

**LAPORAN PRAKTIKUM**  
**ALGORITMA PEMROGRAMAN**  
**“Post Test Pertemuan III: Perulangan”**

Diajukan untuk memenuhi salah satu praktikum Mata Kuliah Algoritma Pemrograman yang  
di ampu oleh:

Dr. Ardiansyah S.T., M.Cs



Disusun Oleh:

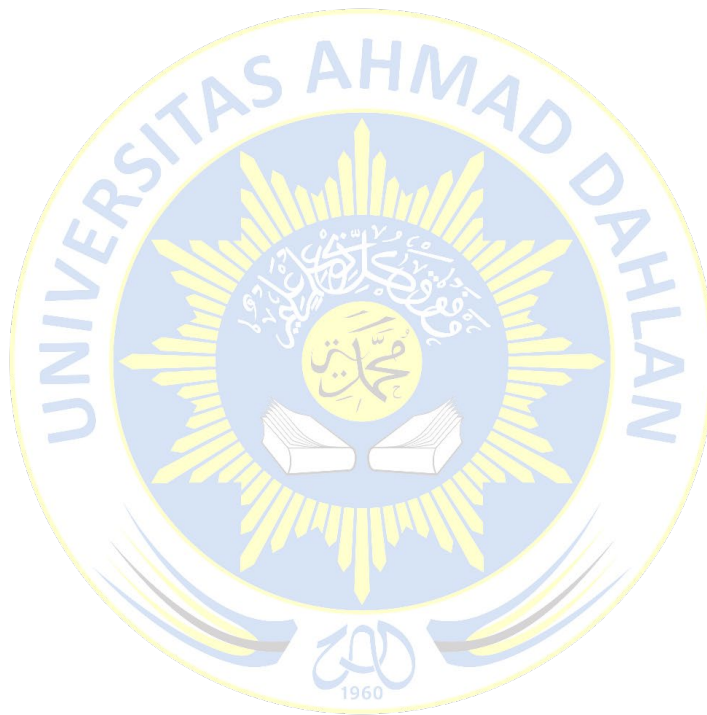
Mohammad Farid Hendianto 2200018401

A / Jumat 13.30 Lab. Jaringan

**PROGRAM STUDI INFORMATIKA**  
**UNIVERSITAS AHMAD DAHLAN**  
**FAKULTAS TEKNOLOGI INDUSTRI**  
**TAHUN 2023**

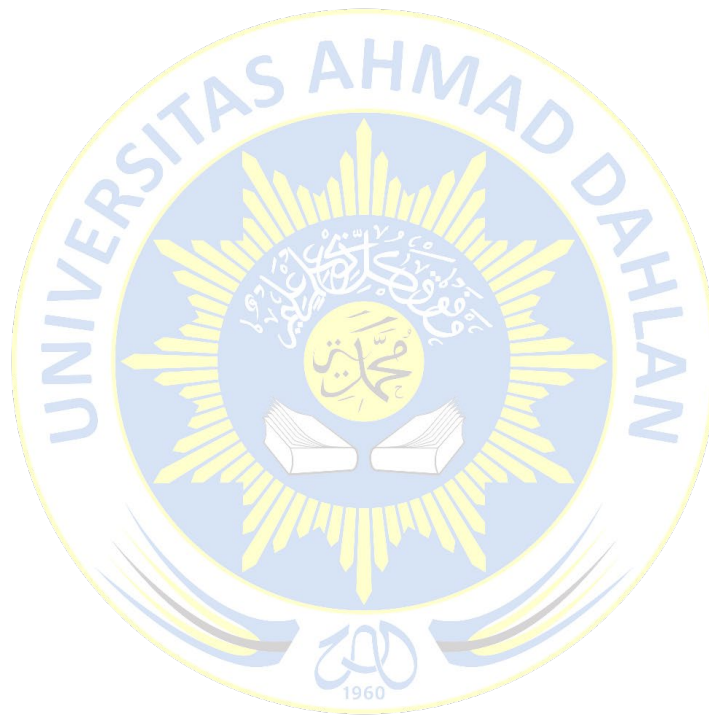
## DAFTAR ISI

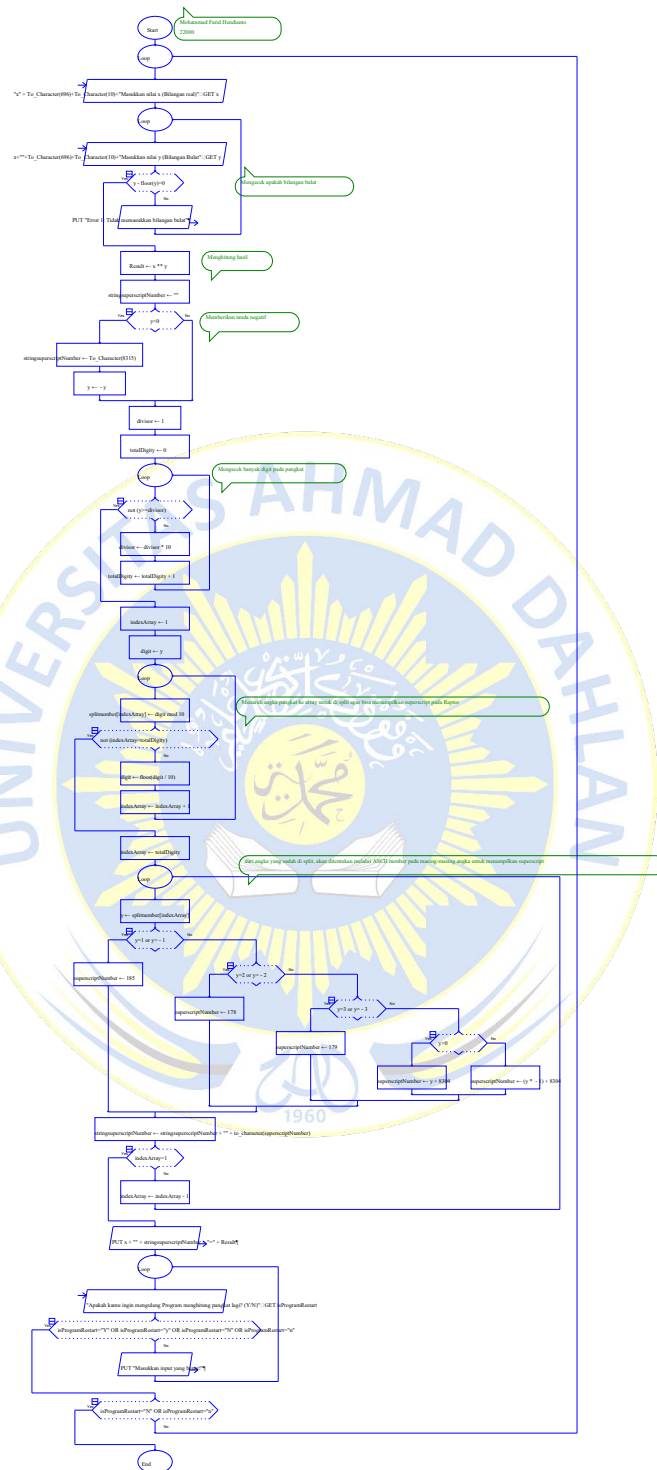
1. Buat lah flowchart untuk menghitung nilai  $x(\text{pangkat})y$  dengan  $x$  bilangan real dan  $y$  bilangan bulat baik negatif maupun positif..... 3
2. Seperti nomor 1, gunakan subprogram dalam flowchart untuk menghitung nilai  $x(\text{pangkat})y$  dengan  $x$  bilangan real dan  $y$  bilangan bulat baik negatif maupun positif..... 7
3. Konversikan hasil dari flowchart nomor 1 dan 2 menjadi program C++..... 12



1. Buat lah flowchart untuk menghitung nilai  $x^y$  dengan  $x$  bilangan real dan  $y$  bilangan bulat baik negatif maupun positif.

Berikut adalah flowchart untuk menghitung nilai  $x^y$ .





Gambar 1 Flowchart hitungPangkat

Berikut adalah contoh outputnya:



```

MasterConsole
Font Font Size Edit Help
2^1=2
2^2=4
2^3=8
2^4=16
2^5=32
2^6=64
2^7=128
2^8=256
2^9=512
2^10=1024
2^11=2048
2^12=4096
2^13=8192
2^-1=0.5000
2^-2=0.2500
2^-3=0.1250
2^-4=0.06250
2^-5=0.03125
2^-6=0.01563
2^-7=0.007813
2^-8=0.003906
2^-9=0.001953
2^-10=0.0009766
2^-11=0.0004883
----Run complete. 963 symbols evaluated.----

```

Gambar 2 Output flowchart hitungPangkat. (Sumber: Penulis)

Flowchart tersebut adalah sebuah program yang digunakan untuk menghitung pangkat suatu bilangan, dengan meminta input nilai x dan y (y sebagai pangkat) dari pengguna dan kemudian menghasilkan output nilai pangkat dari x.

Berikut adalah alur kerja dari flowchart tersebut:

- 1) Program meminta input nilai x (bilangan real) dari pengguna.
- 2) Program meminta input nilai y (bilangan bulat) dari pengguna. Jika input y bukan bilangan bulat, program akan menampilkan pesan error dan meminta input kembali hingga pengguna memasukkan bilangan bulat.
- 3) Program menghitung nilai pangkat dari x dengan menggunakan operator "\*\*\*" pada bahasa pemrograman yang digunakan.
- 4) Program melakukan pengecekan apakah pangkat y negatif atau tidak. Jika iya, program akan menambahkan karakter superscript pada tampilan output sebagai tanda bahwa hasilnya adalah pecahan.
- 5) Program menghitung jumlah digit pada bilangan y untuk mengetahui banyaknya angka yang perlu ditampilkan sebagai superscript pada tampilan output.
- 6) Program memecah angka y menjadi angka-angka yang lebih kecil dan menyimpannya dalam array.
- 7) Program mengonversi setiap angka yang ada di array menjadi karakter superscript sesuai dengan aturan yang telah ditentukan.
- 8) Program menampilkan output berupa nilai pangkat dari bilangan x beserta karakter superscript y (jika ada) pada layar.

- 9) Program menanyakan apakah pengguna ingin mengulang program ini atau tidak. Jika pengguna memilih "Y" atau "y", program akan kembali ke langkah 1. Jika pengguna memilih "N" atau "n", program akan berhenti.

Fitur-fitur dari program ini antara lain:

- Program memastikan input bilangan  $y$  yang dimasukkan pengguna adalah bilangan bulat. Jika tidak, maka akan muncul pesan error hingga pengguna memasukkan bilangan bulat.
- Program menambahkan karakter superscript pada tampilan output jika pangkat  $y$  negatif untuk menandakan bahwa hasilnya adalah pecahan.
- Program mengonversi angka-angka dalam array menjadi karakter superscript sesuai dengan aturan yang telah ditentukan sehingga membuat tampilan output lebih informatif dan mudah dibaca.
- Program memungkinkan pengguna untuk mengulang program kembali jika mereka ingin menghitung pangkat bilangan lain tanpa harus memulai program lagi dari awal.

Pada program tersebut terdapat beberapa bilangan bulat yang dipakai untuk mengekstraksi nilai digit dari bilangan pangkat  $y$ , kemudian mengonversi angka-angka tersebut menjadi karakter superscript sesuai dengan aturan yang telah ditentukan.

Berikut adalah aturan konversi dari angka menjadi karakter superscript yang diterapkan pada program tersebut:

- Jika digit bernilai 1 atau -1, maka karakter superscript yang ditampilkan adalah <sup>1</sup> (nilai ASCII 185).
- Jika digit bernilai 2 atau -2, maka karakter superscript yang ditampilkan adalah <sup>2</sup> (nilai ASCII 178).
- Jika digit bernilai 3 atau -3, maka karakter superscript yang ditampilkan adalah <sup>3</sup> (nilai ASCII 179).

Jika digit lebih besar dari 3 atau lebih kecil dari -3, maka karakter superscript yang ditampilkan dihasilkan dengan menjumlahkan nilai digit tersebut dengan nilai ASCII 8304. Misalnya, jika digit bernilai 4, maka karakter superscript yang ditampilkan adalah <sup>4</sup> (nilai ASCII 8308). Sedangkan jika digit bernilai -4, maka karakter superscript yang ditampilkan adalah <sup>-4</sup> (nilai ASCII 8315 dan 8308).

Dalam program tersebut, setiap nilai digit akan dimasukkan ke dalam variabel  $y$ , kemudian dilakukan pengecekan menggunakan conditional statements untuk menentukan karakter superscript yang tepat. Karakter superscript yang dipilih kemudian dikonversi menjadi kode ASCII menggunakan syntax CharacterVal atau To\_Character dan disimpan dalam variabel superscriptNumber. Setelah itu, karakter superscript tersebut ditambahkan ke dalam variabel stringSuperscriptNumber yang akan digunakan untuk menyusun tampilan output.

Dengan aturan konversi seperti ini, maka karakter superscript yang ditampilkan akan sama untuk semua digit angka yang bersifat superscript pada program tersebut, yaitu <sup>1</sup>, <sup>2</sup>, <sup>3</sup>, atau karakter yang dihasilkan dengan menjumlahkan nilai digit tersebut dengan nilai ASCII 8304.

To\_Character(ASCII) adalah fungsi pada bahasa pemrograman Ada yang digunakan untuk mengubah kode ASCII menjadi karakter yang sesuai. Berikut ini adalah contoh superscript 0-9 pada program di atas dalam bentuk syntax To\_Character:

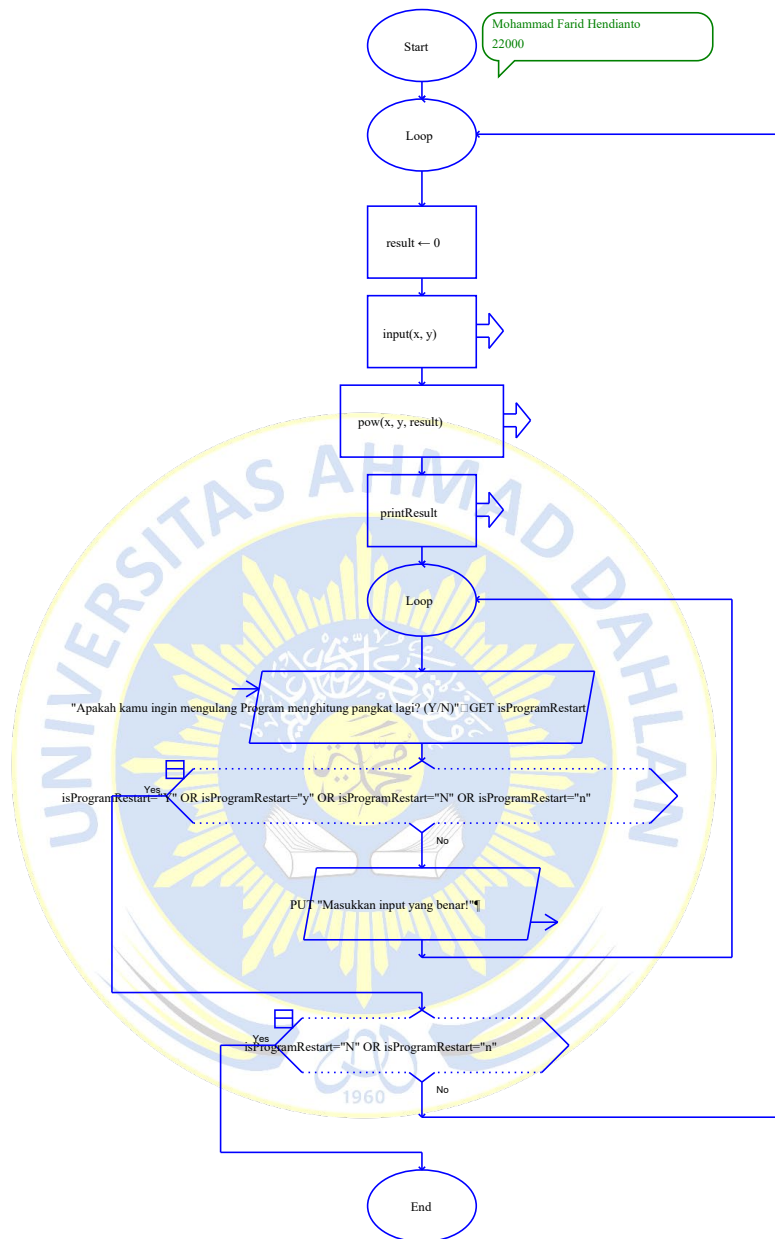
- Superscript <sup>0</sup> : To\_Character(8304)
- Superscript <sup>1</sup> : To\_Character(185)
- Superscript <sup>2</sup> : To\_Character(178)
- Superscript <sup>3</sup> : To\_Character(179)
- Superscript <sup>4</sup> : To\_Character(8308)
- Superscript <sup>5</sup> : To\_Character(8309)
- Superscript <sup>6</sup> : To\_Character(8310)
- Superscript <sup>7</sup> : To\_Character(8311)
- Superscript <sup>8</sup> : To\_Character(8312)
- Superscript <sup>9</sup> : To\_Character(8313)

Nilai kode ASCII untuk superscript berkisar dari 8304 hingga 8313 untuk angka 0 hingga 9. Oleh karena itu, nilai kode tersebut dapat langsung diubah menjadi karakter menggunakan fungsi To\_Character().

2. Seperti nomor 1, gunakan subprogram dalam flowchart untuk menghitung nilai  $x^y$  dengan  $x$  bilangan real dan  $y$  bilangan bulat baik negatif maupun positif.

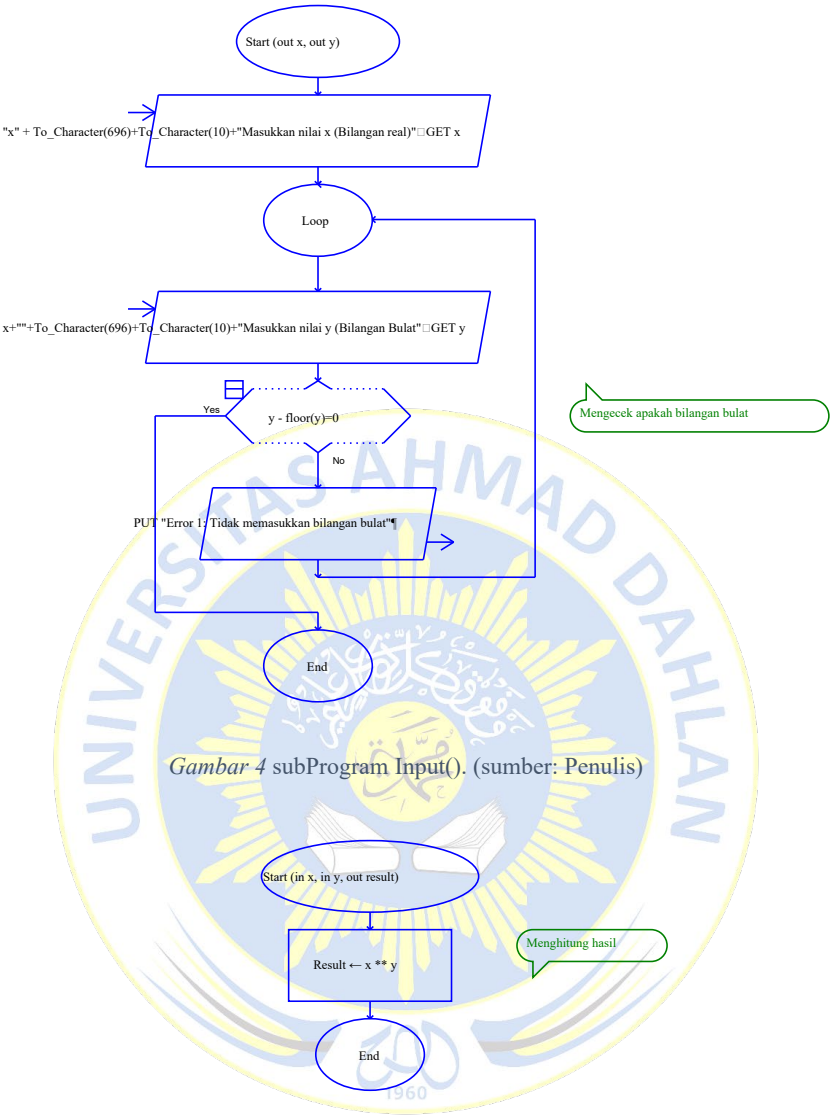
Berikut adalah flowchart untuk menghitung nilai  $x^y$  dengan subProgram.





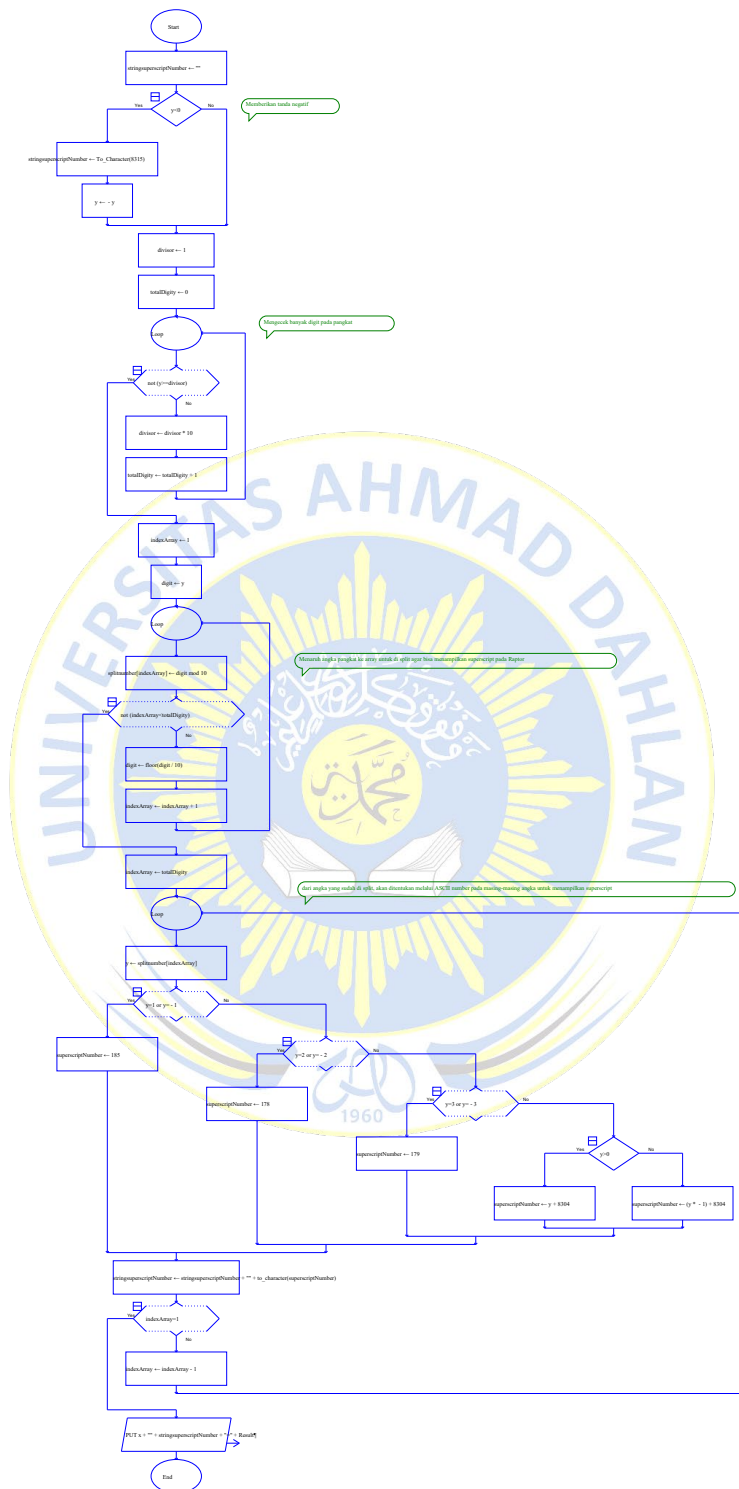
Gambar 3 Flowchart utama hitungPangkat. (sumber: Penulis)





Gambar 4 subProgram Input(). (sumber: Penulis)

Gambar 5 subProgram pow(). (sumber: Penulis)



Gambar 6 subProgram Output(). (Sumber: Penulis)

Flowchart tersebut merupakan implementasi dari subprogram untuk menghitung pangkat suatu bilangan. Subprogram ini terdiri dari dua prosedur yaitu input dan pow.

Prosedur input digunakan untuk meminta input dari pengguna berupa nilai x dan y. Variabel x didefinisikan sebagai out variable yang artinya ia akan diisi dengan nilai yang dimasukkan oleh pengguna. Begitu juga dengan variabel y. Selain itu, pada prosedur input dilakukan pemeriksaan apakah nilai y yang dimasukkan pengguna merupakan bilangan bulat atau tidak. Jika bukan, maka program akan menampilkan pesan error dan meminta pengguna untuk memasukkan bilangan bulat.

Prosedur pow memiliki dua parameter masukan, yaitu x dan y, serta satu parameter keluaran result. Prosedur ini bertugas untuk menghitung hasil perpangkatan x dengan y menggunakan operator `**` dan hasilnya akan disimpan pada variabel result.

Setelah kedua prosedur tersebut dijelaskan, kita dapat melihat bahwa subprogram flowchart\_hitung\_pangkat menggunakan kedua prosedur tersebut untuk menghitung pangkat suatu bilangan.

Pertama-tama, subprogram meminta input dari pengguna berupa nilai x dan y menggunakan prosedur input. Nilai x dan y tersebut kemudian digunakan sebagai parameter masukan untuk prosedur pow sehingga didapatkan hasil perpangkatan yang disimpan pada variabel result.

Selanjutnya, subprogram melakukan konversi nilai y menjadi superscript number yang ditampilkan pada output. Hal ini dilakukan dengan membagi nilai y menjadi angka-angka satuan dan mengkonversinya menjadi karakter superscript menggunakan kode ASCII.

Akhirnya, subprogram menampilkan hasil perpangkatan beserta superscript number dan meminta user untuk memilih apakah ingin mengulang program atau tidak. Jika user memilih untuk mengulang, maka program akan kembali meminta input nilai x dan y. Namun jika user tidak ingin mengulang, program akan berakhir.

Perbedaan utama antara kedua flowchart tersebut adalah penggunaan subProgram pada flowchart yang kedua. Dalam flowchart yang kedua, beberapa bagian dari kode telah dipindahkan ke dalam subProgram untuk memperjelas dan mempermudah struktur program.

Pada flowchart pertama, semua operasi dilakukan di dalam satu procedure, sehingga sulit untuk membedakan fungsionalitas masing-masing bagian dari program. Hal ini dapat menyulitkan pemeliharaan dan pengembangan suatu program di masa depan.

Sementara itu, dalam flowchart yang kedua, subProgram input digunakan untuk mengambil input dari user. SubProgram pow digunakan untuk memangkatkan bilangan x dengan pangkat y. Kode utama hanya berisi loop utama yang mengulang program hingga user memilih untuk keluar. Penggunaan subProgram ini membuat struktur program menjadi lebih jelas dan mempermudah pemeliharaan serta pengembangan di masa depan.

Dengan menggunakan subProgram, kita juga dapat memisahkan fungsi-fungsi yang berbeda ke dalam unit-unit terpisah, yang memudahkan pembagian tugas pada tim pengembang. Jika ada masalah pada salah satu unit, maka hanya unit tersebut yang perlu diperbaiki, tanpa harus memodifikasi seluruh program.

Secara keseluruhan, penggunaan subProgram membuat flowchart lebih mudah dimengerti dan diubah, serta mempermudah perawatan dan pengembangan program di masa depan.

### 3. Konversikan hasil dari flowchart nomor 1 dan 2 menjadi program C++.

Program ini bertujuan untuk menampilkan nilai pangkat dari suatu bilangan real  $x$  dengan bilangan bulat  $y$ . Alur program dimulai dengan meminta input nilai  $x$  dan  $y$  dari pengguna menggunakan fungsi `cin`. Selanjutnya, program melakukan pengecekan apakah nilai  $y$  yang dimasukkan merupakan bilangan bulat atau tidak dengan menggunakan pernyataan `if` pada looping `do-while`.

Jika nilai  $y$  yang dimasukkan bukan bilangan bulat, maka program akan menampilkan pesan error dan meminta pengguna untuk memasukkan nilai  $y$  kembali hingga benar. Setelah nilai  $y$  yang dimasukkan valid, program akan menghitung hasil pangkat menggunakan fungsi `pow()` dan menyimpannya dalam variabel `result`.

Selanjutnya, program akan mengecek apakah  $y$  bernilai negatif atau positif. Jika  $y$  negatif, maka program akan menambahkan superscript `"-"` setelah nilai  $x$  pada output. Kemudian, program akan mengubah nilai  $y$  menjadi array of integer `splitnumber` dengan cara memisahkan setiap digit angka pada nilai  $y$  dan menyimpannya ke dalam elemen array. Selanjutnya, program akan mengubah nilai `splitnumber` menjadi superscript dengan menggunakan pernyataan `switch case`.

Terakhir, program akan menampilkan output berupa nilai  $x$  beserta superscript  $y$  dan hasil pangkat dari  $x^y$ . Output ditampilkan menggunakan fungsi `cout`.

Berikut adalah kodingan C++ `hitungPangkat` tanpa subprogram.

```

1  #include <iostream>
2  #include <windows.h>
3  #include <string.h>
4  #include <cmath>
5  using namespace std;
6  // menampilkan pangkat dengan x sebagai bilangan real dan y sebagai bilangan bulat
7
8  int main(){
9      double x,y,result;
10     cout<<"Masukkan nilai x (Bilangan real) : ";cin>>x;
11     do{
12         cout<<"Masukkan nilai y (Bilangan bulat) : ";cin>>y;
13         if (y-floor(y)!=0){
14             cout<<"Error 1: Tidak memasukkan bilangan bulat"<<endl;y=0;
15         }
16     }
17     while (y==0);
18
19     result=pow(x,y);
20     // mengecek apakah negatif atau positif pada bilangan y
21     // jika negatif akan ditambahkan superscript -
22     cout<<x;
23     if (y<0){
24         SetConsoleOutputCP(65001);cout<<"-";
25     }
26     int totalDigity=0;
27     int digit=y;
28     if (y<0){
29         digit*=-1;
30     }
31     float divisor=1;
32     while (digit/divisor>=1){
33         totalDigity++;
34         divisor*=10;
35     }
36     int splitnumber[totalDigity];
37     //mengubah tipe data y menjadi int
38
39     for (int i=0;i<totalDigity;i++){
40         splitnumber[i]=digit%10;
41         digit/=10;
42     }
43
44     // mengubah split number menjadi superscript
45     while (totalDigity>0){
46         switch (splitnumber[totalDigity-1]){
47             case 0: SetConsoleOutputCP(65001);cout<<"^0";break;
48             case 1: SetConsoleOutputCP(65001);cout<<"^1";break;
49             case 2: SetConsoleOutputCP(65001);cout<<"^2";break;
50             case 3: SetConsoleOutputCP(65001);cout<<"^3";break;
51             case 4: SetConsoleOutputCP(65001);cout<<"^4";break;
52             case 5: SetConsoleOutputCP(65001);cout<<"^5";break;
53             case 6: SetConsoleOutputCP(65001);cout<<"^6";break;
54             case 7: SetConsoleOutputCP(65001);cout<<"^7";break;
55             case 8: SetConsoleOutputCP(65001);cout<<"^8";break;
56             case 9: SetConsoleOutputCP(65001);cout<<"^9";break;
57         }
58         totalDigity--;
59     }
60     cout<<" = "<<result<<endl;
61 }

```

Gambar 7 Kodingan untuk hitungPangkat. (Sumber :penulis)

Program di atas adalah sebuah program yang dibuat untuk menghitung pangkat dari bilangan real (x) dengan bilangan bulat (y), dan menampilkan hasilnya dengan format output yang menggunakan superscript untuk digit-digit pada y. Berikut ini adalah penjelasan detail mengenai alur program dan syntax-syntax yang mungkin asing:

### Library

---

```
#include <iostream>

#include <windows.h>

#include <string.h>

#include <cmath>
```

---

Syntax `#include` merupakan perintah untuk memasukkan library yang akan digunakan dalam program. Pada program di atas terdapat beberapa library yang digunakan, yaitu `iostream`, `windows.h`, `string.h`, dan `cmath`.

### Deklarasi Variabel

---

```
double x,y,result;
```

---

Syntax di atas digunakan untuk mendeklarasikan variabel yang akan digunakan dalam program. Dalam kasus ini, terdapat tiga variabel yang digunakan, yaitu `x`, `y`, dan `result`.

### Input User

---

```
cout<<"Masukkan nilai x (Bilangan real) : ";cin>>x;
```

---

Syntax di atas digunakan untuk meminta input dari user berupa bilangan real `x`. Keyword `cout` digunakan untuk menampilkan text pada console, sedangkan keyword `cin` digunakan untuk mengambil input dari user.

### Validasi Input Y

---

```
do{

    cout<<"Masukkan nilai y (Bilangan bulat) : ";cin>>y;

    if (y-floor(y)!=0){

        cout<<"Error 1: Tidak memasukkan bilangan bulat"<<endl;y=0;
```

---



```

    }
}
while (y==0);

```

Syntax di atas digunakan untuk melakukan validasi input y. Jika user memasukkan nilai y yang bukan bilangan bulat, maka program akan menampilkan pesan error "Error 1: Tidak memasukkan bilangan bulat" dan meminta user menginput kembali nilai y yang benar.

### Perhitungan Pangkat

```
result=pow(x,y);
```

Syntax di atas digunakan untuk menghitung nilai pangkat dari bilangan real x dengan bilangan bulat y menggunakan fungsi pow(). Hasil perhitungan disimpan dalam variabel result.

### Menampilkan Output X

```
cout<<x;
```

Syntax di atas digunakan untuk menampilkan nilai x pada output.

### Menampilkan Superscript

```

if (y<0){
    SetConsoleOutputCP(65001); cout<<"-";
}

```

Syntax di atas digunakan untuk mengecek apakah nilai bilangan bulat y negatif atau positif. Jika y negatif, maka program akan menambahkan superscript "-" pada output menggunakan fungsi SetConsoleOutputCP().

### Menghitung Jumlah Digit Y

```

int totalDigity=0;

int digit=y;

if (y<0){

```



```

    digit*=-1;
}
float divisor=1;
while (digit/divisor>=1){
    totalDigity++;
    divisor*=10;
}

```

---

Syntax di atas digunakan untuk menghitung jumlah digit pada bilangan bulat y. Pertama-tama nilai y dicek terlebih dahulu apakah bernilai negatif atau tidak. Kemudian dilakukan perulangan while untuk membagi y dengan angka 10 sampai hasil bagi kurang dari 1. Setiap kali loop dilakukan, totalDigit akan bertambah 1 dan nilai divisor akan dikalikan 10.

### Memecah Angka Y menjadi Digit-Digit

---

```

int splitnumber[totalDigity];
for (int i=0;i<totalDigity;i++){
    splitnumber[i]=digit%10;
    digit/=10;
}

```

---

Syntax di atas digunakan untuk memecah bilangan bulat y menjadi digit-digit dan menyimpannya dalam array splitnumber. Pada setiap iterasi, digit terakhir pada y diketahui dengan menggunakan modulus 10 dan disimpan pada indeks array splitnumber yang sesuai. Kemudian digit y dibagi dengan angka 10 agar digit selanjutnya dapat diperoleh.

### Mengubah Digit Menjadi Superscript

---

```

while (totalDigity>0){
    switch (splitnumber[totalDigity-1]){
        case 0: SetConsoleOutputCP(65001);cout<<"0";break;
        case 1: SetConsoleOutputCP(65001);cout<<"1";break;
        case 2: SetConsoleOutputCP(65001);cout<<"2";break;
    }
}

```

```

        case 3: SetConsoleOutputCP(65001);cout<<"3";break;
    case 4: SetConsoleOutputCP(65001);cout<<"4";break;
        case 5: SetConsoleOutputCP(65001);cout<<"5";break;
        case 6: SetConsoleOutputCP(65001);cout<<"6";break;
        case 7: SetConsoleOutputCP(65001);cout<<"7";break;
        case 8: SetConsoleOutputCP(65001);cout<<"8";break;
        case 9: SetConsoleOutputCP(65001);cout<<"9";break;
    }
    totalDignity--;
}

```

Syntax di atas digunakan untuk mengubah digit-digit pada bilangan bulat y menjadi superscript. Program menggunakan `switch` statement dan fungsi `SetConsoleOutputCP()` untuk menampilkan superscript.

### Output Hasil Perhitungan

```
cout<<" = "<<result<<endl;
```

Syntax di atas digunakan untuk menampilkan hasil perhitungan pangkat  $x^y$  beserta hasilnya pada output terakhir dengan format " $x^y =$  hasil perhitungan".

Berikut adalah hitungPangkat dengan subprogram.

```

1 #include <iostream>
2 #include <windows.h>
3 #include <string.h>
4 #include <cmath>
5 using namespace std;
6
7 class Pangkat {
8 private:
9     double x,y,result;
10
11 public:
12     void input() {
13         cout << "Masukkan nilai x (Bilangan real) : ";
14         cin >> x;
15         do {
16             cout << "Masukkan nilai y (Bilangan bulat) : ";
17             cin >> y;
18             if (y - floor(y) != 0) {
19                 cout << "Error 1: Tidak memasukkan bilangan bulat" << endl;
20                 y = 0;
21             }
22         } while (y == 0);
23     }
24
25     void calculate() {
26         result = pow(x, y);
27     }
28
29     void display() {
30         // mengecek apakah negatif atau positif pada bilangan y
31         // jika negatif akan ditambahkan superscript -
32         cout << x;
33         if (y < 0) {
34             SetConsoleOutputCP(65001);
35             cout << "-";
36         }
37
38         int totalDigit = 0;
39         int digit = y;
40         if (y < 0) {
41             digit *= -1;
42         }
43         float divisor = 1;
44         while (digit / divisor >= 1) {
45             totalDigit++;
46             divisor *= 10;
47         }
48         int splitNumber[totalDigit];
49         // mengubah tipe data y menjadi int
50
51         for (int i = 0; i < totalDigit; i++) {
52             splitNumber[i] = digit % 10;
53             digit /= 10;
54         }
55
56         // mengubah split number menjadi superscript
57         while (totalDigit > 0) {
58             switch (splitNumber[totalDigit - 1]) {
59                 case 0:
60                     SetConsoleOutputCP(65001);
61                     cout << "0";
62                     break;
63                 case 1:
64                     SetConsoleOutputCP(65001);
65                     cout << "1";
66                     break;
67                 case 2:
68                     SetConsoleOutputCP(65001);
69                     cout << "2";
70                     break;
71                 case 3:
72                     SetConsoleOutputCP(65001);
73                     cout << "3";
74                     break;
75                 case 4:
76                     SetConsoleOutputCP(65001);
77                     cout << "4";
78                     break;
79                 case 5:
80                     SetConsoleOutputCP(65001);
81                     cout << "5";
82                     break;
83                 case 6:
84                     SetConsoleOutputCP(65001);
85                     cout << "6";
86                     break;
87                 case 7:
88                     SetConsoleOutputCP(65001);
89                     cout << "7";
90                     break;
91                 case 8:
92                     SetConsoleOutputCP(65001);
93                     cout << "8";
94                     break;
95                 case 9:
96                     SetConsoleOutputCP(65001);
97                     cout << "9";
98                     break;
99             }
100             totalDigit--;
101         }
102         cout << " = " << result << endl;
103     }
104 };
105
106 int main() {
107     Pangkat pangkat;
108     pangkat.input();
109     pangkat.calculate();
110     pangkat.display();
111     return 0;
112 }

```

Gambar 8 hitungPangkat dengan subprogram dan OOP. (Sumber: Penulis)

Program ini juga bertujuan untuk menampilkan nilai pangkat dari suatu bilangan real  $x$  dengan bilangan bulat  $y$ , namun program ini menggunakan konsep OOP (Object Oriented Programming) dengan membuat class Pangkat. Dalam class Pangkat terdapat tiga subprogram atau method, yaitu `input()`, `calculate()`, dan `display()`.

Subprogram `input()` berfungsi untuk meminta input nilai  $x$  dan  $y$  dari pengguna dengan menggunakan fungsi `cin`. Kemudian, program melakukan pengecekan apakah nilai  $y$  yang dimasukkan merupakan bilangan bulat atau tidak dengan menggunakan pernyataan `if` pada looping `do-while`.

Jika nilai  $y$  yang dimasukkan bukan bilangan bulat, maka program akan menampilkan pesan error dan meminta pengguna untuk memasukkan nilai  $y$  kembali hingga benar. Setelah nilai  $y$  yang dimasukkan valid, nilai  $x$  dan  $y$  akan disimpan dalam variabel `private` di class Pangkat.

Subprogram `calculate()` berfungsi untuk menghitung hasil pangkat menggunakan fungsi `pow()` dengan parameter  $x$  dan  $y$  yang telah diinput sebelumnya. Hasil pangkat akan disimpan dalam variabel `private result`.

Subprogram `display()` berfungsi untuk menampilkan output berupa nilai  $x$  beserta superscript  $y$  dan hasil pangkat dari  $x^y$ . Output ditampilkan menggunakan fungsi `cout`. Subprogram ini juga memiliki kode yang sama dengan program sebelumnya untuk mengecek apakah  $y$  bernilai negatif atau positif, mengubah nilai  $y$  menjadi array of integer `splitnumber`, dan mengubah nilai `splitnumber` menjadi superscript dengan menggunakan pernyataan `switch case`.

Dalam subprogram `main()`, objek dari class Pangkat dibuat dan dipanggil subprogram `input()`, `calculate()`, dan `display()` menggunakan objek tersebut. Terakhir, program mengembalikan nilai 0 dan selesai.

Kedua kodingan memiliki tujuan yang sama untuk menampilkan nilai pangkat dari suatu bilangan real  $x$  dengan bilangan bulat  $y$ . Namun, terdapat perbedaan dalam pendekatan yang digunakan.

Pada kodingan pertama, program menggunakan pendekatan prosedural dengan hanya menggunakan satu subprogram `main()`. Dalam subprogram tersebut, input nilai  $x$  dan  $y$  diminta kepada pengguna melalui fungsi `cin`. Kemudian, dilakukan pengecekan apakah nilai  $y$  merupakan bilangan bulat atau tidak menggunakan looping `do-while`. Jika nilai  $y$  bukan bilangan bulat, maka akan ditampilkan pesan error dan input  $y$  akan diminta kembali. Selanjutnya, program menghitung hasil pangkat menggunakan fungsi `pow()`, mengecek apakah  $y$  bernilai negatif atau positif, mengubah nilai  $y$  menjadi array of integer `splitnumber`, dan mengubah nilai `splitnumber` menjadi superscript dengan menggunakan pernyataan `switch case`. Terakhir, program menampilkan output berupa nilai  $x$  beserta superscript  $y$  dan hasil pangkat dari  $x^y$ .

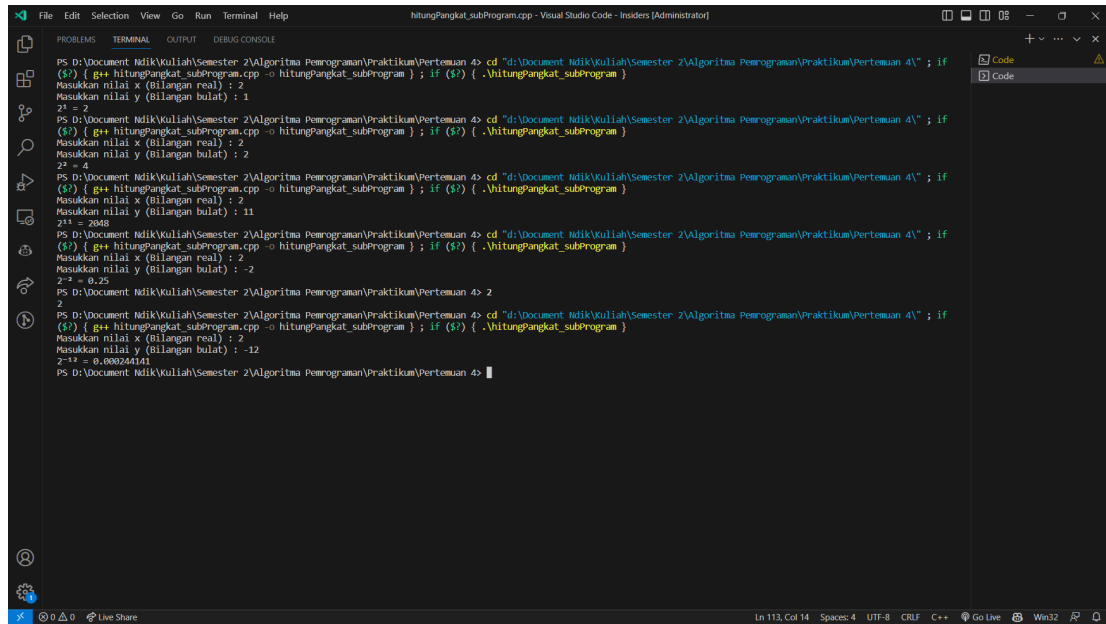
Sementara itu, pada kodingan kedua, program menggunakan pendekatan OOP (Object Oriented Programming) dengan membuat class Pangkat. Dalam class Pangkat terdapat tiga subprogram atau method, yaitu input(), calculate(), dan display(). Subprogram input() berfungsi untuk meminta input nilai x dan y dari pengguna dengan menggunakan fungsi cin. Kemudian, program melakukan pengecekan apakah nilai y yang dimasukkan merupakan bilangan bulat atau tidak dengan menggunakan pernyataan if pada looping do-while. Jika nilai y yang dimasukkan bukan bilangan bulat, maka program akan menampilkan pesan error dan meminta pengguna untuk memasukkan nilai y kembali hingga benar. Setelah nilai y yang dimasukkan valid, nilai x dan y akan disimpan dalam variabel private di class Pangkat. Subprogram calculate() berfungsi untuk menghitung hasil pangkat menggunakan fungsi pow() dengan parameter x dan y yang telah diinput sebelumnya. Hasil pangkat akan disimpan dalam variabel private result. Subprogram display() berfungsi untuk menampilkan output berupa nilai x beserta superscript y dan hasil pangkat dari  $x^y$ . Output ditampilkan menggunakan fungsi cout. Subprogram ini juga memiliki kode yang sama dengan program pertama untuk mengecek apakah y bernilai negatif atau positif, mengubah nilai y menjadi array of integer splitnumber, dan mengubah nilai splitnumber menjadi superscript dengan menggunakan pernyataan switch case.

Kelebihan kodingan kedua adalah penggunaan konsep OOP memudahkan pemeliharaan kode dan pengembangan program yang lebih kompleks dalam skala besar. Selain itu, dengan menggunakan class dan method, program menjadi lebih terstruktur dan lebih mudah dibaca serta dimengerti. Sedangkan kekurangan kodingan kedua adalah program menjadi lebih panjang dan kompleks dibandingkan dengan kodingan pertama, sehingga lebih sulit untuk dipahami bagi pengguna awam atau pemula dalam pemrograman.

Sedangkan untuk kodingan pertama, kelebihanannya adalah program cukup sederhana dan mudah dipahami oleh pengguna pemula, karena hanya menggunakan satu subprogram main(). Namun, kekurangannya adalah program tidak struktural, sehingga lebih sulit untuk dikembangkan dan dipelihara pada skala besar, serta rentan terhadap kesalahan logika pada saat pengembangan.

Berikut adalah outputnya:





```

PS D:\Document\hdk\Kuliah\Semester 2\Algoritma Pemrograman\Praktikum\Pertemuan 4> cd "d:\Document\hdk\Kuliah\Semester 2\Algoritma Pemrograman\Praktikum\Pertemuan 4\"; if ($?) { g++ hitungPangkat_subProgram.cpp -o hitungPangkat_subProgram }; if ($?) { .\hitungPangkat_subProgram }
Masukkan nilai x (Bilangan real) : 2
Masukkan nilai y (Bilangan bulat) : 1
2^1 = 2
PS D:\Document\hdk\Kuliah\Semester 2\Algoritma Pemrograman\Praktikum\Pertemuan 4> cd "d:\Document\hdk\Kuliah\Semester 2\Algoritma Pemrograman\Praktikum\Pertemuan 4\"; if ($?) { g++ hitungPangkat_subProgram.cpp -o hitungPangkat_subProgram }; if ($?) { .\hitungPangkat_subProgram }
Masukkan nilai x (Bilangan real) : 2
Masukkan nilai y (Bilangan bulat) : 2
2^2 = 4
PS D:\Document\hdk\Kuliah\Semester 2\Algoritma Pemrograman\Praktikum\Pertemuan 4> cd "d:\Document\hdk\Kuliah\Semester 2\Algoritma Pemrograman\Praktikum\Pertemuan 4\"; if ($?) { g++ hitungPangkat_subProgram.cpp -o hitungPangkat_subProgram }; if ($?) { .\hitungPangkat_subProgram }
Masukkan nilai x (Bilangan real) : 2
Masukkan nilai y (Bilangan bulat) : 11
2^11 = 2048
PS D:\Document\hdk\Kuliah\Semester 2\Algoritma Pemrograman\Praktikum\Pertemuan 4> cd "d:\Document\hdk\Kuliah\Semester 2\Algoritma Pemrograman\Praktikum\Pertemuan 4\"; if ($?) { g++ hitungPangkat_subProgram.cpp -o hitungPangkat_subProgram }; if ($?) { .\hitungPangkat_subProgram }
Masukkan nilai x (Bilangan real) : 2
Masukkan nilai y (Bilangan bulat) : -2
2^-2 = 0.25
PS D:\Document\hdk\Kuliah\Semester 2\Algoritma Pemrograman\Praktikum\Pertemuan 4> 2
PS D:\Document\hdk\Kuliah\Semester 2\Algoritma Pemrograman\Praktikum\Pertemuan 4> cd "d:\Document\hdk\Kuliah\Semester 2\Algoritma Pemrograman\Praktikum\Pertemuan 4\"; if ($?) { g++ hitungPangkat_subProgram.cpp -o hitungPangkat_subProgram }; if ($?) { .\hitungPangkat_subProgram }
Masukkan nilai x (Bilangan real) : 2
Masukkan nilai y (Bilangan bulat) : -12
2^-12 = 0.000244141
PS D:\Document\hdk\Kuliah\Semester 2\Algoritma Pemrograman\Praktikum\Pertemuan 4>
  
```

Untuk melihat source code, dapat dilihat link github berikut:

<https://github.com/IRedDragonICY/Programming-Algorithms>

