
13          printf("3. YA HONDA");
14      }
15      else{
16          printf("Merek sepeda motor bertipe 4 tag\n\n");
17
18          printf("1. MAHA");
19          printf("2. SUZUKI");
20          printf("3. HONDA");
21      }
22
23      return 0;
24  }
```

Ouput :

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
Code + - [ ] [ ] ...

PS C:\Users\ADVAN> cd "c:\Users\ADVAN\Desktop\" ; if ($?) { gcc perobaan5.c -o perobaan5 } ; if ($?) { .\perobaan5 }
Masukkan tipe tag (2/4) =2
Merek sepeda motor bertipe 2 tag

1. YA MAHA2. YA SUZUKI3. YA HONDA
PS C:\Users\ADVAN\Desktop> cd "c:\Users\ADVAN\Desktop\" ; if ($?) { gcc perobaan5.c -o perobaan5 } ; if ($?) { .\perobaan5 }
Masukkan tipe tag (2/4) =4
Merek sepeda motor bertipe 4 tag

1. MAHA2. SUZUKI3. HONDA
PS C:\Users\ADVAN\Desktop>
```

6) Menentukan satu pilihan yang benar diantara beberapa pilihan yang ada.

Input :

```
C: > Users > ADVAN > Desktop > C percobaan6.c > main()
1  #include <stdio.h>
2  int main()
3  {
4      int valid_operator =1;
5      char operator;
6      float bil1, bil2, hasil;
7
8      printf("Masukkan 2 bilangan dan sebuah operator.\n\n");
9      printf("Dengan format :\n\n");
10     printf("= bilangan-1 operator bilangan-2\n\n");
11
12     scanf("%f %c %f", &bil1, &operator, &bil2);
13
14     if(operator == '*') {
15         hasil = bil1 * bil2;
16     }
17     else if(operator == '/') {
18         hasil = bil1 / bil2;
19     }
20     else if(operator == '+') {
21         hasil = bil1 + bil2;
22     }
23     else if(operator == '-') {
24         hasil = bil1 - bil2;
25     }
26     else valid_operator = 0;
27
28     if(valid_operator) {
29         printf("\n Hasil perhitungan \n\n");
30         printf("%g %c %g = %g \n", bil1, operator, bil2, hasil);
31     }
32     else{
33         printf("\n Operator SALAH!");
34         printf("\n Gunakan operator +, -, /, dan * saja!");
35     }
36     return 0;
37 }
20
```

Output :

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
Masukkan 2 bilangan dan sebuah operator.

Dengan format :

= bilangan-1 operator bilangan-2

3 / 2

Hasil perhitungan

3 / 2 = 1.5
PS C:\Users\ADVAN\Desktop> cd "c:\Users\ADVAN\Desktop\" ; if ($?) { gcc percobaan6.c -o percobaan6 } ; if ($?) { .\percobaan6 }
Masukkan 2 bilangan dan sebuah operator.

Masukkan 2 bilangan dan sebuah operator.

Dengan format :

= bilangan-1 operator bilangan-2

5 < 7

Operator SALAH!
Gunakan operator +, -, /, dan * saja!
```

7) Renggunaan dari else-if dapat diganti dengan statement switch-case.

Input :

```
C:\Users\ADVAN > Desktop > C:\Users\ADVAN\Desktop\percobaan7.c > main()
1  #include <stdio.h>
2  int main()
3  {
4      int valid_operator = 1;
5      char operator;
6      float bil1, bil2, hasil;
7
8      printf("Masukkan 2 bilangan dan sebuah operator.\n\n");
9      printf("Dengan format : \n\n");
10     printf("= bilangan-1 operator bilangan-2\n\n");
11
12     scanf("%f %c %f", &bil1, &operator, &bil2);
13
14     switch(operator) {
15         case '*' : hasil = bil1 * bil2; break;
16         case '/' : hasil = bil1 / bil2; break;
17         case '+' : hasil = bil1 + bil2; break;
18         case '-' : hasil = bil1 - bil2; break;
19         default : valid_operator = 0;
20     }
21     if(valid_operator) {
22         printf("\n Hasil perhitungan \n\n");
23         printf("%g %c %g = %g \n", bil1, operator, bil2, hasil);
24     }
25     else{
26         printf("\n Hasil perhitungan \n\n");
27         printf("%g %c %g = %g \n", bil1, operator, bil2, hasil);
28     }
29     return 0;
30 }
```

Output :

```
Masukkan 2 bilangan dan sebuah operator.

Dengan format :

= bilangan-1 operator bilangan-2

90 * 2

Hasil perhitungan

90 * 2 = 180
PS C:\Users\ADVAN\Desktop>
```

```
Masukkan 2 bilangan dan sebuah operator.

Dengan format :

= bilangan-1 operator bilangan-2

90 < 91

Hasil perhitungan

90 < 91 = 1.12104e-44
```

4. Tugas-Tugas

- 1) Buatlah program untuk menentukan bilangan ganjil atau bilangan genap dari bilangan yang dimasukkan melalui keyboard.

Input :

```
C: > Users > ADVAN > Desktop > C latihan1.c > ...
1  #include <stdio.h>
2
3  int main() {
4      int bilangan;
5
6      printf("Masukkan sebuah bilangan: ");
7      scanf("%d", &bilangan);
8
9      if(bilangan / 2 == 0) {
10         printf("%d adalah bilangan genap\n", bilangan);
11     } else {
12         printf("%d adalah bilangan ganjil\n", bilangan);
13     }
14
15     return 0;
16 }
```

Output :

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\ADVAN> cd "c:\Users\ADVAN\Desktop\" ; if ($?) { gcc latihan1.c -o latihan1 } ; if ($?) { .\latihan1 }
Masukkan sebuah bilangan: 7
7 adalah bilangan ganjil
PS C:\Users\ADVAN\Desktop>
```

- 2) Buatlah program untuk menentukan kelompok suatu karakter yang dimasukkan melalui keyboard. Kelompok karakter tersebut adalah huruf kecil, huruf besar, angka, dan karakter khusus (tanda baca, operator, dsb).

Input :

```
C: > Users > ADVAN > Desktop > C latihan2.c > ...
1  #include <stdio.h>
2
3  int main()
4  {
5      char karakter;
6
7      printf("Masukkan sebuah karakter: ");
8      scanf("%c", &karakter);
9
10     if(karakter >= 'a' && karakter <= 'z') {
11         printf("%c adalah huruf kecil\n", karakter);
12     }
13     else if(karakter >= 'A' && karakter <= 'Z') {
14         printf("%c adalah huruf besar\n", karakter);
15     }
16     else if(karakter >= '0' && karakter <= '9') {
17         printf("%c adalah angka\n", karakter);
18     }
19     else {
20         printf("%c adalah karakter khusus\n", karakter);
21     }
22
23     return 0;
24 }
```

Output :

```
,
Masukkan sebuah karakter: %
'%' adalah karakter khusus
PS C:\Users\ADVAN\Desktop> cd "c:\Users\ADVAN\Desktop\" ; if ($?) { gcc latihan2.c -o latihan2 } ; if ($?) { .\
atihan2 }
Masukkan sebuah karakter: a
'a' adalah huruf kecil
PS C:\Users\ADVAN\Desktop> cd "c:\Users\ADVAN\Desktop\" ; if ($?) { gcc latihan2.c -o latihan2 } ; if ($?) { .\
atihan2 }
Masukkan sebuah karakter: Z
'Z' adalah huruf besar
PS C:\Users\ADVAN\Desktop> cd "c:\Users\ADVAN\Desktop\" ; if ($?) { gcc latihan2.c -o latihan2 } ; if ($?) { .\
atihan2 }
Masukkan sebuah karakter: 4
'4' adalah angka
PS C:\Users\ADVAN\Desktop>
```

3) Buatlah program kalkulator sederhana. Program harus memungkinkan user untuk mengetik ekspresi dengan bentuk: Bilangan Operator

- Operator yang digunakan adalah: * / + - % & | S E
- Operator S memerintah program untuk men-set “accumulator” untuk menyimpan angka yang diketik oleh user.
- Operator E memerintah program untuk mengakhiri program.

Input :

```
C: > Users > ADVAN > Desktop > C:\latihan3.c > main()
1  #include <stdio.h>
2
3  int main() {
4      double accumulator = 0;
5      double number;
6      char operator;
7
8      printf("Kalkulator Sederhana\n");
9      printf("Gunakan format: Bilangan Operator\n");
10     printf("Operator yang tersedia: +, -, *, /, %, &, |, S, E\n");
11
12     while (1) {
13         printf("Masukkan ekspresi: ");
14         scanf("%lf %c", &number, &operator);
15
16         if (operator == 'S') {
17             accumulator = number;
18             printf("Accumulator diset ke: %.2f\n", accumulator);
19         } else if (operator == 'E') {
20             printf("Program dihentikan.\n");
21             break;
22         } else {
23             switch (operator) {
24                 case '+':
25                     accumulator += number;
26                     printf("Hasil: %.2f\n", accumulator);
27                     break;
28                 case '-':
29                     accumulator -= number;
30                     printf("Hasil: %.2f\n", accumulator);
31                     break;
32
33                 case '*':
34                     accumulator *= number;
35                     printf("Hasil: %.2f\n", accumulator);
36                     break;
37                 case '/':
38                     if (number != 0) {
39                         accumulator /= number;
40                         printf("Hasil: %.2f\n", accumulator);
41                     } else {
42                         printf("Error: Pembagian dengan nol.\n");
43                     }
44                     break;
45                 case '%':
46                     if ((int)number != 0) {
47                         accumulator = (int)accumulator % (int)number;
48                         printf("Hasil: %.2f\n", accumulator);
49                     } else {
50                         printf("Error: Pembagian dengan nol.\n");
51                     }
52                     break;
53                 case '&':
54                     accumulator = (int)accumulator & (int)number;
55                     printf("Hasil: %.2f\n", accumulator);
56                     break;
57                 case '|':
58                     accumulator = (int)accumulator | (int)number;
59                     printf("Hasil: %.2f\n", accumulator);
60                     break;
61             }
62         }
63     }
64 }
```

```

60         default:
61             printf("Operator tidak dikenal.\n");
62             break;
63     }
64 }
65 }
66
67 return 0;
68 }

```

Output :

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\ADVAN> cd "c:\Users\ADVAN\Desktop\" ; if ($?) { gcc latihan3.c -o latihan3 } ; if ($?) { .\latihan3 }
Kalkulator Sederhana
Gunakan format: Bilangan Operator
Operator yang tersedia: +, -, *, /, %, &, |, S, E
Masukkan ekspresi: 255 S
Accumulator diset ke: 255.00
Masukkan ekspresi: 10 /
Hasil: 25.50
Masukkan ekspresi: 15 -
Hasil: 10.50
Masukkan ekspresi: 0 E
Program dihentikan.
PS C:\Users\ADVAN\Desktop>

```

- 4) Buatlah program untuk menentukan suatu tahun kabisat atau bukan yang dimasukkan melalui keyboard, dimana tahun dibatasi mulai dari tahun 1900 sampai dengan tahun 2005.

Input :

```

> Users > ADVAN > Desktop > C latihan4.c > main()
1  #include <stdio.h>
2  int main()
3  {
4      int tahun;
5
6      printf("Masukkan tahun (1900 - 2005): ");
7      scanf("%d", &tahun);
8
9      if (tahun >= 1900 && tahun <= 2005) {
10         if ((tahun % 4 == 0 && tahun % 100 != 0) || (tahun % 400 == 0)) {
11             printf("%d adalah tahun kabisat.\n", tahun);
12         } else {
13             printf("%d bukan tahun kabisat.\n", tahun);
14         }
15     } else {
16         printf("Tahun tidak valid.\n");
17     }
18
19     return 0;
20 }

```

Output :

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\ADVAN> cd "c:\Users\ADVAN\Desktop\" ; if ($?) { gcc latihan4.c -o latihan4 } ; if ($?) { .\latihan4 }
Masukkan tahun (1900 - 2005): 2001
2001 bukan tahun kabisat.
PS C:\Users\ADVAN\Desktop>

```

5) Buatlah program untuk menghitung determinan (D) dan mencari akar-akar dari persamaan kuadrat: $Ax^2 + Bx + C = 0$ dimana $D = B^2 - 4AC$.

- Jika $D = 0$, maka akar-akarnya kembar, yaitu $x_1 = x_2 = -(B/2A)$
- Jika $D > 0$, maka akar-akarnya berbeda, yaitu:
 $x_1 = (-B + \sqrt{D})/2A$ dan $x_2 = (-B - \sqrt{D})/2A$
- Jika $D < 0$, maka akar-akarnya imajiner berlainan, yaitu
 $x_1 = [(-B + \sqrt{D})/2A]j$ dan $x_2 = [(-B - \sqrt{D})/2A]j$

Input :

```
C:\> Users > ADVAN > Desktop > C latihan5.c > main()
1  #include <stdio.h>
2  #include <math.h>
3
4  int main() {
5      float A, B, C, D, x1, x2;
6
7      printf("Masukkan nilai A: ");
8      scanf("%f", &A);
9
10     printf("Masukkan nilai B: ");
11     scanf("%f", &B);
12
13     printf("Masukkan nilai C: ");
14     scanf("%f", &C);
15
16     D = B * B - 4 * A * C;
17
18     if (D > 0) {
19         x1 = (-B + sqrt(D)) / (2 * A);
20         x2 = (-B - sqrt(D)) / (2 * A);
21         printf("Akar-akar berbeda: x1 = %.2f, x2 = %.2f\n", x1, x2);
22     } else if (D == 0) {
23         x1 = x2 = -B / (2 * A);
24         printf("Akar-akar kembar: x1 = x2 = %.2f\n", x1);
25     } else {
26         x1 = (-B + sqrt(-D)) / (2 * A);
27         x2 = (-B - sqrt(-D)) / (2 * A);
28         printf("Akar-akar imajiner: x1 = %.2f + %.2fi, x2 = %.2f - %.2fi\n", x1, sqrt(-D), x2, sqrt(-D));
29     }
30
31     return 0;
32 }
```

Output :

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  Code
PS C:\Users\ADVAN> cd "c:\Users\ADVAN\Desktop\" ; if ($?) { gcc latihan5.c -o latihan5 } ; if ($?) { .\lat
Masukkan nilai A: 4
Masukkan nilai B: 5
Masukkan nilai C: 2
Akar-akar imajiner: x1 = -0.29 + 2.65i, x2 = -0.96 - 2.65i
PS C:\Users\ADVAN\Desktop> cd "c:\Users\ADVAN\Desktop\" ; if ($?) { gcc latihan5.c -o latihan5 } ; if ($?)
Masukkan nilai A: 0
Masukkan nilai B: 0
Masukkan nilai C: 0
Akar-akar kembar: x1 = x2 = -nan(ind)
PS C:\Users\ADVAN\Desktop> cd "c:\Users\ADVAN\Desktop\" ; if ($?) { gcc latihan5.c -o latihan5 } ; if ($?)
Masukkan nilai A: 2
Masukkan nilai B: 5
Masukkan nilai C: 1
Akar-akar berbeda: x1 = -0.22, x2 = -2.28
```

KESIMPULAN

Pada praktikum kali ini, kami telah berhasil membuat program kalkulator sederhana menggunakan bahasa pemrograman C dengan memanfaatkan struktur control if - else. Program ini dirancang untuk menerima input dari pengguna dalam format "Bilangan Operator", di mana pengguna dapat melakukan berbagai operasi aritmatika serta mengatur nilai accumulator.

Praktikum ini memberikan pemahaman yang lebih baik tentang penggunaan struktur control if - else, serta pengelolaan input dan output dalam bahasa C. Selain itu, kami juga belajar tentang pentingnya penanganan kesalahan dalam pengembangan perangkat lunak untuk meningkatkan pengalaman pengguna.