

Laporan Praktikum Pemrograman

Pertemuan Ke-2

Nama: Bryan Al Hilal Siregar

NIM: 24/541712/PA/22990

Github Link: <https://github.com/bryanalsiregar/praktikum-pemrograman/tree/main/Tugas%20Chapter%202>

Identifikasi Masalah

1. You are tasked with creating a payslip for an employee. The payslip should include the following details:

- Name: The name of the employee.
- Gross Salary: The gross salary of the employee, given as a percentage of the base salary.
- Tax: A flat tax rate of 20%.
- Installment: A fixed installment amount.
- Insurance: A fixed insurance amount.

The payslip should be displayed in a table format, and your program should calculate the net salary after tax, installment, and insurance deductions.

Requirements:

- Calculate the gross salary from the percentage.
- Deduct the tax (20%) from the gross salary.
- Subtract the fixed installment and insurance amounts.
- Display the payslip in a table format.

Example Output:

Payslip for Employee -----

Name: John Doe
 Gross Salary: Rp15.000.000,00
 Tax (20%): Rp3.000.000,00
 Installment: Rp200.000,00
 Insurance: Rp150.000,00
 Net Salary: Rp11.650.000,00

2. You are given a quadratic equation of the form:

$$ax^2 + bx + c = 0$$

where a, b, and c are coefficients. Your task is to write a C++ program to solve the quadratic equation and find the values of x.

Penjelasan Singkat Mengenai Source Code Test Case

Code	Penjelasan
<pre>// Header yang dipakai untuk menjalankan #include<iostream> // Untuk mengautr I/O Stream #include<string> // Untuk menggunakan tipe string dan manipulasinya #include<algorithm> // Untuk menggunakan algoritma yang sudah dibuat #include<math.h> // Untuk menggunakan fungsi matematika yang tidak tersedia di C++ #include<sstream> // Untuk menetapkan presisi angka desimal #include<iomanip> // Untuk mengatur presisi angka desimal // Penggunaan score std using namespace std; // Penyederhaana tipe typedef unsigned long long ull;</pre>	<p>Awalan yang harus digunakan pada test case untuk kebutuhan dan juga penggunaan scope dan juga penyederhaan tipe</p>
<pre>// Fungsi ini untuk menghilangkan digit nol di belakang desimal string removeTrailingZeros(float value) { // Deklarasi variabel ostringstream oss;</pre>	<p>Fungsi ini digunakan untuk menghilangkan angka nol di belakang desimal supaya sesuai dengan yang diinginkan tester. Fungsi ini untuk membantu</p>

```

    // Mengatur presisi angka desimal nilai secara
    tetap
    oss << fixed << setprecision(10) << value;
    // Mengubah nilai itu menjadi tipe string
    string str = oss.str();

    // Menghapus nol dibelakang desimal
    str.erase(str.find_last_not_of('0') + 1,
string::npos);

    // Menghilangkan titik jika desimal berada
    akhir
    if (str.back() == '.') {
        str.pop_back();
    }

    // Keluarkan hasil penghilangan digit nol di
    belakang desimal
    return str;
}

```

pada penyelesaian problem 2:
Quadratic Equation Solver

```

// Fungsi ini untuk menambahkan titik pada format
ribuan
string thousandSeparator(int n)
{
    // Membuat inisial output
    string ans = "";

    // Mengubah angka pada input menjadi string
    string num = to_string(n);

    // Melakukan penambahan titik koma setelah tiga
    digit dari belakang angka
    // Dilakukan counting untuk mencari tahu apakah
    sudah pada digit ke-3 dari belakang
    // Penambahan titik koma dan tiga digit pada
    variabel ans
    int count = 0;
    for (int i = num.size() - 1;
        i >= 0; i--) {
        count++;
        ans.push_back(num[i]);

        if (count == 3) {
            ans.push_back('.');

```

Fungsi ini digunakan untuk memberikan “.” Tiap tiga digit pada angka sehingga akan menghasilkan output string yang sesuai dengan output yang diinginkan.

```

        count = 0;
    }
}

// Output harus dibalikkan sebab pencarian tiga
digit belakang angka
reverse(ans.begin(), ans.end());

// Pengecekan apakah panjang variabel ans
habis dibagi 4
// Jika iya maka terjadi penghapusan pada
variabel ans
if (ans.size() % 4 == 0) {
    ans.erase(ans.begin());
}

// Output hasil dari pemisah ribuan
return ans;
}

```

```

// Problem 1
// Fungsi untuk penyelesaian permasalahan Payslip
Calculation
string paySlip(string name, ull salary, ull
installment, ull insurance) {
    // Deklarasi variabel
    ull tax;
    string outStr;

    // Menghitung pajak dari variabel salary
    tax = salary*0.2;

    // Membuat output yang diinginkan oleh tester
    outStr = "Payslip for Employee\n-----
-----\nName: "
        +name+"\nGross Salary:
Rp"+thousandSeparator(salary)+" ,00\nTax (20%):
Rp"+thousandSeparator(tax)+
        ",00\nInstallment:
Rp"+thousandSeparator(installment)+" ,00\nInsurance:
Rp"+
        thousandSeparator(insurance)+" ,00\nNet Salary:
Rp"+thousandSeparator(salary-tax-installment-
insurance)+

```

Fungsi merupakan penyelesaian pada problem 1: Payslip Calculation. Fungsi ini merupakan penyederhanaan dan penyesuaian file problem_1.cpp pada link github supaya sesuai dengan format tester.

<pre> ",00"; return outStr; } </pre>	
<pre> // Problem 2 Quadratic Equation Solver // Fungsi untuk penyelesaian permasalahan Quadratic Equation Solver string realRoots(float a, float b, float c) { // Deklarasi variabel yang dibutuhkan float disc; string s_x_1, s_x_2, outRoots; // Mencari Diskriminan disc = pow(b,2)-4*a*c; // Jika diskriminan lebih dari 0 maka output ada dua penyelesaian if (disc > 0) return removeTrailingZeros((- b+sqrt(disc))/(2*a))+ " "+removeTrailingZeros((-b- sqrt(disc))/(2*a)); // Jika diskriminan sama dengan 0 maka output ada satu penyelesaian if (disc == 0) return removeTrailingZeros((-b)/(2*a)); // Jika diskriminan kurang dari nol maka output Tidak ada return "Tidak ada"; } </pre>	<p>Fungsi merupakan penyelesaian pada problem 2: Quadratic Equation Solver. Fungsi ini merupakan penyederhanaan dan penyesuaian file problem_2.cpp pada link github supaya sesuai dengan format tester.</p>
<pre> // Bagian ini untuk mengetes problem 1 yaitu Payslip Calculation if (paySlip("Adi", 1000000, 100000, 100000) == "Payslip for Employee\n----- -\nName: Adi\nGross Salary: Rp1.000.000,00\nTax (20%): Rp200.000,00\nInstallment: Rp100.000,00\nInsurance: Rp100.000,00\nNet Salary: Rp600.000,00") {cout << "Payslip Calculation Test 1: SUCCESS" << endl;} else {cout << "Payslip Calculation Test 1: FAILED" << endl;} if (</pre>	<p>Bagian ini merupakan tester untuk problem 1: Payslip Calculation</p>

```

        paySlip("Bryan", 0, 0, 0) ==
        "Payslip for Employee\n-----
-\nName: Bryan\nGross Salary: Rp0,00\nTax (20%):
Rp0,00\nInstallment: Rp0,00\nInsurance: Rp0,00\nNet
Salary: Rp0,00"
    ) {cout << "Payslip Calculation Test 2:
SUCCESS" << endl;} else {cout << "Payslip
Calculation Test 2: FAILED" << endl;}
    if (
        paySlip("Caca", 1000000000, 1, 1) ==
        "Payslip for Employee\n-----
-\nName: Caca\nGross Salary:
Rp1.000.000.000,00\nTax (20%):
Rp200.000.000,00\nInstallment: Rp1,00\nInsurance:
Rp1,00\nNet Salary: Rp799.999.998,00"
    ) {cout << "Payslip Calculation Test 3:
SUCCESS" << endl;} else {cout << "Payslip
Calculation Test 3: FAILED" << endl;}

```

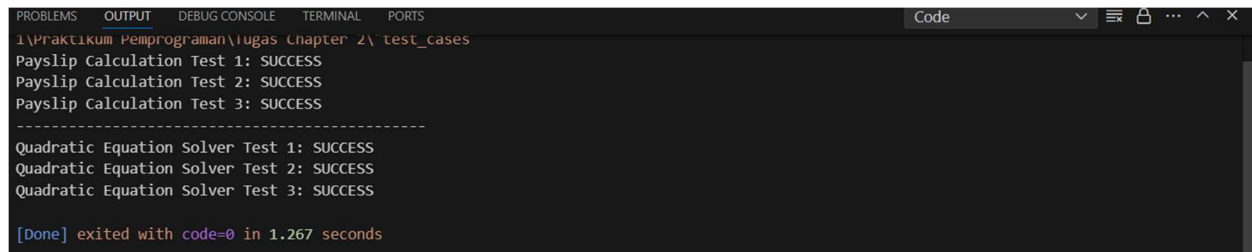
```

if (
    realRoots(2, -1, -15) == "3 -2.5"
) {cout << "Quadratic Equation Solver Test 1:
SUCCESS" << endl;} else {cout << "Quadratic
Equation Solver Test 1: FAILED" << endl;}
    if (
        realRoots(1, 2, 5) == "Tidak ada"
    ) {cout << "Quadratic Equation Solver Test 2:
SUCCESS" << endl;} else {cout << "Quadratic
Equation Solver Test 2: FAILED" << endl;}
    if (
        realRoots(1, -4, 4) == "2"
    ) {cout << "Quadratic Equation Solver Test 3:
SUCCESS" << endl;} else {cout << "Quadratic
Equation Solver Test 3: FAILED" << endl;}

```

Bagian ini untuk mengetes permasalahan problem 2: Quadratic Equation Solver.

Output yang dihasilkan



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Code
1\Praktikum Pemrograman\ugas Chapter 2\ test_cases
Payslip Calculation Test 1: SUCCESS
Payslip Calculation Test 2: SUCCESS
Payslip Calculation Test 3: SUCCESS
-----
Quadratic Equation Solver Test 1: SUCCESS
Quadratic Equation Solver Test 2: SUCCESS
Quadratic Equation Solver Test 3: SUCCESS
[Done] exited with code=0 in 1.267 seconds
```

Kesimpulan

Pada praktikum pemrograman ini diberikan dua permasalahan dan satu permasalahan mengenai *test code*. Permasalahan itu meliputi: pencetakan *payslip* dengan pajak tetap sebesar 20%, penghitungan penyelesaian persamaan kuadrat dengan kasus-kasusnya, serta pembuatan test code untuk kedua permasalahan sebelumnya. Saat pengerjaan, disimpan segala file yang dibutuhkan pada tautan Github yang ada pada identitas di atas. Saat mengoding menggunakan banyak referensi di internet meliputi: GeeksForGeeks, StackOverflow, serta ChatGPT untuk penyelesaian dan penyesuaian dengan output yang diinginkan atau diperintahkan.

Pengerjaan Problem 1: Payslip Calculation menggunakan metode cetak dan menghitung pajak yang dibutuhkan. Pengerjaan Problem 2: Quadratic Equation Solver menggunakan metode cetak dan menghitung diskriminan serta penggunaan rumus Kuadrat.

Saat mengetes koding, kedua masalah tersebut menggunakan fungsi yang disesuaikan input dan outputnya. Kedua masalah tersebut disesuaikan dalam bentuk fungsi dari file-file yang tertera pada tautan Github. File pada tautan Github tersebut merupakan proses pengerjaan yang lebih lengkap dan sistematis dengan memperhatikan *prompt*.