

LAPORAN TUGAS BESAR
PRAKTIKUM ALGORITMA DAN PEMROGRAMAN
D' AUTO PARTS

*Laporan ini disusun untuk memenuhi syarat mata kuliah Praktikum Algoritma dan
Pemrograman Program Studi S1 Teknik Telekomunikasi Universitas Telkom*



Disusun oleh :
KELOMPOK 2 KELAS TT-45-05

Anggota :

1. Muhammad Rafi Ediananta (1101210178)
2. Raden Roro Nisrina Hasna Firdaus Irawati (1101213084)
3. Salwa Aulia Farida Yudistira (1101213236)
4. Mauritius Sahala Syafaat Manurung (1101213335)

FAKULTAS TEKNIK ELEKTRO
TELKOM UNIVERSITY
BANDUNG
2021

DAFTAR ISI

DAFTAR ISI	i
DAFTAR GAMBAR	ii
BAB I	1
PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Batasan Masalah.....	2
1.3 Rumusan Masalah	2
1.4 Tujuan dan Manfaat	2
1.5 Deskripsi Pembagian Tugas Kelompok	3
BAB II.....	4
KAJIAN PUSTAKA	4
2.1 Algoritma	4
2.2 Tipe Data dan Komponen	5
2.3 Percabangan	8
2.4 Perulangan.....	11
2.5 Fungsi.....	15
2.6 Searching.....	19
2.7 File Sekuensial	22
2.8 Algoritma Rekursif.....	34
BAB III.....	36
HASIL DAN ANALISIS	36
3.1 Deskripsi program	36
3.2 Screenshot Fitur – Fitur Program	37
3.3 Flowchart.....	45
3.4 Analisis Program.....	52
BAB IV	53
PENUTUP.....	53
4.1 Kesimpulan	53
4.2 Saran.....	53
LAMPIRAN.....	54

DAFTAR GAMBAR

Gambar 1. 1 Standar Flowchart.....	5
Gambar 1. 2 Struktur Bahasa C	5
Gambar 1. 3 Tipe Data	7
Gambar 1. 4 Operator Boolean.....	8
Gambar 1. 5 If dengan satu aksi	9
Gambar 1. 6 If dengan dua aksi.....	9
Gambar 1. 7 Case Variable of	11
Gambar 1. 8 Contoh While Loop	12
Gambar 1. 9 Contoh Do while Loop	12
Gambar 1. 10 Contoh program For Loop.....	13
Gambar 1. 11 Contoh program perulangan dengan 2 tipe data	13
Gambar 1. 12 Contoh Post & Pre Increment	14
Gambar 1. 13 Contoh penggunaan Break & Continue	15
Gambar 1. 14 Struktur Fungsi	16
Gambar 1. 15 Cara pemanggilan Fungsi	16
Gambar 1. 16 Penulisan fungsi setelah main program	17
Gambar 1. 17 Penulisan fungsi sebelum main program	17
Gambar 1. 18 Fungsi dengan parameter reference	18
Gambar 1. 19 parameter pass by value dan pass by reference	19
Gambar 1. 20 Pointer.....	19
Gambar 1. 21 Sequential Search.....	20
Gambar 1. 22 Binary Search	21
Gambar 1. 23 Pendeklarasian File.....	22
Gambar 1. 24 Bentuk dasar membuka file	22
Gambar 1. 25 Mode fopen.....	23
Gambar 1. 26 Contoh Penggunaan fungsi fopen.....	23
Gambar 1. 27 Bentuk dasar untuk membaca file.....	24

Gambar 1. 28 Menulis dan membaca data string	25
Gambar 1. 29 Bentuk dasar fget	25
Gambar 1. 30 Membaca data string dari file teks menggunakan fgetc	26
Gambar 1. 31 Menambahkan data string ke dalam file teks	26
Gambar 1. 32 Bentuk dasar menulis dan membaca file biner	27
Gambar 1. 33 Contoh program menulis & membaca file biner	28
Gambar 1. 34 Menambahkan data pada file biner.....	29
Gambar 1. 35 Bentuk dasar fseek	29
Gambar 1. 36 Referensi Whence	30
Gambar 1. 37 Rewind function	30
Gambar 1. 38 Mengakses file biner secara acak	31
Gambar 1. 39 Bentuk dasar fungsi remove	31
Gambar 1. 40 Bentuk dasar fungsi rename.....	32
Gambar 1. 41 Menghapus data dalam file biner.....	33
Gambar 1. 42 Bentuk dasar fclose.....	33
Gambar 1. 43 Bentuk dasar mengosongkan buffer	33
Gambar 1. 44 Menulis dan menempatkan indikator posisi file.....	34
Gambar 1. 45 Contoh Algoritma Rekursif	35
Gambar 1. 46 Penggunaan Algoritma Rekursif.....	35
Gambar 2. 1 Bagian Dealer	37
Gambar 2. 2 Melihat dan Menghapus data customer	38
Gambar 2. 3 Menambah data Barang	39
Gambar 2. 4 Menghapus Data Barang	40
Gambar 2. 5 Role Customer	41
Gambar 2. 6 List Barang.....	42
Gambar 2. 7 Cari barang sesuai kategori.....	43
Gambar 2. 8 Checkout atau Pemesanan	44
Gambar 2. 9 Flowchart	51

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan dan kemajuan zaman di era globalisasi sekarang ini, mengantarkan kehidupan ke arah yang lebih modern dan serba mudah serta manusia dituntut mengikuti perkembangan ilmu pengetahuan dan teknologi. Segala aspek kehidupan saat ini sangat erat kaitannya dengan penerapan konsep dari ilmu pengetahuan dan teknologi. Dalam keberlangsungan kehidupan, teknologi digunakan untuk menyederhanakan hal yang kompleks. Untuk menyederhanakan hal tersebut juga membutuhkan proses yang lebih kompleks dari masalah yang disederhanakan. Salah satunya ialah ilmu pemrograman yang diterapkan untuk menyederhanakan hal yang kompleks.

Ilmu pemrograman bertujuan untuk menjalankan serangkaian kode perintah program yang akan dijalankan. Semakin banyak perangkat lunak (software) dan semakin meningkatnya kecanggihan perangkat keras (hardware). Sekarang ini, perangkat komputer berguna sebagai alat bantu yang berperan dalam menyelesaikan suatu pekerjaan. Tersedianya bahasa-bahasa pemrograman komputer seperti C, C++, Pascal, Basic, dan lain-lain telah mendorong para ahli komputer untuk merancang sebuah program yang dapat dipergunakan sebagai alat bantu dalam menyelesaikan sebuah pekerjaan dalam bidang tertentu, seperti yang akan kami buat dalam laporan ini yaitu *D' Auto Parts* yang merupakan sebuah program yang digunakan untuk jual beli suku cadang kendaraan bermotor secara online.

Dalam rangka menyelesaikan proses perkuliahan untuk semester yang kedua, mahasiswa Teknik Telekomunikasi mengikuti mata kuliah Praktikum Dasar Komputer Algoritma dan Pemrograman, dengan tujuan untuk mengembangkan kemampuan *programming*. Bahasa yang dilakukan dalam mata kuliah ini adalah C dan terdapat 11 modul yang harus dikuasai oleh mahasiswa. Untuk menuntaskan mata kuliah ini mahasiswa diwajibkan untuk menyelesaikan sebuah tugas akhir yang mengimplementasikan fungsi dari setiap modul yang telah diselesaikan. Tugas akhir ini adalah membuat program sesuai dengan yang telah diperintahkan.

1.2 Batasan Masalah

- Dalam membuat program *D'Auto Parts* kami menggunakan bahasa C.
- Aplikasi yang kami gunakan untuk memprogram adalah Dev C.
- *D' Auto Parts* merupakan sebuah program yang digunakan untuk jual beli suku cadang kendaraan bermotor secara online.
- Pada program ini terdapat 2 akun yaitu dealer dan customer.

1.3 Rumusan Masalah

Berdasarkan uraian judul di atas, maka permasalahan yang akan dikaji adalah sebagai berikut :

- a. Apa saja materi yang diimplementasikan dalam program *D' Auto Parts*?
- b. Bagaimana deskripsi program *D' Auto Parts*?
- c. Bagaimana cara kerja program *D' Auto Parts*?
- d. Bagaimana hasil dari program *D' Auto Parts* saat dijalankan?
- e. Bagaimana flowchart dari program *D' Auto Parts*?

1.4 Tujuan dan Manfaat

Sesuai dengan masalah yang diangkat, tujuan dari tugas besar ini adalah :

- a. Mengetahui materi apa saja yang diimplementasikan dalam program *D' Auto Parts*.
- b. Mengetahui deskripsi program *D' Auto Parts*.
- c. Mengetahui cara kerja program *D' Auto Parts*.
- d. Mengetahui hasil dari program *D' Auto Parts* saat dijalankan.
- e. Mengetahui flowchart dari program *D' Auto Parts*.

Manfaat dan keutamaan dari tugas besar ini antara lain sebagai berikut :

1. Manfaat Secara Teoritis

Penulisan secara teoritis diharapkan dapat menambah wawasan dan pemahaman teori terhadap materi yang terdapat dalam Modul Praktikum Alpro Semester Genap 2021-2022 sebagaimana diterapkan pengaplikasiannya pada program yang kami kerjakan berjudul *D' Auto Parts*.

2. Manfaat Secara Praktis

Dengan adanya laporan tugas besar dan program *D' Auto Parts* ini diharapkan dapat meningkatkan kemampuan kita memahami materi yang telah diberikan selama kegiatan Praktikum Algoritma dan Pemrograman dan mengaplikasikannya dalam kehidupan sehari-hari. Diharapkan kedepannya kami juga dapat mengembangkan program *D' Auto Parts* supaya dapat membantu pelaku usaha jual beli suku cadang kendaraan bermotor.

1.5 Deskripsi Pembagian Tugas Kelompok

Muhammad Rafi Ediananta	Raden Roro Nisrina Hasna	Salwa Aulia Farida Yudistira	Mauritius M.
Mengerjakan codingan bagian fitur login register dealer customer dan fitur hapus dan melihat customer, mengerjakan badan program. Menyusun laporan tugas besar.	Mengerjakan flowchart.	Mengerjakan codingan bagian fitur browse barang, menambah dan menghapus barang, cari barang dalam kategori, dan checkout barang. Menyusun laporan tugas besar.	Menyusun laporan tugas besar.

BAB II

KAJIAN PUSTAKA









2.1 Algoritma

Algoritma adalah serangkaian langkah yang tepat, terperinci, dan terbatas dalam menyelesaikan masalah. Tepat artinya langkah tersebut selalu benar untuk menyelesaikan masalah. Terperinci artinya langkah tersebut detail dan dapat dieksekusi oleh komputer. Terbatas artinya suatu saat langkah tersebut harus berhenti. Jika langkah “tidak pernah” berhenti namanya looping forever.

Dalam menyelesaikan tugas besar ini kami menggunakan algoritma diagram alur atau flowchart. Flowchart adalah suatu standar untuk menggambarkan suatu proses. Setiap langkah dalam algoritma dinyatakan dengan sebuah simbol dan aliran setiap langkah (dari suatu langkah ke langkah lain) dinyatakan dengan garis yang dilengkapi panah. Berikut ini beberapa petunjuk yang harus diperhatikan dalam pembuatan flowchart :

1. Flowchart digambarkan dari halaman atas ke bawah dan dari kiri ke kanan.
2. Aktivitas yang digambarkan harus didefinisikan secara hati-hati dan definisi ini harus dapat dimengerti oleh pembacanya.
3. Kapan aktivitas dimulai dan berakhir harus ditentukan secara jelas.
4. Setiap langkah dari aktivitas harus diuraikan dengan menggunakan deskripsi kata kerja, misalkan menghitung pajak penjualan.
5. Setiap langkah dari aktivitas harus berada pada urutan yang benar.
6. Lingkup dan range dari aktivitas yang sedang digambarkan harus ditelusuri dengan hati-hati. Percabangan-percabangan yang memotong aktivitas yang sedang digambarkan tidak perlu digambarkan pada flowchart yang sama. Simbol konektor harus digunakan dan percabangannya diletakan pada halaman yang terpisah atau hilangkan seluruhnya bila percabangannya tidak berkaitan dengan sistem.
7. Gunakan simbol-simbol flowchart yang standar.

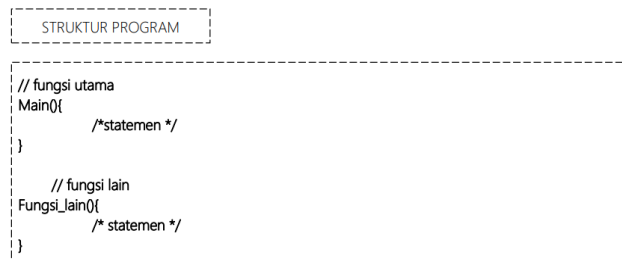
Berikut adalah simbol standar untuk flowchart.

Bentuk	Nama Simbol	Deskripsi
	Flow Line	Menyatakan jalannya arus suatu proses .
	On-Page Connector	Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama .
	Off-Page Connector	Menyatakan sambungan dari proses ke proses lainnya di halaman yang berbeda .
	Process	Menyatakan suatu tindakan (proses) yang dilakukan oleh komputer .
	Decision	Menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan, yaitu YA atau TIDAK.
	Pre-defined Process	Menyatakan prosedur lain yang telah dibuat dalam diagram.
	Terminator	Menyatakan suatu permulaan atau pengakhiran dari suatu program .
	Input / Output	Menyatakan proses <i>input</i> atau <i>output</i> tanpa tergantung jenis peralatannya .

Gambar 1. 1 Standar Flowchart

2.2 Tipe Data dan Komponen

1. Struktur Bahasa Pemrograman C Dalam bahasa C, struktur programnya terdiri dari fungsi-fungsi. Berikut merupakan struktur umum dari bahasa C.



Gambar 1. 2 Struktur Bahasa C

2. Variabel dan Konstanta

Variabel dan konstanta adalah suatu nama yang menyatakan tempat dalam memori komputer dan berisi sebuah nilai. Isi dari variabel dapat diubah sewaktu-waktu ketika program dieksekusi, sedangkan konstanta tidak dapat diubah ketika program dieksekusi. Hal-hal penting yang harus diperhatikan dalam penamaan variabel dan konstanta antara lain :

- harus diawali dengan huruf alfabet (A..Z, a..z) atau garis bawah (_);
- karakter selanjutnya bisa berupa huruf, digit atau karakter garis bawah (_);

- panjang maksimal identifier adalah 32 karakter, jika lebih maka yang dianggap adalah 32 karakter awal;
- tidak boleh menggunakan spasi;
- tidak boleh menggunakan operator aritmatika (+ - / * %);
- tidak boleh menggunakan karakter-karakter khusus seperti : , ; # @ (.) & dan \$;
- bahasa C bersifat case-sensitive, sehingga huruf kapital dan non-kapital dibedakan.

3. Tipe Data

Tipe data adalah jenis dari suatu variabel yang digunakan dalam algoritma. Berikut adalah beberapa macam tipe data.

- Int : Menyatakan bilangan bulat antara -32768 sampai +32767 Contoh : 1 0 3 -5 -7
- Float : Menyatakan bilangan real (bilangan asli) dari 10⁻³⁸ sampai dengan 10³⁸. Contoh : 1.1, 1.3, -9.2
- Char : Menyatakan sebuah karakter. Contoh : A, f, 9, atau *.
- Bool : Menyatakan nilai logika : true atau false.

Type	Bits	Minimal Range
char	8	-127 to 127
unsigned char	8	0 to 255
signed char	8	-127 to 127
int	16 or 32	-32,768 to 32,767
unsigned int	16 or 32	0 to 65,535
signed int	16 or 32	Same as int
short int	16	-32,768 to 32,767
unsigned short int	16	0 to 65,535
signed short int	16	Same as short int
long int	32	-2,147,483,648 to 2,147,483,647
long long int	64	$-(2^{63} - 1)$ to $2^{63} - 1$ (Added by C99)
signed long int	32	Same as long int
unsigned long int	32	0 to 4,294,967,295
unsigned long long int	64	$2^{64} - 1$ (Added by C99)
float	32	1E-37 to 1E+37 with six digits of precision
double	64	1E-37 to 1E+37 with ten digits of precision
long double	80	1E-37 to 1E+37 with ten digits of precision

Table 2-1. All Data Types Defined by the C Standard

Gambar 1. 3 Tipe Data

4. Tipe Data Bentukan

Tipe bentukan adalah tipe data yang dibuat sendiri oleh pengguna. Tipe ini dibuat karena ada relasi antar variabel yang bila digabungkan mempunyai suatu maksud yang sama. Untuk membuat tipe data bentukan, digunakan kata kunci “struct”. Salah satu contoh dari tipe data bentukan adalah mahasiswa, dimana mahasiswa mempunyai nama, nim, jurusan, dan nilai.

5. Penjelasan Tambahan

→ Berkas header (berekstensi .h) adalah berkas yang berisi prototype fungsi, definisi konstanta, dan definisi variabel. Prototype fungsi berupa judul suatu fungsi yang `#include int main () { printf (“Hello World \n”); return 0; }` Laboratorium Dasar Komputer Telkom University 10 dilengkapi dengan argumen dan tipe argumen serta tipe nilai balik untuk fungsi yang memberikan nilai balik.

- preprocessor `#include` adalah suatu perintah yang digunakan untuk mengatur compiler agar membaca berkas header yang disertakan di belakang kata `include` saat pelaksanaan kompilasi.
- `main()` adalah fungsi yang akan dijalankan pertama kali ketika program dieksekusi. Kata `int` di depan `main ()` menyatakan bahwa program memberikan nilai balik yang bertipe `int`.
- Nilai balik program ditentukan oleh pernyataan `return`. Pada program tersebut, `return 0` menyatakan bahwa nilai balik program adalah nol. Nilai nol biasa digunakan untuk menyatakan bahwa program berhasil melaksanakan tugas yang diembannya.
- Setiap pernyataan dalam bahasa C ditulis dengan diakhiri tanda titik-koma (;). Pada program di atas, ada dua buah pernyataan, yaitu : `printf ("Hello World \n");` dan `return 0;`
- `"\n"` akan menyebabkan kursor berpindah ke baris berikut setelah `Hello World` dioutputkan.

2.3 Percabangan

Operasi boolean adalah operasi logika yang hanya akan menghasilkan nilai `true` atau `false`. Operasi ini akan sering digunakan untuk menentukan aksi yang akan dilakukan oleh program (dalam percabangan maupun perulangan).

Operator boolean		Contoh	Keterangan
C	Algoritma		
<code>==</code>	<code>=</code>	<code>a == b</code>	Bernilai true jika a sama dengan b
<code><</code>	<code><</code>	<code>a < b</code>	Bernilai true jika nilai a kurang dari b
<code><=</code>	<code>≤</code>	<code>a <= b</code>	Bernilai true jika nilai a kurang dari / sama dengan b
<code>></code>	<code>></code>	<code>a > b</code>	Bernilai true jika nilai a lebih dari b
<code>>=</code>	<code>≥</code>	<code>a >= b</code>	Bernilai true jika nilai a lebih dari / sama dengan b
<code>!=</code>	<code>≠</code>	<code>a != b</code>	Bernilai true jika nilai a tidak sama dengan b

Gambar 1. 4 Operator Boolean

Percabangan :

Percabangan adalah metode pemilihan suatu aksi yang akan dijalankan oleh program.

Terdapat dua jenis percabangan, yaitu :

1. if [kondisi] -- [aksi]

else – [aksi]

Pada metode ini aksi akan dilakukan ketika kondisi bernilai true.

a. Satu Aksi

C	Algoritma
<pre>#include <stdio.h> int bil; void main() { printf("Angka : "); scanf("%d",&bil); if (bil%2==0) printf("Bilangan Genap"); else printf("Bilangan Ganjil"); }</pre>	<pre>Program if Satu Aksi Kamus : bil : integer Algoritma : Output('Masukkan nilai : ') Input(bil) if(bil mod 2 = 0) then Output('Bilangan genap') else output("Bilangan Ganjil")</pre>

Gambar 1. 5 If dengan satu aksi

Program di atas akan menampilkan “Bilangan Genap” jika nilai bil yang dimasukkan bernilai genap (bilangan genap berarti habis dibagi dua atau dapat ditulis $bil \% 2 = 0$)

b. Dua Aksi

C
<pre>#include <stdio.h> #include <string.h> char uname[50],pass[50]; int main() { printf("Username : "); gets(uname); printf("Password : "); gets(pass); if (strcmp(uname,"Daskom")==0 && strcmp(pass,"Kece")==0) { printf("Login Sukses \n"); printf("Silahkan Lanjutkan\n"); } else printf("Login Gagal\n"); return 0; }</pre>

Gambar 1. 6 If dengan dua aksi

Sama halnya dengan program sebelumnya, aksi dalam if akan dilakukan ketika kondisi bernilai true, sebaliknya bila kondisi bernilai false maka aksi pada else yang akan dilakukan.

Program di atas akan menampilkan output “Login Sukses” jika username dan password yang dimasukkan sesuai dengan yang ada di baris program di atas, yaitu “Daskom” dan “Kece”.

Note :

- Jika aksi (pada if atau else) yang akan dikerjakan lebih dari satu, maka tambahkan “{” (sebelum aksi pertama) dan “}” (setelah aksi terakhir).
- Anda tidak bisa menggunakan tanda “==” untuk membandingkan string. Perbandingan dilakukan menggunakan fungsi strcmp(). Fungsi strcmp() terdapat pada header #include. Sehingga perlu menyertakan header #include pada program menggunakan perintah #include di awal program agar fungsi strcmp() dapat terbaca.
- Pada fungsi strcmp(string1, string2) terdapat beberapa return value, diantaranya :
 - 0 : jika string1 sama dengan string2
 - <0 : jika string1 lebih kecil dari string2
 - >0 : jika string1 lebih besar dari string2

2. case [variable] of

C	Algoritma
<pre>#include <stdio.h> int i; void main () { printf("Masukkan urutan hari :"); scanf("%d",&i); switch (i) { case 1:printf("Senin");break; case 2:printf("Selasa");break; case 3:printf("Rabu");break; case 4:printf("Kamis");break; case 5:printf("Jumat");break; case 6:printf("Sabtu");break; case 7:printf("Minggu");break; default : printf("Angka di luar urutan hari");break; } }</pre>	<p>Program contoh Case Of</p> <p>Kamus :</p> <p>i : integer</p> <p>Algoritma :</p> <p>Output("Masukkan urutan hari : '")</p> <p>Input(i)</p> <p>Depend on (i)</p> <p>i = 1 : output("Senin"); i = 2 : output("Selasa"); i = 3 : output("Rabu"); i = 4 : output("Kamis"); i = 5 : output("Jumat"); i = 6 : output("Sabtu"); i = 7 : output("Minggu"); else output("Angka di luar urutan hari");</p>

Gambar 1. 7 Case Variable of

Dari contoh di atas aksi yang akan dilakukan tergantung dengan nilai yang terdapat pada variabel i, ketika i bernilai 1 maka program akan menampilkan kata “senin” begitupun selanjutnya sampai angka 7. Default berfungsi jika case 1 - 7 tidak terpenuhi, maka perintah default akan dijalankan. Break berfungsi untuk menghentikan aksi yang dilakukan sampai pada case tersebut sehingga aksi pada case selanjutnya tidak diproses.

2.4 Perulangan

Perulangan dalam pemrograman digunakan untuk mengulangi sebuah blok kode tertentu sampai memenuhi kondisi yang diinginkan. Berikut adalah jenis-jenis perulangan dalam bahasa pemrograman C:

1. While loop

Pada metode ini, aksi akan dilakukan selama kondisi bernilai true dan akan berhenti jika kondisi menjadi false. Pada perulangan ini jumlah perulangan belum diketahui secara pasti.

Deklarasi	C	Algoritma	Output
<pre>while(kondisi) { // aksi // aksi // aksi }</pre>	<pre>#include <stdio.h> int i; void main() { i=1; while (i<=10) { printf("%d \n",i); i=i+1; } }</pre>	Program contoh	1
		While	2
		Kamus :	3
		i : integer	4
		Algoritma :	5
		i ← 1	6
		While (i ≤ 10) do	7
		Output(i)	8
		i ← i + 1	9
			10

Gambar 1. 8 Contoh While Loop

Pada contoh di atas terdapat aksi “i = i + 1” yang akan mengubah output kondisi “i <= 10” menjadi false (pada saat i > 10) sehingga perulangan akan berhenti. Artinya “i wajib bernilai <= 10” agar true selain itu false.

2. Do while loop

Pada metode perulangan ini aksi akan dilakukan minimal satu kali, hal ini disebabkan adanya pengecekan kondisi setelah aksi dilakukan untuk pertama kali. Aksi akan terus dilakukan selama kondisi bernilai true dan berhenti ketika kondisi bernilai false. Sehingga, meskipun kondisi bernilai false, program akan tetap melakukan aksi sebanyak satu kali. Pada perulangan ini jumlah perulangan belum diketahui secara pasti.

Deklarasi	C	Algoritma	Output
<pre>do{ // aksi // aksi // aksi }while(kondisi)</pre>	<pre>#include <stdio.h> int i; void main (){ i=1; do{ printf("%d \n",i); i++; }while (i<10); }</pre>	Program contoh Do	1
		While	2
		Kamus :	3
		i : integer	4
		Algoritma :	5
		i ← 1	6
		repeat	7
		Output(i)	8
		i ← i + 1	9
		until (i > 10)	

Gambar 1. 9 Contoh Do while Loop

Catatan:

Pernyataan i++ identik dengan i=i+1;

3. For loop

Pada perulangan ini jumlah perulangan telah diketahui dengan pasti (Counted Loop). Perulangan ini digunakan jika kita sudah mengetahui berapa perulangan yang akan dilakukan.

Deklarasi	C	Algoritma	Output
<pre>for(inisialisasi; kondisi; update statement){ // aksi // aksi // aksi }</pre>	<pre>#include <stdio.h> int i; void main(){ for (i=1; i<8; i++) printf("%d \n",i); }</pre>	Program contoh For Loop	1
		Kamus :	2
		i : integer	3
			4
		Algoritma :	5
		i traversal [1..7]	6
		Output(i)	7

Gambar 1. 10 Contoh program For Loop

Catatan:

Jika ingin mencacah turun (misal 10 → 1) maka gunakan for(i=10; i>=1; i--)

Contoh program untuk memasukkan beberapa data menggunakan perulangan :

C	Keterangan
<pre>#include <stdio.h> #include <stdlib.h> int nilai,n,i; char nama[100]; void main() { system("cls"); printf("Banyak Data : ");scanf("%d",&n); for (i=1;i<=n;i++) { printf("Nilai"); scanf("%d",&nilai); getchar(); printf("Nama : "); gets(nama); } }</pre>	<p>scanf() memiliki kelemahan yaitu meninggalkan "\n" (baris baru) pada memori <i>input</i>, sehingga jika terdapat gets() setelah scanf(), gets() tersebut tidak akan bekerja. Solusinya adalah menambahkan getchar() setelah scanf() yang berfungsi untuk mengambil "\n" pada memori <i>input</i> sehingga gets() dapat bekerja.</p>
<p>system("cls"); berfungsi untuk membersihkan layar program. system("cls") terdapat pada library stdlib.h</p>	<p>Fungsi getchar() terdapat pada library stdio.h</p>

Gambar 1. 11 Contoh program perulangan dengan 2 tipe data

Pre-Increment dan Post-Increment

Kedua istilah ini merupakan cara untuk menambahkan nilai pada suatu variabel.

- post-increment mengembalikan nilai ke nilai yang belum mengalami operasi penambahan (nilai dioutputkan terlebih dahulu, kemudian ditambah).

→ pre-increment mengembalikan nilai ke nilai yang telah mengalami operasi penambahan (nilai ditambahkan terlebih dahulu, kemudian dioutputkan).

Perbedaan dari keduanya akan terlihat berbeda ketika menggunakan perintah output seperti contoh dibawah :

Contoh Program Increment	Output
<pre>#include<stdio.h> void main() { int i=0; // Contoh Post-Increment printf("\ni = %d",i++); // Contoh Pre-Increment printf("\ni = %d",++i); }</pre>	<pre>i = 0 i = 2</pre>

- Keterangan:

Pada perintah printf pertama menghasilkan nilai i=0 karena menggunakan post-increment (nilai dioutputkan dahulu, kemudian ditambahkan), lalu nilai i = 0 berubah menjadi i = 1.

Pada perintah printf kedua menghasilkan nilai i=1 karena menggunakan pre-increment (nilai ditambahkan dahulu, kemudian dioutputkan), lalu nilai i = 1 berubah menjadi i = 2.

Gambar 1. 12 Contoh Post & Pre Increment

Catatan:

Sebaliknya jika digunakan untuk mengurangi nilai pada suatu variabel maka disebut decrement, contoh: i-- atau --i.

Perintah Kontrol

Perintah kontrol digunakan pada perulangan untuk mengubah hasil eksekusi perulangan dari keadaan normalnya. Dibawah ini merupakan perintah-perintah kontrol pada perulangan :

Perintah	Contoh Program	Output	Keterangan
Perintah Break	<pre>#include<stdio.h> int i; void main(){ for(i=1;i<10;i++){ if(i==7) break; printf("%d\n",i); } }</pre>	1 2 3 4 5 6	Perintah break berfungsi untuk menghentikan proses perulangan yang sedang berlangsung. Pada contoh di samping perulangan akan berhenti saat i = 7.
Perintah Continue	<pre>#include<stdio.h> int i; void main(){ for(i=1;i<10;i++){ if(i==7) continue; printf("%d\n",i); } }</pre>	1 2 3 4 5 6 8 9	Perintah continue berfungsi untuk melewati/skip perulangan yang dimaksud. Pada contoh disamping program akan melanjutkan perulangan saat i = 7.

Gambar 1. 13 Contoh penggunaan Break & Continue

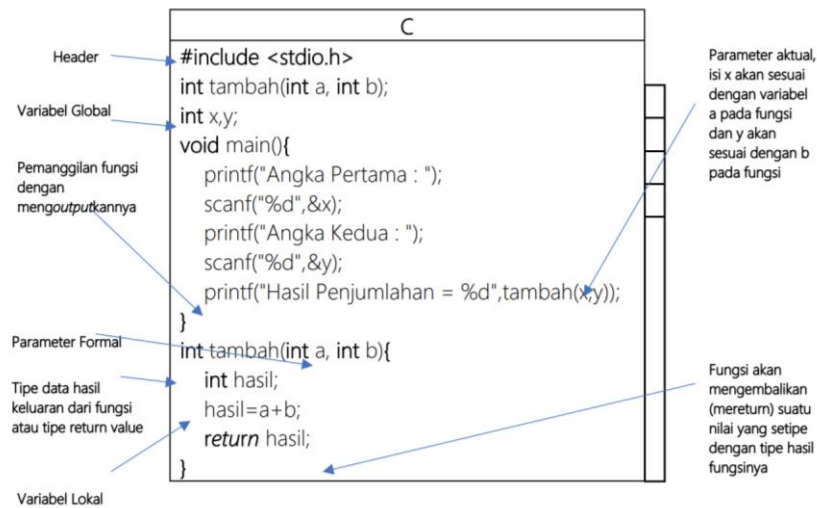
2.5 Fungsi

Fungsi merupakan subprogram yang sangat bermanfaat dalam pemrograman, terutama untuk program besar. Misalnya, kita memasukkan fungsi rumus luas prisma segiempat. Maka kita hanya perlu menuliskan rumus luas prisma satu kali pada fungsi tanpa harus menuliskan berulang kali pada program utama.

Berikut istilah-istilah yang sering digunakan dalam fungsi :

- Variabel lokal : variabel yang hanya dapat digunakan untuk algoritma didalam fungsi tersebut saja
- Variabel global : variabel yang dapat digunakan baik dalam program utama maupun pada fungsi
- Parameter aktual : parameter yang disertakan saat pemanggilan fungsi
- Parameter formal : parameter yang diletakkan pada header fungsi

Struktur Fungsi



Gambar 1. 14 Struktur Fungsi

Fungsi (function)

Fungsi adalah sebuah potongan kode yang ditujukan untuk menjalankan tugas yang spesifik, tugas tersebut membutuhkan informasi input yang disebut parameter dan biasanya suatu fungsi akan mengembalikan sebuah nilai (return value). Dalam penulisan fungsi diberikan nama/identifier, parameter formal (nilai masukan yang diberi juga nama dan dijelaskan tipenya), dan dideklarasikan tipe dari fungsi tersebut. Fungsi dapat dibuat sebelum atau sesudah penulisan program utama. Bila dibuat sesudah penulisan main, maka harus menuliskan prototype fungsi di bagian atas program.

Cara Pemanggilan Fungsi	C	Algoritma
<i>Assignment</i>	<code>a = tambah(x,y)</code>	<code>a ← tambah(x,y)</code>
<i>Output (printf)</i>	<code>printf("%d",tambah(x,y))</code>	<code>Output(tambah(x,y))</code>

Gambar 1. 15 Cara pemanggilan Fungsi

a. Penulisan fungsi setelah main program

C	Algoritma
#include <stdio.h>	Program penjumlahan
int tambah(int, int); /* Prototype Fungsi */	Kamus :
void main(){	x, y : integer
int x,y;	Algoritma :
printf("Angka Pertama : "); scanf("%d",&x);	Output('Angka pertama : ')
printf("Angka Kedua : "); scanf("%d",&y);	Input(x)
printf("Hasil Penjumlahan = %d",tambah(x,y));	Output('Angka kedua : ')
}	Input(y)
int tambah(int a, int b){	function tambah (a,b :
int c;	integer) → integer
c=a+b;	Kamus :
return c;	c : integer
}	Algoritma :
	c → a + b

Gambar 1. 16 Penulisan fungsi setelah main program

Cara penulisan seperti ini membutuhkan sebuah prototype fungsi, tujuannya adalah sebagai abstrak supaya fungsi dapat dikenali oleh seluruh program. Prototype fungsi hanya membutuhkan tipe data dari parameternya saja, jadi kita tidak harus menuliskan variabelnya.

b. Penulisan fungsi sebelum main program

C	Algoritma
#include <stdio.h>	Program penjumlahan
int tambah(int a, int b){	Kamus :
int c;	x, y : integer
c=a+b;	Algoritma :
return c;	Output('Angka pertama : ')
}	Input(x)
void main(){	Output('Angka kedua : ')
int x,y;	Input(y)
printf("Angka Pertama : ");	Output('Hasil Penjumlahan=',
scanf("%d",&x);	tambah(x,y))
printf("Angka Kedua : ");	function tambah (a,b :
scanf("%d",&y);	integer) → integer
printf("Hasil Penjumlahan = %d",tambah(x,y));	Kamus :
}	c : integer
	Algoritma :
	c ← a + b
	→ c

Gambar 1. 17 Penulisan fungsi sebelum main program

Fungsi Void/Prosedur (Void Function)

Fungsi void adalah sebuah fungsi yang tidak mengembalikan nilai atau tidak memiliki return value. Fungsi ini banyak digunakan pada 2 kasus berikut:

1. Menampilkan informasi pada layar agar dapat dibaca oleh user.
2. Fungsi dengan parameter reference, yaitu dengan mengirimkan alamat memori suatu variable kedalam parameter.

C	Output
<pre>#include <stdio.h> void bilangan(int a){ if(a>0) printf("Positive"); else if(a<0) printf("Negative"); else printf("Zero"); } void main(){ int x; printf("Angka: "); scanf("%d",&x); bilangan(x); }</pre>	<p>Contoh 1: Angka: 10 Positive</p> <p>Contoh 2: Angka: -22 Negative</p> <p>Contoh 3: Angka: 0 Zero</p>

Gambar 1. 18 Fungsi dengan parameter reference

Parameter

Parameter adalah data yang dimasukkan ke dalam sebuah fungsi yang kemudian akan digunakan pada proses yang ada di dalam fungsi tersebut. Ada 2 cara untuk memasukkan parameter pada bahasa C:

- Pass by value, artinya data yang dilewati hanya nilai dari suatu variabel kemudian disimpan pada variabel lokal pada fungsi. Dengan kata lain cara ini menggunakan 2 variabel yang berbeda, sehingga variabel pada parameter aktual tidak bisa diubah dalam fungsi
- Pass by reference, artinya data yang dilewati adalah alamat memori dari suatu variabel. Dengan kata lain cara ini menggunakan variabel yang sama, sehingga mengubah nilai variabel dalam fungsi dapat mengubah nilai variabel aslinya.

Berikut contoh parameter pass by value dan parameter pass by reference :

Parameter <i>pass by value</i>	Parameter <i>pass by reference</i>
<pre>#include <stdio.h> void tukar (int , int); void main(){ int a,b; a=27; b=7; tukar(a, b); printf("Hasil Penukaran : \n"); printf("a = %d \n",a); printf("b = %d",b); } void tukar (int x, int y){ int temp; temp=x; x=y; y=temp; }</pre>	<pre>#include <stdio.h> void tukar (int* , int*); void main(){ int a,b; a=27; b=7; tukar(&a, &b); // Pass by reference printf("Hasil Penukaran : \n"); printf("a = %d \n",a); printf("b = %d",b); } void tukar (int*x, int*y){ int temp; temp=*x; *x=*y; *y=temp; }</pre>
<p>Output : a=27 , b=7 Nilai variabel tidak tertukar karena variabel pada parameter aktual dan variabel pada fungsi adalah kedua variabel yang berbeda.</p>	<p>Output : a=7 , b=27 Cara untuk membuat parameter <i>pass by reference</i> adalah dengan menggunakan <i>asterisk</i> (*). Sedangkan cara pemanggilan fungsinya dengan menggunakan <i>ampersand</i> (&) di depan variabel <i>reference</i>-nya.</p>

Gambar 1. 19 parameter pass by calue dan pass by reference

Pointer

Pointer adalah fitur pada C yang digunakan untuk mengakses dan manipulasi alamat memori suatu variabel. Pointer memiliki 2 operator yaitu:

1. Dereference operator (*), untuk memberikan nilai (value) dari suatu variabel.
2. Reference operator (&), untuk memberikan alamat memori dari suatu variabel.

<pre>#include <stdio.h> int main(){ int var = 5; printf("Value: %d\n", var); printf("Address: %u", &var); }</pre>	<p>Output: Value: 5 Address: 2358860</p> <p><i>/* Nilai alamat memori dapat berbeda-beda setiap saat */</i></p>
---	--

Gambar 1. 20 Pointer

2.6 Searching

1. Sequential search

Sequential search adalah metode pencarian yang paling sederhana, dimana pencarian dilakukan secara terurut dari data pertama sampai data yang dicari ditemukan. Data dibandingkan satu per satu dari awal sampai data ditemukan atau tidak ditemukan.

Misalkan terdapat sebuah data pada array A: 5 3 1 8 4, dan yang dicari adalah angka 1, maka program akan mengecek data di setiap indeks satu persatu dan data akan ditemukan pada indeks kedua atau pada urutan ketiga.

- jika data A[mid] lebih besar dari data yang dicari maka akan dilakukan penyempitan pada area pencarian (nilai high menjadi mid - 1)
- jika data A[mid] lebih kecil dari data yang dicari maka akan dilakukan penyempitan pada area pencarian (nilai low menjadi mid + 1)

C	Algoritma
<pre>#include<stdio.h> int a[100]; int i,n,cari; main () { printf("Banyak data : "); scanf("%d",&n); for (i=0;i<=n-1;i++) { printf("Data ke-%d :",i+1); scanf("%d",&a[i]); } printf("Data yang ingin dicari : "); scanf("%d",&cari); //proses sequential search i=0; while ((i<=n-1) && (a[i]!=cari)) i=i+1; if (a[i]==cari) printf("Data ditemukan pada urutan : %d",i+1); else printf("Data tidak ditemukan"); }</pre>	<p>Program <i>Sequential Search</i></p> <p>Kamus :</p> <p>a : array [1..100] of integer i, n, cari : integer</p> <p>Algoritma :</p> <p>Output('Masukkan banyak data : ')</p> <p>Input(n)</p> <p>i traversal [1..n]</p> <p>Output('Masukkan data ke-',i,' : ');</p> <p>Input(a_i)</p> <p>Output('Data yang ingin dicari : ')</p> <p>Input(cari)</p> <p>i ← 1</p> <p>while (i ≤ n) and (a_i ≠ cari) do</p> <p>i ← i + 1;</p> <p>if(a_i = cari) then</p> <p> Output('Data ditemukan pada urutan : ',i)</p> <p>else</p> <p> Output('Data tidak ditemukan')</p>

Gambar 1. 21 Sequential Search

2. Binary Search

Metode ini hanya dapat dilakukan pada kumpulan data yang sudah diurutkan terlebih dahulu. Jika terdapat N buah data pada array A, data yang akan dicari akan dibandingkan dengan data A[mid], dimana mid adalah indeks tengah dari array A (low = 0, high = N-1, mid = (high + low) div 2 atau (0+N-1) div 2). Anggap data terurut secara ascending:

- jika data A[mid] lebih besar dari data yang dicari maka akan dilakukan penyempitan pada area pencarian (nilai high menjadi mid - 1)

- jika data A[mid] lebih kecil dari data yang dicari maka akan dilakukan penyempitan pada area pencarian (nilai low menjadi mid + 1)

C	Algoritma
<pre>#include <stdio.h> int a[100]; int i,n,cari,idx,low,high, mid; main () { printf("Banyak data : "); scanf("%d",&n); for (i=0;i<n;i++) { printf("Data ke-%d : ",i+1); scanf("%d",&a[i]); } printf("Data yang dicari : "); scanf("%d",&cari); //data diurutkan terlebih dahulu //isi bagian ini dengan proses sorting //proses binary search low=0; high=n-1; idx=-1; while ((low<=high) && (idx== -1)) { mid=(low+high)/2; if (a[mid]==cari) idx=mid; else if (a[mid]<cari) low=mid+1; else high=mid-1; } if (idx!= -1) printf("Data ditemukan pada indeks : %d",idx); else printf("Data tidak ditemukan"); }</pre>	<pre>Program Binary Search Kamus : a : array [1..100] of integer i, n, cari, idx, low, high, mid : integer Algoritma : Output("Masukkan banyak data : ") Input(n) i traversal [1..n] Output("Masukkan data ke-'i' : "); Input(ai) Output("Data yang ingin dicari : ") Input(cari) { ----- } { -- proses sorting -- } { ----- } low ← 1 high ← n idx ← 0 while (low ≤ high) and (idx = 0) do mid ← (low + high) div 2 if(a_{mid} = cari) then idx ← mid else if(a_{mid} < cari) then low ← mid + 1 else high ← mid - 1 if(idx ≠ 0) then Output("Data ditemukan pada indeks : ',i) else Output("Data tidak ditemukan")</pre>

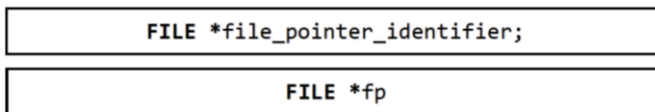
Gambar 1. 22 Binary Search

2.7 File Sekuensial

File sekuensial digunakan agar hasil dari program yang tereksekusi dapat disimpan secara permanen di dalam perangkat penyimpanan mana pun, bukan sementara menghilang ketika program ditutup atau dihentikan. Tujuannya adalah agar data yang sudah tersimpan dapat digunakan kembali.

Bahasa C menyediakan sebuah struktur yang dapat digunakan untuk menyimpan semua informasi yang dibutuhkan untuk mengendalikan aliran data suatu file, yaitu struktur FILE (sering disebut juga sebagai file handle) dan dideklarasikan di dalam file header standard `stdio.h`. Struktur tersebut akan menyimpan informasi aliran arus suatu file yang meliputi indikator posisi file, sebuah pointer buffer (jika ada), sebuah indikator error yang menyimpan ketika proses kesalahan baca/tulis terjadi, sebuah indikator end-of-file yang menyimpan ketika akhir dari suatu file telah tercapai, serta informasi lainnya yang dibutuhkan.

Adapun cara pertama yang harus dilakukan untuk mendapatkan informasi dari struktur FILE tersebut adalah mendeklarasikan sebuah pointer yang akan menunjuk ke struktur FILE. Pointer semacam ini dinamakan dengan pointer file. Berikut ini bentuk umum dari pendeklarasian pointer ke file yang dimaksud di atas.



```
FILE *file_pointer_identifler;
```

```
FILE *fp
```

Gambar 1. 23 Pendeklarasian File

Misalnya, kita akan mendeklarasikan pointer dengan nama `fp` untuk menunjuk ke struktur FILE, maka sintaks penulisannya adalah sebagai berikut.

Tahapan Operasi File

Operasi pada file meliputi tiga tahapan utama.

1. Membuka/mengaktifkan file
2. Memproses file
3. Menutup file

1. Membuka File

Sebelum file dapat diproses (dibaca atau ditulis), file haruslah dibuka terlebih dahulu. Untuk keperluan ini, ada fungsi yang bernama `fopen`. Berikut adalah bentuk dasarnya.



```
FILE *fopen (const char * restrict filename, const char * restrict mode);
```

Gambar 1. 24 Bentuk dasar membuka file

Deskripsi

- **filename**, berupa nama dari file (string) yang akan dibuka.
- **mode**, berupa tipe operasi (string) yang akan dilakukan terhadap file.
- Untuk selanjutnya bentuk dasar di dalam modul ini terdapat di dalam file `stdio.h`.
- Fungsi tersebut akan mengembalikan pointer yang menunjuk pada objek (FILE) untuk mengendalikan aliran data suatu file. Jika operasi pembukaan file gagal, `fopen` akan mengembalikan pointer **NULL** (sebuah konstanta pointer null yang didefinisikan berdasarkan implementasi suatu platform).

Berikut adalah mode dari `fopen` yang terdapat di dalam bahasa C.

Mode	Arti
"r"	Membuka file untuk proses pembacaan (<i>reading</i>). Jika file belum ada, maka file akan gagal dibuka, sedangkan jika file ada, maka file dapat dibaca dengan indikator posisi file mula-mula diposisikan pada awal file.
"w"	Membuka file untuk proses penulisan (<i>writing</i>). Jika file belum ada, maka file akan dibuat. Sedangkan jika file sudah ada, semua isi dari file tersebut akan dihapus tanpa ada peringatan terlebih dahulu. Terakhir, jika file berhasil dibuka, maka file dapat ditulis dengan indikator posisi file mula-mula diposisikan pada awal file.
"a"	Membuka file untuk proses penambahan (<i>appending</i>). Jika file belum ada, maka file akan dibuat. Sedangkan, jika file ada, maka akan menyebabkan proses penulisan data pada file akan ditulis pada bagian akhir dari suatu file, meskipun adanya intervensi fungsi pemosisian file (akan dibahas nanti pada modul ini). Terakhir, jika file berhasil dibuka, maka indikator posisi file mula-mula akan diposisikan berdasarkan implementasi suatu <i>platform</i> apakah di awal atau di akhir file.
"r+"	Sama seperti "r", tetapi dapat juga digunakan untuk proses penulisan.
"w+"	Sama seperti "w", tetapi dapat juga digunakan untuk proses pembacaan.
"a+"	Sama seperti "a", tetapi dapat juga digunakan untuk proses pembacaan.

Gambar 1. 25 Mode `fopen`

Sebagai contoh, berikut adalah contoh dari penggunaan fungsi **fopen**.

```
fp = fopen("TES.TXT", "w");
```

Gambar 1. 26 Contoh Penggunaan fungsi `fopen`

Catatan:

- Jika argumen **mode** terdapat simbol '+' di dalamnya, maka itu dinamakan sebagai mode **update**. Proses masukan (membaca dari file) dan keluaran (menulis ke file) dapat dilaksanakan pada aliran arus suatu file yang bersangkutan.
- Proses penulisan tidak boleh langsung diikuti oleh proses penulisan tanpa adanya pemanggilan intervensi fungsi **fflush** atau fungsi pemosisian file (**fseek**, **fsetpos**, atau **rewind**), kecuali proses pembacaan telah mencapai akhir dari suatu file (end-of-file).

2. Memproses file

Pada saat file dibuka, file bisa diperlakukan sebagai file biner atau file teks. File biner adalah file yang datanya diproses dalam bentuk blok-blok memori yang terdiri dari byte, seperti memori pada RAM. Sedangkan, file teks adalah file yang datanya akan diproses dalam bentuk karakter.

Sebuah file teks biasanya dipakai untuk menyimpan data bertipe karakter atau string. Sedangkan, file biner biasanya dipakai untuk menyimpan data kompleks, seperti struktur (struct)

Hal yang dapat dilakukan untuk bekerja dengan file biner adalah dengan menambah karakter tambahan, yaitu “**b**” pada argumen **mode** dari fungsi **fopen**. Sementara itu, untuk bekerja dengan file teks tidak perlu ditambahkan karakter apa pun di dalamnya.

→ File teks

Dengan **r**, **w**, **a** atau mode update dengan **r+**, **w+**, **a+**.

→ File biner

Dengan **rb**, **wb**, **ab** atau mode update dengan **rb+**, **wb+**, **ab+**, atau **r+b**, **w+b**, **a+b**.

A. File teks

1. Menulis dan Membaca Data String pada File Teks

Untuk menulis data string pada file, ingat bahwa fungsi printf biasa digunakan untuk menulis ke layar, sedangkan fungsi fprintf dapat digunakan untuk menulis pada file. Selanjutnya, untuk membaca data string pada file, ada fungsi yang bernama fgets. Berikut bentuk dasarnya.

<pre>int fprintf(FILE * restrict stream, const char * restrict format, ...);</pre>
<pre>char *fgets(char * restrict s, int n, FILE * restrict stream);</pre>

Gambar 1. 27 Bentuk dasar untuk membaca file

Deskripsi

- **s** adalah data string.
- **stream** adalah pointer **FILE**.
- **n** mendefinisikan berapa banyak karakter yang akan disimpan pada string **s**.
- Fungsi **fprintf** digunakan untuk menulis string **s** ke dalam file berdasarkan string yang ditunjuk oleh **format** untuk menentukan argumen-argumen (...) yang selanjutnya akan dikonversikan menjadi output pada file. Fungsi ini akan mengembalikan jumlah karakter yang ditulis, atau nilai negatif jika output atau error proses pengkodean karakter (encode) pada file terjadi.

- Fungsi **fgets** digunakan untuk membaca string dari file ke dalam string **s**. Jika akhir dari file telah tercapai maka **s** akan tetap tidak berubah dan fungsi akan mengembalikan nilai **NULL**. Namun jika error terjadi, isi dari string **s** tidak dapat ditentukan dan fungsi akan mengembalikan nilai **NULL**, Jika berhasil maka akan mengembalikan string **s**.

Menulis Data String ke dalam File Teks	Membaca Data String dari File Teks
<pre>#include <stdio.h> char string[255]; int main(){ FILE *f_teks; f_teks = fopen("contoh.txt", "w"); gets(string); fprintf(f_teks, "%s\n", string); fclose (f_teks); return 0; }</pre>	<pre>#include <stdio.h> char string[255]; int main(){ FILE *f_teks; f_teks = fopen("contoh.txt", "r"); while (fgets(string, sizeof(string), f_teks) != NULL) printf("%s",string); fclose (f_teks); return 0; }</pre>

Gambar 1. 28 Menulis dan membaca data string

Sebagai catatan, argumen **format** yang digunakan pada fungsi **fprintf**, yaitu **"%s\n"** memiliki karakter baris baru (newline) di dalamnya. Karakter baris baru tersebut harus ditulis secara manual ke dalam file, tanpa itu, bukannya membuat file teks yang terdiri lebih dari satu baris dengan benar, file akan ditulis hanya dalam satu baris saja. Selain menggunakan fungsi **fgets**, **fgetc** juga dapat digunakan untuk membaca data string. Perbedaannya adalah **fgets** digunakan untuk membaca dalam basis string, sedangkan **fgetc** digunakan membaca dalam basis karakter (ingat bahwa string adalah kumpulan dari karakter, dengan kata lain adalah array of char). Berikut adalah bentuk dasarnya.

```
int fgetc(FILE *stream);
```

Gambar 1. 29 Bentuk dasar fgetc

Fungsi **fgets** mengambil karakter dari dalam file dengan tipe **unsigned char** (memiliki jangkauan 0 – 255 nilai dalam nilai ASCII) yang kemudian dikonversikan menjadi nilai **int** positif. Fungsi tersebut mengembalikan karakter dari file yang ditunjuk oleh **stream**. Jika error pembacaan terjadi, maka akan mengembalikan **EOF**.

```

Membaca Data String dari File Teks Menggunakan fgetc
#include <stdio.h>
int karakter;
int main(){
    FILE *f_teks;
    f_teks = fopen("contoh.txt", "r");
    while( (karakter=fgetc(f_teks)) != EOF)
        printf("%c", karakter);
    fclose (f_teks);
    return 0;
}

```

Gambar 1. 30 Membaca data string dari file teks menggunakan *fgetc*

EOF adalah sebuah konstanta negatif **int** yang didefinisikan berdasarkan implementasi suatu platform, nilai ini biasanya dikembalikan oleh beberapa fungsi seperti **fgetc** untuk mengindikasikan akhir dari suatu file, dengan kata lain, tidak ada data yang dapat dibaca dari karakter tersebut. Oleh karena itu, untuk membaca dengan benar menggunakan **fgetc**, jangan mendeklarasikan variabel cadangan untuk menyimpan karakter sebagai **char**, tetapi deklarasikanlah dengan **int**.

2. Menambahkan Data pada File Teks

Metode untuk menambahkan data ke dalam file hampir sama dengan metode untuk menulis data. Perbedaannya terdapat pada argumen **filemode** di dalam fungsi **fopen** yang sedang dibuka oleh file. Sederhana saja, ubah **filemode** “**w**” atau “**w+**” menjadi “**a**” atau “**a+**” untuk menambahkan.

```

Menambahkan Data String ke dalam File Teks
#include <stdio.h>
char string[255];
int main(){
    FILE *f_teks;
    f_teks = fopen("contoh.txt", "a");
    gets(string);
    fprintf(f_teks, "%s\n", string);
    fclose (f_teks);
    return 0;
}

```

Gambar 1. 31 Menambahkan data string ke dalam file teks

B. File biner

1. Menulis dan Membaca Data pada File Biner

Ada dua fungsi untuk menulis atau membaca sebuah data dalam bentuk blok memori (terdiri dari byte), misalnya untuk menulis tipe data apa pun seperti **float** atau **struct**. Kedua fungsi tersebut dinamakan **fread** dan **fwrite**. Berikut adalah bentuk dasarnya.

```
size_t fread(void * restrict ptr, size_t size, size_t nmemb, FILE * restrict stream);
```

```
size_t fwrite(const void * restrict ptr, size_t size, size_t nmemb, FILE * restrict stream);
```

Gambar 1. 32 Bentuk dasar menulis dan membaca file biner

Deskripsi

- **ptr** adalah sebuah pointer yang menunjuk ke daerah memori yang akan digunakan sebagai tempat penyimpanan data sementara untuk membaca dari file (untuk **fread**) atau untuk menulis data ke dalam file (untuk **fwrite**).
- **size** adalah ukuran dari satu blok memori dalam byte yang akan dibaca atau ditulis.
- **nmemb** adalah jumlah dari blok-blok memori yang akan dibaca atau ditulis dalam satu waktu.
- **stream** adalah file pointer yang menunjuk ke **FILE**.
- **size_t**, adalah tipe data integer spesial yang dapat menyimpan ukuran dari suatu objek (seperti variabel, array, tipe data dan apa pun yang dapat menyimpan nilai) yang disimpan dalam memori dengan satuan byte. Tipe data ini juga merupakan dari operator **sizeof**.
- Kedua fungsi tersebut mengembalikan nilai jumlah dari blok-blok memori yang berhasil dibaca (untuk **fread**) atau ditulis (untuk **fwrite**) dalam satu waktu.
- Untuk dapat menulis atau membaca data lebih dari satu blok memori (berdasarkan nilai **nmemb**), sebuah array harus digunakan sebagai tempat penyimpanan sementara sebagai input argumen **ptr**.

Menulis Data ke dalam File Biner	Membaca Data dari File Biner
<pre> #include <stdio.h> int main(){ FILE *f_struktur; int n, i; struct { char Menu[30]; double Harga; } daftar; f_struktur=fopen("Daftar Harga.dat","wb"); printf("Banyaknya menu: "); scanf("%d", &n); getchar(); for (i=1; i<=n; i++){ printf("Menu : "); gets(daftar.Menu); printf("Harga : "); scanf("%lf", &daftar.Harga); getchar(); fwrite(&daftar,sizeof(daftar),1,f_struktur); } fclose(f_struktur); return 0; } </pre>	<pre> #include <stdio.h> int main(){ FILE *f_struktur; struct { char Menu[30]; double Harga; } daftar; f_struktur = fopen("Daftar Harga.dat", "rb"); while(fread(&daftar,sizeof(daftar),1,f_struktur)==1){ printf("Menu : %s\n", daftar.Menu); printf("Harga : %lf\n", daftar.Harga); } fclose(f_struktur); return 0; } </pre>

Gambar 1. 33 Contoh program menulis & membaca file biner

2. Menambahkan Data pada File Biner

Metode untuk menambahkan data ke dalam file biner hampir sama dengan metode untuk menulis data. Tentu saja, gunakan **filemode** “wb”, “wb+”, atau “w+b” untuk menulis dan “ab”, “ab+”, atau “a+b” untuk menambahkan.

Menambahkan Data ke dalam File Biner
<pre>#include <stdio.h> int main(){ FILE *f_struktur; int n, i; struct { char Menu [30]; double Harga; } daftar; f_struktur = fopen("Daftar Harga.dat", "ab"); printf("Silahkan masukkan banyaknya menu tambahan: "); scanf("%d", &n); getchar(); for (i=1; i<=n; i++){ printf("Menu : "); gets(daftar.Menu); printf("Harga : "); scanf("%lf", &daftar.Harga); getchar(); fwrite(&daftar, sizeof (daftar), 1, f_struktur); } fclose(f_struktur); return 0; }</pre>

Gambar 1. 34 Menambahkan data pada file biner

3. Mengakses File Biner Secara Acak

Bahasa C menyediakan fasilitas yang memungkinkan untuk membaca dan menulis file secara acak dengan mengubah indikator posisi file. Dengan adanya fasilitas ini, misalkan bahwa sebuah data yang berada di tengah file akan dibaca, data tersebut dapat dibaca secara langsung tanpa harus diakses satu per satu dimulai dari awal file.

Untuk keperluan ini, ada fungsi yang bernama **fseek**. Berikut bentuk dasarnya:

<pre>int fseek(FILE *stream, long int offset, int whence);</pre>
--

Gambar 1. 35 Bentuk dasar fseek

Deskripsi

- Jika operasi gagal, maka akan mengembalikan nilai selain nol yang didefinisikan berdasarkan implementasi suatu platform.
- **stream** adalah file pointer yang menunjuk ke **FILE**.
- **offset** adalah nilai penyesuaian byte terhadap referensi **whence**.
- **whence** adalah referensi indikator posisi file.

- Ingat, setelah pemanggilan **fseek** yang berhasil, operasi selanjutnya dalam aliran data update dapat menjadi input (membaca dari file) atau output (menulis ke dalam file).

Berikut adalah tabel dari referensi **whence**.

Macro	Referensi Lokasi File
SEEK_SET	Awal file
SEEK_CUR	Indikator posisi file saat ini
SEEK_END	Akhir file

Gambar 1. 36 Referensi Whence

Macro-macro tersebut didefinisikan berdasarkan implementasi suatu platform yang merupakan konstanta bilangan bulat dengan nilai-nilai yang berbeda.

Sebagai

contoh:

1. <code>fseek(fp, 3, SEEK_SET);</code>	Indikator posisi file akan diubah 3 byte setelah awal dari file.
2. <code>fseek(fp, -3, SEEK_END);</code> <i>Perhatikan tanda negatif.</i>	Indikator posisi file akan diubah 3 byte sebelum akhir dari file.
3. <code>fseek(fp, 3, SEEK_CUR);</code>	Indikator posisi file akan diubah 3 byte setelah indikator posisi file saat ini.
4. <code>fseek(fp, -sizeof(DATA), SEEK_CUR);</code> <i>Perhatikan tanda negatif.</i>	Indikator posisi file akan diubah berdasarkan ukuran dari objek DATA sebelum indikator posisi file saat ini.

Perlu diingat bahwa argumen **offset** juga dapat menerima nilai negatif.

Extra, the rewind function:

<code>void rewind(FILE *stream);</code>

Gambar 1. 37 Rewind function

Deskripsi

- Fungsi **rewind** mengubah indikator posisi file untuk aliran data yang ditunjuk oleh **stream** menjadi bagian awal file. Hal tersebut sama saja dengan **(void)fseek(stream, 0L, SEEK_SET)** kecuali indikator error pada aliran data juga dihapus.
- Fungsi **rewind** tidak mengembalikan nilai.

Berikut adalah contoh penggunaan dari fungsi **fseek**.

Mengakses File Biner Secara Acak
<pre>#include <stdio.h> int main(){ FILE *f_struktur; int no_struct; long int offset_byte;</pre>

```

struct {
    char Menu [30];
    double Harga;
} daftar;

f_struktur = fopen("Daftar Harga.dat", "rb");
printf("Silahkan masukkan nomor urutan yang ada pada menu: ");
scanf("%d", &no_struct);
offset_byte = (no_struct - 1) * sizeof(daftar);
fseek(f_struktur, offset_byte, SEEK_SET);

if (fread(&daftar, sizeof (daftar), 1, f_struktur) == 0){
    printf("Menu yang dicari tidak dapat ditemukan pada daftar.\n");
} else {
    printf("Menu : %s\n",daftar.Menu);
    printf("Harga: %lf\n",daftar.Harga);
}

fclose(f_struktur);
return 0;
}

```

Indikator file posisi akan diposisikan tepat sebelum data yang nanti akan ditampilkan pada layar.

Gambar 1. 38 Mengakses file biner secara acak

4. Menghapus Data dalam File Biner

Untuk menghapus data dengan benar, ada beberapa cara untuk melakukannya. Tetapi hanya ada salah satu yang akan dijelaskan pada modul ini. Berikut adalah prosedur yang benar untuk melakukannya.

1. Buka file utama dengan mode baca (read).
2. Buatlah file cadangan dan buka file tersebut dengan mode tulis (write).
3. Salin semua data dari file utama ke dalam file cadangan, kecuali data yang akan dihapus.
4. Tutup file utama dan file cadangan.
5. Hapus file utama.
6. Ubah nama file file cadangan berdasarkan nama file file utama.

Bahasa C juga menyediakan fungsi untuk menghapus dan mengubah nama suatu file, secara berurutan dengan nama **remove** dan **rename**. Berikut adalah bentuk dasarnya.

```
int remove(const char *filename);
```

Gambar 1. 39 Bentuk dasar fungsi remove

Fungsi **remove** menyebabkan file dengan nama string yang ditunjuk oleh **filename**, sehingga tidak dapat lagi diakses oleh file dengan nama tersebut. Dengan kata lain, file akan hilang.

```
int rename(const char *old, const char *new);
```

Gambar 1. 40 Bentuk dasar fungsi rename

Fungsi **rename** menyebabkan file dengan nama string yang ditunjuk oleh **old** untuk kemudian dikenali menjadi nama yang diberikan oleh string yang ditunjuk oleh **new**.

Fungsi **remove** dan **rename** mengembalikan bilangan bulat nol jika operasi berhasil dan selainnya yang didefinisikan berdasarkan implementasi suatu platform jika gagal.

Menghapus Data dalam File Biner

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

int main(){
    FILE *f_struktur;
    FILE *f_struktur2;

    char dicari[50];

    struct {
        char Menu[30];
        double Harga;
    } daftar;

    f_struktur = fopen("Daftar Harga.dat", "rb");
    f_struktur2 = fopen("Daftar Harga2.dat", "wb");

    printf("Silahkan masukkan nama menu yang akan dihapus: "); gets(dicari);

    while (fread(&daftar,sizeof (daftar),1, f_struktur)==1){
```

```

        if (strcmp(daftar.Menu, dicari)!=0){
            fwrite(&daftar, sizeof(daftar), 1, f_struktur2);
        }
    }

    fclose(f_struktur);
    fclose(f_struktur2);

    remove("Daftar Harga.dat");
    rename("Daftar Harga2.dat", "Daftar Harga.dat");

    return 0;
}

```

Gambar 1. 41 Menghapus data dalam file biner

5. Menutup file

Untuk menutup file, ada fungsi yang bernama **fclose**. Bentuk umumnya seperti berikut.

```
int fclose(FILE *stream);
```

Gambar 1. 42 Bentuk dasar fclose

Deskripsi

- **stream** adalah file pointer yang menunjuk ke **FILE**.
- Jika fungsi **fclose** berhasil, maka akan menghilangkan aliran data file yang ditunjuk oleh **stream** dan menutup file yang bersangkutan. Data buffer untuk proses penulisan yang belum ditulis dalam aliran suatu data akan ditulis terlebih dahulu, tetapi data buffer untuk proses pembacaan yang belum dibaca akan dibuang.
- Fungsi **fclose** mengembalikan nilai nol jika aliran data berhasil ditutup, atau **EOF** jika ada error yang terdeteksi.

6. Mengosongkan Buffer pada Aliran Dat

```
int fflush(FILE *stream);
```

Gambar 1. 43 Bentuk dasar mengosongkan buffer

Deskripsi

- **stream** adalah file pointer yang menunjuk ke **FILE**.
- Data buffer untuk proses penulisan yang belum ditulis ke dalam aliran suatu data akan ditulis terlebih dahulu.

- Jika **operasi terakhir bukan input** (membaca dari file), fungsi **fflush** akan menyebabkan data buffer untuk proses penulisan yang belum ditulis dalam aliran suatu data akan segera ditulis ke dalam file. Selain itu, perilaku dari fungsi ini tidak terdefinisi (**undefined**).
- Fungsi **fflush** akan menetapkan indikator error untuk aliran data dan mengembalikan nilai **EOF** jika error proses penulisan terjadi. Selain itu akan mengembalikan nilai nol.

7. Menulis dan Menempatkan indikator posisi file

```
int fgetpos(FILE * restrict stream, fpos_t * restrict pos);

int fsetpos(FILE *stream, const fpos_t *pos);
```

Gambar 1. 44 Menulis dan menempatkan indikator posisi file

Deskripsi

- **stream** adalah file pointer yang menunjuk ke **FILE**.
- **fpos_t** adalah tipe data yang dapat menyimpan informasi yang dibutuhkan untuk menentukan setiap posisi secara unik di dalam suatu file.
- **fgetpos** akan menyimpan indikator posisi file di dalam objek yang ditunjuk oleh **pos** dan **fsetpos** akan mengubah indikator posisi file berdasarkan nilai dari objek yang ditunjuk oleh **pos**.
- Kedua fungsi tersebut akan mengembalikan bilangan bulat nol jika operasi berhasil dan selainnya yang didefinisikan berdasarkan implementasi suatu platform jika gagal.

2.8 Algoritma Rekursif

Rekursi adalah algoritma yang melakukan pemanggilan terhadap dirinya sendiri. Algoritma yang bersifat rekursi disebut rekursif. Rekursi hanya dapat digunakan pada prosedur dan fungsi, hal ini dikarenakan hanya prosedur dan fungsi saja yang dapat dipanggil. Pada umumnya, rekursi dapat diilustrasikan sebagai algoritma perulangan biasa (iteratif) tetapi dengan menggunakan fungsi dan prosedur. Di dalam rekursi terdapat dua istilah penting, yaitu basis dan rekurens. Pertama, basis merupakan kondisi-kondisi yang akan menghentikan proses rekursif. Sementara rekurens adalah perintah yang akan memanggil algoritma rekursif itu sendiri (dalam hal ini prosedur atau fungsi).

<pre>#include <stdio.h> int nFaktorial(int n); int x, hasil; int main(){ printf("FUNGSI FAKTORIAL\n"); printf("Silahkan masukkan angka: "); scanf("%d",&x); getchar(); hasil = nFaktorial(x); printf("Hasil faktorial: %d", hasil); return 0; } int nFaktorial(int n){ if (n==1 n==0){ return 1; } else if (n>1) { return n * nFaktorial(n-1); } }</pre>	<p>Basis → kondisi yang akan menyebabkan proses rekursif berhenti → $n = 0$ atau $n = 1$</p> <p>Rekurens → kondisi yang akan menyebabkan proses rekursif terus berlanjut dengan cara memanggil dirinya sendiri → $nFaktorial(n-1)$;</p> <p>Asumsikan jika <i>input</i> $n = 3$ maka :</p> <p>$nFaktorial = 3 * nFaktorial(2)$ $nFaktorial = 2 * nFaktorial(1)$ $nFaktorial = 1 * nFaktorial(0)$ $nFaktorial = 1$</p> <p>Sehingga, $nFaktorial = 3 * 2 * 1 * 1 = 6$</p>
--	--

Gambar 1. 45 Contoh Algoritma Rekursif

Sebagai tambahan, berikut ini adalah contoh lain dari penggunaan algoritma rekursif.

Input Data Mahasiswa	Cek Bilangan Prima
<pre>#include <stdio.h> struct mahasiswa{ char nama[20],nim[20]; } mhs; int n; void input(int x); int main(){ printf("Banyak Mahasiswa : "); scanf("%d", &n); getchar(); input(n); return 0; } void input(int x){ if (x > 0){ printf("Nama : "); gets(mhs.nama); printf("NIM : "); gets(mhs.nim); x=x-1; input(x); } }</pre>	<pre>#include <stdio.h> int cekprima(int p); int n; int main(){ printf("masukkan angka :"); scanf("%d",&n); getchar(); if (n > 1){ int p = cekprima(n-1); if(p == 1){ printf("%d adalah bilangan prima\n",n); } else { printf("%d bukan bilangan prima\n",n); } } else { printf("%d bukan bilangan prima\n",n); } return 0; } int cekprima(int p){ if(p == 1){ return 1; } else if (n%p == 0) { return 0; } else { return cekprima(p-1); } }</pre>

Gambar 1. 46 Penggunaan Algoritma Rekursif

BAB III HASIL DAN ANALISIS

3.1 Deskripsi program

D' Auto Parts merupakan sebuah program yang digunakan untuk jual beli suku cadang kendaraan bermotor secara online dimana alamat dan nomor telepon dealer terpampang pada bagian menu login. Pada program ini terdapat 2 akun yaitu dealer dan customer. Setiap akun diharuskan untuk login terlebih dahulu; dealer dapat menambah dan menghapus barang, customer diharuskan membuat ID dan password terlebih dahulu kemudian melakukan registrasi dengan mengisi nama, alamat, dan nomor telepon.

Fitur

Dealer:

- Login (ID, Password)
- Menghapus dan melihat akun customer
- Menambahkan dan menghapus barang sesuai kategori yang ada dengan harganya dan jenis kendaraan bermotornya (kategori: mesin, ban, velg, aksesoris, jok/kursi)

Customer:

- Login (nama, alamat, nomor telepon, ID, password)
- Cari barang pada kategori yang ada (kategori: mesin, ban, velg, aksesoris, jok/kursi)
- Membeli barang dari kategori dan barang yang ada
- Opsi pembelian COD ke alamat dealer

3.2 Screenshot Fitur – Fitur Program

Role Dealer



Gambar 2. 1 Bagian Dealer

Melihat dan menghapus data customer

```
===== Dealer =====
1. Melihat dan menghapus data Customer
2. Menambah Data Barang
3. Menghapus Data Barang
4. Data Barang
5. Log Out
Masukkan nomor Pilihan :
_
```

```
===== Data Customer =====
1. ID : maurits
   Password : maurits
   Nama : maurits
   Alamat : Makassar, Sulawesi Selatan
   No. Telp : 085201010101
-----
2. ID : salwa
   Password : salwa
   Nama : salwa
   Alamat : Jombang
   No. Telp : 081234567890
-----
Masukkan ID yang akan dihapus : maurits_
```

```
===== Data Customer =====
1. ID : salwa
   Password : salwa
   Nama : salwa
   Alamat : Jombang
   No. Telp : 081234567890
-----
Press any key to continue . . . _
```

Gambar 2. 2 Melihat dan Menghapus data customer

Menambah data barang

```
-----
===== Dealer =====
-----
1. Melihat dan menghapus data Customer
2. Menambah Data Barang
3. Menghapus Data Barang
4. Data Barang
5. Log Out
Masukkan nomor Pilihan :

===== Data Barang =====
1. Nama barang : aki
   Kategori    : mesin
   Jenis       : mobil
   Harga       : Rp. 10000000
2. Nama barang : dunlop
   Kategori    : ban
   Jenis       : mobil
   Harga       : Rp. 5000000
3. Nama barang : chemco
   Kategori    : velg
   Jenis       : motor
   Harga       : Rp. 2500000
4. Nama barang : spakboard
   Kategori    : aksesoris
   Jenis       : motor
   Harga       : Rp. 500000
5. Nama barang : cover kulit
   Kategori    : jok
   Jenis       : mobil
   Harga       : Rp. 3000000
Press any key to continue . . .
```

Gambar 2. 3 Menambah data Barang

Menghapus data barang

```
===== Data Barang =====
1. Nama barang : aki
   Kategori    : mesin
   Jenis       : mobil
   Harga       : Rp. 10000000
2. Nama barang : dunlop
   Kategori    : ban
   Jenis       : mobil
   Harga       : Rp. 5000000
3. Nama barang : chemco
   Kategori    : velg
   Jenis       : motor
   Harga       : Rp. 2500000
4. Nama barang : spakboard
   Kategori    : aksesoris
   Jenis       : motor
   Harga       : Rp. 500000
5. Nama barang : cover kulit
   Kategori    : jok
   Jenis       : mobil
   Harga       : Rp. 3000000
Masukkan kategori yang akan dihapus : jok
```

```

===== Data Barang =====
1. Nama barang : aki
   Kategori    : mesin
   Jenis       : mobil
   Harga       : Rp. 10000000

2. Nama barang : dunlop
   Kategori    : ban
   Jenis       : mobil
   Harga       : Rp. 5000000

3. Nama barang : chenco
   Kategori    : velg
   Jenis       : motor
   Harga       : Rp. 2500000

4. Nama barang : spakboard
   Kategori    : aksesoris
   Jenis       : motor
   Harga       : Rp. 500000

Press any key to continue . . .

```

Gambar 2. 4 Menghapus Data Barang

Role Customer

```

=====
===== D'Auto Parts =====
=====
Menu Utama
=====
1. Dealer
2. Customer
Masukkan nomor Pilihan :
2

```

```

=====
===== Customer =====
=====
1. login
2. Registrasi
3. Kembali
Masukkan nomor Pilihan :
2

```

```
=====
==== Registrasi ====
=====
Nama      : maurits
Alamat    : Makassar, Sulawesi Selatan
No. Telepon : 085201010101
ID        : maurits
Password  : maurits

=====
==== Customer ====
=====
1. Login
2. Registrasi
3. Kembali
Masukkan nomor Pilihan :
1

=====
==== LOGIN CUSTOMER ====
ID : maurits
Password : maurits_
```

Gambar 2. 5 Role Customer

List barang

```
.....
**** Customer ****
.....
1. List Barang
2. Cari barang sesuai Kategori
3. Checkout atau Pemesanan
4. Log Out
Masukkan nomor Pilihan :
1

Nama Barang : aki
Kategori : mesin
Jenis : mobil
Harga : 10000000

Nama Barang : dunlop
Kategori : ban
Jenis : mobil
Harga : 5000000

Nama Barang : chemco
Kategori : velg
Jenis : motor
Harga : 2500000

Nama Barang : spakboard
Kategori : aksesoris
Jenis : motor
Harga : 500000

Press any key to continue . . .
```

Gambar 2. 6 List Barang

Cari barang sesuai kategori

```
.....
**** Customer ****
.....
1. List Barang
2. Cari barang sesuai Kategori
3. Checkout atau Pemesanan
4. Log Out
Masukkan nomor Pilihan :
2
```

```

===== CARI BARANG =====
Daftar Kategori
0. Batal
1. Mesin
2. Ban
3. Velg
4. Aksesoris
5. Jok
6. Kembali
Masukkan pilihan anda : 1
2. Nama barang      : aki
   Kategori         : mesin
   Jenis            : mobil
   Harga            : Rp. 10000000

Press any key to continue . . .

```

```

===== CARI BARANG =====
Daftar Kategori
0. Batal
1. Mesin
2. Ban
3. Velg
4. Aksesoris
5. Jok
6. Kembali
Masukkan pilihan anda : 2
3. Nama barang      : dunlop
   Kategori         : ban
   Jenis            : mobil
   Harga            : Rp. 5000000

Press any key to continue . . .

```

```

===== CARI BARANG =====
Daftar Kategori
0. Batal
1. Mesin
2. Ban
3. Velg
4. Aksesoris
5. Jok
6. Kembali
Masukkan pilihan anda : 3
4. Nama barang      : chenco
   Kategori         : velg
   Jenis            : motor
   Harga            : Rp. 2500000

Press any key to continue . . .

```

```

===== CARI BARANG =====
Daftar Kategori
0. Batal
1. Mesin
2. Ban
3. Velg
4. Aksesoris
5. Jok
6. Kembali
Masukkan pilihan anda : 4
5. Nama barang      : spakboard
   Kategori         : aksesoris
   Jenis            : motor
   Harga            : Rp. 500000

Press any key to continue . . .

```

Gambar 2. 7 Cari barang sesuai kategori

Checkout atau Pemesanan

```
-----
==== Customer ====
-----
1. List Barang
2. Cari barang sesuai Kategori
3. Checkout atau Pemesanan
4. Log Out
Masukkan nomor Pilihan :
3

Nama Barang : aki
Kategori : mesin
Jenis : mobil
Harga : 10000000

Nama Barang : dunlop
Kategori : ban
Jenis : mobil
Harga : 5000000

Nama Barang : chemco
Kategori : velg
Jenis : motor
Harga : 2500000

Nama Barang : spakboard
Kategori : aksesoris
Jenis : motor
Harga : 500000

Masukkan Nama Barang yang akan dibeli : dunlop

===== Proses pembelian anda telah berhasil =====
===== Silahkan lakukan transaksi COD =====
===== Di alamat : Jl. Airlangga No.100 Jombang =====

Press any key to continue . . .

===== Data Barang =====
Nama barang : aki
Kategori : mesin
Jenis : mobil
Harga : Rp. 10000000

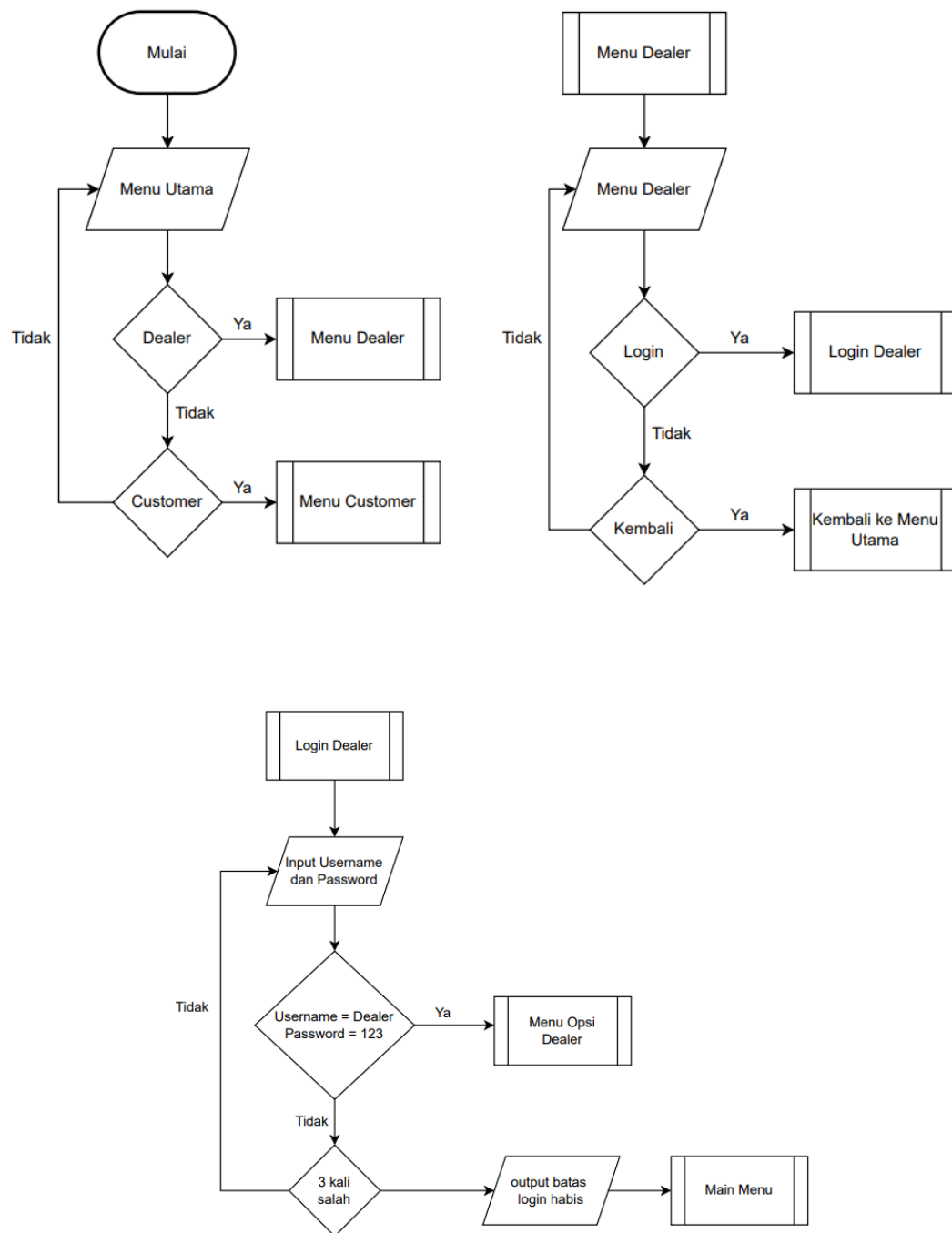
Nama barang : chemco
Kategori : velg
Jenis : motor
Harga : Rp. 2500000

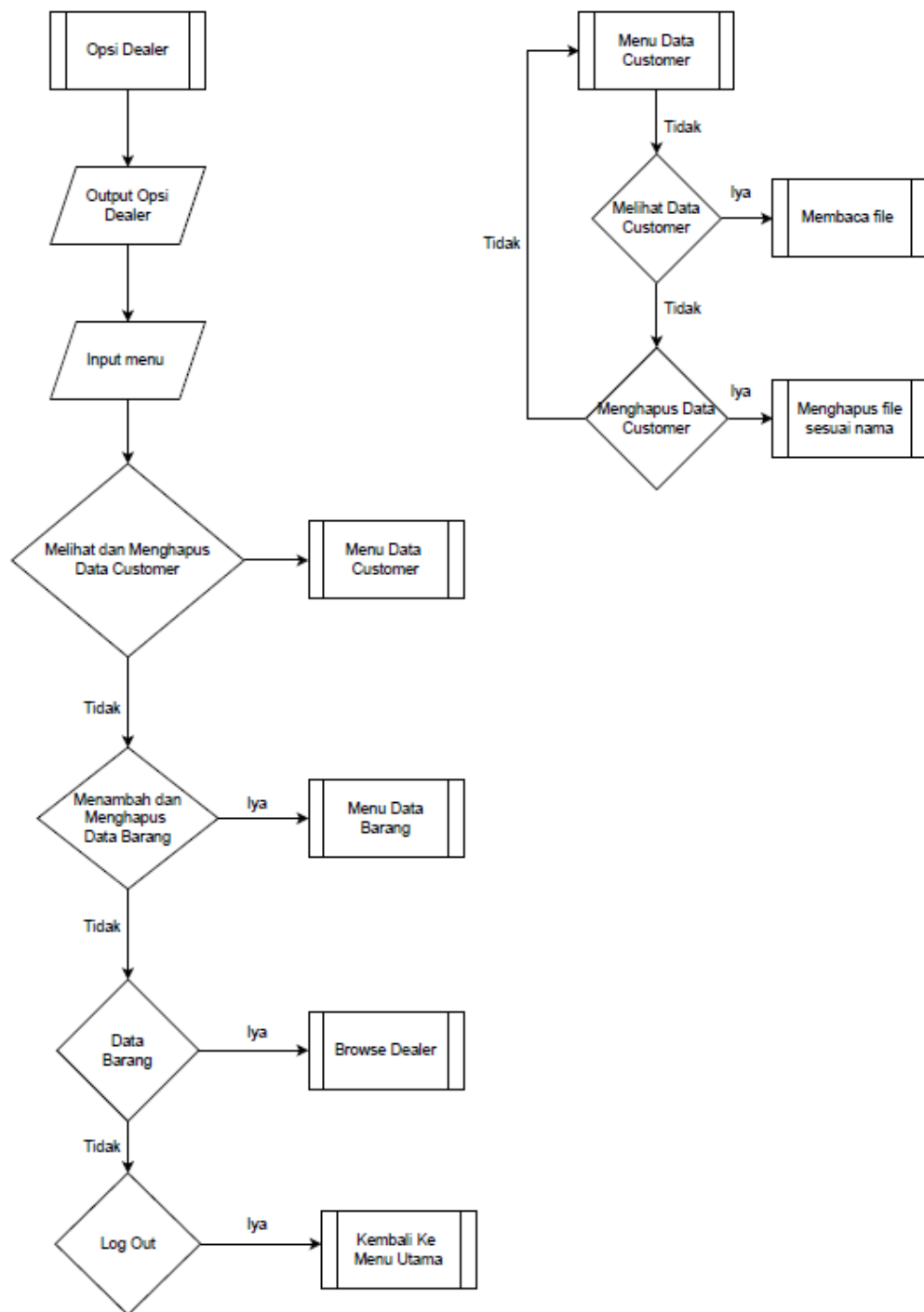
Nama barang : spakboard
Kategori : aksesoris
Jenis : motor
Harga : Rp. 500000

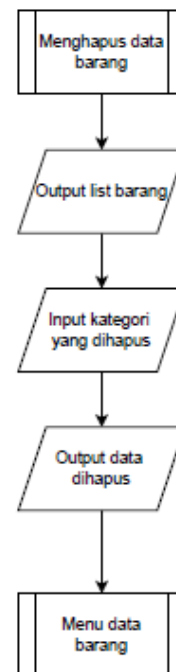
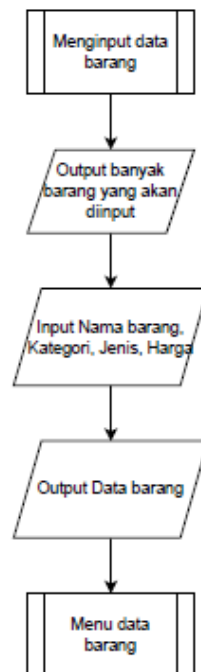
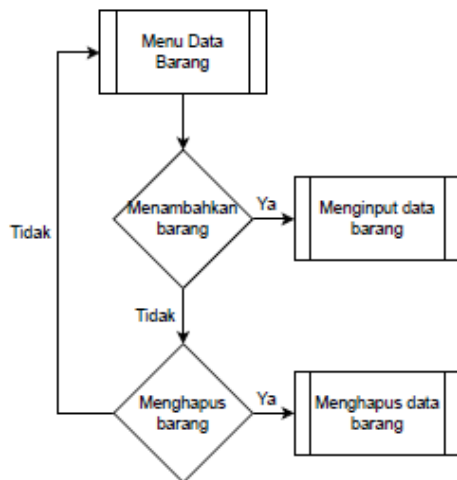
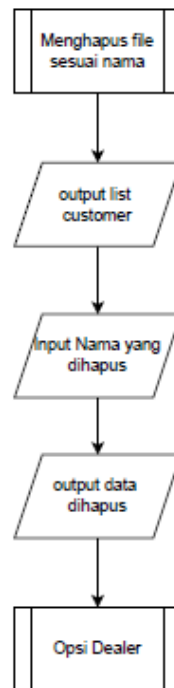
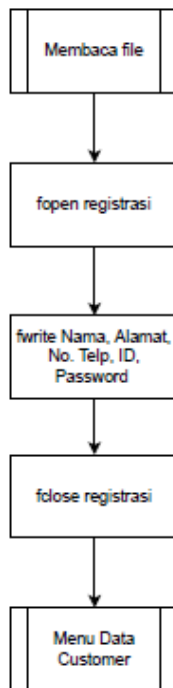
Press any key to continue . . .
```

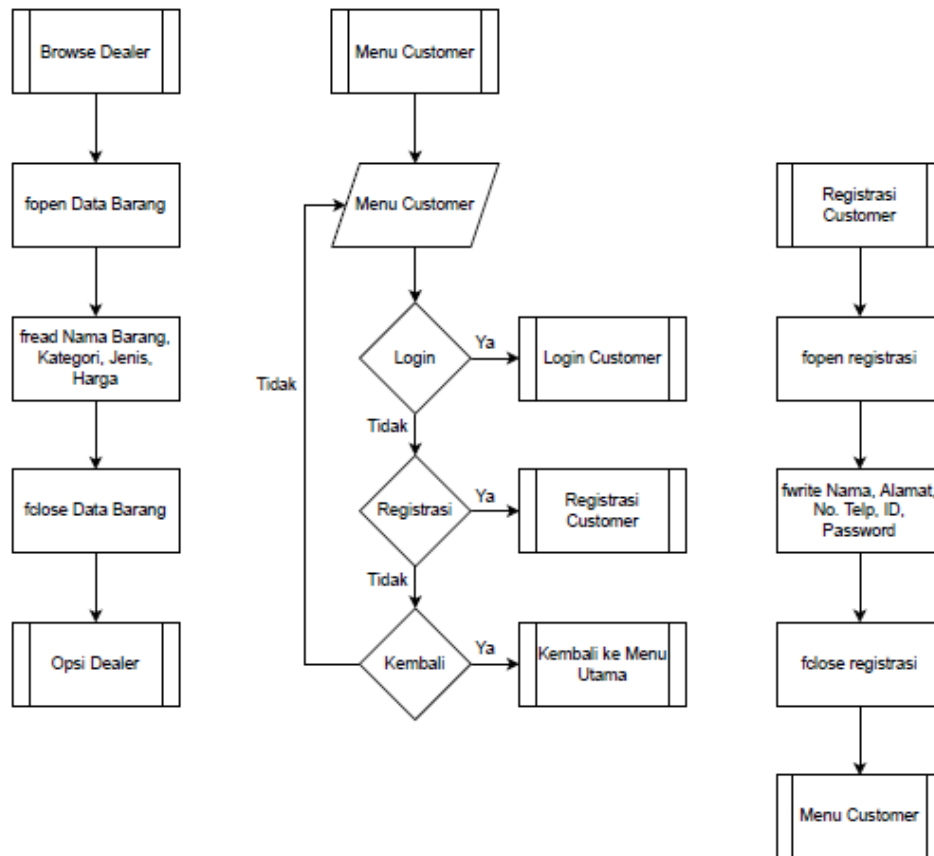
Gambar 2. 8 Checkout atau Pemesanan

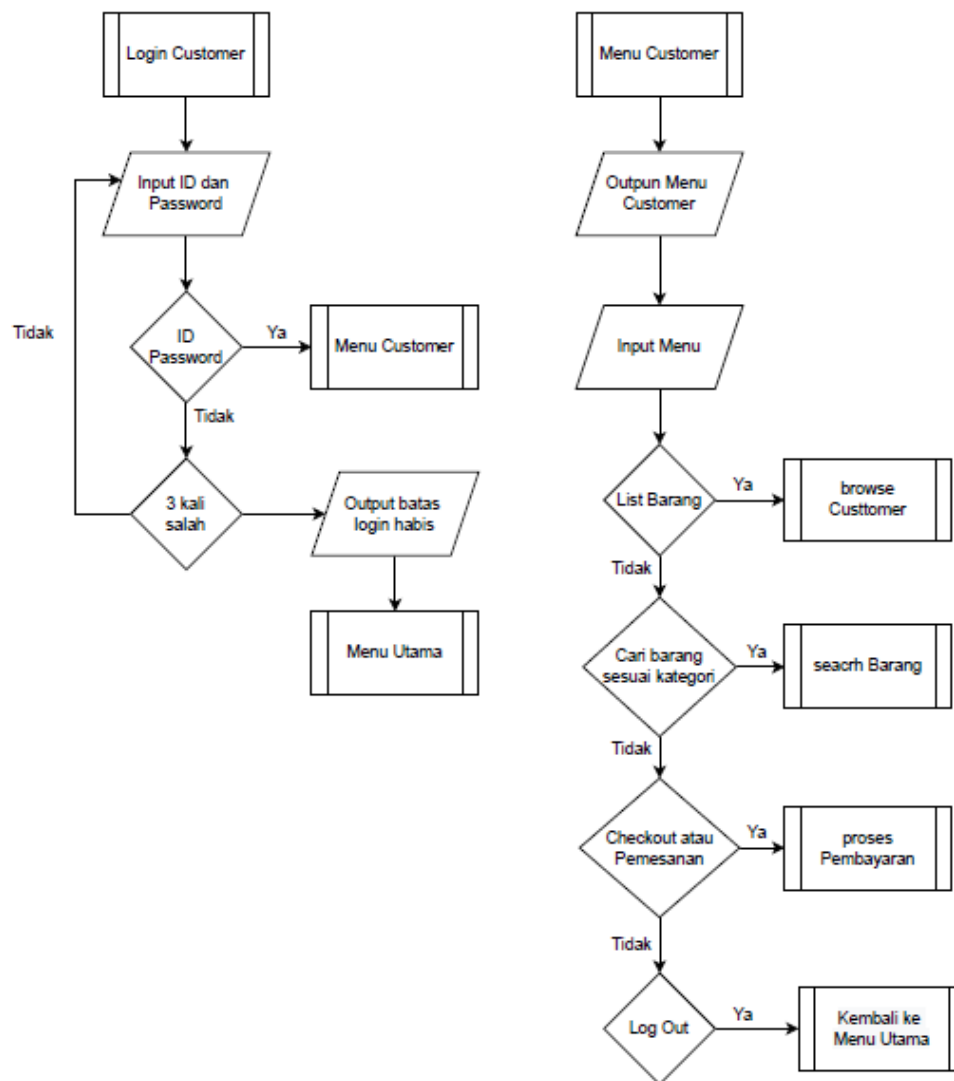
3.3 Flowchart

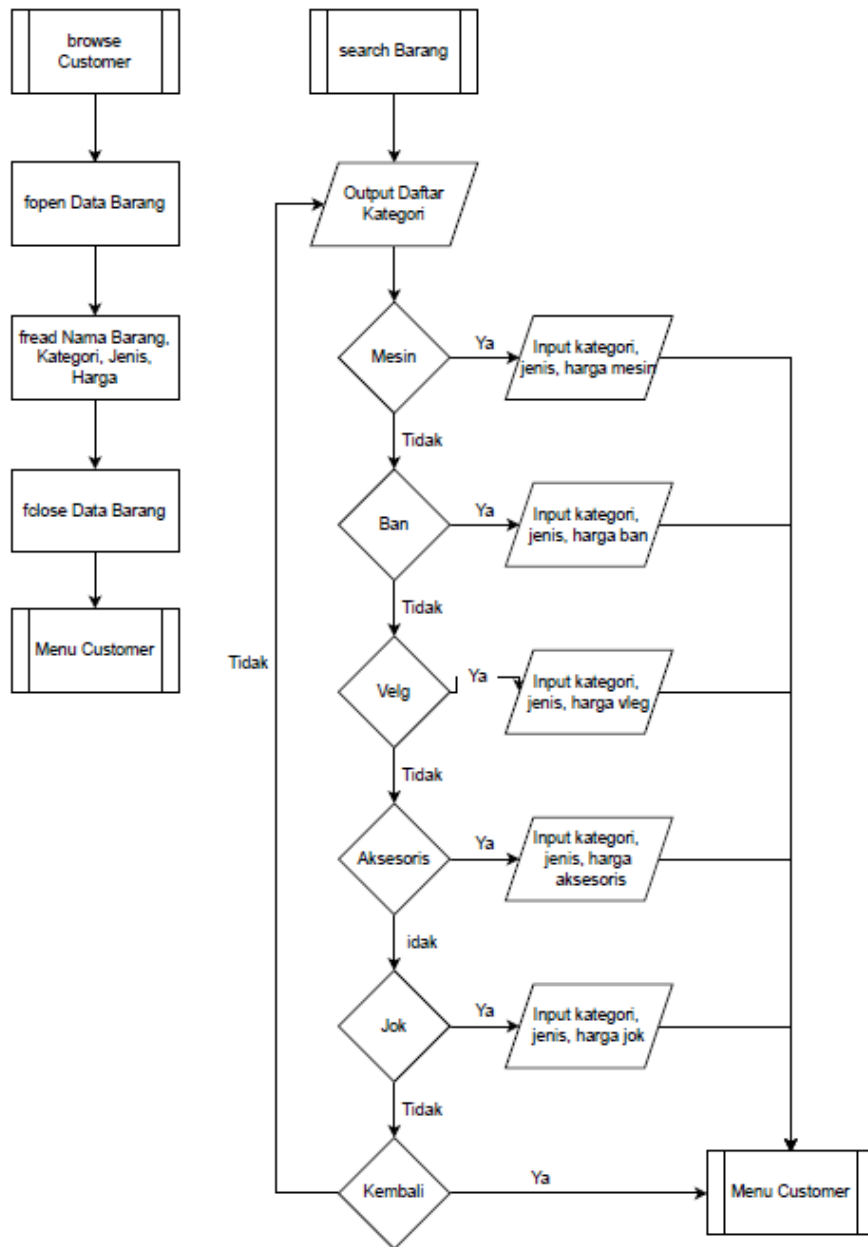


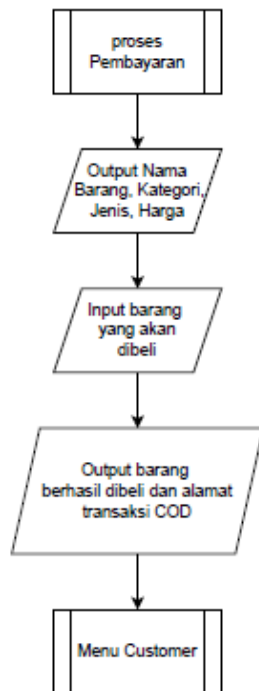












Gambar 2. 9 Flowchart

3.4 Analisis Program

Program D' Auto Parts merupakan pengaplikasian bahasa C dari materi yang terdapat di Modul Praktikum Alpro Semester Genap 2021-2022. Dalam program D' Auto Parts terdapat 2 role user yaitu Dealer (sebagai admin) dan customer. Masing-masing role memiliki hak akses fitur yang berbeda. Untuk mengakses fitur-fitur tersebut admin hanya perlu login dan customer perlu melakukan registrasi untuk dapat login. Namun bagi customer yang telah melakukan registrasi, selanjutnya hanya perlu login. Pihak dealer (admin) dapat melihat dan menghapus akun customer, menambahkan dan menghapus barang sesuai kategori yang ada dengan harganya dan jenis kendaraan bermotornya (kategori: mesin, ban, velg, aksesoris, jok/kursi), dan melihat data barang yang telah ditambahkan. Pihak customer dapat mencari barang pada kategori yang ada (kategori: mesin, ban, velg, aksesoris, jok/kursi), membeli barang dari kategori dan barang yang ada, dan opsi pembelian COD ke alamat dealer. Dalam program D' Auto Parts tidak menggunakan materi bab array, bab sorting, dan bab algoritma rekursif. Program D' Auto Parts banyak menggunakan materi bab file sekuensial. Materi bab file sekuensial yang dipakai rata-rata adalah proses penulisan (writing), pembacaan (reading), penambahan (appending), dan hapus data dalam file biner. Dalam proses pemrograman fitur menghapus customer kami mengalami kendala dimana syntax yang kami gunakan sudah betul dan sama, saat dicompile juga tidak mengalami error namun saat kami akan menghapus data customer data tersebut tidak terhapus. Setelah diamati sambil diubah-ubah ternyata tetap tidak mau terhapus. Akhirnya kami mencoba mengganti nama file yang awalnya register menjadi f_register, saat dicoba kembali data customer berhasil terhapus.

BAB IV PENUTUP

4.1 Kesimpulan

Program ini bertujuan untuk memudahkan customer dalam memilih kendaraan bermotor yang akan dibeli beserta cara pembayarannya. Dan memudahkan dealer maupun customer untuk memilih pelayanan yang akan dipilih seperti menambahkan atau menghapus barang.

4.2 Saran

Diharapkan bagi mahasiswa yang ingin mengembangkan program ini agar mencari sumber informasi yang lebih baik guna dapat memenuhi kebutuhan manusia yang lebih efisien dan member pemahaman yang lebih baik untuk pembaca.

LAMPIRAN



LABORATORIUM DASAR KOMPUTER TELKOM UNIVERSITY


Gedung P/N Ruang P304-N109 Jalan Telekomunikasi 1 Terusan Buah Batu,
Bandung



Nama Asisten : Priyagung Yudho Aufar Rizky (PNG)

Judul Tubes : D'Auto Parts

Asistensi:

Tanggal: Senin, 06 Juni 2022	Tanda Tangan Asisten : 
Kelas/Kelompok: TT-05/2	

Pembahasan :

Mengganti bentuk dan jalannya program sebanyak 50%.

Memperbaiki cari barang dalam kategori dan fitur menambah dan menghapus barang.



LABORATORIUM DASAR KOMPUTER TELKOM UNIVERSITY


Gedung P/N Ruang P304-N109 Jalan Telekomunikasi 1 Terusan Buah Batu,
Bandung



Nama Asisten : Priyagung Yudho Aufar Rizky (PNG)

Judul Tubes : D'Auto Parts

Asistensi:

Tanggal: Senin, 08 Juni 2022	Tanda Tangan Asisten : 
Kelas/Kelompok: TT-05/2	

Pembahasan :

Memperbaiki check out barang dan memperbaiki fitur hapus dan lihat customer.

Membahas laporan tentang latar belakang, Batasan masalah, manfaat, dan table.



LABORATORIUM DASAR KOMPUTER TELKOM UNIVERSITY


Gedung P/N Ruang P304-N109 Jalan Telekomunikasi 1 Terusan Buah Batu,
Bandung



Nama Asisten : Priyagung Yudho Aufar Rizky (PNG)

Judul Tubes : D'Auto Parts

Asistensi:

Tanggal: Senin, 10 Juni 2022	Tanda Tangan Asisten : 
Kelas/Kelompok: TT-05/2	

Pembahasan :

Memperbaiki bagian fitur melihat dan menghapus data customer.

Memisahkan fitur menambah dan menghapus barang.