

RESPONSI
PRAKTIKUM DASAR PEMROGRAMAN
2023/2024

IDENTITAS

Nama : Muhammad Riza Zaidan
NIM : L0223031
Kelas : B Sains Data
Judul Program : Note for Lutpi

DOKUMENTASI PROGRAM

SOURCE CODE

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

#define MAX_CATATAN 100

struct Catatan {
    char judul[50];
    char isi[200];
};

struct Catatan catatan[MAX_CATATAN];

int jumlahCatatan = 0;

// Fungsi untuk menambahkan catatan baru
void tambahCatatan() {

    printf("\nMENAMBAHKAN CATATAN:\n");

    if (jumlahCatatan < MAX_CATATAN) {
        char judulBaru[50];

        // Meminta pengguna untuk memasukkan judul catatan
        printf("\nMasukkan judul catatan: ");
        scanf(" %[^\\n]", judulBaru);

        // Mengecek apakah judul sudah ada dalam catatan
```

```

        for (int i = 0; i < jumlahCatatan; i++) {
            if (strcmp(catatan[i].judul, judulBaru) == 0) {
                // Jika judul sudah ada, meminta pengguna untuk mengganti judul
                printf("Judul '%s' sudah ada.\nMohon masukkan judul baru: ", judulBaru);
                scanf(" %[^\\n]", judulBaru);
                // Kembali memeriksa apakah judul baru sudah ada
                i = -1; // Mengulang pemeriksaan dari awal
            }
        }

        // Memasukkan isi catatan setelah mendapatkan judul yang unik
        printf("Masukkan isi catatan: \\n");
        scanf(" %[^\\n]", catatan[jumlahCatatan].isi);

        // Menyimpan judul yang sudah diverifikasi ke dalam catatan
        strcpy(catatan[jumlahCatatan].judul, judulBaru);

        jumlahCatatan++;
        printf("\\n[Catatan berhasil ditambahkan!]\\n");
    } else {
        printf("\\n[Maaf, batas maksimum catatan telah tercapai.]\\n");
    }
}

// Fungsi untuk menampilkan semua catatan
void tampilkanSemuaCatatan() {

    printf("\\nSEMUA CATATAN LUTPI:\\n");

    if (jumlahCatatan > 0) {
        for (int i = 0; i < jumlahCatatan; i++) {
            printf("\\nJudul: %s\\n", catatan[i].judul);
            printf("Isi: %s\\n", catatan[i].isi);
        }
    } else {
        printf("\\n[ Belum ada catatan yang dimasukkan. ]\\n");
    }
}

// Fungsi untuk menampilkan satu catatan berdasarkan judul
void tampilkanSatuCatatan() {

    printf("\\nMENAMPILKAN CATATAN:\\n");

    char cariJudul[50];
    printf("\\nMasukkan judul catatan yang ingin ditampilkan: ");
    scanf(" %[^\\n]", cariJudul);

```

```

    struct Catatan *ptrCatatan = NULL;

    for (int i = 0; i < jumlahCatatan; i++) {
        if (strcmp(catatan[i].judul, cariJudul) == 0) {
            ptrCatatan = &catatan[i];
            break;
        }
    }

    if (ptrCatatan != NULL) {
        printf("\nCATATAN YANG DICARI:\n");
        printf("\nJudul: %s\n", ptrCatatan->judul);
        printf("Isi: %s\n", ptrCatatan->isi);
    } else {
        printf("\n[Catatan dengan judul '%s' tidak ditemukan.]\n", cariJudul);
    }
}

// Fungsi untuk mengganti isi catatan
void gantiCatatan() {

    printf("\nMENGGANTI ISI CATATAN:\n");

    char cariJudul[50];
    printf("\nMasukkan judul catatan yang ingin diganti: ");
    scanf(" %[^\\n]", cariJudul);

    struct Catatan *ptrCatatan = NULL;

    for (int i = 0; i < jumlahCatatan; i++) {
        if (strcmp(catatan[i].judul, cariJudul) == 0) {
            ptrCatatan = &catatan[i];
            break;
        }
    }

    if (ptrCatatan != NULL) {
        printf("Masukkan isi catatan yang baru: \n");
        scanf(" %[^\\n]", ptrCatatan->isi);
        printf("\n[Catatan berhasil diganti!]\n");
    } else {
        printf("\n[Catatan dengan judul '%s' tidak ditemukan.]\n", cariJudul);
    }
}

// Fungsi untuk menghapus catatan

```

```

void hapusCatatan() {

    printf("\nMENGHAPUS CATATAN:\n");

    char cariJudul[50];
    printf("\nMasukkan judul catatan yang ingin dihapus: ");
    scanf(" %[^\\n]", cariJudul);

    int catatanDitemukan = 0;

    for (int i = 0; i < jumlahCatatan; i++) {
        if (strcmp(catatan[i].judul, cariJudul) == 0) {
            // Menggeser catatan ke sebelumnya untuk menutup celah yang dihapus
            for (int j = i; j < jumlahCatatan - 1; j++) {
                strcpy(catatan[j].judul, catatan[j + 1].judul);
                strcpy(catatan[j].isi, catatan[j + 1].isi);
            }
            jumlahCatatan--;
            printf("\n[Catatan berhasil dihapus!]\n");
            catatanDitemukan = 1;
            break;
        }
    }

    if (!catatanDitemukan) {
        printf("\n[Catatan dengan judul '%s' tidak ditemukan.]\n", cariJudul);
    }
}

int main() {
    int pilihan;

    while (1) {
        printf("\n===== NOTE FOR LUTPI =====\n");
        printf("\nMENU:");
        printf("\n1. Tambah Catatan");
        printf("\n2. Tampilkan Semua Catatan");
        printf("\n3. Tampilkan Satu Catatan");
        printf("\n4. Ganti Catatan");
        printf("\n5. Hapus Catatan");
        printf("\n6. Keluar\n");
        printf("\nPilih menu (1-6): ");
        scanf("%d", &pilihan);

        switch (pilihan) {
            case 1:
                tambahCatatan();

```

```

        break;
    case 2:
        tampilkanSemuaCatatan();
        break;
    case 3:
        tampilkanSatuCatatan();
        break;
    case 4:
        gantiCatatan();
        break;
    case 5:
        hapusCatatan();
        break;
    case 6:
        printf("\nProgram diberhentikan. Dadah Lutpi!");
        exit(0);
    default:
        printf("\nPerintah tidak valid. Silakan masukkan perintah yang benar.\n");
    }
}

return 0;
}

```

OUTPUT

```
===== NOTE FOR LUTPI =====
```

```
MENU:
```

1. Tambah Catatan
2. Tampilkan Semua Catatan
3. Tampilkan Satu Catatan
4. Ganti Catatan
5. Hapus Catatan
6. Keluar

```
Pilih menu (1-6): 1
```

```
MENAMBAHKAN CATATAN:
```

```
Masukkan judul catatan: TUGAS DP8
```

```
Masukkan isi catatan:
```

```
Responsi Dasar Pemrograman 1: Membuat Program Catatan untuk Lutpi
```

```
[Catatan berhasil ditambahkan!]
```

===== NOTE FOR LUTPI =====

MENU:

1. Tambah Catatan
2. Tampilkan Semua Catatan
3. Tampilkan Satu Catatan
4. Ganti Catatan
5. Hapus Catatan
6. Keluar

Pilih menu (1-6): 1

MENAMBAHKAN CATATAN:

Masukkan judul catatan: TUGAS OAK7

Masukkan isi catatan:

Responsi Arsitektur Komputer 1: Komponen Komputer

[Catatan berhasil ditambahkan!]

===== NOTE FOR LUTPI =====

MENU:

1. Tambah Catatan
2. Tampilkan Semua Catatan
3. Tampilkan Satu Catatan
4. Ganti Catatan
5. Hapus Catatan
6. Keluar

Pilih menu (1-6): 2

SEMUA CATATAN LUTPI:

Judul: TUGAS DP8

Isi: Responsi Dasar Pemrograman 1: Membuat Program Catatan untuk Lutpi

Judul: TUGAS OAK7

Isi: Responsi Arsitektur Komputer 1: Komponen Komputer

===== NOTE FOR LUTPI =====

MENU:

1. Tambah Catatan
2. Tampilkan Semua Catatan
3. Tampilkan Satu Catatan

```
4. Ganti Catatan
5. Hapus Catatan
6. Keluar

Pilih menu (1-6): 3

MENAMPILKAN CATATAN:

Masukkan judul catatan yang ingin ditampilkan: TUGAS DP8

CATATAN YANG DICARI:

Judul: TUGAS DP8
Isi: Responsi Dasar Pemrograman 1: Membuat Program Catatan untuk Lutpi

===== NOTE FOR LUTPI =====

MENU:
1. Tambah Catatan
2. Tampilkan Semua Catatan
3. Tampilkan Satu Catatan
4. Ganti Catatan
5. Hapus Catatan
6. Keluar

Pilih menu (1-6): 4

MENGGANTI ISI CATATAN:

Masukkan judul catatan yang ingin diganti: TUGAS OAK7
Masukkan isi catatan yang baru:
Responsi Arsitektur Komputer 1: Makalah Komponen Komputer

[Catatan berhasil diganti!]

===== NOTE FOR LUTPI =====

MENU:
1. Tambah Catatan
2. Tampilkan Semua Catatan
3. Tampilkan Satu Catatan
4. Ganti Catatan
5. Hapus Catatan
6. Keluar

Pilih menu (1-6): 5
```

MENGHAPUS CATATAN:

Masukkan judul catatan yang ingin dihapus: TUGAS DP8

[Catatan berhasil dihapus!]

===== NOTE FOR LUTPI =====

Menu:

1. Tambah Catatan
2. Tampilkan Semua Catatan
3. Tampilkan Satu Catatan
4. Ganti Catatan
5. Hapus Catatan
6. Keluar

Pilih perintah (1-6): 6

Program diberhentikan. Dadah Lutpi!

INSTRUKSI PEMAKAIAAN

1. Tambah Catatan:

- Pengguna diminta untuk memasukkan judul catatan.
- Dilakukan pengecekan apakah judul sudah ada, jika ya, pengguna diminta untuk mengganti judul.
- Pengguna diminta untuk memasukkan isi catatan.

2. Tampilkan Semua Catatan:

- Semua catatan yang telah dimasukkan akan ditampilkan.

3. Tampilkan Satu Catatan:

- Pengguna diminta untuk memasukkan judul catatan yang ingin ditampilkan.
- Satu catatan dengan judul yang sesuai akan ditampilkan.

4. Ganti Catatan:

- Pengguna diminta untuk memasukkan judul catatan yang ingin diganti.
- Jika judul ditemukan, pengguna diminta untuk memasukkan isi catatan yang baru.

5. Hapus Catatan:

- Pengguna diminta untuk memasukkan judul catatan yang ingin dihapus.
- Jika judul ditemukan, catatan dengan judul tersebut akan dihapus.

6. Keluar Program:

- Program akan keluar dan berhenti berjalan.

FITUR-FITUR

1. Penggunaan Library:

```
C: > Users > muham > Documents > 2. UNS > 1. FATISDA > 1. SAINS DATA > 1. DASPEM > PRAKTIKUM > Responsi1_DP_L0223031 > C Responsi1_DP_L0223031 2ndVersion.c > Catatan > C
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4
```

#include <stdio.h>: Untuk fungsi input/output standar.

#include <stdlib.h>: Untuk fungsi umum, seperti exit().

#include <string.h>: Untuk fungsi-fungsi yang berhubungan dengan manipulasi string.

2. Batas Maksimum Catatan (MAX_CATATAN):

```
4
5  #define MAX_CATATAN 100
6
11
12  struct Catatan catatan[MAX_CATATAN];
13
14  int jumlahCatatan = 0;
15
```

Program memiliki batasan jumlah maksimum catatan yang dapat dimasukkan oleh pengguna. Ini menghindari potensi masalah memori yang tidak perlu.

3. Pengecekan Duplikasi Judul:

```
27
28      // Mengecek apakah judul sudah ada dalam catatan
29      for (int i = 0; i < jumlahCatatan; i++) {
30          if (strcmp(catatan[i].judul, judulBaru) == 0) {
31              // Jika judul sudah ada, meminta pengguna untuk mengganti judul
32              printf("Judul '%s' sudah ada.\nMohon masukkan judul baru: ", judulBaru);
33              scanf("%s", judulBaru);
34              // Kembali memeriksa apakah judul baru sudah ada
35              i = -1; // Mengulang pemeriksaan dari awal
36          }
37      }
38
```

Program memastikan bahwa setiap judul catatan harus unik. Jika judul yang dimasukkan sudah ada, pengguna diminta untuk menggantinya.

4. Penggunaan Struct:

```
6
7  struct Catatan {
8      char judul[50];
9      char isi[200];
10 };
11
```

Struktur data Catatan digunakan untuk mengelompokkan informasi terkait judul dan isi catatan menjadi satu unit logis.

5. Looping Menu Utama:

```
154
155 while (1) {
156     printf("\n==== NOTE FOR LUTPI =====\n");
157     printf("\nMENU:");
158     printf("\n1. Tambah Catatan");
159     printf("\n2. Tampilkan Semua Catatan");
160     printf("\n3. Tampilkan Satu Catatan");
161     printf("\n4. Ganti Catatan");
162     printf("\n5. Hapus Catatan");
163     printf("\n6. Keluar\n");
164     printf("\nPilih menu (1-6): ");
165     scanf("%d", &pilihan);
166
167     switch (pilihan) {
168         case 1:
169             tambahCatatan();
170             break;
171         case 2:
172             tampilkanSemuaCatatan();
173             break;
174         case 3:
175             tampilkanSatuCatatan();
176             break;
177         case 4:
178             gantiCatatan();
179             break;
180         case 5:
181             hapusCatatan();
182             break;
183         case 6:
184             printf("\nProgram dihentikan. Dadah Lutpi!");
185             exit(0);
186         default:
187             printf("\nPerintah tidak valid. Silakan masukkan perintah yang benar.\n");
188     }
189 }
```

Program dirancang agar berjalan dalam loop sehingga pengguna dapat terus memilih dan menjalankan operasi hingga memutuskan untuk keluar.

6. Penggunaan Switch:

```
166
167 switch (pilihan) {
168     case 1:
169         tambahCatatan();
170         break;
171     case 2:
172         tampilkanSemuaCatatan();
173         break;
174     case 3:
175         tampilkanSatuCatatan();
176         break;
177     case 4:
178         gantiCatatan();
179         break;
180     case 5:
181         hapusCatatan();
182         break;
183     case 6:
184         printf("\nProgram dihentikan. Dadah Lutpi!");
185         exit(0);
186     default:
187         printf("\nPerintah tidak valid. Silakan masukkan perintah yang benar.\n");
188 }
189 }
```

Switch digunakan untuk menentukan tindakan yang diambil oleh program berdasarkan pilihan menu yang dimasukkan oleh pengguna.

7. Penanganan Kesalahan Input:

```

185         exit(0);
186     default:
187         printf("\nPerintah tidak valid. Silakan masukkan perintah yang benar.\n");
188     }

```

Program memberikan pesan jika pengguna memasukkan pilihan menu yang tidak valid, meningkatkan kegunaan program dan memberikan pengalaman pengguna yang lebih baik.

NO	MATERI	PERAN
1.	Variabel, Konstanta, dan Tipe Data	<p><i>int, char</i></p> <p>Variabel ini digunakan untuk menyimpan nilai-nilai seperti pilihan menu dan judul catatan.</p> <p><i>struct Catatan</i></p> <p>Digunakan untuk membuat tipe data baru yang menyimpan informasi judul dan isi catatan.</p>
2.	Array	<p><i>catatan[MAX_CATATAN]</i></p> <p>Array ini berperan sebagai wadah untuk menyimpan catatan-catatan yang dibuat oleh pengguna. Ukuran maksimum array didefinisikan oleh konstanta MAX_CATATAN.</p>
3.	Function	<p><i>tambahCatatan()</i> <i>tampilkanSemuaCatatan()</i> <i>tampilkanSatuCatatan()</i> <i>gantiCatatan()</i> <i>hapusCatatan()</i></p> <p>Fungsi-fungsi ini memecah tugas-tugas dalam program menjadi bagian-bagian yang lebih kecil, meningkatkan keterbacaan kode.</p>
4.	Conditional dan Looping	<p><i>if-else</i> <i>switch-case</i> <i>while</i></p> <p>Penggunaan <i>conditional</i> dan <i>looping</i> ini membuat alur kerja yang terorganisir yang memungkinkan pengguna untuk terus memilih opsi menu tanpa harus menjalankan program lagi dari awal. Ini memberikan kemampuan untuk membuat keputusan (misalnya, menambahkan catatan atau tidak) atau melakukan repetisi/perulangan (seperti loop untuk menampilkan semua catatan).</p>

5.	Input dan Output	<i>printf</i> <i>scanf</i> Digunakan untuk berinteraksi dengan pengguna, mencetak pesan ke layar, dan membaca input dari pengguna.
6.	Pointer	<i>struct Catatan *ptrCatatan</i> Penggunaan pointer memungkinkan program untuk menunjuk pada catatan tertentu saat mencari atau mengganti catatan, memanfaatkan konsep pointer untuk mengakses dan memanipulasi data dengan lebih efisien.