

LAPORAN PRAKTIKUM

“POST TEST 6: BILANGAN BULAT”

Diajukan untuk memenuhi salah satu praktikum Mata Kuliah Matematika Diskrit yang di
ampu oleh:

Nur Rochmah Dyah PA, S.T., M.Kom



Disusun Oleh:

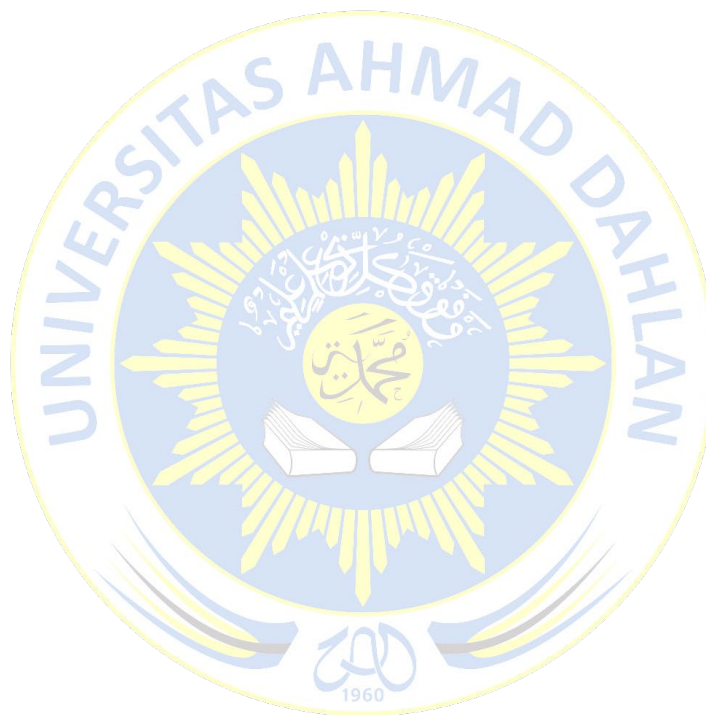
Mohammad Farid Hendianto 2200018401

Selasa 12.00-13.30

PROGRAM STUDI INFORMATIKA
UNIVERSITAS AHMAD DAHLAN
FAKULTAS TEKNOLOGI INDUSTRI
TAHUN 2023

DAFTAR SOAL

1. Ubahlah program pencarian fpb dan kpk dalam modul kedalam bentuk fungsi rekrusif. 3



1. Ubahlah program pencarian fpb dan kpk dalam modul kedalam bentuk fungsi rekursif.

*Untuk nim genap membuat program fpb. (Walaupun nim ganjil, saya mengerjakan FPB juga)

*Untuk nim ganjil membuat program kpk.

Jawab:

Karena kodingan di dalam modul memiliki banyak kesalahan, berikut kodingan sebelumnya yang sudah dibenarkan dan lebih efisien pada langkah praktikum sebelumnya.



```

1  #include <iostream>
2
3  using namespace std;
4
5  class Math {
6  public:
7      // // Algoritma Brute Force
8      // int getKPK(int num1, int num2) {
9      //     if (i % num1 == 0 && i % num2 == 0) {
10         //         return i;
11         //     }
12         //     return getKPK(num1, num2 + 1);
13         // }
14
15         // int getFPB(int num1, int num2) {
16         //     if (num2 == 0) {
17         //         return num1;
18         //     } else {
19         //         return getFPB(num2, num1 % num2);
20         //     }
21         // }
22         // Algoritma Euclidean
23         int getKPK(int num1, int num2) {
24             int fpb = getFPB(num1, num2);
25             return (num1 * num2) / fpb;
26         }
27
28         int getFPB(int num1, int num2) {
29             while (num2 != 0) {
30                 int temp = num2;
31                 num2 = num1 % num2;
32                 num1 = temp;
33             }
34             return num1;
35         }
36     };
37
38     class Menu {
39     public:
40         int displayMenu() {
41             int choice;
42             cout << "Masukkan pilihan anda:" << endl;
43             cout << "1. Menentukan KPK" << endl;
44             cout << "2. Menentukan FPB" << endl;
45             cout << "3. Keluar" << endl;
46             cout << "> ";
47             cin >> choice;
48             return choice;
49         }
50         void inputNum(int& num1, int& num2) {
51             cout << "Masukkan bilangan pertama: ";
52             cin >> num1;
53             cout << "Masukkan bilangan kedua: ";
54             cin >> num2;
55         }
56         void pressEnterToContinue() {
57             cout << "Tekan Enter untuk melanjutkan...";
58             cin.ignore();
59             cin.get();
60         }
61     };

```

```
63 class Program {
64 public:
65     void run() {
66         Math math;
67         Menu menu;
68         int choice;
69         int num1, num2;
70
71         while (true) {
72             system("cls");
73             choice = menu.displayMenu();
74
75             system("cls");
76             switch (choice) {
77                 case 1:
78                     cout << "Menentukan KPK" << endl;
79                     menu.inputNum(num1, num2);
80
81                     cout << "Bilangan pertama: " << num1 << endl;
82                     cout << "Bilangan kedua : " << num2 << endl;
83                     cout << "KPK: " << math.getKPK(num1, num2) << endl;
84                     break;
85
86                 case 2:
87                     cout << "Menentukan FPB" << endl;
88                     menu.inputNum(num1, num2);
89
90                     cout << "Bilangan pertama: " << num1 << endl;
91                     cout << "Bilangan kedua: " << num2 << endl;
92                     cout << "FPB: " << math.getFPB(num1, num2) << endl;
93                     break;
94
95                 case 3:
96                     cout << "Keluar dari program..." << endl;
97                     return;
98
99                 default:
100                     cout << "Pilihan tidak valid!" << endl;
101             }
102
103             menu.pressEnterToContinue();
104         }
105     }
106 };
107
108 int main() {
109     Program program;
110     program.run();
111     return 0;
112 }
```

Gambar 1 Kodingan sebelumnya berbentuk fungsi iteratif. (Sumber: Penulis)

Berikut adalah kodingan yang sudah diubah menjadi fungsi rekursif, beberapa algoritma yang digunakan yaitu KPK menggunakan algoritma pendekatan brute force, sedangkan FPB menggunakan algoritma euclidean.

```
1  #include <iostream>
2  using namespace std;
3
4  class Math {
5  public:
6
7      int getKPK(int num1, int num2, int i = 1) {
8          if (i % num1 == 0 && i % num2 == 0) {
9              return i;
10         }
11         return getKPK(num1, num2, i+1);
12     }
13
14     int getFPB(int num1, int num2) {
15         if (num2 == 0) {
16             return num1;
17         }
18         else {
19             return getFPB(num2, num1 % num2);
20         }
21     }
22 };
23
24 class Menu {
25 public:
26     int displayMenu() {
27         int choice;
28         cout << "Masukkan pilihan anda:" << endl
29             << "1. Menentukan KPK" << endl
30             << "2. Menentukan FPB" << endl
31             << "3. Keluar" << endl
32             << "> ";
33         cin >> choice;
34         return choice;
35     }
36     void inputNum(int& num1, int& num2) {
37         cout << "Masukkan bilangan pertama: ";
38         cin >> num1;
39         cout << "Masukkan bilangan kedua: ";
40         cin >> num2;
41     }
42     void pressEnterToContinue() {
43         cout << "Tekan Enter untuk melanjutkan...";
44         cin.ignore();
45         cin.get();
46     }
47 };
48
```

```

48
49 class Program {
50 public:
51     void run() {
52         Math math;
53         Menu menu;
54         int choice;
55         int num1, num2;
56
57         while (true) {
58             system("cls");
59             choice = menu.displayMenu();
60
61             system("cls");
62             switch (choice) {
63                 case 1:
64                     cout << "Menentukan KPK" << endl;
65                     menu.inputNum(num1, num2);
66
67                     cout << "Bilangan pertama: " << num1 << endl;
68                     cout << "Bilangan kedua : " << num2 << endl;
69                     cout << "KPK: " << math.getKPK(num1, num2, 1) << endl;
70                     break;
71
72                 case 2:
73                     cout << "Menentukan FPB" << endl;
74                     menu.inputNum(num1, num2);
75
76                     cout << "Bilangan pertama: " << num1 << endl;
77                     cout << "Bilangan kedua : " << num2 << endl;
78                     cout << "FPB: " << math.getFPB(num1, num2) << endl;
79                     break;
80
81                 case 3:
82                     cout << "Keluar dari program..." << endl;
83                     return;
84
85                 default:
86                     cout << "Pilihan tidak valid!" << endl;
87             }
88
89             menu.pressEnterToContinue();
90         }
91     }
92 };
93
94 int main() {
95     Program program;
96     program.run();
97     return 0;
98 }

```

Gambar 2 Program mencari FPB dan KPK dengan fungsi rekursif. (Sumber: Penulis)

```

1 #include <iostream>
2 using namespace std;
3
4 class Math {
5 public:
6
7     int get KPK(int num1, int num2, int i = 1) {
8         if (1 % num1 == 0 && 1 % num2 == 0) {
9             return i;
10        }
11        return getKPK(num1, num2, i+1);
12    }
13
14    int getFPB(int num1, int num2) {
15        if (num2 == 0) {
16            return num1;
17        }
18        else {
19            return getFPB(num2, num1 % num2);
20        }
21    }
22 }
23
24 class Menu {
25 public:
26     int displayMenu() {
27         int choice;
28         cout << "Masukkan pilihan anda:" << endl;
29         << "1. Menentukan KPK" << endl;
30         << "2. Menentukan FPB" << endl;
31         << "3. Keluar" << endl;
32         << "> ";
33         cin >> choice;
34         return choice;
35     }
36     void inputNum(int& num1, int& num2) {
37         cout << "Masukkan bilangan pertama: ";
38         cin >> num1;
39         cout << "Masukkan bilangan kedua: ";
40         cin >> num2;
41     }
42     void pressEnterToContinue() {
43         cout << "Tekan enter untuk melanjutkan...";
44         cin.ignore();
45         cin.get();
46     }
47 };
48
49 class Program {
50 public:
51     void run() {
52         Math math;
53         Menu menu;
54         int choice;
55         int num1, num2;
56
57         while (true) {
58             system("cls");
59             choice = menu.displayMenu();
60
61             system("cls");
62             switch (choice) {
63                 case 1:
64                 cout << "Menentukan KPK" << endl;
65                 menu.inputNum(num1, num2);
66                 cout << "Bilangan pertama: " << num1 << endl;
67                 cout << "Bilangan kedua: " << num2 << endl;
68                 cout << "KPK: " << math.getKPK(num1, num2, 1) << endl;
69                 break;
70                 case 2:
71                 cout << "Menentukan FPB" << endl;
72                 menu.inputNum(num1, num2);
73                 cout << "Bilangan pertama: " << num1 << endl;
74                 cout << "Bilangan kedua: " << num2 << endl;
75                 cout << "FPB: " << math.getFPB(num1, num2) << endl;
76                 break;
77                 case 3:
78                 cout << "Keluar dari program..." << endl;
79                 return;
80                 default:
81                 cout << "Pilihan tidak valid!" << endl;
82             }
83             menu.pressEnterToContinue();
84         }
85     }
86 };
87
88 int main() {
89     Program program;
90     program.run();
91     return 0;
92 }

```

Gambar 3 Tampilan kode di Visual Studio Code. (Sumber: Penulis)

Program yang dijabarkan pada kodingan tersebut adalah program kalkulator sederhana untuk menentukan KPK (Kelipatan Persekutuan Terkecil) dan FPB (Faktor Persekutuan Terbesar) dari dua bilangan bulat. KPK dan FPB sangat penting digunakan dalam berbagai bidang seperti matematika, fisika, dan teknologi, sehingga program ini sangat berguna bagi mereka yang membutuhkan perhitungan KPK dan FPB secara cepat dan mudah.

Alur kerja program dimulai dengan penggunaan tiga buah class yaitu Math, Menu, dan Program. Class Math berfungsi sebagai kelas yang berisi fungsi-fungsi matematika untuk menghitung KPK dan FPB dari dua bilangan bulat. Fungsi getKPK berfungsi untuk mendapatkan nilai KPK dari dua bilangan, dan fungsi getFPB adalah untuk mendapatkan nilai FPB dari dua bilangan. Class Menu berfungsi untuk menampilkan menu pilihan dan menerima input dari pengguna. Sedangkan class Program merupakan inti dari program ini, karena menjadi kontrol utama dari seluruh program.

Pada saat program dijalankan, pertama-tama program akan memanggil fungsi run() dari class Program. Di dalam fungsi run(), program akan membuat objek dari class Math dan class Menu yang kemudian akan menggunakan objek-objek tersebut untuk menjalankan program. Setelah itu, program akan masuk ke dalam loop while yang digunakan untuk menampilkan menu pilihan kepada pengguna dan melakukan perhitungan KPK atau FPB sesuai dengan pilihan pengguna.

Pada setiap iterasi dari loop while, program akan memanggil fungsi displayMenu() dari class Menu untuk menampilkan menu pilihan kepada pengguna. Setelah itu, program akan menerima input pilihan dari pengguna melalui fungsi cin dan menyimpannya pada variabel choice. Kemudian, program akan melakukan switch case untuk menentukan tindakan apa yang harus dilakukan berdasarkan pilihan pengguna.

Jika pilihan pengguna adalah 1, maka program akan memanggil fungsi inputNum() dari class Menu untuk menerima input dua buah bilangan bulat dari pengguna, kemudian akan melakukan perhitungan KPK menggunakan fungsi getKPK() dari class Math dengan parameter yang sesuai. Setelah itu, program akan menampilkan hasil perhitungan KPK ke layar. Hal yang sama juga dilakukan jika pilihan pengguna adalah 2, namun kali ini program akan melakukan perhitungan FPB menggunakan fungsi getFPB() dari class Math. Jika pilihan pengguna adalah 3, maka program akan keluar dari loop while dan program akan berakhir. Jika pilihan pengguna tidak valid (tidak 1, 2, atau 3), maka program akan menampilkan pesan kesalahan.

Setelah melakukan tindakan sesuai dengan pilihan pengguna, program akan memanggil fungsi pressEnterToContinue() dari class Menu untuk menunggu pengguna menekan tombol enter sebelum melanjutkan program ke iterasi selanjutnya dari loop while. Fungsi pressEnterToContinue() ini digunakan agar program tidak langsung keluar setelah menampilkan hasil perhitungan KPK atau FPB.

Fitur-fitur yang ada di dalam program ini antara lain adalah:

- Menampilkan menu pilihan kepada pengguna. Program ini menampilkan menu pilihan menentukan KPK atau FPB kepada pengguna pada awal program dijalankan. Pengguna dapat memilih opsi ketiga untuk keluar dari program.
- Menerima input dari pengguna Program ini menggunakan fungsi cin untuk menerima input dua bilangan bulat dari pengguna yang akan digunakan dalam perhitungan KPK atau FPB.

- Menghitung KPK dan FPB dari dua bilangan bulat. Program ini menggunakan class Math untuk menghitung KPK dan FPB dari dua bilangan bulat yang diberikan oleh pengguna. Fungsi `getKPK()` digunakan untuk menghitung KPK, sedangkan fungsi `getFPB()` digunakan untuk menghitung FPB.
- Menampilkan hasil perhitungan ke layar. Setelah melakukan perhitungan KPK atau FPB, program ini akan menampilkan hasilnya ke layar sehingga pengguna dapat melihat hasil perhitungan tersebut.
- Memberikan kesempatan kepada pengguna untuk mengulang perhitungan. Setelah menampilkan hasil perhitungan KPK atau FPB, program akan meminta pengguna untuk menekan tombol enter sebelum melanjutkan ke iterasi selanjutnya dari loop while. Hal ini memberikan kesempatan kepada pengguna untuk mengulang perhitungan KPK atau FPB jika diperlukan.
- Menangani input yang tidak valid. Program ini memiliki mekanisme untuk menangani pilihan pengguna yang tidak valid dengan menampilkan pesan kesalahan.
- Modular. Program ini menggunakan tiga buah class yaitu Math, Menu, dan Program, sehingga memudahkan pengembangan program di masa depan. Jika ada fitur-fitur baru yang ingin ditambahkan, dapat dilakukan pada satu dari ketiga class tersebut tanpa merusak struktur program yang sudah ada.

Pada bagian class Math terdapat dua buah fungsi yaitu `getKPK()` dan `getFPB()`. Fungsi `getKPK()` berfungsi untuk mendapatkan nilai KPK dari dua bilangan bulat yang diberikan, sedangkan fungsi `getFPB()` berfungsi untuk mendapatkan nilai FPB dari dua bilangan bulat yang diberikan.

Fungsi `getKPK()` menggunakan konsep rekursi untuk mencari nilai KPK. Pertama-tama, fungsi akan menerima tiga parameter yaitu `num1`, `num2`, dan `i`. Parameter `i` digunakan untuk mencari nilai KPK secara incremental mulai dari 1. Pada setiap iterasi, fungsi akan mengecek apakah nilai `i` merupakan kelipatan dari `num1` dan `num2`. Jika ya, maka nilai `i` akan menjadi nilai KPK dan akan dikembalikan oleh fungsi. Jika tidak, maka fungsi akan memanggil dirinya sendiri dengan nilai `i+1` sebagai parameter hingga didapatkan nilai KPK yang sesuai.


Fungsi `getFPB()` juga menggunakan konsep rekursi untuk mencari nilai FPB. Fungsi akan menerima dua parameter yaitu `num1` dan `num2`. Pada setiap iterasi, fungsi akan mengecek apakah nilai `num2` adalah 0. Jika ya, maka nilai `num1` akan menjadi nilai FPB dan akan dikembalikan oleh fungsi. Jika tidak, maka fungsi akan memanggil dirinya sendiri dengan `num2` sebagai `num1` dan `num1 modulo num2` sebagai `num2` hingga didapatkan nilai FPB yang sesuai.

Pada bagian class Menu terdapat tiga buah fungsi yaitu `displayMenu()`, `inputNum()`, dan `pressEnterToContinue()`. Fungsi `displayMenu()` berfungsi untuk menampilkan menu pilihan kepada pengguna dan mengembalikan pilihan yang dipilih oleh pengguna. Fungsi `inputNum()` berfungsi untuk menerima input dua bilangan bulat dari pengguna. Fungsi `pressEnterToContinue()` berfungsi untuk memberikan kesempatan kepada pengguna untuk mengulang perhitungan atau keluar dari program.

Pada bagian class Program terdapat satu buah fungsi yaitu run(). Fungsi ini merupakan inti dari program karena menjadi kontrol utama dari seluruh program. Fungsi run() menggunakan objek-objek dari class Math dan class Menu untuk menjalankan program. Fungsi ini juga menggunakan loop while untuk menampilkan menu pilihan kepada pengguna dan melakukan perhitungan KPK atau FPB sesuai dengan pilihan pengguna. Selain itu, fungsi ini juga menangani input yang tidak valid dan memberikan kesempatan kepada pengguna untuk mengulang perhitungan jika diperlukan.

Berikut adalah cara kerja perhitungan KPK dan FPB.

KPK



```
1 int getKPK(int num1, int num2, int i = 1) {  
2     if (i % num1 == 0 && i % num2 == 0) {  
3         return i;  
4     }  
5     return getKPK(num1, num2, i+1);  
6 }  
7
```

Gambar 4 Fungsi rekursif getKPK(). (Sumber: Penulis)

Fungsi getKPK() merupakan sebuah fungsi yang digunakan untuk mencari nilai Kelipatan Persekutuan Terkecil (KPK) dari dua bilangan bulat yang diberikan. Fungsi ini menggunakan metode rekursi dan parameter default untuk mencari KPK tersebut.

Algoritma yang digunakan pada fungsi ini adalah metode brute force dengan menggunakan rekursi untuk mencari KPK (Kelipatan Persekutuan Terkecil) dari dua bilangan bulat. Metode brute force adalah metode yang dilakukan dengan melakukan pengecekan secara sistematis pada setiap nilai kemungkinan hingga ditemukan solusi yang diinginkan.

Pada fungsi getKPK(), algoritma brute force dilakukan dengan melakukan pengecekan pada setiap angka secara bertambah satu hingga ditemukan sebuah angka yang merupakan kelipatan dari kedua bilangan yang dimasukkan sebagai parameter fungsi. Pengecekan ini dilakukan dengan menggunakan operator modulo, yaitu jika i habis dibagi oleh num1 dan num2 maka i tersebut merupakan KPK dari num1 dan num2.

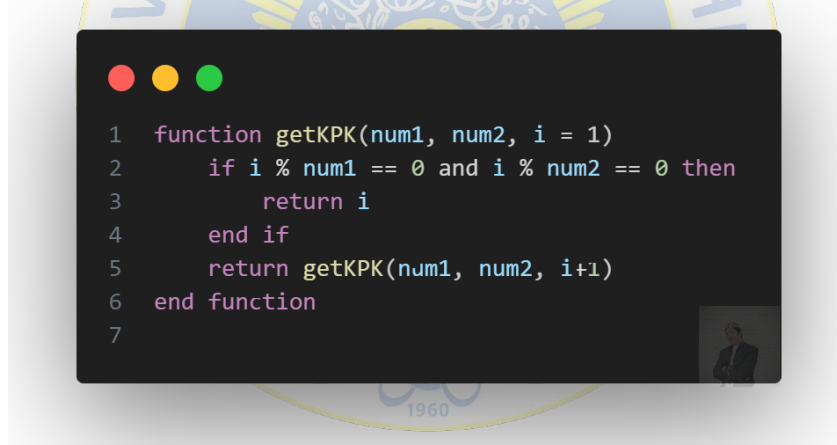
Metode brute force ini cukup efektif untuk mencari KPK dari dua bilangan kecil, namun akan menjadi tidak efektif ketika bilangan semakin besar karena memerlukan waktu yang lebih lama untuk menemukan angka yang diinginkan.

Dalam kasus ini, sebaiknya kita menggunakan algoritma yang lebih efisien seperti Algoritma Euclidean atau Algoritma Steins untuk mencari KPK dari dua bilangan yang sangat besar.

Algoritma yang digunakan pada fungsi getKPK() adalah sebagai berikut:

- 1) Fungsi menerima tiga parameter yaitu num1, num2, dan i.
- 2) Fungsi melakukan pengecekan apakah nilai i adalah kelipatan dari num1 dan num2 dengan menggunakan operator modulo (%).
- 3) Jika ya, maka nilai i akan dikembalikan oleh fungsi sebagai nilai KPK.
- 4) Jika tidak, maka fungsi akan memanggil dirinya sendiri dengan nilai i+1 sebagai parameter hingga didapatkan nilai KPK yang sesuai.

Pseudocode dari algoritma di atas adalah sebagai berikut:



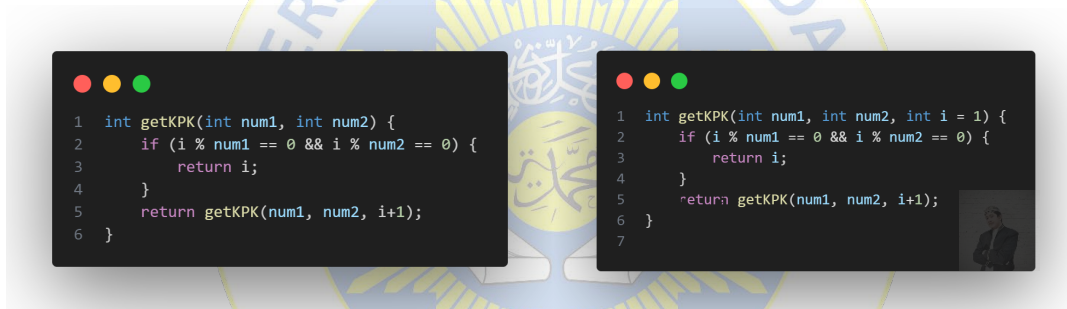
Gambar 5 Pseudocode getKPK(). (Sumber: Penulis)

Cara kerja dari fungsi getKPK() adalah sebagai berikut:

- 1) Fungsi menerima dua bilangan bulat yaitu num1 dan num2 sebagai input.
- 2) Fungsi memanggil dirinya sendiri dengan nilai i=1 sebagai parameter.
- 3) Pada setiap iterasi, fungsi akan mengecek apakah nilai i adalah kelipatan dari num1 dan num2 dengan menggunakan operator modulo (%).
- 4) Jika ya, maka fungsi akan mengembalikan nilai i sebagai nilai KPK.
- 5) Jika tidak, maka fungsi akan memanggil dirinya sendiri dengan nilai i+1 sebagai parameter hingga didapatkan nilai KPK yang sesuai.

Contoh cara kerja dari fungsi getKPK() adalah sebagai berikut:

- 1) Misalkan kita ingin mencari KPK dari bilangan 6 dan 8.
- 2) Fungsi getKPK() dipanggil dengan parameter num1=6 dan num2=8.
- 3) Pada iterasi pertama, nilai i=1 dan bukan merupakan kelipatan dari 6 dan 8. Fungsi memanggil dirinya sendiri dengan nilai i=2.
- 4) Pada iterasi kedua, nilai i=2 dan bukan merupakan kelipatan dari 6 dan 8. Fungsi memanggil dirinya sendiri dengan nilai i=3.
- 5) Pada iterasi ketiga, nilai i=3 dan bukan merupakan kelipatan dari 6 dan 8. Fungsi memanggil dirinya sendiri dengan nilai i=4.
- 6) Pada iterasi keempat, nilai i=4 dan bukan merupakan kelipatan dari 6 dan 8. Fungsi memanggil dirinya sendiri dengan nilai i=5.
- 7) Pada iterasi kelima, nilai i=5 dan bukan merupakan kelipatan dari 6 dan 8. Fungsi memanggil dirinya sendiri dengan nilai i=6.
- 8) Pada iterasi keenam, nilai i=6 dan merupakan kelipatan dari 6 dan 8. Fungsi mengembalikan nilai i=6 sebagai nilai KPK.
- 9) Sehingga hasil akhir dari fungsi getKPK(6, 8) adalah 6.



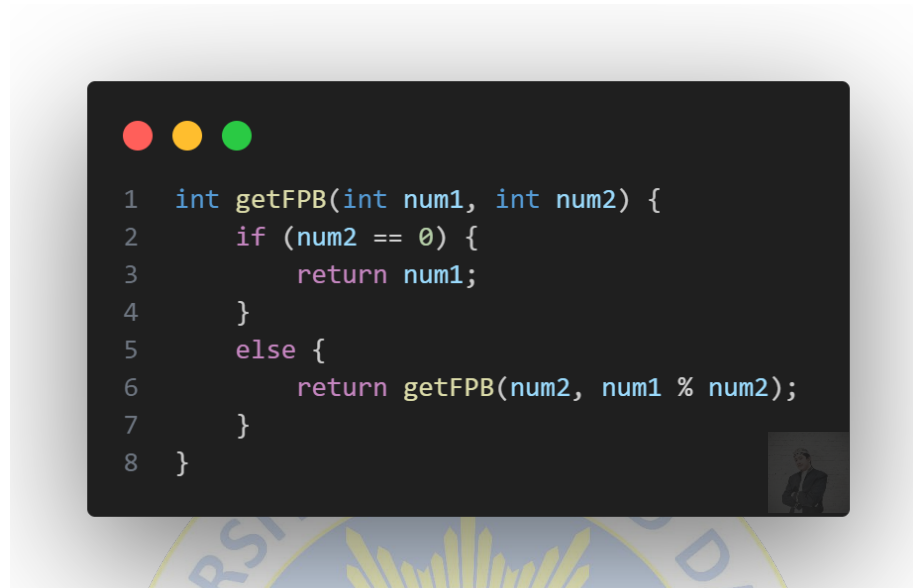
Gambar 6 Kiri menggunakan fungsi iteratif sedangkan kanan menggunakan fungsi rekursif. (Sumber: Penulis)

Kelebihan dari penggunaan fungsi getKPK() dengan metode rekursi adalah kode programnya lebih sederhana dan mudah dipahami. Selain itu, penggunaan konsep rekursi pada fungsi getKPK() memungkinkan untuk mencari nilai KPK dari dua bilangan bulat yang sangat besar dengan cara yang efisien.

Namun, penggunaan metode rekursi pada fungsi getKPK() juga memiliki beberapa kekurangan, yaitu:

- Kemungkinan terjadi stack overflow jika nilai i terlalu besar atau jika terjadi pemanggilan rekursi yang terlalu dalam.
- Penggunaan metode rekursi pada fungsi getKPK() dapat mengakibatkan program menjadi lebih lambat karena perlu melakukan pemanggilan fungsi secara berulang-ulang.

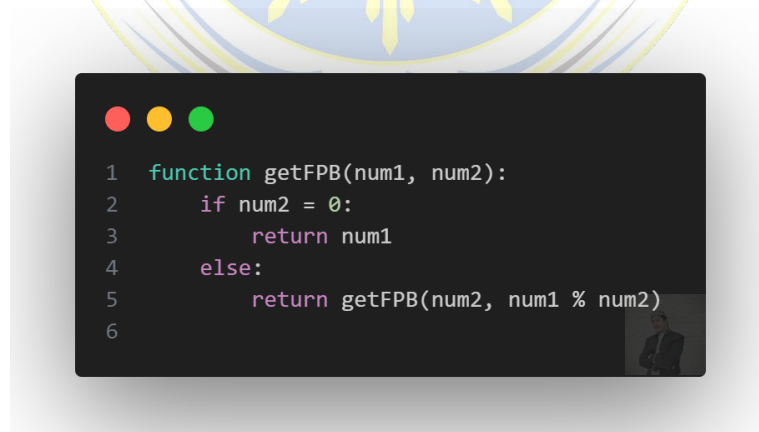
FPB



Gambar 7 Fungsi rekursif getFPB(). (Sumber: Penulis)

Fungsi getFPB adalah sebuah fungsi dalam bahasa pemrograman C++ yang digunakan untuk mencari faktor persekutuan terbesar (FPB) dari dua bilangan bulat. Faktor persekutuan terbesar adalah bilangan bulat positif terbesar yang dapat membagi habis kedua bilangan input tersebut. Fungsi ini menggunakan rekursi, yaitu teknik pemanggilan fungsi dirinya sendiri hingga mencapai kondisi akhir tertentu.

Berikut adalah pseudocode dari fungsi getFPB:



Gambar 8 Pseudocode getKPK(). (Sumber: Penulis)

Algoritma yang digunakan dalam fungsi getFPB sangat sederhana dan mudah dipahami. Fungsi ini menerima dua parameter, yaitu num1 dan num2, yang merupakan bilangan bulat yang ingin dicari FPB-nya. Pertama-tama, algoritma akan melakukan pengecekan apakah num2 sama dengan nol. Jika iya, maka nilai num1 akan direturn karena FPB dari num1 dan 0 adalah num1 itu sendiri. Jika tidak, maka algoritma akan melakukan rekursi dengan memanggil kembali fungsi getFPB dengan parameter num2 dan sisa pembagian num1 dengan num2 ($\text{num1} \% \text{num2}$).

Setiap kali fungsi getFPB dipanggil kembali secara rekursif, nilai num2 akan menjadi nilai num1 yang baru dan nilai $\text{num1} \% \text{num2}$ akan menjadi nilai num2 yang baru. Ini dilakukan hingga num2 sama dengan nol, sehingga nilai num1 yang ditemukan pada akhirnya adalah FPB dari num1 dan num2.

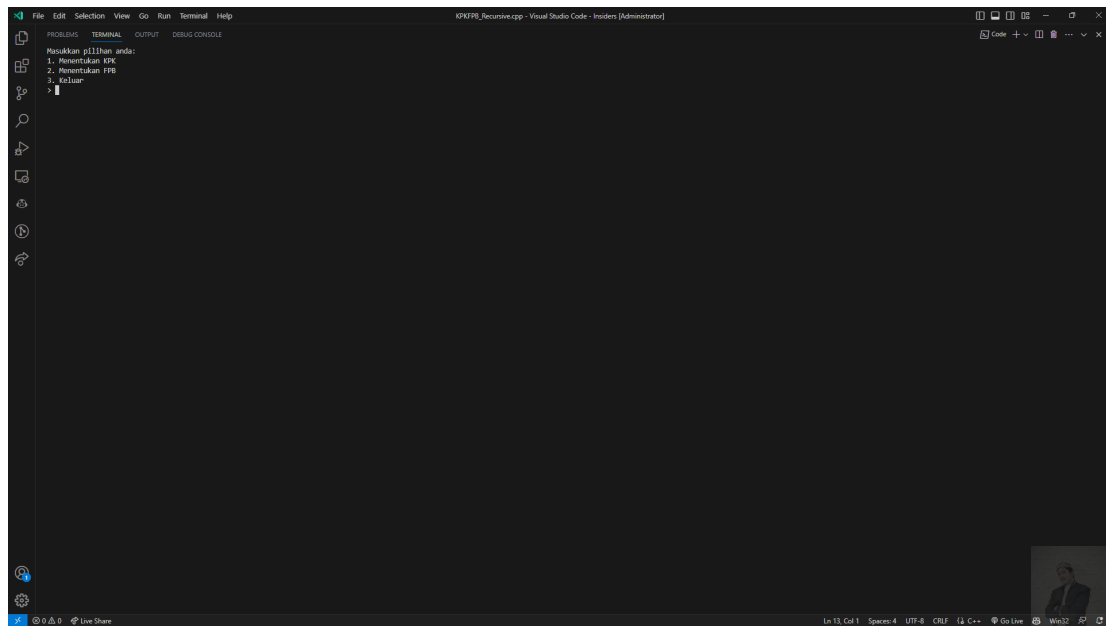
Misalkan kita ingin mencari FPB dari dua bilangan bulat yaitu 24 dan 18. Maka, kita dapat menggunakan fungsi getFPB dan memasukkan nilai 24 dan 18 sebagai parameter.

Pertama-tama, algoritma akan melakukan pengecekan apakah num2 sama dengan nol. Karena num2 tidak sama dengan nol, maka algoritma akan melakukan rekursi dengan memanggil kembali fungsi getFPB dengan parameter num2 (nilai 18) dan sisa pembagian num1 dengan num2 ($24 \% 18 = 6$).

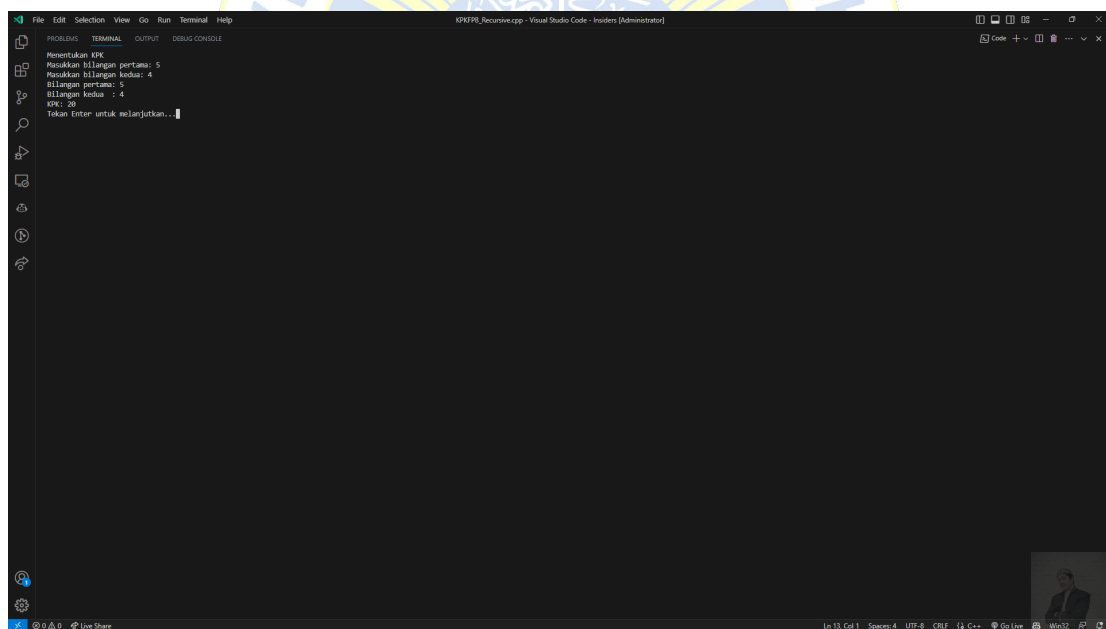
Kemudian, fungsi getFPB dipanggil lagi secara rekursif dengan parameter num2 (6) dan sisa pembagian num1 dengan num2 ($18 \% 6 = 0$). Karena num2 sama dengan nol pada pemanggilan ini, maka nilai num1 (6) akan direturn sebagai hasil FPB dari 24 dan 18.

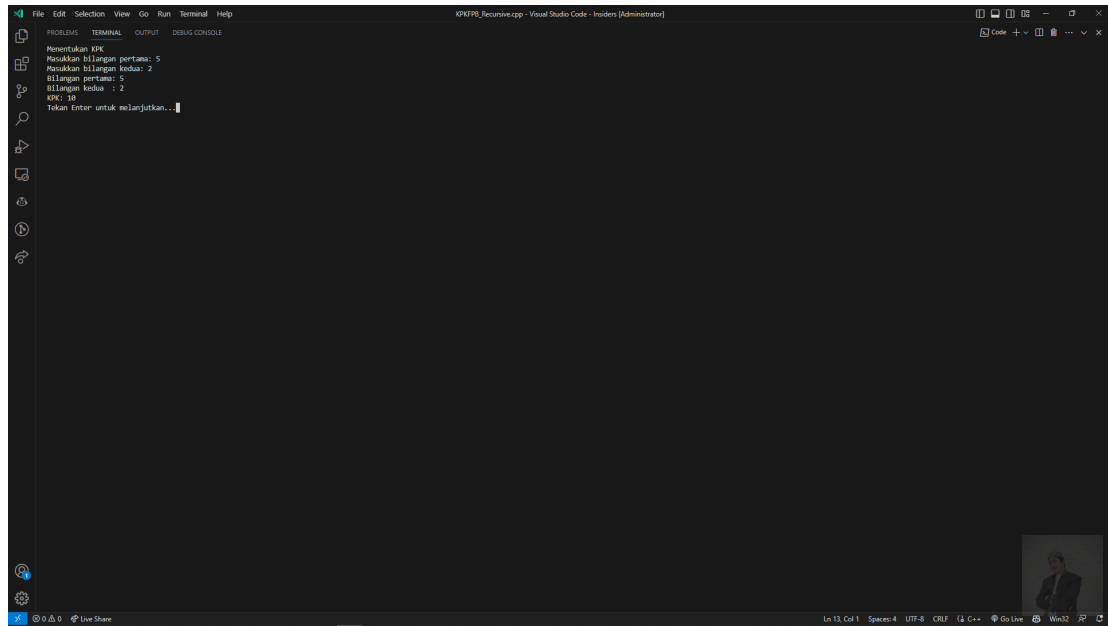
Algoritma yang digunakan dalam fungsi getFPB memiliki kompleksitas waktu $O(\log n)$ di mana n merupakan bilangan terbesar dari kedua input num1 dan num2. Hal ini dikarenakan setiap kali fungsi getFPB dipanggil secara rekursif, ukuran masalah akan berkurang sebesar faktor 2 hingga mencapai kondisi dasar. Sebagai contoh, jika num1 dan num2 masing-masing bernilai 32, maka fungsi getFPB akan dipanggil sebanyak 5 kali hingga mencapai kondisi dasar (1 dan 0). Oleh karena itu, kompleksitas waktu algoritma ini tidak terlalu besar dan dapat digunakan untuk mencari FPB dari bilangan bulat yang sangat besar.

Berikut adalah output program



Gambar 9 Tampilan Menu. (Sumber: Penulis)

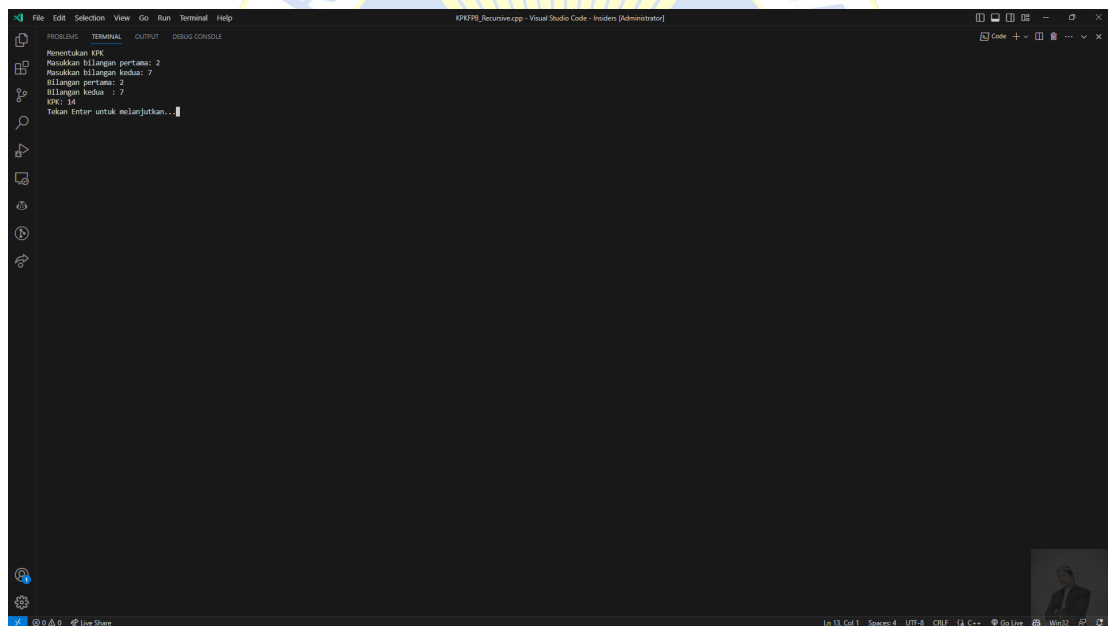




Visual Studio Code interface showing a terminal window with the following output:

```

Menentukan KPK
Masukkan bilangan pertama: 5
Masukkan bilangan kedua: 2
Bilangan pertama: 5
Bilangan kedua : 2
KPK: 10
Tekan Enter untuk melanjutkan...
  
```



Visual Studio Code interface showing a terminal window with the following output:

```

Menentukan KPK
Masukkan bilangan pertama: 2
Masukkan bilangan kedua: 7
Bilangan pertama: 2
Bilangan kedua : 7
KPK: 14
Tekan Enter untuk melanjutkan...
  
```



```

File Edit Selection View Go Run Terminal Help
KPKFPB_recursive.cpp - Visual Studio Code - Insiders [Administrator]
PROBLEMS TERMINAL OUTPUT DEBUG CONSOLE

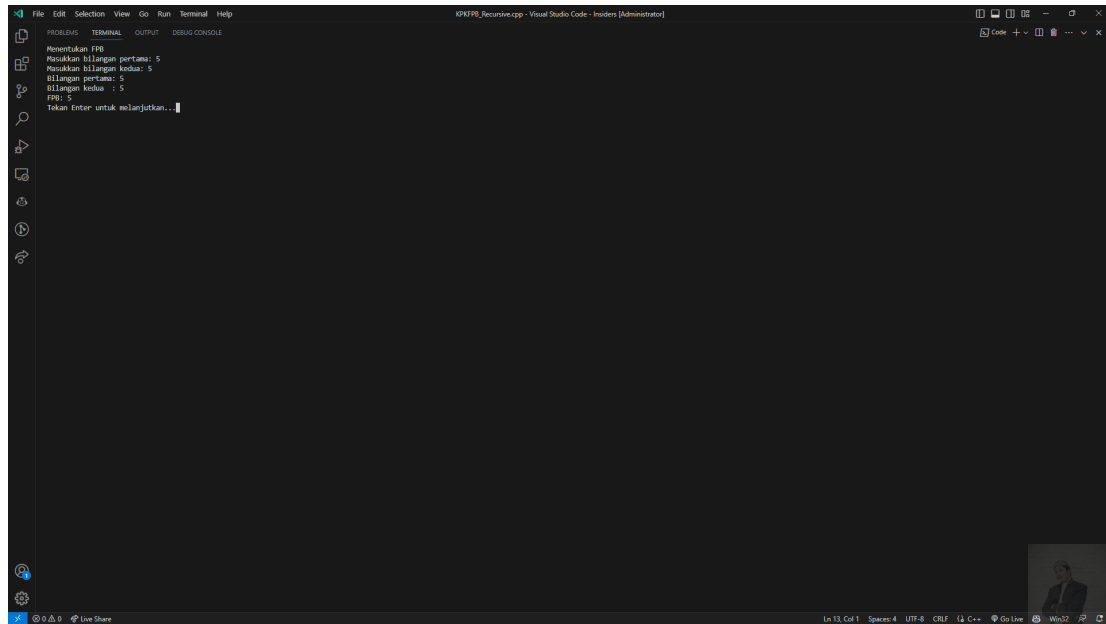
Mencari KPK
Masukkan bilangan pertama: 2
Masukkan bilangan kedua: 6
Bilangan pertama : 2
Bilangan kedua : 6
KPK: 6
Tekan Enter untuk melanjutkan...
  
```

Gambar 10 Contoh mencari KPK. (Sumber: Penulis)

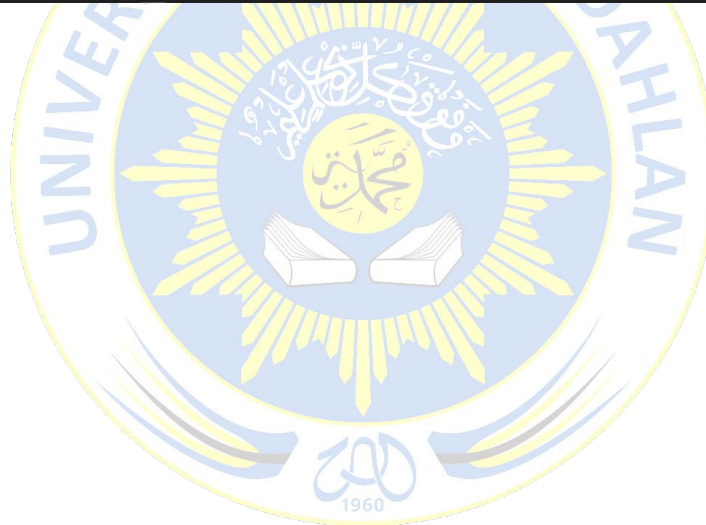
```

File Edit Selection View Go Run Terminal Help
KPKFPB_recursive.cpp - Visual Studio Code - Insiders [Administrator]
PROBLEMS TERMINAL OUTPUT DEBUG CONSOLE

Mencari FPB
Masukkan bilangan pertama: 5
Masukkan bilangan kedua: 4
Bilangan pertama: 5
Bilangan kedua : 4
FPB: 1
Tekan Enter untuk melanjutkan...
  
```



```
File Edit Selection View Go Run Terminal Help
KPK7PB_recursive.cpp - Visual Studio Code - Insiders (Administrator)
PROBLEMS TERMINAL OUTPUT DEBUG CONSOLE
Menentukan FPB
Masukkan bilangan pertama: 5
Masukkan bilangan kedua: 5
Bilangan pertama: 5
Bilangan kedua : 5
FPB: 5
Tekan Enter untuk melanjutkan...
```



The image displays two screenshots of a Visual Studio Code terminal window. The top screenshot shows the program 'KPK7PB_Recursive.cpp' running. The terminal output is as follows:

```

Menentukan FPB
Masukkan bilangan pertama: 6
Masukkan bilangan kedua: 2
Bilangan pertama: 6
Bilangan kedua : 2
FPB: 2
Tekan Enter untuk melanjutkan...

```

The bottom screenshot shows the same program running with different inputs. The terminal output is as follows:

```

Menentukan FPB
Masukkan bilangan pertama: 7
Masukkan bilangan kedua: 2
Bilangan pertama: 7
Bilangan kedua : 2
FPB: 1
Tekan Enter untuk melanjutkan...

```

Gambar 11 Contoh output FPB. (Sumber: Penulis)

Untuk mengakses kodingan, dapat mengakses link github berikut.

<https://github.com/IRedDragonICY/Matematika-Diskrit>