Tugas Praktikum Algoritma dan Struktur Data

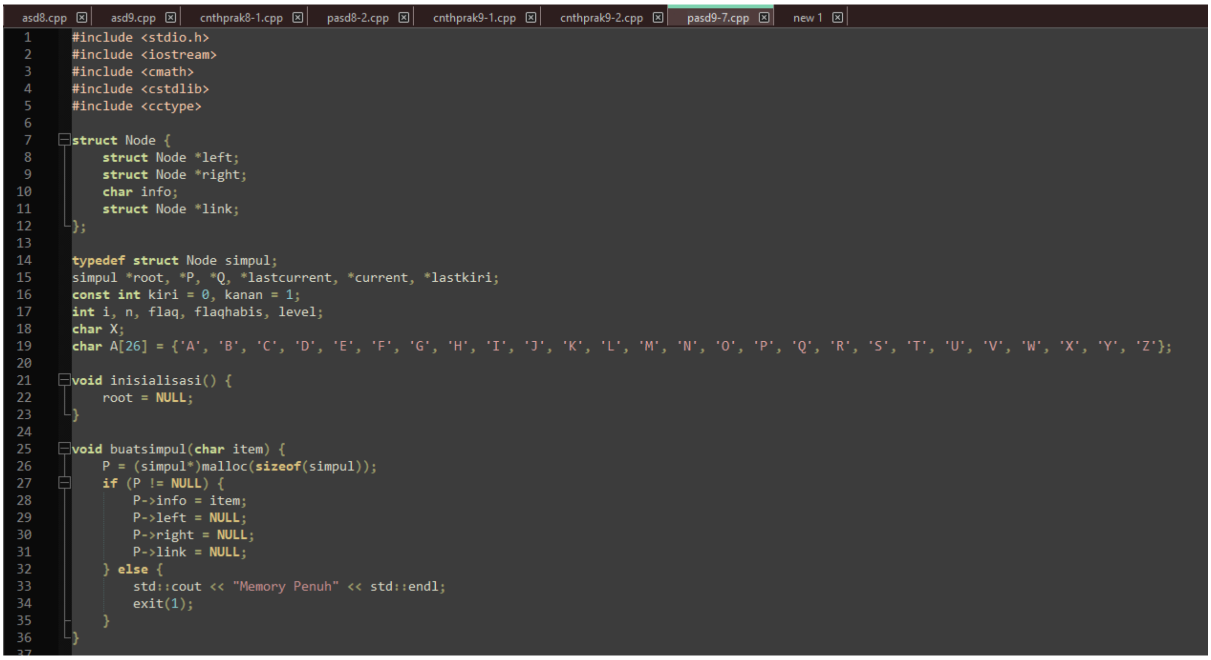
Nama : Agil Deriansyah Hasan
Nim : 4522210125

Dosen Pengajar :

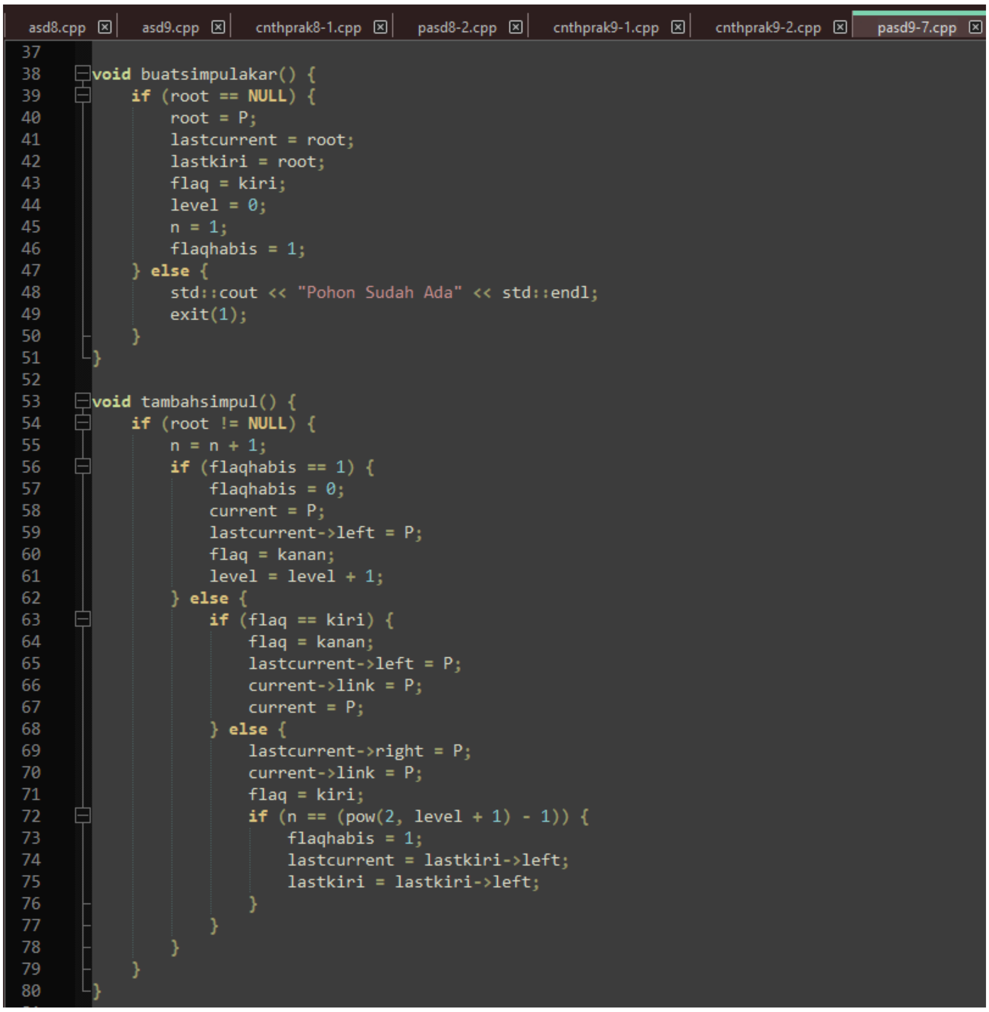
Dra.SRI REZEKI CANDRA NURSARI,M.Kom
Prak. Algoritma dan Struktur Data - I

S1-Teknik Informatika
Fakultas Teknik
Universitas Pancasila 2023/2024

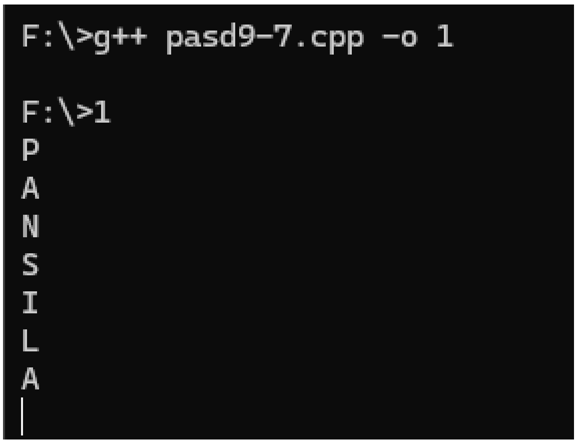
pasd9-7



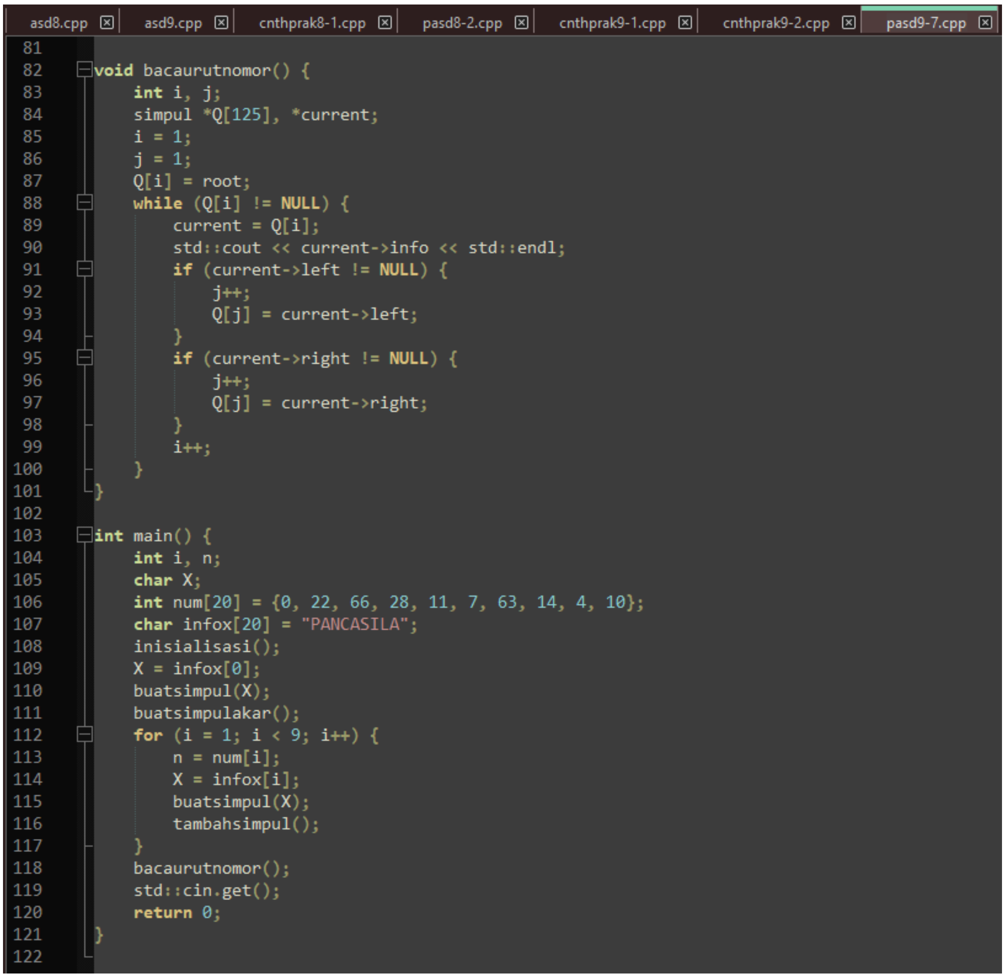
```
#include <stdio.h>
#include <iostream>
#include <cmath>
#include <cstdlib>
#include <cctype>

struct Node {
    struct Node *left;
    struct Node *right;
    char info;
    struct Node *link;
};

typedef struct Node simpul;
simpul *root, *P, *Q, *lastcurrent, *current, *lastkiri;
const int kiri = 0, kanan = 1;
int i, n, flaq, flaqhabis, level;
char X;
char A[26] = {'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z'};

void inisialisasi() {
    root = NULL;
}

void buatsimpul(char item) {
    P = (simpul*)malloc(sizeof(simpul));
    if (P != NULL) {
        P->info = item;
        P->left = NULL;
        P->right = NULL;
        P->link = NULL;
    } else {
        std::cout << "Memory Penuh" << std::endl;
        exit(1);
    }
}
```



```

void buatsimpulakar() {
    if (root == NULL) {
        root = P;
        lastcurrent = root;
        lastkiri = root;
        flaq = kiri;
        level = 0;
        n = 1;
        flaqhabis = 1;
    } else {
        std::cout << "Pohon Sudah Ada" << std::endl;
        exit(1);
    }
}

void tambahsimpul() {
    if (root != NULL) {
        n = n + 1;
        if (flaqhabis == 1) {
            flaqhabis = 0;
            current = P;
            lastcurrent->left = P;
            flaq = kanan;
            level = level + 1;
        } else {
            if (flaq == kiri) {
                flaq = kanan;
                lastcurrent->left = P;
                current->link = P;
                current = P;
            } else {
                lastcurrent->right = P;
                current->link = P;
                flaq = kiri;
                if (n == (pow(2, level + 1) - 1)) {
                    flaqhabis = 1;
                    lastcurrent = lastkiri->left;
                    lastkiri = lastkiri->left;
                }
            }
        }
    }
}
```



```
F:\>g++ pasd9-7.cpp -o 1

F:\>1
P
A
N
S
I
L
A
```



```cpp

void bacaurutnomor() {
    int i, j;
    simpul *Q[125], *current;
    i = 1;
    j = 1;
    Q[i] = root;
    while (Q[i] != NULL) {
        current = Q[i];
        std::cout << current->info << std::endl;
        if (current->left != NULL) {
            j++;
            Q[j] = current->left;
        }
        if (current->right != NULL) {
            j++;
            Q[j] = current->right;
        }
        i++;
    }
}

int main() {
    int i, n;
    char X;
    int num[20] = {0, 22, 66, 28, 11, 7, 63, 14, 4, 10};
    char infox[20] = "PANCASILA";
    inisialisasi();
    X = infox[0];
    buatsimpul(X);
    buatsimpulakar();
    for (i = 1; i < 9; i++) {
        n = num[i];
        X = infox[i];
        buatsimpul(X);
        tambahsimpul();
    }
    bacaurutnomor();
    std::cin.get();
    return 0;
}

```

Pseudocode :

```
Kamus/Deklarasi Variabel fungsi buatsimpul
item : char

Algoritma/Deskrips fungsi buatsimpul(item)
P = (simpul*)malloc(sizeof(simpul));
if (P != NULL)
        P->info = item
        P->left = NULL
        P->right = NULL
        P->link = NULL
else
        print "Memory Penuh"
endif

Kamus/Deklarasi Variabel fungsi buatsimpulakar
-

Algoritma/Deskripsi fungsi buatsimpulakar
if (root == NULL)
        root = P
        lastcurrent = root
        lastkiri = root
        flaq = kiri
        level = 0
        n = 1
        flaqhabis = 1
else
        print "Pohon Sudah Ada"
endif
```

```
Kamus/Deklarasi Variabel fungsi tambahsimpul
-

Algoritma/Deskripsi fungsi tambahsimpul
if (root != NULL)
      n = n + 1
      if (flaqhabis == 1)
         flaqhabis = 0
         current = P
         lastcurrent->left = P
         flaq = kanan
         level = level + 1
       else
         if (flaq == kiri)
            flaq = kanan
            lastcurrent->left = P
            current->link = P
            current = P
         else
            lastcurrent->right = P
            current->link = P
            flaq = kiri
            if (n == (pow(2, level + 1) - 1))
               flaqhabis = 1
               lastcurrent = lastkiri->left
               lastkiri = lastkiri->left
endif
endif
endif
endif
```

```
Kamus/Deklarasi Variabel fungsi bacaurutnomer
i , j = int

Algoritma/Deskripsi fungsi bacaurutnomer
simpul *Q[125], *current
   i = 1
   j = 1
   Q[i] = root
   while (Q[i] != NULL)
      current = Q[i]
      print current->info
      if (current->left != NULL)
         j++
         Q[j] = current->left
        endif
      if (current->right != NULL)
         j++
         Q[j] = current->right
        endif
i++
endwhile
```

```
Kamus/Deklarasi Variabel fungsi utama
i, n, flaq, flaqhabis, level = int
X = char
num [20] = int
infox[20] = int
kiri,kanan = const int

Algoritma/Deskripsi fungsi utama
struct Node {left, right, info, link}
num[20] = {0, 22, 66, 28, 11, 7, 63, 14, 4, 10}
infox[20] = "PANCASILA"
inisialisasi()
   X = infox[0]
   buatsimpul(X)
   buatsimpulakar()
   for (i = 1; i < 9; i++)
      n = num[i]
      X = infox[i]
      buatsimpul(X)
      tambahsimpul()
    endfor
   bacaurutnomor()
```

Algoritma :

```
1. Membuat fungsi inisialisasi
2. root = NULL
3. Memabu fungsi buatsimpul ( item )
4. P = (simpul*)malloc(sizeof(simpul))
5. Jika (P != NULL) maka kerjakan baris 6 s.d 9 kalau tidak baris 10
6. P->info = item
7. P->left = NULL
8. P->right = NULL
9. P->link = NULL
10. print "Memory Penuh"
11. Membuat fungsi buatsimpulakar
12. Jika (root == NULL) maka kerjakan baris 13 s.d 19 kalau tidak baris 20
13. root = P
14. lastcurrent = root
15. lastkiri = root
16. flaq = kiri
17. level = 0
18. n = 1
19. flaqhabis = 1
20. "Pohon Sudah Ada"
21. Membuat fungsi tambahsimpul
22. Jika (root != NULL) maka kerjakan baris 23 s.d 41
23. n = n + 1
24. Jika (flaqhabis == 1) maka kerjakan baris 25 s.d 29 kalau tidak 30 s.d 25 s.d 41
25. flaqhabis = 0
26. current = P
27. lastcurrent->left = P
28. flaq = kanan
29. level = level + 1
30. Jika (flaq == kiri) maka kerjakan baris 31 s.d 34 kalau tidak 31 s.d 41
31. flaq = kanan
32. lastcurrent->left = P
33. current->link = P
34. current = P
35. lastcurrent->right = P
36. current->link = P
37. flaq = kiri
38. Jika (n == (pow(2, level + 1) - 1)) maka kerjakan baris 39 s.d 41
39. flaqhabis = 1
40. lastcurrent = lastkiri->left
41. lastkiri = lastkiri->left
42. Membuat fungsi bacaurutnomer
43. i = 1
44. j = 1
45. Q[i] = root
46. Selama (Q[i] != NULL) Maka kerjakan baris 47 s.d 55
47. current = Q[i]
48. Mencetak/Menampilkan Nilai Current-> info
49. Jika (current->left != NULL) maka kerjakan baris 50 s.d 51
50. j++
51. Q[j] = current->left
52. Jika (current->right != NULL) maka kerjakan baris 53 s.d 54
53. j++
54. Q[j] = current->right
55. i++
56. Membuat fungsi utama
57. Deklarasi struktur (struct{left, right, info, link}
58. num[20] = {0, 22, 66, 28, 11, 7, 63, 14, 4, 10}
59. infox[20] = "PANCASILA"
60. Memanggil fungsi inisialisasi
61. X= infox[0]
62. Memanggil fungsi buatsimpul(X)
63. Memanggil fungsi buatsimpulakar
64. Selama (i = 1) maka kerjakan baris 65 s.d 69
65. n = num[i]
66. X = infox[i]
67. buatsimpul(X)
68. tambahsimpul()
69. i++
70.Memanggil fungsi bacaurrutnomer
71. Selesai
```