

LAPORAN PRAKTIKUM

“LANGKAH PRAKTIKUM 9: Penerapan Kombinatorik”

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

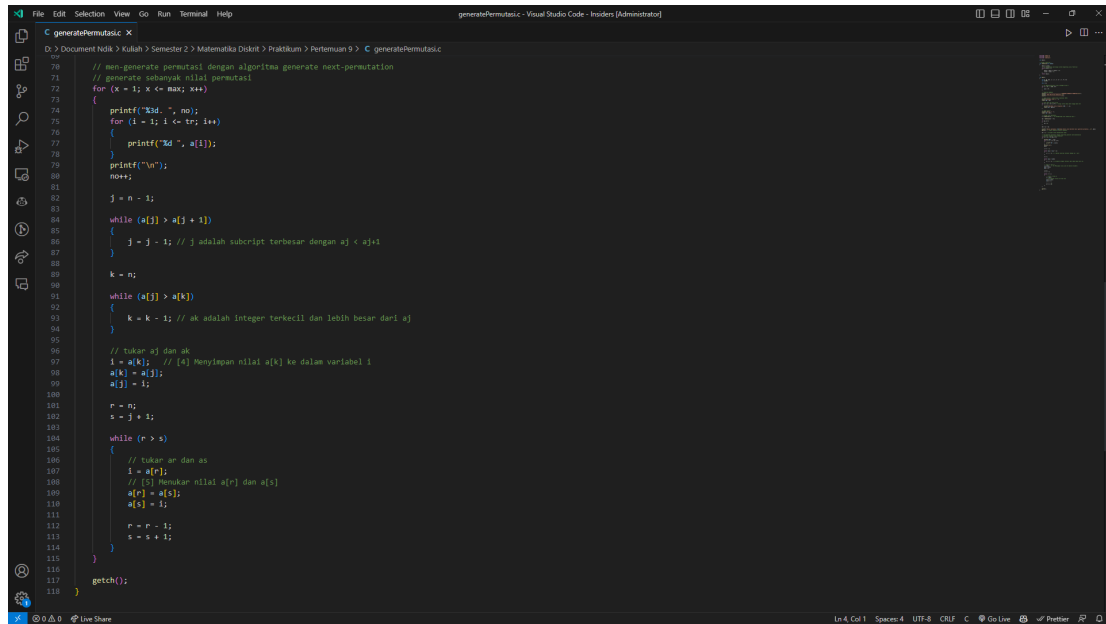
PROGRAM STUDI INFORMATIKA
UNIVERSITAS AHMAD DAHLAN
FAKULTAS TEKNOLOGI INDUSTRI
TAHUN 2023

Berikut adalah tampilan kodingan di Visual Studio Code.

```

1 // generatePermUtasc.c
2 #include <stdio.h>
3 #include <conio.h>
4 #include <stdlib.h>
5
6 int hasil;
7
8 // Fungsi faktorial :
9 int faktorial(int nilai)
10 {
11     hasil = nilai;
12     // [1] Mengawakan perulangan untuk menghitung nilai faktorial
13     while (nilai > 1)
14     {
15         hasil = hasil * (nilai - 1);
16         nilai = nilai - 1;
17     }
18     return hasil;
19 }
20
21 int main()
22 {
23     int p, nq, max, x, i, j, n, k, r, s, tr, m;
24     int a[100];
25     div_t xx;
26
27     // [2] Menginisialisasi array a dengan nilai 0
28     for (i = 0; i < 100; i++)
29     {
30         a[i] = 0;
31     }
32
33     // Tampilan pembuka
34     printf("-----\nPROGRAM GENERATE PERMUTASI\n\n");
35     printf("Oleh Mhd. Farid Hendianto\n");
36     printf("-----\n");
37
38     // input nilai n (jumlah data maksimal 100)
39     printf("Masukkan nilai n : ");
40     scanf("%d", &n);
41
42     // input data ke dalam array
43     for (i = 1; i <= n; i++) // ulangi untuk semua data hingga data ke-n
44     {
45         printf("Masukkan nilai himpunan a[%d] : ", i);
46         scanf("%d", &a[i]);
47     }
48
49     // input nilai r
50     printf("\nNilai r : ");
51     scanf("%d", &r);
52
53     // hitung nilai faktorial
54     p = faktorial(n); // [3] Menghitung nilai faktorial dari n
55
56     nq = faktorial(n - tr);
57
58     if (nq == 0)
59     {
60         nq = 1;
61     }
62
63     max = p / nq;
64
65     printf("\nNilai permutasi: %d\nTekan Enter untuk melihat hasil generate permutasi...\n", max);
66     getch(); // fungsi membaca karakter keyboard
67
68     no = 1; // variabel untuk menampilkan nomor
69
70     // men-generate permutasi dengan algoritma generate next-permutation
71     // generate sebanyak nilai permutasi
72     for (x = 1; x <= max; x++)
73     {
74         printf("%d. ", no);
75         for (i = 1; i <= tr; i++)
76         {
77             printf("%d ", a[i]);
78         }
79         printf("\n");
80         no++;
81
82         j = n - 1;
83
84         while (a[j] > a[j + 1])
85         {
86             j = j - 1; // j adalah subscript terbesar dengan a[j] < a[j+1]
87         }
88
89         k = n;
90
91         while (a[j] > a[k])
92         {
93             k = k - 1; // ak adalah integer terkecil dan lebih besar dari aj
94         }
95
96         // tukar aj dan ak
97         i = a[k]; // [4] Menyimpan nilai a[k] ke dalam variabel i
98         a[k] = a[j];
99     }

```



```
70 // men-generate permutasi dengan algoritma generate next-permutation
71 // generate sebanyak nilai permutasi
72 for (x = 1; x <= mas; x++)
73 {
74     printf("%3d. ", no);
75     for (i = 1; i <= tr; i++)
76     {
77         printf("%d ", a[i]);
78     }
79     printf("\n");
80     next;
81
82     j = n - 1;
83
84     while (a[j] > a[j + 1])
85     {
86         j = j - 1; // j adalah subscript terbesar dengan a[j] < a[j+1]
87     }
88     k = n;
89
90     while (a[j] > a[k])
91     {
92         k = k - 1; // k adalah integer terkecil dan lebih besar dari a[j]
93     }
94
95     // tukar a[j] dan a[k]
96     i = a[k]; // [k] menyimpan nilai a[k] ke dalam variabel i
97     a[k] = a[j];
98     a[j] = i;
99
100     r = n;
101     s = j + 1;
102
103     while (r > s)
104     {
105         // tukar ar dan as
106         i = a[r];
107         // [s] Menukar nilai a[r] dan a[s]
108         a[r] = a[s];
109         a[s] = i;
110
111         r = r - 1;
112         s = s + 1;
113     }
114 }
115
116 getch();
117 }
```

Gambar 1 Tampilan kodingan penerapan kombinatorik di Visual Studio Code. (Sumber: Penulis)

Berikut tampilan source code yang sesuai dengan modul

```

1  #include <stdio.h>
2  #include <conio.h>
3  #include <stdlib.h>
4
5  int hasil;
6
7  // Fungsi faktorial :
8  int faktorial(int nilai)
9  {
10     hasil = nilai;
11     // [1] Menggunakan perulangan untuk menghitung nilai faktorial
12     while (nilai > 1)
13     {
14         hasil = hasil * (nilai - 1);
15         nilai = nilai - 1;
16     }
17     return hasil;
18 }
19
20 int main()
21 {
22     int p, nq, max, x, i, j, n, k, r, s, tr, no;
23     int a[100];
24
25     div_t xx;
26
27     // [2] Menginisialisasi array a dengan nilai 0
28     for (i = 0; i < 100; i++)
29     {
30         a[i] = 0;
31     }
32
33     // Tampilan pembuka
34     printf("-----\nPROGRAM GENERATE PERMUTASI\n\n");
35     printf("Oleh Moh.Farid Hendianto \n");
36     printf("-----\n");
37
38     // input nilai n (jumlah data <maksimal 100>)
39     printf("Masukkan nilai n : ");
40     scanf("%d", &n);
41
42     // input data ke dalam array
43     for (i = 1; i <= n; i++) // ulangi untuk semua data hingga data ke-n
44     {
45         printf("Masukkan nilai himpunan a[%d] : ", i);
46         scanf("%d", &a[i]);
47     }
48
49     // input nilai r
50     printf("\nNilai r: ");
51     scanf("%d", &tr);
52
53     // hitung nilai permutasi
54     p = faktorial(n); // [3] Menghitung nilai faktorial dari n
55
56     nq = faktorial(n - tr);
57
58     if (nq == 0)
59     {
60         nq = 1;
61     }

```

```

70 // men-generate permutasi dengan algoritma generate next-permutation
71 // generate sebanyak nilai permutasi
72 for (x = 1; x <= max; x++)
73 {
74     printf("=. ", no);
75     for (i = 1; i <= tr; i++)
76     {
77         printf("%d ", a[i]);
78     }
79     printf("\n");
80     no++;
81
82     j = n - 1;
83
84     while (a[j] > a[j + 1])
85     {
86         j = j - 1; // j adalah subcript terbesar dengan aj < aj+1
87     }
88
89     k = n;
90
91     while (a[j] > a[k])
92     {
93         k = k - 1; // ak adalah integer terkecil dan lebih besar dari aj
94     }
95
96     // tukar aj dan ak
97     i = a[k]; // [4] Menyimpan nilai a[k] ke dalam variabel i
98     a[k] = a[j];
99     a[j] = i;
100
101     r = n;
102     s = j + 1;
103
104     while (r > s)
105     {
106         // tukar ar dan as
107         i = a[r];
108         // [5] Menukar nilai a[r] dan a[s]
109         a[r] = a[s];
110         a[s] = i;
111
112         r = r - 1;
113         s = s + 1;
114     }
115 }
116
117 getch();
118 }

```

Gambar 2 Kodingan yang sesuai pada modul. (Sumber: Penulis)

Program ini adalah program dalam bahasa C yang digunakan untuk menghasilkan permutasi dari suatu himpunan angka.

Tujuan dari program ini adalah untuk menghasilkan permutasi dari suatu himpunan angka. Fungsi utama dari program ini adalah untuk menghitung nilai permutasi dan menampilkan hasil permutasi tersebut. Program ini menggunakan algoritma generate next-permutation untuk menghasilkan permutasi.

Berikut adalah langkah-langkah yang dilakukan oleh program:

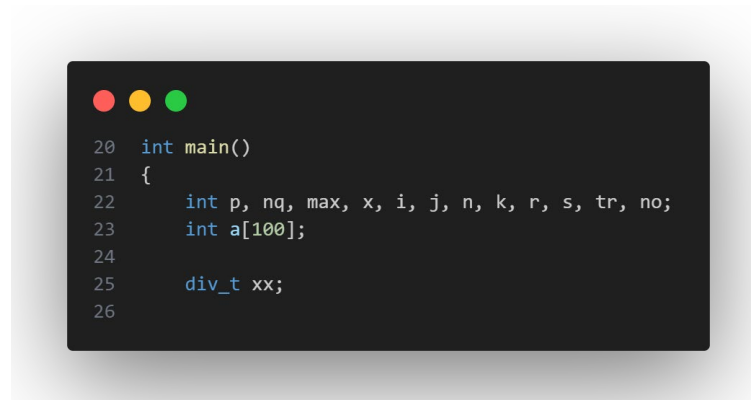
- 1) Menginisialisasi array a dengan nilai 0.
- 2) Menampilkan tampilan pembuka dan meminta input nilai n (jumlah angka yang akan dipermutasi) dan nilai r (jumlah angka yang akan diambil dari himpunan angka).
- 3) Meminta input angka-angka yang akan dipermutasi dan menyimpannya dalam array a.
- 4) Menghitung nilai permutasi dengan menggunakan fungsi faktorial.
- 5) Menampilkan jumlah permutasi yang mungkin dan menunggu input dari pengguna untuk melanjutkan.
- 6) Menghasilkan permutasi dengan menggunakan algoritma generate next-permutation dan menampilkan hasil permutasi tersebut.



Gambar 3 Library-library yang digunakan pada program. (Sumber: Penulis)

Program ini menggunakan beberapa fungsi dari library standar C, seperti printf, scanf, dan getch. Fungsi printf digunakan untuk menampilkan teks dan variabel ke layar, fungsi scanf digunakan untuk membaca input dari pengguna, dan fungsi getch digunakan untuk membaca karakter dari keyboard tanpa menampilkan karakter tersebut ke layar.

Program ini adalah program dalam bahasa C yang digunakan untuk menghasilkan permutasi dari suatu himpunan angka. Berikut ini adalah penjelasan mengenai inisialisasi variabel dan kegunaan masing-masing variabel:



Gambar 4 Inisialisasi variabel pada program. (Sumber: Penulis)

- 1) `int p, nq, max, x, i, j, n, k, r, s, tr, no;` - Inisialisasi variabel-variabel bertipe integer yang akan digunakan dalam program.
 - `p`: Variabel untuk menyimpan hasil permutasi.
 - `nq`: Variabel untuk menyimpan hasil faktorial dari $(n - tr)$.
 - `max`: Variabel untuk menyimpan jumlah maksimum permutasi yang akan dihasilkan.
 - `x`: Variabel untuk mengontrol iterasi dalam loop permutasi.
 - `i, j, k, r, s`: Variabel indeks dan bantuan untuk loop dan operasi dalam program.
 - `n`: Variabel untuk menyimpan jumlah elemen dalam himpunan.
 - `tr`: (target `r`) Variabel untuk menyimpan nilai `r` yang merupakan jumlah elemen yang akan diambil dalam permutasi.
 - `no`: Variabel untuk menampilkan nomor urut permutasi yang dihasilkan.
- 2) `int a[100];` - Inisialisasi array `a` dengan ukuran 100 yang akan digunakan untuk menyimpan elemen-elemen himpunan angka.
- 3) `div_t xx;` - Inisialisasi variabel `xx` dengan tipe data `div_t`. Namun, dalam program ini, variabel `xx` tidak digunakan sama sekali, jadi sebenarnya tidak perlu diinisialisasi.
- 4) `int hasil;` - Inisialisasi variabel `hasil` dengan tipe data integer secara global

Program ini akan meminta input jumlah elemen dalam himpunan (`n`), elemen-elemen himpunan, dan nilai `r` (jumlah elemen yang akan diambil dalam permutasi). Kemudian, program akan menghitung jumlah permutasi yang mungkin dan menampilkan hasil permutasi tersebut. Dalam program tersebut, masing-masing variabel memiliki peran yang berbeda dalam menghitung dan menampilkan permutasi dari himpunan angka. Variabel-variabel tersebut digunakan dalam berbagai operasi, seperti perhitungan faktorial, perhitungan jumlah permutasi, dan proses permutasi itu sendiri.

Program tersebut menggunakan fungsi faktorial. Berikut adalah penjelasan pada fungsi faktorial.



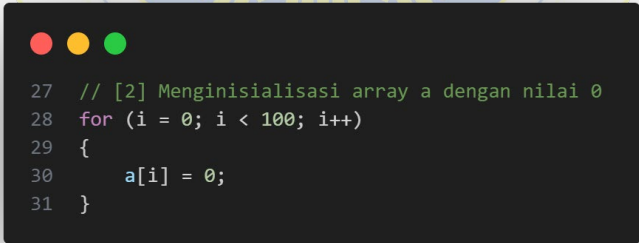
```
7 // Fungsi faktorial :  
8 int faktorial(int nilai)  
9 {  
10     hasil = nilai;  
11     // [1] Menggunakan perulangan untuk menghitung nilai faktorial  
12     while (nilai > 1)  
13     {  
14         hasil = hasil * (nilai - 1);  
15         nilai = nilai - 1;  
16     }  
17     return hasil;  
18 }
```

Gambar 5 Fungsi faktorial. (Sumber: Penulis)

Fungsi faktorial adalah sebuah fungsi yang menghitung nilai faktorial dari sebuah bilangan bulat. Faktorial dari sebuah bilangan bulat adalah hasil perkalian dari bilangan tersebut dengan semua bilangan bulat positif yang lebih kecil darinya.

Fungsi faktorial di atas mengambil satu parameter, yaitu nilai, yang merupakan bilangan bulat yang akan dihitung faktorialnya. Fungsi ini menggunakan perulangan while untuk menghitung nilai faktorial dari bilangan tersebut. Setiap iterasi, nilai faktorial dikalikan dengan nilai - 1 dan nilai dikurangi 1. Perulangan berhenti ketika nilai kurang dari atau sama dengan 1. Hasil faktorial kemudian dikembalikan oleh fungsi.

Berikut adalah penjelasan alur program lebih lengkap:



```
27 // [2] Menginisialisasi array a dengan nilai 0  
28 for (i = 0; i < 100; i++)  
29 {  
30     a[i] = 0;  
31 }
```

Gambar 6 Inisialisasi array a dengan value 0. (Sumber: Penulis)

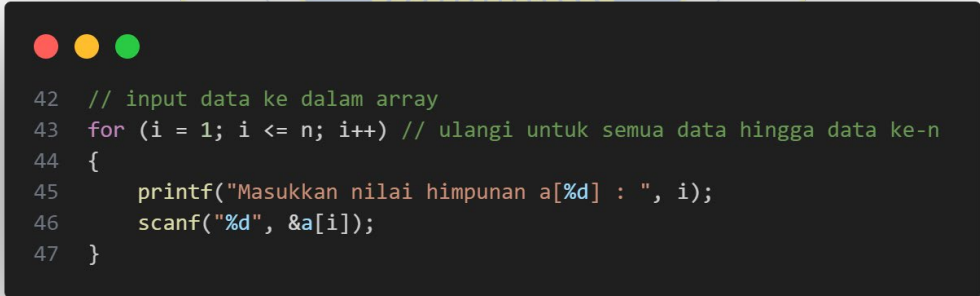
Kodingan tersebut merupakan sebuah loop for yang digunakan untuk menginisialisasi setiap elemen pada array a dengan nilai 0. Loop tersebut akan berjalan sebanyak 100 kali, dimulai dari indeks ke-0 hingga indeks ke-99. Setiap kali loop dijalankan, nilai pada elemen array a pada indeks yang sedang diiterasi akan diubah menjadi 0.



```
33 // Tampilan pembuka
34 printf("-----\nPROGRAM GENERATE PERMUTASI\n\n");
35 printf(" Oleh Moh. Farid Hendianto \n");
36 printf("-----\n");
37
38 // input nilai n (jumlah data <maksimal 100>)
39 printf("Masukkan nilai n : ");
40 scanf("%d", &n);
```

Gambar 7 Output tampilan pertama, dan input nilai n. (Sumber: Penulis)

Kode tersebut merupakan bagian dari program generate permutasi yang ditulis dalam bahasa C. Bagian pertama dari kode tersebut adalah tampilan pembuka program yang menampilkan judul program dan nama pembuat program. Bagian kedua dari kode tersebut adalah input nilai n yang dimasukkan oleh pengguna melalui keyboard. Nilai n tersebut merupakan jumlah data yang akan digenerate permutasinya. Nilai n harus berupa bilangan bulat dan tidak boleh lebih dari 100. Nilai n yang dimasukkan oleh pengguna akan disimpan dalam variabel n.

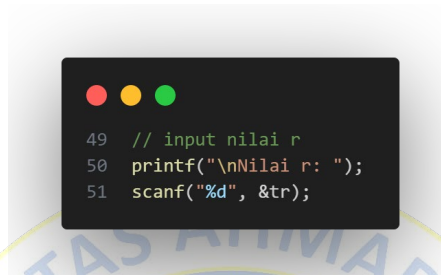


```
42 // input data ke dalam array
43 for (i = 1; i <= n; i++) // ulangi untuk semua data hingga data ke-n
44 {
45     printf("Masukkan nilai himpunan a[%d] : ", i);
46     scanf("%d", &a[i]);
47 }
```

Gambar 8 Input data ke dalam array,. (Sumber: Penulis)

Kode tersebut merupakan bagian dari program untuk menghasilkan permutasi dari suatu himpunan. Kode tersebut bertujuan untuk meminta input nilai dari pengguna dan menyimpannya ke dalam array a.

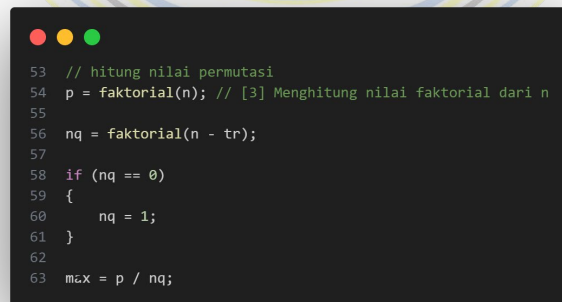
Pertama, program akan melakukan perulangan for sebanyak n kali, di mana n adalah jumlah elemen yang akan diisi ke dalam array. Setiap kali melakukan perulangan, program akan menampilkan pesan "Masukkan nilai himpunan $a[i]$: " di layar, di mana i adalah indeks dari elemen array yang akan diisi. Kemudian, program akan menunggu input dari pengguna menggunakan fungsi `scanf()`. Input yang diberikan oleh pengguna akan disimpan ke dalam elemen array $a[i]$. Setelah semua elemen array terisi, program akan melanjutkan ke bagian selanjutnya untuk menghasilkan permutasi dari himpunan tersebut.



```
49 // input nilai r
50 printf("\nNilai r: ");
51 scanf("%d", &tr);
```

Gambar 9 Menginput nilai r. (Sumber: Penulis)

Kode tersebut bertujuan untuk meminta input nilai r dari pengguna dan menyimpannya ke dalam variabel `tr`. Pertama, program akan menampilkan pesan "Nilai r: " di layar menggunakan fungsi `printf()`. Pesan tersebut bertujuan untuk meminta pengguna untuk memasukkan nilai r . Kemudian, program akan menunggu input dari pengguna menggunakan fungsi `scanf()`. Input yang diberikan oleh pengguna akan disimpan ke dalam variabel `tr` menggunakan operator `&` untuk mengambil alamat memori dari variabel tersebut. Setelah nilai r berhasil dimasukkan oleh pengguna dan disimpan ke dalam variabel `tr`, program akan melanjutkan ke bagian selanjutnya untuk menghasilkan permutasi dari himpunan.

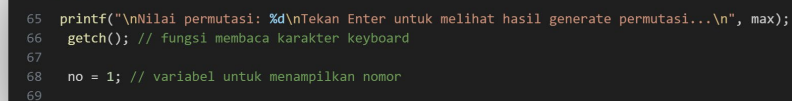


```
53 // hitung nilai permutasi
54 p = faktorial(n); // [3] Menghitung nilai faktorial dari n
55
56 nq = faktorial(n - tr);
57
58 if (nq == 0)
59 {
60     nq = 1;
61 }
62
63 max = p / nq;
```

Gambar 10 Perhitungan nilai permutasi. (Sumber: Penulis)

Kode tersebut bertujuan untuk menghitung nilai permutasi dari suatu himpunan dengan menggunakan rumus permutasi. program akan memanggil fungsi `faktorial(n)` untuk menghitung nilai faktorial dari n . Hasil faktorial dari n akan disimpan ke dalam variabel `p`.

Faktorial dari n digunakan untuk menghitung jumlah permutasi dari himpunan. Selanjutnya, program akan menghitung nilai faktorial dari $n - tr$ menggunakan fungsi faktorial($n - tr$). Hasil faktorial dari $n - tr$ akan disimpan ke dalam variabel nq . Variabel nq digunakan untuk menghitung jumlah permutasi yang memenuhi syarat nilai r yang dimasukkan oleh pengguna. Kemudian, program akan melakukan pengecekan apakah nilai nq sama dengan 0. Jika nilai nq sama dengan 0, maka variabel nq akan diubah nilainya menjadi 1. Hal ini dilakukan untuk menghindari terjadinya pembagian dengan nilai 0 pada perhitungan nilai max . Terakhir, program akan menghitung nilai max dengan membagi nilai p dengan nilai nq . Nilai max merupakan jumlah permutasi yang memenuhi syarat nilai r yang dimasukkan oleh pengguna.



```
65 printf("\nNilai permutasi: %d\nTekan Enter untuk melihat hasil generate permutasi...\n", max);
66 getch(); // fungsi membaca karakter keyboard
67
68 no = 1; // variabel untuk menampilkan nomor
69
```

Gambar 11 Menampilkan nilai permutasi. (Sumber: Penulis)

Kode tersebut bertujuan untuk menampilkan nilai permutasi yang telah dihitung sebelumnya dan meminta pengguna untuk menekan tombol Enter untuk melihat hasil generate permutasi. Pertama, program akan menampilkan pesan "Nilai permutasi: " di layar menggunakan fungsi printf(). Pesan tersebut akan menampilkan nilai permutasi yang telah dihitung sebelumnya dan disimpan ke dalam variabel max . Nilai max merupakan jumlah permutasi yang memenuhi syarat nilai r yang dimasukkan oleh pengguna. Kemudian, program akan menampilkan pesan "Tekan Enter untuk melihat hasil generate permutasi..." di layar. Pesan tersebut bertujuan untuk meminta pengguna untuk menekan tombol Enter untuk melihat hasil generate permutasi. Setelah itu, program akan menunggu pengguna menekan tombol Enter menggunakan fungsi getch(). Fungsi getch() akan membaca karakter keyboard yang ditekan oleh pengguna dan mengembalikan nilai karakter tersebut. Setelah pengguna menekan tombol Enter, program akan melanjutkan ke bagian selanjutnya untuk menghasilkan permutasi dari himpunan. Selanjutnya, program akan menginisialisasi variabel no dengan nilai 1. Variabel no digunakan untuk menampilkan nomor urut dari setiap permutasi yang dihasilkan.

```

70 // men-generate permutasi dengan algoritma generate next-permutation
71 // generate sebanyak nilai permutasi
72 for (x = 1; x <= max; x++)
73 {
74     printf("=. ", no);
75     for (i = 1; i <= tr; i++)
76     {
77         printf("%d ", a[i]);
78     }
79     printf("\n");
80     no++;
81
82     j = n - 1;
83
84     while (a[j] > a[j + 1])
85     {
86         j = j - 1; // j adalah subcript terbesar dengan aj < aj+1
87     }
88
89     k = n;
90
91     while (a[j] > a[k])
92     {
93         k = k - 1; // ak adalah integer terkecil dan lebih besar dari aj
94     }
95
96     // tukar aj dan ak
97     i = a[k]; // [4] Menyimpan nilai a[k] ke dalam variabel i
98     a[k] = a[j];
99     a[j] = i;
100
101     r = n;
102     s = j + 1;
103
104     while (r > s)
105     {
106         // tukar ar dan as
107         i = a[r];
108         // [5] Menukar nilai a[r] dan a[s]
109         a[r] = a[s];
110         a[s] = i;
111
112         r = r - 1;
113         s = s + 1;
114     }
115 }
116
117 getch();

```

Gambar 12 Menggenerate nilai permutasi. (Sumber: Penulis)

Kode tersebut bertujuan untuk menghasilkan permutasi dari himpunan dengan menggunakan algoritma generate next-permutation.

Pertama, program akan melakukan perulangan for sebanyak max kali, di mana max merupakan jumlah permutasi yang telah dihitung sebelumnya. Setiap kali melakukan perulangan, program akan menampilkan nomor urut dari permutasi menggunakan fungsi printf() dan variabel no. Selanjutnya, program akan melakukan perulangan for sebanyak tr kali, di mana tr merupakan nilai r yang dimasukkan oleh pengguna. Setiap kali melakukan perulangan, program akan menampilkan elemen array a dengan indeks i menggunakan fungsi printf(). Setelah menampilkan elemen array a, program akan menghitung nilai j dengan mengurangi n dengan 1. Selanjutnya, program akan melakukan perulangan while untuk mencari nilai j yang merupakan subscript terbesar dengan $a[j] < a[j+1]$. Setelah nilai j ditemukan, program akan menghitung nilai k dengan menginisialisasi k dengan nilai n. Selanjutnya, program akan melakukan perulangan while untuk mencari nilai k yang merupakan integer terkecil dan lebih besar dari $a[j]$. Setelah nilai k ditemukan, program akan menukar nilai $a[j]$ dengan nilai $a[k]$ menggunakan variabel i sebagai variabel penampung sementara. Selanjutnya, program akan menginisialisasi nilai r dengan n dan nilai s dengan j+1. Kemudian, program akan melakukan perulangan while untuk menukar nilai $a[r]$ dengan nilai $a[s]$ hingga nilai r lebih kecil atau sama dengan nilai s.

Setelah selesai menghasilkan permutasi, program akan menunggu pengguna menekan tombol Enter menggunakan fungsi getch(). Fungsi getch() akan membaca karakter keyboard yang ditekan oleh pengguna dan

Berikut adalah outputnya.

```

C generatePermutasi.c
17
18 {
19     if (nq == 0)
20     {
21         nq = 1;
22     }
23     max = p / nq;
24     printf("\nNilai permutasi: Tekan Enter untuk melihat hasil generate permutasi...\n", max);
25     getch(); // fungsi membaca karakter keyboard
26     no = 1; // variabel untuk menampilkan nomor
27 }
28
PROBLEMS TERMINAL OUTPUT DEBUG CONSOLE COMMENTS
PS D:\Document Ndk\Kulia\Semester 2\Matematika Diskrit\Praktikum\Portemon 9> cd "d:\Document Ndk\Kulia\Semester 2\Matematika Diskrit\Praktikum\Portemon 9\"; if ($?) { gcc generatePermutasi.c -o generatePermutasi }; if ($?) { .\generatePermutasi }
PROGRAM GENERATE PERMUTASI
Oleh Nuh, Farid Hendianto
-----
Masukkan nilai n : 3
Masukkan nilai himpunan a[1] : 1
Masukkan nilai himpunan a[2] : 2
Masukkan nilai himpunan a[3] : 3
Nilai r : 2
Nilai permutasi: 6
Tekan Enter untuk melihat hasil generate permutasi...
1. 1 2
2. 1 3
3. 2 1
4. 2 3
5. 3 1
6. 3 2
  
```

Gambar 13 Output programnya. (Sumber: Penulis)

Output program tersebut adalah hasil dari program generate permutasi yang telah dijalankan dengan input nilai n dan himpunan a yang dimasukkan oleh pengguna.

Program akan menampilkan pesan "PROGRAM GENERATE PERMUTASI" dan "Oleh Moh. Farid Hendianto" di layar sebagai header dari program.

Selanjutnya, program akan meminta pengguna untuk memasukkan nilai n dan himpunan a menggunakan fungsi `printf()` dan `scanf()`. Setelah nilai n dan himpunan a dimasukkan, program akan meminta pengguna untuk memasukkan nilai r menggunakan fungsi `printf()` dan `scanf()`.

Setelah nilai r dimasukkan, program akan menghitung nilai permutasi menggunakan rumus permutasi dan menampilkan nilai permutasi tersebut di layar menggunakan fungsi `printf()`. Program akan meminta pengguna untuk menekan tombol Enter untuk melihat hasil generate permutasi menggunakan fungsi `getch()`.

Setelah pengguna menekan tombol Enter, program akan menampilkan hasil generate permutasi dari himpunan a dengan jumlah permutasi sebanyak nilai \max yang telah dihitung sebelumnya. Setiap permutasi akan ditampilkan dengan nomor urut dan elemen-elemen himpunan a yang telah diacak.

Terakhir, program akan menunggu pengguna menekan tombol Enter menggunakan fungsi `getch()` sebelum program berakhir.

Untuk mengakses kodingan, dapat melihat link github berikut.

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