**TUGAS DESAIN DAN ANALISA ALGORITMA (KJ001)**

**Dosen : Muhamad Bahrul Ulum,S.Kom,M.Kom**

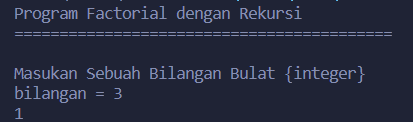


**DISUSUN OLEH:**

Gavin Adlan Hidayat – 20220801093

1. Factorial
2. #include <iostream>
3. using namespace std;
4. int factorial (int n){
5. if(n==1){
6. *return* 1;
7. } else {
8. *return* n=factorial(n=1);
9. }
10. }
11. int main(){
12. int bil;
13. cout<<"Program Factorial dengan Rekursi"<<endl;
14. cout<<"=========================================="<<endl;
15. cout<<endl;
16. cout<<"Masukan Sebuah Bilangan Bulat {integer}"<<endl;
17. cout<<"bilangan = ";cin>>bil;
18. cout<<factorial(bil);
19. }

Output:



2. Fibonaci

#include <iostream>

int fibonacci(int n) {

    if (n <= 1)

*return* n;

    else

*return* fibonacci(n - 1) + fibonacci(n - 2);

}

int main() {

    int n;

    std::cout << "Masukkan jumlah bilangan Fibonacci yang ingin ditampilkan: ";

    std::cin >> n;

    std::cout << "Urutan Fibonacci: ";

    for (int i = 0; i < n; i++) {

        std::cout << fibonacci(i) << " ";

    }

*return* 0;

}

Output:



3. Combinasi

#include <iostream>

*// Fungsi untuk menghitung kombinasi*

unsigned long long combination(int n, int r) {

    if (r > n || n < 0 || r < 0)

*return* 0;

*// Menggunakan metode iteratif*

    unsigned long long result = 1;

    for (int i = 1; i <= r; i++) {

        result \*= n;

        result /= i;

        n--;

    }

*return* result;

}

int main() {

    int n, r;

    std::cout << "Masukkan nilai n: ";

    std::cin >> n;

    std::cout << "Masukkan nilai r: ";

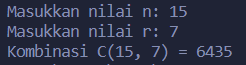
    std::cin >> r;

    std::cout << "Kombinasi C(" << n << ", " << r << ") = " << combination(n, r) << std::endl;

*return* 0;

}

Output:



4. Permutasi

#include <iostream>

using namespace std;

unsigned long long factorial(int n) {

    if (n == 0 || n == 1)

*return* 1;

    unsigned long long result = 1;

    for (int i = 2; i <= n; i++) {

        result \*= i;

    }

*return* result;

}

unsigned long long permutation(int n, int r) {

    if (r > n)

*return* 0;

    unsigned long long numerator = factorial(n);

    unsigned long long denominator = factorial(n - r);

*return* numerator / denominator;

}

unsigned long long combination(int n, int r) {

    if (r > n)

*return* 0;

    unsigned long long numerator = factorial(n);

    unsigned long long denominator = factorial(r) \* factorial(n - r);

*return* numerator / denominator;

}

int main() {

    int n, r;

    cout << "Masukkan nilai n: ";

    cin >> n;

    cout << "Masukkan nilai r: ";

    cin >> r;

    if (n < 0 || r < 0) {

        cout << "Masukkan bilangan non-negatif." << endl;

    } else {

        unsigned long long permutationResult = permutation(n, r);

        unsigned long long combinationResult = combination(n, r);

        cout << n << "P" << r << " = " << permutationResult << endl;

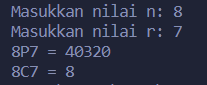
        cout << n << "C" << r << " = " << combinationResult << endl;

    }

*return* 0;

}

Output:



5. deret N

#include <iostream>

using namespace std;

int main() {

    int n;

    cout << "Masukkan jumlah angka dalam deret: ";

    cin >> n;

    if (n < 1) {

        cout << "Masukkan bilangan bulat positif." << endl;

*return* 1; *// Exit the program with an error code*

    }

    cout << "Deret " << n << " angka pertama: ";

    int totalSum = 0;

    for (int i = 0; i < n; i++) {

        int num = n + i;

        totalSum += num;

        cout << num;

        if (i != n - 1) {

            cout << ", ";

        }

    }

    cout << endl;

    cout << "Total sum: " << totalSum << endl;

*return* 0;

}

Output:



6. deret 2n

#include <iostream>

using namespace std;

int main() {

    int n;

    cout << "Masukkan jumlah angka dalam deret: ";

    cin >> n;

    if (n < 1) {

        cout << "Masukkan bilangan bulat positif." << endl;

    } else {

        cout << "Deret " << n << " angka pertama (2^n): ";

        for (int i = 0; i < n; i++) {

            cout << (1 << i); *// Equivalent to 2^i*

            if (i != n - 1) {

                cout << ", ";

            }

        }

        cout << endl;

    }

*return* 0;

}

Output:

