

Jawaban

1. Tuliskan 4 perintah berbeda yang mengurangi nilai n sebanyak 1.

- `angka = angka -1 ;`
- `angka-- ;`
- `--angka ;`
- `angka -=1 ;`

2. Evaluasi setiap ekspresi berikut, misalkan nilai $m=24$ dan nilai $n=7$

- $m - 8 - n \Rightarrow 24 - 8 - 7 = 9$
- $m = n = 3 \Rightarrow m=3 \text{ dan } n=3$
- $m \% n \Rightarrow 3$
- $m \% n++ \Rightarrow 3$
- $m \% ++n \Rightarrow 0$
- $++m - n-- \Rightarrow 18$
- $m += n -= 2 \Rightarrow m=29 \text{ dan } n=5$

Jawaban

3. Tuliskan jadi satu perintah dari dua operasi berikut (*lihat Operator assignment*):
- $z = z - (x+y) \Rightarrow z = z - (x+y)$
 - $y = y+1 \Rightarrow y += 1$
4. Tentukan hasilnya dan uraikan perintah berikut dengan nilai $m=5$ dan $n=2$
- a. $m *= n++; \Rightarrow 10$
 - b. $m += --n; \Rightarrow 6$

Perintah $m *= n++$;

$n++$ adalah operator post-increment. Ini berarti nilai n digunakan dalam operasi terlebih dahulu, dan kemudian n di-increment.

Langkah-langkahnya:

1. Gunakan nilai n yang saat ini adalah 2 dalam operasi $m *= n++$.
2. $m *= n$ berarti $m = m * n$. Dengan nilai awal $m = 5$ dan $n = 2$, perintah ini akan menghitung $m = 5 * 2 = 10$.
3. Setelah operasi, n di-increment menjadi 3.
4. Jadi, setelah perintah ini:

• **$m = 10$**

• **$n = 3$**

Perintah $m += --n$;

$--n$ adalah operator pre-decrement. Ini berarti nilai n dikurangi terlebih dahulu, dan kemudian nilai baru dari n digunakan dalam operasi.

Langkah-langkahnya:

1. $--n$ mengurangi nilai n dari 2 menjadi 1, dan kemudian nilai 1 digunakan dalam operasi.
2. $m += --n$ berarti $m = m + --n$. Dengan nilai $m = 5$ dan $--n = 1$, perintah ini akan menghitung $m = 5 + 1 = 6$.
3. Jadi, setelah perintah ini:

• **$m = 6$**

• **$n = 1$**

Jawaban

5. Tentukan hasil dari perintah-perintah berikut:

- `int x, y, z;`
- `x = y = z = 5; => x=5, y=5, z=5`
- `x *= y += z -= 1; => 45`

Perhatikan urutan pengerjaan:

- **`z -= 1`**: Mengurangi nilai z dengan 1.
 - z sebelumnya adalah 5, jadi `z -= 1` menjadi 4.
 - Sekarang `z = 4`.
- **`y += z`**: Menambahkan nilai z yang baru ke y.
 - y sebelumnya adalah 5, dan z yang baru adalah 4.
 - Jadi `y += 4` menjadi 9.
 - Sekarang `y = 9`.
- **`x *= y`**: Mengalikan nilai x dengan y yang baru.
 - x sebelumnya adalah 5, dan y yang baru adalah 9.
 - Jadi `x *= 9` menjadi 45.
 - Sekarang `x = 45`.

6. Seorang programmer menuliskan pernyataan sebagai berikut, setelah mendefinisikan semua nama yang dipakai sebagai tipe real/float
- $\text{dua} = \text{tujuh} + \text{lima}$
 - $\text{Cinta} = \text{Toto} + \text{Tita}$
- Apa komentar anda ?
- Jawaban : Kedua pernyataan tersebut valid, namun nama variabel seperti dua, tujuh, lima, Cinta, Toto, dan Tita tidak sejalan dengan konvensi penamaan yang umum. Dalam pemrograman praktis terbaik, nama variabel sebaiknya deskriptif dan mencerminkan tujuan atau tipe data yang diwakili.

Soal 7

Dibaca nama dan jam kerja pegawai, harus dihitung honor pegawai tersebut jika upahnya perjam adalah Rp. 5.000,- Perhatikan bahwa upah perjam setiap pegawai tidak sama, dan perubahan upah tidak sesering perubahan kurs.

Algoritma Hitung_Honor_Pegawai

{Menghitung honor pegawai berdasarkan upah per jam}

Deklarasi

string nama

int jamKerja

int upahPerJam , honor

Deskripsi

INPUT nama

INPUT jamKerja

upahPerJam = 5000

honor = jamKerja * upahPerJam

OUTPUT "Nama pegawai: " + nama

OUTPUT "Honor pegawai: " + honor

```
/* Nama program      : Hitung_Honor_Pegawai
   Nama              : Erick
   NPM               :
   Tanggal buat      : 13 Sep 2024
   Deskripsi         : Menghitung honor pegawai
   berdasarkan upah per jam
   ***** /
#include <iostream>
#include <string>
using namespace std;

int main() {
    string nama;
    int jamKerja;
    const int upahPerJam = 5000;

    cout << "Masukkan nama pegawai: ";
    getline(cin, nama);
    cout << "Masukkan jam kerja: ";
    cin >> jamKerja;

    int honor = jamKerja * upahPerJam;

    cout << "Nama pegawai: " << nama << endl;
    cout << "Honor pegawai: Rp " << honor << endl;

    return 0;
}
```

Soal 8

Dibaca 3 buah bilangan bulat yang mewakili tiga buah tahanan dalam Ohm : R1, R2 dan R3, harus dihitung dan dituliskan tahanan total yang dihasilkan jika ketiganya dipasang seri dan parallel

Algoritma Hitung_Tahanan_Total

{Menghitung Tahanan Total berdasarkan tiga buah tahanan}

Deklarasi

int R1, R2, R3

Deskripsi

INPUT R1

INPUT R2

INPUT R3

$\text{tahananSeri} = R1 + R2 + R3$

$\text{tahananParallel} = 1 / ((1 / R1) + (1 / R2) + (1 / R3))$

OUTPUT "Tahanan total seri: " + tahananSeri

OUTPUT "Tahanan total paralel: " + tahananParallel

```
/* Nama program      : Hitung_Tahanan_Total
   Nama              : Erick
   NPM              :
   Tanggal buat      : 13 Sep 2024
   Deskripsi         : Menghitung Tahanan Total berdasarkan
tiga buah tahanan
   ***** /
#include <iostream>
using namespace std;

int main() {
    int R1, R2, R3;

    cout << "Masukkan tahanan R1 (Ohm): ";
    cin >> R1;
    cout << "Masukkan tahanan R2 (Ohm): ";
    cin >> R2;
    cout << "Masukkan tahanan R3 (Ohm): ";
    cin >> R3;

    int tahananSeri = R1 + R2 + R3;
    double tahananParallel = 1.0 / ((1.0 / R1) + (1.0 / R2) + (1.0 / R3));

    cout << "Tahanan total seri: " << tahananSeri << " Ohm" << endl;
    cout << "Tahanan total paralel: " << tahananParallel << " Ohm" << endl;

    return 0;
}
```