

Implementasi Process Specification (PSPEC)



Disusun oleh:
Kelompok RetoGen

Fathan Arya Maulana	103012300083
Rafiq Labib	103012330353
Dzaky Alfari	103012300391
M. Rifqi Dzaky Azhad	103012330009

Implementasi dan Pengujian Perangkat Lunak

Program Studi S1 Informatika

Fakultas Informatika

Universitas Telkom

Bandung 2025/2026

Soal 1 - C (M. Rifqi Dzaky Azhad)

Soal:

DETERMINE TRIANGLE

Dari masukan 3 bilangan a, b, c bebas (boleh bulat, boleh pecahan), segitiga apa dapat dibangun?

- ❑ Jika ada yang negatif atau 0, tidak ada segitiga dapat dibangun.
- ❑ Jika bilangan yang terbesar lebih besar atau sama dengan penjumlahan dua bilangan lainnya yang lebih kecil, tidak ada segitiga dapat dibangun.
- ❑ Jika $a=b$ atau $b=c$ atau $a=c$ (namun tidak sama dengan salah satu yang lain maka segitiga SAMA KAKI. (ISOSCELES)
- ❑ Jika $a=b$ dan $b=c$ maka segitiga SAMA SISI. (EQUILATERAL)
- ❑ Jika kuadrat bilangan terbesar = penjumlahan dari kuadrat dua bilangan lainnya, maka SEGITIGA SIKU-SIKU. (RIGHT TRIANGLE)
- ❑ Jika bukan sama kaki, sama sisi atau siku-siku, namun bilangan terbesarnya lebih kecil daripada penjumlahan dua bilangan lainnya, maka SEGITIGA BEBAS.

SOAL

- ❑ 1. Buat code (Pascal/C/PHP) untuk masukan bilangan bulat
- ❑ 2. Buat untuk bilangan pecahan dengan ketelitian 0.01 (1%). Ketelitian 1% berarti selisih panjang sisi-sisinya tidak lebih dari 1%. Selisih lebih dari 1% dianggap beda. Selisih 1% atau kurang dianggap sama.

Code:

Implementasi bahasa pemrograman C untuk menerima input 3 bilangan (integer dan/atau double), memproses ketiga bilangan tersebut dan menentukan validitas sebagai panjang sisi suatu segitiga dan jenis segitiga apa.

```
1 void determineTriangleInteger(int a, int b, int c, FILE 'output') {
2     printf("\n=== BILANGAN BULAT ===\n");
3     printf("Input: a=%d, b=%d, c=%d\n", a, b, c);
4
5     fprintf(output, "\n=== BILANGAN BULAT ===\n");
6     fprintf(output, "Input: a=%d, b=%d, c=%d\n", a, b, c);
7
8     if (a <= 0 || b <= 0 || c <= 0) {
9         printf("Output: TIDAK ADA SEGITIGA (ada bilangan <= 0)\n");
10        fprintf(output, "Output: TIDAK ADA SEGITIGA (ada bilangan <= 0)\n");
11        return;
12    }
13
14    int max = a, sum;
15    if (b > max) max = b;
16    if (c > max) max = c;
17
18    sum = a + b + c - max;
19
20    if (max >= sum) {
21        printf("Output: TIDAK ADA SEGITIGA (tidak memenuhi ketidaksamaan segitiga)\n");
22        fprintf(output, "Output: TIDAK ADA SEGITIGA (tidak memenuhi ketidaksamaan segitiga)\n");
23        return;
24    }
25
26    if (a == b && b == c) {
27        printf("Output: SEGITIGA SAMA SISI (EQUILATERAL)\n");
28        fprintf(output, "Output: SEGITIGA SAMA SISI (EQUILATERAL)\n");
29        return;
30    }
31
32    if (a == b || b == c || a == c) {
33        printf("Output: SEGITIGA SAMA KAKI (ISOSCELES)\n");
34        fprintf(output, "Output: SEGITIGA SAMA KAKI (ISOSCELES)\n");
35        return;
36    }
37
38    int a2 = a * a;
39    int b2 = b * b;
40    int c2 = c * c;
41
42    if (a2 + b2 == c2 || a2 + c2 == b2 || b2 + c2 == a2) {
43        printf("Output: SEGITIGA SIKU-SIKU (RIGHT TRIANGLE)\n");
44        fprintf(output, "Output: SEGITIGA SIKU-SIKU (RIGHT TRIANGLE)\n");
45        return;
46    }
47
48    printf("Output: SEGITIGA BEBAS (SCALENE)\n");
49    fprintf(output, "Output: SEGITIGA BEBAS (SCALENE)\n");
50 }
```

```

1 void determineTriangleFloat(double a, double b, double c, FILE *output) {
2     printf("\n=== BILANGAN PECAHAN (Toleransi 1%) ===\n");
3     printf("Input: a=%.2f, b=%.2f, c=%.2f\n", a, b, c);
4
5     fprintf(output, "\n=== BILANGAN PECAHAN (Toleransi 1%) ===\n");
6     fprintf(output, "Input: a=%.2f, b=%.2f, c=%.2f\n", a, b, c);
7
8     if (a <= 0 || b <= 0 || c <= 0) {
9         printf("Output: TIDAK ADA SEGITIGA (ada bilangan <= 0)\n");
10        fprintf(output, "Output: TIDAK ADA SEGITIGA (ada bilangan <= 0)\n");
11        return;
12    }
13
14    double max = a, sum;
15    if (b > max) max = b;
16    if (c > max) max = c;
17
18    sum = a + b + c - max;
19
20    if (max >= sum) {
21        printf("Output: TIDAK ADA SEGITIGA (tidak memenuhi ketidaksamaan segitiga)\n");
22        fprintf(output, "Output: TIDAK ADA SEGITIGA (tidak memenuhi ketidaksamaan segitiga)\n");
23        return;
24    }
25
26    double tolerance = 0.01;
27
28    int isEqual(double x, double y) {
29        double larger = (x > y) ? x : y;
30        if (larger == 0) return (x == 0 && y == 0);
31        return (fabs(x - y) / larger <= tolerance);
32    }
33
34    int ab_equal = isEqual(a, b);
35    int bc_equal = isEqual(b, c);
36    int ac_equal = isEqual(a, c);
37
38    if (ab_equal && bc_equal) {
39        printf("Output: SEGITIGA SAMA SISI (EQUILATERAL)\n");
40        fprintf(output, "Output: SEGITIGA SAMA SISI (EQUILATERAL)\n");
41        return;
42    }
43
44    if (ab_equal || bc_equal || ac_equal) {
45        printf("Output: SEGITIGA SAMA KAKI (ISOSCELES)\n");
46        fprintf(output, "Output: SEGITIGA SAMA KAKI (ISOSCELES)\n");
47        return;
48    }
49
50    double a2 = a * a;
51    double b2 = b * b;
52    double c2 = c * c;
53
54    int isPythagorean1 = isEqual(a2 + b2, c2);
55    int isPythagorean2 = isEqual(a2 + c2, b2);
56    int isPythagorean3 = isEqual(b2 + c2, a2);
57
58    if (isPythagorean1 || isPythagorean2 || isPythagorean3) {
59        printf("Output: SEGITIGA SIKU-SIKU (RIGHT TRIANGLE)\n");
60        fprintf(output, "Output: SEGITIGA SIKU-SIKU (RIGHT TRIANGLE)\n");
61        return;
62    }
63
64    printf("Output: SEGITIGA BEBAS (SCALENE)\n");
65    fprintf(output, "Output: SEGITIGA BEBAS (SCALENE)\n");
66 }

```

Input:

Tiga bilangan bertipe data Integer atau Double.

Output:

Suatu string yang akan di print di terminal, diantaranya:

- TIDAK ADA SEGITIGA
- SEGITIGA SAMA SISI
- SEGITIGA SAMA KAKI
- SEGITIGA SIKU SIKU
- SEGITIGA BEBAS

Contoh Running Code:

Saat code dijalankan, menampilkan menu dan dapat memilih jenis input:

- Bilangan Bulat
- Bilangan Pecahan

Contoh penggunaan dengan opsi “Bilangan Bulat” dan input $a = 2$, $b = 3$, dan $c = 4$.

```
PROGRAM DETERMINE TRIANGLE
=====

1. Bilangan Bulat
2. Bilangan Pecahan (Toleransi 1%)
3. Exit
Pilih mode: 1

Masukkan 3 bilangan bulat:
a: 2
b: 3
c: 4

=== BILANGAN BULAT ===
Input: a=2, b=3, c=4
Output: SEGITIGA BEBAS (SCALENE)
```

Contoh penggunaan dengan opsi “Bilangan Pecahan (Toleransi 1%)” dan input $a = 5.21$, $b = 3.72$, dan $c = 9.22$.

```
PROGRAM DETERMINE TRIANGLE
=====

1. Bilangan Bulat
2. Bilangan Pecahan (Toleransi 1%)
3. Exit
Pilih mode: 2

Masukkan 3 bilangan pecahan:
a: 5.21
b: 3.72
c: 9.22

=== BILANGAN PECAHAN (Toleransi 1%) ===
Input: a=5.21, b=3.72, c=9.22
Output: TIDAK ADA SEGITIGA (tidak memenuhi ketidaksamaan segitiga)

Lanjutkan program? (y/n):
```

Soal 2 - C (Dzaky Alfari)

Soal:

Buatkan kode program dalam bahasa pascal/C/Java untuk narasi program menghitung biaya servis komputer dalam bentuk *structured English* berikut ini:

```
DO CASE  
CASE computer-amount = 1 OR 2  
set base-fee = $50 and additional-fee = 0  
CASE computer-amount = 3 to 10  
set base-fee = $100 and additional-fee = $10 per peripheral  
CASE computer-amount > 10  
set base-fee = $500 and additional-fee = $10 per peripheral  
IF service-time is NOT in business hours  
base-fee is doubled  
IF customer is willing to drop off and pick up  
total-base-fee is reduced to one-half
```

Code:

Implementasi dalam bahasa pemrograman C untuk menerima empat parameter input yang terdiri dari dua bilangan bertipe data integer dan dua nilai boolean yang direpresentasikan sebagai integer. Input integer pertama menentukan jumlah komputer yang akan diservis, sedangkan input integer kedua menentukan jumlah peripheral. Sementara itu, input boolean pertama mengindikasikan apakah layanan servis dilakukan dalam jam operasional bisnis, dan input boolean kedua menyatakan kesediaan pelanggan untuk melakukan drop-off dan pick-up perangkat. Keempat parameter input tersebut kemudian diproses untuk menghitung komponen biaya dasar dan biaya tambahan, yang pada akhirnya menghasilkan total biaya servis komputer yang harus dibayarkan oleh pelanggan.

```
#include <stdio.h>

int main() {
    int computer_amount, service_time, is_business_hours, is_willing_dropoff;
    int base_fee, additional_fee, total_base_fee, total_fee;
    int peripheral_count;
    printf("=== SERVIS KOMPUTER ===\n");
    printf("Jumlah komputer: ");
    scanf("%d", &computer_amount);
    printf("Jumlah peripheral: ");
    scanf("%d", &peripheral_count);
    printf("Maktu servis (0 = di luar jam bisnis, 1 = dalam jam bisnis): ");
    scanf("%d", &is_business_hours);
    printf("Pelanggan bersedia drop off & pick up (0 = tidak, 1 = ya): ");
    scanf("%d", &is_willing_dropoff);
    if (computer_amount == 1 || computer_amount == 2) {
        base_fee = 50;
        additional_fee = 0;
    }
    else if (computer_amount >= 3 && computer_amount <= 10) {
        base_fee = 100;
        additional_fee = 10 * peripheral_count;
    }
    else if (computer_amount > 10) {
        base_fee = 500;
        additional_fee = 10 * peripheral_count;
    }
    else {
        printf("Jumlah komputer tidak valid!\n");
        return 1;
    }
}
```

```
total_base_fee = base_fee;
if (!is_business_hours) {
    total_base_fee = base_fee * 2;
}
if (is_willing_dropoff) {
    total_base_fee = total_base_fee / 2;
}
total_fee = total_base_fee + additional_fee;

printf("\n=== RINCIAN BIAYA SERVIS ===\n");
printf("Base fee: $d\n", base_fee);
printf("Additional fee (peripheral): $d\n", additional_fee);
if (!is_business_hours) {
    printf("Biaya diluar jam bisnis: base fee dikalikan 2\n");
}
if (is_willing_dropoff) {
    printf("Diskon drop off & pick up: base fee dikurangi 50%\n");
}
printf("Total base fee: $d\n", total_base_fee);
printf("Total biaya servis: $d\n", total_fee);
return 0;
}
```

Input:

Dua bilangan bertipe data integer dan Dua bilangan bertipe data boolean yang direpresentasikan sebagai integer

Output:

Rincian dari biaya service mulai dari biaya dasar, biaya tambahan, diskon drop off, dan biaya diluar jam kerja

Contoh Running Code:

Saat kode dijalankan program akan meminta :

- Jumlah komputer
- Jumlah peripheral
- Apakah di jam kerja?
- Apakah bersedia drop off dan pick up

Lalu hasil yang dikeluarkan akan berbeda:

Diluar jam kerja

```
=== SERVIS KOMPUTER ===
Jumlah komputer: 3
Jumlah peripheral: 3
Waktu servis (0 = di luar jam bisnis, 1 = dalam jam bisnis): 0
Pelanggan bersedia drop off & pick up (0 = tidak, 1 = ya): 1

=== RINCIAN BIAYA SERVIS ===
Base fee: $100
Additional fee (peripheral): $30
Biaya diluar jam bisnis: base fee dikalikan 2
Diskon drop off & pick up: base fee dikurangi 50%
Total base fee: $100
Total biaya servis: $130
```

Dalam jam kerja:

```
=== SERVIS KOMPUTER ===
Jumlah komputer: 3
Jumlah peripheral: 3
Waktu servis (0 = di luar jam bisnis, 1 = dalam jam bisnis): 1
Pelanggan bersedia drop off & pick up (0 = tidak, 1 = ya): 1

=== RINCIAN BIAYA SERVIS ===
Base fee: $100
Additional fee (peripheral): $30
Diskon drop off & pick up: base fee dikurangi 50%
Total base fee: $50
Total biaya servis: $80
```

Soal 3 - C (Rafiq Labib)

Soal:

Precondition 1

Customer arrives with **account-number** matching an account number in **ACCOUNTS**, whose **status-code** is set to "valid."

Postcondition 1

Invoice is produced containing **account-number** and **amount-of-sale**.

Precondition 2

Precondition 1 fails for any reason (**account-number** can't be found on **ACCOUNTS** or **status-code** is not equal to "valid").

Postcondition 2

Error message is produced.

Code:

Program ini merupakan implementasi bahasa pemrograman C untuk menerima input satu bilangan integer sebagai representasi ID akun. Jika ID akun berhasil ditemukan, program akan mengembalikan informasi nomor akun dan jumlah penjualan yang dilakukan. Jika ID akun tidak berhasil ditemukan, program akan mengembalikan pesan error yang memberitahukan bahwa ID akun tidak valid.

```
#include <stdio.h>
#include <string.h>
#define MAX_ACCOUNTS 5

typedef struct {
    int account_number;
    char status_code[10];
} Account;

Account ACCOUNTS[MAX_ACCOUNTS] = {
    {1001, "valid"},
    {1002, "invalid"},
    {1003, "valid"},
    {1004, "valid"},
    {1005, "suspended"}
};

int isEqual(char str1[], char str2[]) {
    int i = 0;
    while (str1[i] != '\0') {
        if (str1[i] != str2[i]) {
            return 0;
        }
        i++;
    }
    if (str2[i] == '\0') {
        return 1;
    }
    return 0;
};
```

```
int verifyAccount(int account_number) {
    for (int i = 0; i < MAX_ACCOUNTS; i++) {
        if (ACCOUNTS[i].account_number == account_number) {
            if (isEqual(ACCOUNTS[i].status_code, "valid")) {
                return 1;
            }
        }
    }
    return 0;
}

void produceInvoice(int account_number, float amount) {
    printf("\n=== INVOICE SERVIS KOMPUTER ===\n");
    printf("Account Number: %d\n", account_number);
    printf("Amount of Sale: $%.2f\n", amount);
    printf("=====\n");
}

void produceErrorMessage() {
    printf("\n=== ERROR ===\n");
    printf("Akun tidak ditemukan atau tidak valid!\n");
    printf("tolong cek nomor akun atau status akun.\n");
    printf("=====\n");
}

int main() {
    int input_account;
    float service_amount = 150.75;
    printf("Masukan nomor akun: ");
    scanf("%d", &input_account);

    if (verifyAccount(input_account)) {
        produceInvoice(input_account, service_amount);
    } else {
        produceErrorMessage();
    }

    return 0;
}
```

Input:

Satu bilangan integer sebagai representasi ID akun.

Output

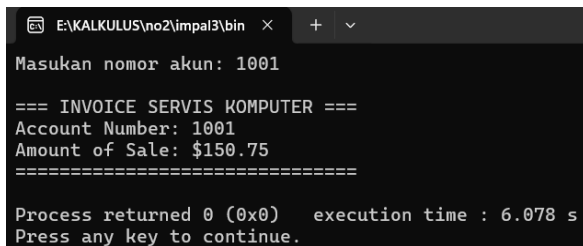
Keluaran berupa string berdasarkan dua kondisi berikut:

- Jika ID akun berhasil ditemukan , akan mengembalikan string berupa informasi nomor akun dan jumlah penjualan yang dilakukan
- Jika ID akun tidak berhasil ditemukan, akan mengembalikan string berupa pesan error yang memberitahukan bahwa ID akun tidak valid.

Contoh Running Code:

Ketika code dijalankan. program akan meminta masukan berupa string sebagai ID akun.

Contoh ketika memasukkan ID akun yang terdaftar/ada.

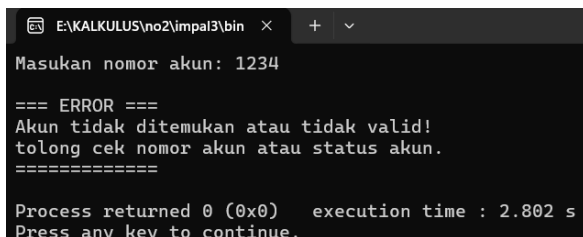


```
E:\KALKULUS\yno2\impal3\bin x + v
Masukan nomor akun: 1001

=== INVOICE SERVIS KOMPUTER ===
Account Number: 1001
Amount of Sale: $150.75
=====

Process returned 0 (0x0)   execution time : 6.078 s
Press any key to continue.
```

Contoh ketika memasukkan ID akun yang tidak terdaftar/tidak ada.



```
E:\KALKULUS\yno2\impal3\bin x + v
Masukan nomor akun: 1234

=== ERROR ===
Akun tidak ditemukan atau tidak valid!
tolong cek nomor akun atau status akun.
=====

Process returned 0 (0x0)   execution time : 2.802 s
Press any key to continue.
```


Soal 2 - Pascal (Fathan Arya Maulana)

Soal:

Precondition 1

Customer arrives with **account-number** matching an account number in **ACCOUNTS**, whose **status-code** is set to "valid."

Postcondition 1

Invoice is produced containing **account-number** and **amount-of-sale**.

Precondition 2

Precondition 1 fails for any reason (**account-number** can't be found on **ACCOUNTS** or **status-code** is not equal to "valid").

Postcondition 2

Error message is produced.

Code:

Implementasi dalam bahasa pemrograman Pascal untuk menerima empat parameter input: dua bilangan integer (jumlah komputer dan peripheral) serta dua nilai boolean yang direpresentasikan sebagai integer (status jam operasional bisnis dan kesediaan drop-off/pick-up). Keempat parameter tersebut diproses untuk menghitung biaya dasar dan biaya tambahan, menghasilkan total biaya servis komputer yang harus dibayarkan pelanggan.

```
program ServisKomputer;

var
  computer_amount, service_time, is_business_hours, is_willing_dropoff: Integer;
  base_fee, additional_fee, total_base_fee, total_fee: Integer;
  peripheral_count: Integer;

begin
  WriteLn('=== SERVIS KOMPUTER ===');

  Write('Jumlah komputer: ');
  ReadLn(computer_amount);

  Write('Jumlah peripheral: ');
  ReadLn(peripheral_count);

  Write('Waktu servis (0 = di luar jam bisnis, 1 = dalam jam bisnis): ');
  ReadLn(is_business_hours);

  Write('Pelanggan bersedia drop off & pick up (0 = tidak, 1 = ya): ');
  ReadLn(is_willing_dropoff);

  if (computer_amount = 1) or (computer_amount = 2) then
  begin
    base_fee := 50;
    additional_fee := 0;
  end
  else if (computer_amount >= 3) and (computer_amount <= 10) then
  begin
    base_fee := 100;
    additional_fee := 10 * peripheral_count;
  end
  else if computer_amount > 10 then
  begin
    base_fee := 500;
    additional_fee := 10 * peripheral_count;
  end
  else
  begin
    WriteLn('Jumlah komputer tidak valid!');
    Halt(1);
  end;
end;
```

```
total_base_fee := base_fee;

if is_business_hours = 0 then
begin
    total_base_fee := base_fee * 2;
end;

if is_willing_dropoff = 1 then
begin
    total_base_fee := total_base_fee div 2;
end;

total_fee := total_base_fee + additional_fee;

writeln;
writeln('=== RINCIAN BIAYA SERVIS ===');
writeln('Base fee: $', base_fee);
writeln('Additional fee (peripheral): $', additional_fee);

if is_business_hours = 0 then
begin
    writeln('Biaya diluar jam bisnis: base fee dikalikan 2');
end;

if is_willing_dropoff = 1 then
begin
    writeln('Diskon drop off & pick up: base fee dikurangi 50%');
end;

writeln('Total base fee: $', total_base_fee);
writeln('Total biaya servis: $', total_fee);
end.
```

Input:

Dua bilangan bertipe data integer dan Dua bilangan bertipe data boolean yang direpresentasikan sebagai integer

Output:

Rincian dari biaya service mulai dari biaya dasar, biaya tambahan, diskon drop off, dan biaya diluar jam kerja

Contoh Running Code:

Saat kode dijalankan program akan meminta :

- Jumlah komputer
- Jumlah peripheral
- Apakah di jam kerja
- Apakah bersedia drop off dan pick up

Lalu hasil yang dikeluarkan akan berbeda:

Diluar jam kerja

Contoh Running Code:

Contoh ketika di dalam jam bisnis.

```
=== SERVIS KOMPUTER ===  
Jumlah komputer: 3  
Jumlah peripheral: 3  
Waktu servis (0 = di luar jam bisnis, 1 = dalam jam bisnis): 1  
Pelanggan bersedia drop off & pick up (0 = tidak, 1 = ya): 1  
  
=== RINCIAN BIAYA SERVIS ===  
Base fee: $100  
Additional fee (peripheral): $30  
Diskon drop off & pick up: base fee dikurangi 50%  
Total base fee: $50  
Total biaya servis: $80
```

Contoh ketika inputan diluar jam bisnis.

```
=== SERVIS KOMPUTER ===  
Jumlah komputer: 3  
Jumlah peripheral: 3  
Waktu servis (0 = di luar jam bisnis, 1 = dalam jam bisnis): 0  
Pelanggan bersedia drop off & pick up (0 = tidak, 1 = ya): 1  
  
=== RINCIAN BIAYA SERVIS ===  
Base fee: $100  
Additional fee (peripheral): $30  
Biaya diluar jam bisnis: base fee dikalikan 2  
Diskon drop off & pick up: base fee dikurangi 50%  
Total base fee: $100  
Total biaya servis: $130
```

Link Github: <https://github.com/FathanAM0/IMPAL-Kelompok-RetoGen>