

TUGAS JOBSHEET 04
PRAKTIKUM STRUKTUR DATA



DOSEN PENGAMPU:
Vera Irma Delianti, S.Pd., M.Pd.T.

OLEH:
M. ilham
23343008

PROGRAM STUDI INFORMATIKA
DEPARTEMEN TEKNIK ELEKTRONIKA
FAKULTAS TEKNIK
UNIVERSITAS NEGERI PADANG
2024

1. implementasi-array-dari-stack.c

a. SOURCE CODE

```
/* Nama file   : implementasi array dari stack
Pembuat       : M. Ilham 23343008
Tgl pembuatan : 26 February 2024*/

#include <stdio.h>

int stack[100], i, j, choice = 0, n, top = -1;
void push();
void pop();
void show();

int main(){

    int choice = 0;
    printf("*****Stack operations using
array*****\n");

    printf("\n*****
*****");

    printf("\nEnter the number of elements in the
stack : ");
    scanf("%d", &n);
    printf("\n-----
-----");

    while(choice != 4){
        printf("\n\nChoose one from the below
options...\n");
        printf("\n 1. Push\n 2. Pop\n 3. Show\n 4.
Exit");
        printf("\nEnter your choice : ");
        scanf("%d", &choice);
        switch(choice){
            case 1:
```

```

        push();
        break;
    case 2:
        pop();
        break;
    case 3:
        show();
        break;
    case 4:
        printf("Exiting...");
        break;
    default:
        printf("Please Enter valid
choice");

        break;
    }
}
return 0;
}

void push(){
    int val;
    if(top == n - 1){
        printf("\nOverflow");
    }
    else {
        printf("Enter the value?");
        scanf("%d", &val);
        top = top + 1;
        stack[top] = val;
    }
}

void pop(){
    if(top == -1){
        printf("Underflow");
    }
}

```

```

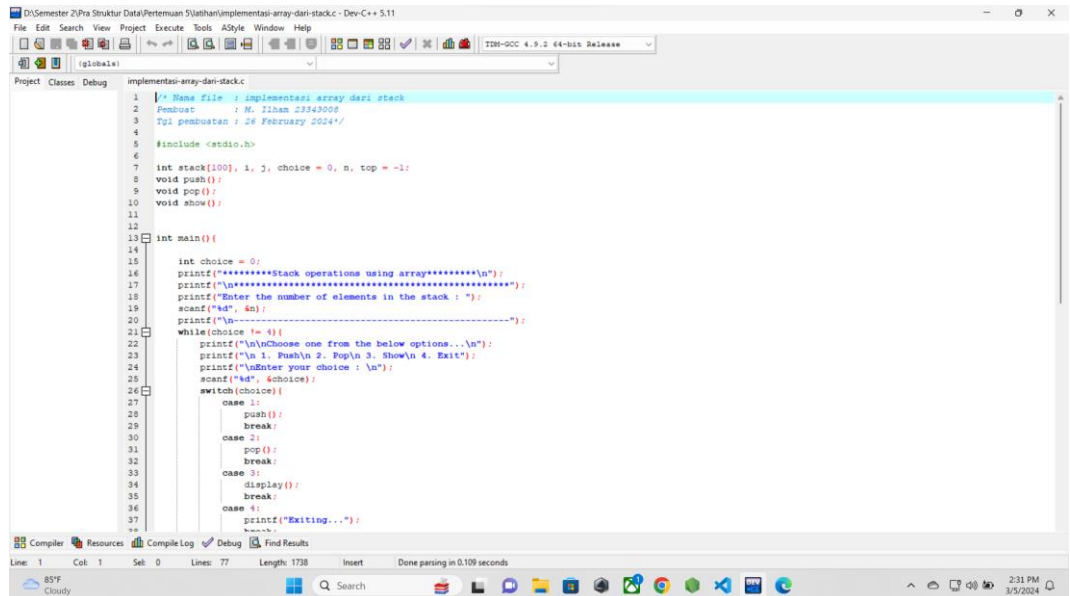
        else {
            top = top - 1;
        }
    }

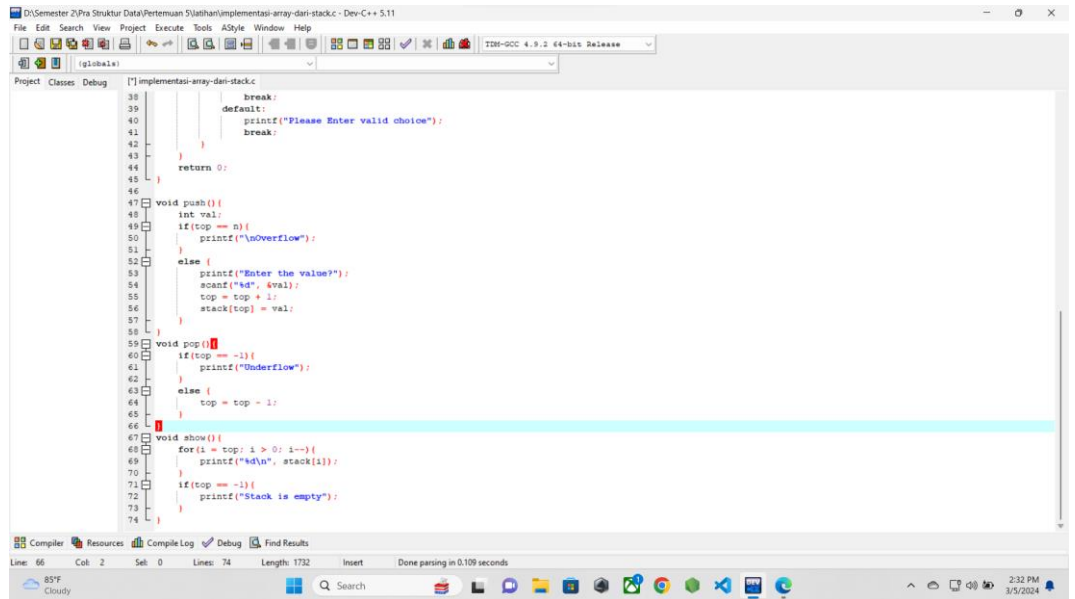
void show(){
    for(i = top; i >= 0; i--){
        printf("%d\n", stack[i]);
    }

    if(top == -1){
        printf("Stack is empty");
    }
}

```

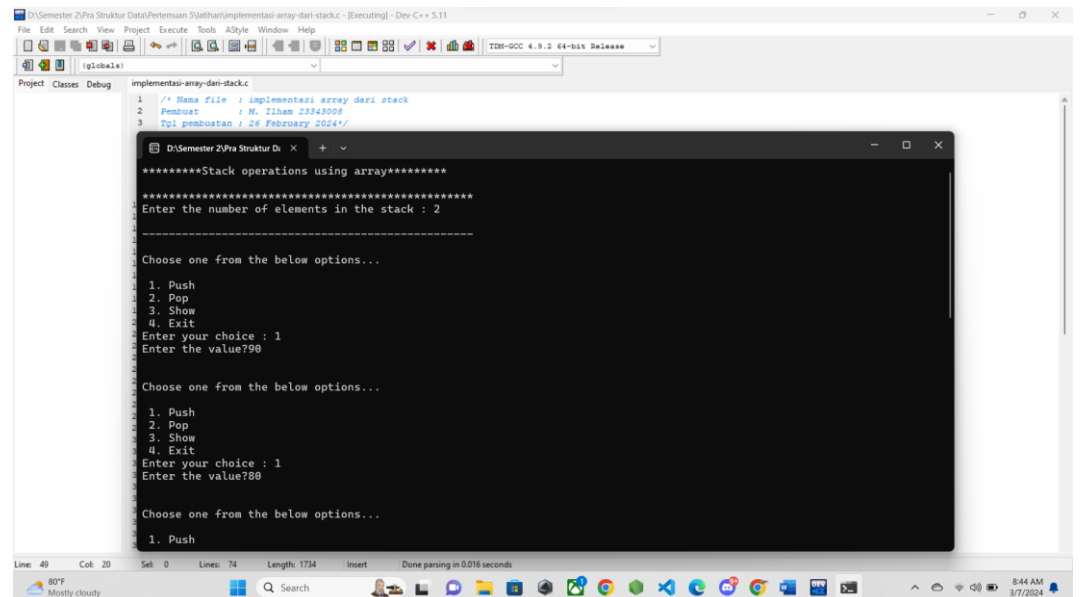
b. SCREENSHOT PROGRAM





```
38         break;
39     default:
40         printf("Please Enter valid choice");
41         break;
42     }
43 }
44 return 0;
45 }
46
47 void push() {
48     int val;
49     if (top == 0) {
50         printf("\nOverflow");
51     }
52     else {
53         printf("Enter the value?");
54         scanf("%d", &val);
55         top = top + 1;
56         stack[top] = val;
57     }
58 }
59
60 void pop() {
61     if (top == -1) {
62         printf("Underflow");
63     }
64     else {
65         top = top - 1;
66     }
67 }
68
69 void show() {
70     for (i = top; i > 0; i--) {
71         printf("%d\n", stack[i]);
72     }
73     if (top == -1) {
74         printf("Stack is empty");
75     }
76 }
```

c. SCREENSHOT OUTPUT



```
1 // Nama file : implementasi array dari stack
2 Pembuat : M. Ilham 20342000
3 Tgl pembuatan : 28 February 2024*/

*****Stack operations using array*****
*****
Enter the number of elements in the stack : 2
-----
Choose one from the below options...
1. Push
2. Pop
3. Show
4. Exit
Enter your choice : 1
Enter the value:790

Choose one from the below options...
1. Push
2. Pop
3. Show
4. Exit
Enter your choice : 1
Enter the value:780

Choose one from the below options...
1. Push
```

The image contains two screenshots of a C++ program running in Dev-C++. The program implements a stack using an array. The first screenshot shows the program starting with a menu: 1. Push, 2. Pop, 3. Show, 4. Exit. The user enters choice 3, and the program displays '80' and '90'. Then, the user enters choice 1, and the program displays 'Overflow'. The second screenshot shows the program continuing with the same menu. The user enters choice 2, and the program displays 'Overflow'. Then, the user enters choice 3, and the program displays '80' and '90'. Finally, the user enters choice 1, and the program displays 'Overflow'.

```
1 // Name file : implementasi array dari stack
2 // Pembuat : M. Ilham 23343008
3 // Tgl pembuatan : 26 February 2024*/

2. Pop
3. Show
4. Exit
Enter your choice : 3
80
90

Choose one from the below options...
1. Push
2. Pop
3. Show
4. Exit
Enter your choice : 1
Overflow

Choose one from the below options...
1. Push
2. Pop
3. Show
4. Exit
Enter your choice : 2
Overflow

Choose one from the below options...
1. Push
2. Pop
3. Show
4. Exit
Enter your choice : 3
80
90

Choose one from the below options...
1. Push
2. Pop
3. Show
4. Exit
Enter your choice : 1
Overflow
```

d. PENJELASAN PROGRAM

Pada program ini, akan di implementasikan stack dengan array, saat element baru di tambahkan, ia akan diletakkan di element paling belakang. Saat element di hapus, element di paling belakang jugalah yang di hapus, namun karena ini menggunakan alokasi memori otomatis berarti memori akan di alokasikan di stack sehingga tidak bisa melakukan pembebasan memori, karena itu pada fungsi pop hanya akan menghapus akses ke element paling belakang.

Saat melakukan show, maka akan di tampilkan satu persatu dari element paling akhir hingga paling awal sehingga terkesan ditampilkan dari element paling atas tumpukan.

2. implementasi-linked-list-dari-stack.c

a. SOURCE CODE

```
/* Nama file   : implementasi linked list dari Stack
Pembuat      : M. Ilham 23343008
Tgl pembuatan : 26 February 2024*/

#include <stdio.h>

void push();
void pop();
void display();

struct node {
    int val;
    struct node *next;
};

struct node *head;

int main() {

    int choice = 0;
    printf("*****Stack operations using linked
list*****\n");
    printf("\n-----
-----");
    while(choice != 4) {
        printf("\n\nChoose one from the below
options...\n");
```

```

        printf("\n 1. Push\n 2. Pop\n 3. Show\n 4.
Exit");
        printf("\nEnter your choice : \n");
        scanf("%d", &choice);
        switch(choice){
            case 1:
                push();
                break;
            case 2:
                pop();
                break;
            case 3:
                display();
                break;
            case 4:
                printf("Exiting...");
                break;
            default:
                printf("Please Enter valid
choice");
                break;
        }
    }

    return 0;
}

void push(){
    int val;
    struct node *ptr = (struct node
*)malloc(sizeof(struct node));
    if(ptr == NULL){
        printf("just and only able to push the
element");
    }
    else {

```



```

        printf("Enter the value : ");
        scanf("%d", &val);
        if(head == NULL){
            ptr->val = val;
            ptr->next = NULL;
            head = ptr;
        }
        else {
            ptr->val = val;
            ptr->next = head;
            head = ptr;
        }
        printf("Item pushed");
    }
}

```

```

void pop(){
    int item;
    struct node *ptr;
    if(head == NULL){
        printf("Underflow");
    }
    else {
        item = head->val;
        ptr = head;
        head = head->next;
        free(ptr);
        printf("Item popped");
    }
}

```

```

void display(){
    int i;
    struct node *ptr;
    ptr = head;
    if(ptr == NULL){

```

```

        printf("Stack is empty\n");
    }
else {
    printf("Printing Stack elements\n");
    while(ptr != NULL){
        printf("%d\n", ptr->val);
        ptr = ptr->next;
    }
}
}

```

b. SCREENSHOT PROGRAM

```

1  /* Nama file : implementasi linked list dari Stack
2  Pembuat : M. Ilham 23342008
3  Tgl pembuatan : 26 February 2024 */
4
5  #include <stdio.h>
6
7  void push();
8  void pop();
9  void display();
10
11 struct node {
12     int val;
13     struct node *next;
14 }
15
16 struct node *head;
17
18 int main() {
19     int choice = 0;
20     printf("*****Stack operations using linked list*****\n");
21     printf("\n-----");
22     while(choice != 4) {
23         printf("\nChoose one from the below options...\n");
24         printf("\n 1. Push\n 2. Pop\n 3. Show\n 4. Exit");
25         printf("\nEnter your choice : \n");
26         scanf("%d", &choice);
27         switch(choice) {
28             case 1:
29                 push();
30                 break;
31             case 2:
32                 pop();
33                 break;
34             case 3:
35                 display();
36                 break;
37             case 4:
38                 printf("Exiting...");
39         }
40     }
41 }

```

```

40     break;
41     default:
42         printf("Please Enter valid choice");
43         break;
44 }
45
46 return 0;
47 }
48
49 void push() {
50     int val;
51     struct node *ptr = (struct node *)malloc(sizeof(struct node));
52     if(ptr == NULL) {
53         printf("Just and only able to push the element");
54     }
55     else {
56         printf("Enter the value : ");
57         scanf("%d", &val);
58         if(head == NULL) {
59             ptr->val = val;
60             ptr->next = NULL;
61             head = ptr;
62         }
63         else {
64             ptr->val = val;
65             ptr->next = NULL;
66             head = ptr;
67             printf("Item pushed");
68         }
69     }
70 }
71
72 void pop() {
73     int item;
74     struct node *ptr;
75     if(head == NULL) {
76         printf("Underflow");
77     }
78 }

```

```
64 else {
65     ptr->val = val;
66     ptr->next = NULL;
67     head = ptr;
68     printf("Item pushed\n");
69 }
70 }
71
72
73 void pop() {
74     int item;
75     struct node *ptr;
76     if(head == NULL) {
77         printf("Underflow\n");
78     }
79     else {
80         item = head->val;
81         ptr = head;
82         head = head->next;
83         free(ptr);
84         printf("Item popped\n");
85     }
86 }
87
88 void display() {
89     int i;
90     struct node *ptr;
91     ptr = head;
92     if(ptr == NULL) {
93         printf("Stack if empty\n");
94     }
95     else {
96         printf("Printing Stack elements\n");
97         while(ptr != NULL) {
98             printf("%d\n", ptr->val);
99             ptr = ptr->next;
100         }
101     }
102 }
```

c. SCREENSHOT OUTPUT

```
*****Stack operations using linked list*****
Choose one from the below options...
1. Push
2. Pop
3. Show
4. Exit
Enter your choice :
1
Enter the value : 1
Item pushed
Choose one from the below options...
1. Push
2. Pop
3. Show
4. Exit
Enter your choice :
1
Enter the value : 20
Item pushed
Choose one from the below options...
1. Push
2. Pop
```

```
33 pop():  
34 break;  
35 case 3:  
2. Pop  
3. Show  
4. Exit  
Enter your choice :  
1  
Enter the value : 30  
Item pushed  
Choose one from the below options...  
1. Push  
2. Pop  
3. Show  
4. Exit  
Enter your choice :  
3  
Printing Stack elements  
30  
20  
1  
Choose one from the below options...  
1. Push  
2. Pop  
3. Show  
4. Exit  
Enter your choice :  
2
```

```
1. Push  
2. Pop  
3. Show  
4. Exit  
Enter your choice :  
2  
Item popped  
Choose one from the below options...  
1. Push  
2. Pop  
3. Show  
4. Exit  
Enter your choice :  
2  
Item popped  
Choose one from the below options...  
1. Push  
2. Pop  
3. Show  
4. Exit  
Enter your choice :  
3  
Printing Stack elements  
1
```

d. PENJELASAN PROGRAM

Pada program ini digunakan linkedlist, sehingga saat melakukan push element baru, element akan diletakkan di node paling depan karena dengan linkedlist kita bisa lebih leluasa meletakkan posisi dari node. Pada saat pop, element paling depan yaitu headnya akan di hapus sehingga nodenya tidak bisa di akses lagi. Lalu posisi head di atur ke node selanjutnya. Pada saat show, akan ditampilkan element secara berurutan dengan menggunakan while loop sehingga node ditampilkan dari yang terbaru atau paling atas ke yang paling bawah

3. polindrom.c

a. SOURCE CODE

```
/* Nama file   : polindrom
Pembuat       : M. Ilham 23343008
Tgl pembuatan : 26 February 2024*/

#include <stdio.h>
#include <string.h>

#define MAXSTACK 100
#define MAXCOLUMN 30
char stack[MAXSTACK][MAXCOLUMN];
int top = -1;

void check(){
    int i, j;
    int isPolyndrome = 1;
    int length = strlen(stack[top]);
    for(i = 0, j = length - 1; i < j; i++, j--){
        if(stack[top][i] != stack[top][j]){
            isPolyndrome = 0;
            break;
        }
    }
    if(!isPolyndrome) printf("Kalimat tersebut
    bukan polyndrom\n");
    else printf("Kalimat tersebut adalah
    polyndrom\n");
}

void push(char kalimat[]){
    if(top == MAXSTACK) puts("Penuh");
    else {
        if (kalimat[strlen(kalimat) - 1] == '\n')
            kalimat[strlen(kalimat) - 1] = '\0';
        ++top;
    }
}
```

```

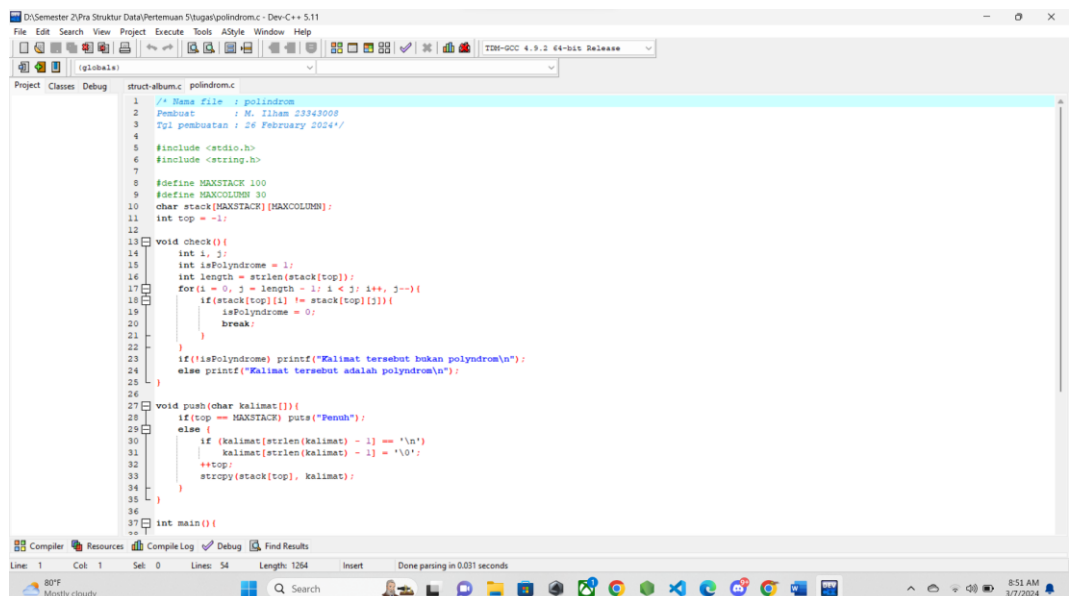
        strcpy(stack[top], kalimat);
    }
}

int main(){
    char kalimat[MAXCOLUMN];
    char run;
    do {
        printf("\nSelamat datang di program
menentukan kalimat polyndrom dengan stack\n");
        printf("Masukkan kalimat : ");
        fgets(kalimat, sizeof(kalimat), stdin);
        push(kalimat);
        check();
        printf("Ulang ? y/n : ");
        scanf("%c", &run);
        getchar();
    } while(run == 'y' || run == 'Y');

    return 0;
}

```

b. SCREENSHOT PROGRAM



```
18 if(stack[top][i] != stack[top][j]){
19     isPolindrome = 0;
20     break;
21 }
22
23 if(!isPolindrome) printf("Kalimat tersebut bukan polyndrom\n");
24 else printf("Kalimat tersebut adalah polyndrom\n");
25
26
27 void push(char kalimat){
28     if(top == MAXSTACK) puts("Penuh");
29     else {
30         if (kalimat[strlen(kalimat) - 1] == '\n')
31             kalimat[strlen(kalimat) - 1] = '\0';
32         ++top;
33         strcpy(stack[top], kalimat);
34     }
35 }
36
37 int main(){
38     char kalimat[MAXCOLON];
39     char run;
40
41     do {
42         printf("\nSelamat datang di program menentukan kalimat polyndrom dengan stack\n");
43         printf("Masukkan kalimat : ");
44         fgets(kalimat, sizeof(kalimat), stdin);
45         push(kalimat);
46         check();
47         printf("Ulang ? y/n : ");
48         scanf("%c", &run);
49         getch();
50     } while(run == 'y' || run == 'Y');
51
52     return 0;
53 }
```

c. SCREENSHOT OUTPUT

```
16 int length = strlen(stack[top]);
17 for(i = 0, j = length - 1; i < j; i++, j--){
18     if(stack[top][i] != stack[top][j]){
19         isPolindrome = 0;
20         break;
21     }
22 }
23 if(!isPolindrome) printf("Kalimat tersebut bukan polyndrom\n");
24 else printf("Kalimat tersebut adalah polyndrom\n");
25
26
27 void push(char kalimat){
28     if(top == MAXSTACK) puts("Penuh");
29     else {
30         if (kalimat[strlen(kalimat) - 1] == '\n')
31             kalimat[strlen(kalimat) - 1] = '\0';
32         ++top;
33         strcpy(stack[top], kalimat);
34     }
35 }
36
37 int main(){
38     char kalimat[MAXCOLON];
39     char run;
40
41     do {
42         printf("\nSelamat datang di program menentukan kalimat polyndrom dengan stack\n");
43         printf("Masukkan kalimat : ");
44         fgets(kalimat, sizeof(kalimat), stdin);
45         push(kalimat);
46         check();
47         printf("Ulang ? y/n : ");
48         scanf("%c", &run);
49         getch();
50     } while(run == 'y' || run == 'Y');
51
52     return 0;
53 }
```

d. PENJELASAN PROGRAM

Pada program ini untuk mencari sebuah kalimat polindrom atau tidak. Pertama akan diminta kalimat yang akan diperiksa, lalu tiap karakter dari string yang di input akan dimasukkan kedalam variabel array dua dimensi bernama stack. Lalu untuk memeriksa apakah kalimatnya polindrom dengan method check().

Didalam method check akan diperiksa apakah element awal dan akhirnya sama, lalu akan dicek setiap iterasi, jika ada kalimat yang berbeda program pengecekan dengan loop akan berhenti dan di dapatkan

hasil bukan polindrom, jika sampai akhir berjalannya program looping tidak ada kalimat yang berbeda, kalimat dinyatakan sebagai polindrom.