DOKUMENTASI PENGEMBANGAN APLIKASI
JASA SEWA MOBIL
DENGAN NAMA "RENTURE"

Oleh:

Akhtar Abdurrasyid Nitisastra

Assyifa Azzahra

Gede Bramanta

Gerrald Nathaniel Syarif

Matthew Avrillio

Yehezkiel Yefta Valastra

SEPTEMBER 2021

KATA PENGANTAR

Puji syukur ke hadirat Tuhan Yang Maha Esa. Berkat rahmat dan hidayah-Nya, kami dapat menyelesaikan tugas proyek akhir bertemakan pengembangan aplikasi jasa sewa mobil bernama “Renture” dengan tepat waktu.

Laporan ini kami susun untuk memenuhi tugas bootcamp algoritma. Selain itu, makalah ini bertujuan menambah wawasan tentang alur pemrograman dalam aplikasi terkait bagi para pembaca dan juga bagi penulis.

Penulis mengucapkan terima kasih kepada para kakak pembimbing bootcamp. Ucapan terima kasih juga disampaikan kepada semua pihak yang telah membantu diselesaikannya projek ini.

Kami menyadari bahwa laporan dokumentasi ini masih jauh dari sempurna. Oleh sebab itu, kami mengharapkan saran dan kritik yang membangun demi perbaikan di masa yang akan mendatang.

Jakarta, September 2021

Penulis

DAFTAR ISI

HALAMAN SAMPUL i

KATA PENGANTAR ii

DAFTAR ISI iii

BAB I PENDAHULUAN 1

BAB II PEMBAHASAN 2

A. Header 2
B. Struct 2
C. Main Function 3
D. Displaying Selection of Cars 5
E. Search Car 6
F. Shopping Cart 8
G. Sort Cars 11
H. Feedback 13

BAB III PENUTUP 16

DAFTAR PUSTAKA 17

BAB I
PENDAHULUAN

Seiring dengan berkembangnya teknologi informasi dan komunikasi, agar dapat mempermudah hidup manusia, maka haruslah diciptakan sebuah sistem yang dapat digunakan dengan mudah oleh semua kalangan baik usia remaja, usia dewasa bahkan usia lanjut.

Untuk itu, sistem tersebut sudah selayaknya dan seharusnya dibuat dengan jelas supaya dapat dimengerti. Kami sangat mengerti dan memahami berbagai permasalahan yang dihadapi calon konsumen pada saat ingin memesan layanan jasa rental mobil. Entah itu kondisi mobil yang kurang bagus, kebersihan kurang terjaga, pengantaran mobil yang sering kali terjadi keterlambatan dan lain-lainnya.

Renture dibuat dengan maksud mempermudah dan menjamin setiap pelanggan yang melakukan order untuk rental mobil kami. Kami memberikan banyak pilihan mobil yang dapat anda pilih sesuai dengan kebutuhan anda. Tujuan utama kami membuat aplikasi ini adalah murni untuk melayani anda sebagai calon konsumen kami dan memberikan pengalaman terbaik yang tidak mungkin anda bisa lupakan saat menggunakan jasa rental mobil kami yang dapat diatur secara fleksibel dengan satu sentuhan.

BAB II
PEMBAHASAN

A. Header

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

Sebelum memulai eksekusi file menggunakan fungsi main, perlu disisipkan header pada awal projek cpp terlebih dahulu dengan format "# include <header>". Karimishaq (2019) mendefinisikan header sebagai "file yang berisi deklarasi variabel, function, tipe data, dll." Apabila header tidak dicakupi dalam file, beberapa fungsi tidak akan berjalan ketika program dikompilasi dan dijalankan. Maka dari itu, diperlukan dua file header di dalam projek.

File header pertama yang dibutuhkan adalah stdio.h. Berdasarkan artikel yang ditulis oleh Wildan (2018), "Stdio (STandarD Input Output) merupakan header yang berisi fungsi-fungsi, makro dan tipe yang digunakan untuk melakukan operasi input output." Stdio.h menggunakan apa yang dinamakan sebagai streams untuk mendefinisikan fungsi-fungsi berkaitan dengan data input-output seperti printf dan scanf. Standar streams sendiri terbagi menjadi stdin (standard input), stdout (standard output), dan stderr (standard error).

Di samping itu, dibutuhkan header untuk mengakses fungsi-fungsi berkaitan dengan string, antara lain strcmp, strcpy, dan strlen. Hal ini karena string tidak memiliki tipe data secara langsung dalam bahasa pemrograman C. Situs www.duniailkom.com menyatakan, "String adalah tipe data yang menampung kumpulan karakter, seperti "aku", "kamu" atau "Duniailkom"" (Andre, 2018). Pada dasarnya, string dalam program C merupakan sebuah array yang terdiri atas beberapa karakter. Maka dari itu, diperlukan string.h untuk mempersingkat kode yang dibuat.

B. Struct

```
struct car {
        char carName[200], type[100];
        int seat;
        long int price;
        float rating;
};
```

```
struct date {
        int d, m, y;
};
```

```
struct carOrder {
        char userName[200], carName[400];
        struct date pickUp;
        struct date dropOff;
};
```

Sebelum membuat pernyataan dalam fungsi main, perlu dideklarasikan beberapa tipe data struct. Menurut www.petanikode.com, “Structure atau struct adalah kumpulan dari beberapa variabel dengan beragam tipe data yang dibungkus dalam satu variabel” (Muhardian, 2021). Struct yang pertama merupakan struct car berisikan atribut nama mobil (carName), tipenya (type), berapa bangku (seat), harga sewaan per hari (price), serta rating tiap mobil dengan skor tertinggi 5.0 (rating). Struct yang kedua merupakan struct date berisikan atribut tanggal (d), bulan (b), dan tahun (y). Struct terakhir adalah struct carOrder yang memiliki atribut-atribut nama penyewa (username), nama mobil (carName), struct tanggal sewa (struct date pickup), dan struct tanggal kembali (struct date dropOff).

C. Main Function

```c
int main(){
	short unsigned int choice;
	struct car selection [6];
	FILE *fp = fopen("./carSelectionData.txt","r");
	for(int i = 0; i<6; i++){
		fscanf(fp, "%[^#]#%[^#]#%d#%ld#%f\n", selection[i].carName, selection[i].type, &selection[i].seat, &selection[i].price, &selection[i].rating);
	}
	fclose(fp);
	header();
	do{
		menu();
		bool found = false;
		scanf("%d", &choice);
		getchar();
		(0 <= choice && choice <= 9)? printf(""): puts("The number can only be in the range of 0 to 9");

		switch(choice){
			case 1:{
				DisplayAll(selection);
				break;
			}
			case 2:{
				char x [200];
				struct car sResult;
				printf("Search Car: ");
				scanf("%[^\n]", x);
				sResult = SearchCar(selection, 0, 5, x);
				for(int i = 0; i<6; i++){
					if(strcmp(selection[i].carName, sResult.carName) == 0){
						found = true;
					}
				}
				if (found == true){
					puts("Found");
					printf("Car:                    %s\n", sResult.carName);
					printf("Type:                   %s\n", sResult.type);
					printf("Seats:                  %d\n", sResult.seat);
					printf("Daily Price:            %ld\n", sResult.price);
					printf("Rating:                 %.1f / 5.0\n", sResult.rating);
					printf("\n");
				}
				else
					puts("Not found");
				break;
			}
			case 3:{
				int choice2;
				int *c2;
				c2 = &choice2;
				do{
					case3Menu(c2);
					switch(choice2){
						case 1:{
							updateOrder();
							break;
						}
						case 2:{
							DisplayOrder(selection);
							break;
						}
					}
				}while(choice2 != 0);
				break;
			}
			case 4:{
				rentinput(selection);
				break;
			}
			case 5:{
				char c = 'O';
				while(c != 'Y' and c != 'N'){
					puts("Are you sure? (Y / N)");
					scanf("%c", &c);
					getchar();
				}
				if(c == 'Y'){
					dlete();
				}
				else if(c == 'N'){
					printf("");
				}
				break;
			}
			case 6:{
				splitName(selection, 0, 6-1);
				puts("Sorting successful!\nDisplay all selection to see results.");
				break;
			}
			case 7:{
				splitPrice(selection, 0, 6-1);
				puts("Sorting successful!\nDisplay all selection to see results.");
				break;
			}
			case 8:{
				DisplayFeedback();
				break;
			}
			case 9:{
				AddFeedback();
				break;
			}
		}
		border();
	}while(choice != 0);

	footer();
	return 0;
}
```

Untuk mengeksekusi program, tentu dibutuhkan fungsi main. Main merupakan "function utama yang akan menjadi kepala dari program dimana semua eksekusi pernyataan berawal" (Fajar, 2018). Dalam fungsi main program, terdapat beberapa fungsi yang menjalankan dekorasi pada header berupa ASCII art, menu, serta footer yang dijalankan apabila program berakhir.

Pada main function, perlu dideklarasikan array of structs dengan panjang 6 bernama struct car selection untuk menyimpan daftar mobil sewaan. Kemudian, dibuat pointer FILE *fp yang menunjuk ke file carSelectionData.txt dengan mode r untuk membuka txt file yang akan dibaca. Berikutnya, for loop diatur agar loop sebanyak 6 kali dan dalamnya diberi pernyataan fscanf untuk membaca isi txt file. Sesudah isi file dibaca, fclose dijalankan agar stream yang dilakukan oleh fopen berhenti.

Fungsi main juga terdapat kode pembangun menu utama program. Untuk menjalankan menu, perlu dibuat loop agar program tidak langsung berhenti ketika memberi input. Dalam program ini, do-while loop digunakan. Alasan digunakannya loop tersebut digunakan dibandingkan for atau while ialah karena do-while akan menjalankan pernyataan di dalamnya setidaknya sekali sehingga tidak ada kemungkinan bahwa menu tidak akan dijalankan. Ahmad Muhardian dari situs www.petanikode.com juga menulis bahwa do/while digunakan apabila syarat perulangan berkaitan dengan hasil perhitungan dalam blok kode (2021).

Dalam loop do-while, tentu dijalankan fungsi void main yang memunculkan menu kemudian pengguna aplikasi harus memberi input berupa angka 0-9 yang akan disetor ke variable choice. Ketika input selain yang sudah ditentukan, program akan menyuruh pengguna untuk memberi input lagi. Karena menu yang dijalankan akan membutuhkan banyak persyaratan terkait input pengguna, penggunaan if-else akan menjadi kurang efektif. Oleh karena itu, switch case diimplementasikan dan pada setiap case diberi perintah break. Setiawan Dimas menyatakan, "Setiap case harus diakhiri dengan perintah break tujuannya adalah memberitahu kepada program bahwa case sudah terpenuhi sehingga tidak perlu mengecek pada case berikutnya." (2020). Apabila angka nol diinput, pengguna akan keluar dari program. Jika program sukses, aplikasi akan mengembalikan nilai nol.

D. Displaying Selection of Cars

Apabila nilai choice sama dengan 1, switch case 1 akan dijalankan dan fungsi void DisplayAll akan dipanggil dengan parameter array struct car selection. Dalam fungsi tersebut, fungsi display dijalankan secara berulang melalui for loop sebanyak 6 kali agar

semua jenis mobil dapat dikeluarkan pada aplikasi. Ketika loop selesai dijalankan, fungsi void DisplayAll akan kembali ke switch case 1 kemudian menjalankan perintah break yang akan menghentikan pengecekan pada case-case berikutnya sehingga do-while akan looping dan aplikasi akan kembali pada menu.

```
void display(struct car selection[], int i){
        printf("Car:                    %s\n", selection[i].carName);
        printf("Type:                   %s\n", selection[i].type);
        printf("Seats:                  %d\n", selection[i].seat);
        printf("Daily Price:            %ld\n", selection[i].price);
        printf("Rating:                 %.1f / 5.0\n", selection[i].rating);
        printf("\n");
}
```

```
void DisplayAll(struct car selection[]){
        border();
        for(int i = 0; i<6; i++){
                display(selection, i);
        }
}
```

E. Search Car

```
struct car SearchCar(struct car selection[], int l, int r, char x[200]){
        struct car sResult;
        // search result
        while (l <= r){
                int mid = (l + r) / 2;
//              printf("%d %d %d \n", l, mid, r);
                if(strcmp(selection[mid].carName, x) == 0){
                        sResult = selection[mid];
                        return sResult;
                }

                if(strcmp(x, selection[mid].carName) < 0){
                        return SearchCar(selection, l, mid-1, x);
                }

                if(strcmp(x, selection[mid].carName) > 0){
                        return SearchCar(selection, mid+1, r, x);
                }
        }
        return sResult;
}
```

Jika nilai choice sama dengan 2, pengguna akan diminta untuk mengisi mobil yang ingin dicari. Dalam case 2, terdapat 2 variabel yang harus dideklarasikan terlebih dahulu, yakni bool found dan char x [200]. Pada fungsi scanf, perlu juga diberi notasi [^\n] untuk mengindikasikan bahwa scanf berhenti ketika menemukan escape character new line. Kemudian, struct car sResult dideklarasikan untuk menyetor bilangan yang akan dikembalikan oleh struct car SearchCar.

Pada fungsi struct car SearchCar, diperlukan parameter array struct car selection, int l yakni indeks paling bawah dalam array of structs tersebut, int r yakni indeks paling atas, serta char x [200] yakni hasil scan mobil yang ingin dicari. Terlebih dahulu, dideklarasikan struct car sResult untuk menyimpan rekor mobil yang dicari. SearchCar menggunakan algoritma binary search, yang didefinisikan sebagai “algoritma pencarian pada array/list dengan elemen terurut, yang dilakukan dengan memotong array menjadi dua bagian secara terus menerus hingga nilai yang dicari ditemukan” (Alza, 2020). Dengan ini, proses akan memakan waktu yang lebih sedikit, namun array yang diperiksa sudah harus terurut ketika dicek.

Dalam proses binary search ini, nilai l dan r akan diperiksa melalui while loop. Selama l kurang dari sama dengan r, loop akan tetap berjalan. Dalam loop, akan dibuat variable mid, yakni indeks tengah interval l-r, lalu dicek apakah string pada variable x sama dengan atribut carName pada indeks tengah array of struct selection melalui fungsi

strcmp (string compare). Berdasarkan redaksi.pens.ac.id, dinyatakan bahwa strcmp() akan menghasilkan bilangan:

1. Negatif, bila nilai ASCII string pertama kurang dari string kedua.
2. Nol, bila nilai ASCII string pertama sama dengan string kedua.
3. Positif, bila nilai ASCII string pertama lebih banyak dari string kedua. (Pramuja, 2019)

Apabila strcmp mengembalikan nilai nol, atribut-atribut sResult akan memiliki nilai sama dan sResult akan dikembalikan ke fungsi main. Jika nilai strcmp negatif, dikembalikan nilai fungsi SearchCar dengan interval baru dari indeks terendah array hingga indeks tengah dikurangi satu. Sebaliknya, jika nilai strcmp positif, dikembalikan nilai SearchCar dengan interval dari indeks tengah ditambah satu hingga indeks tertinggi array.

Ketika nilai SearchCar dikembalikan, sResult diperiksa lagi dengan strcmp antara indeks ke- i selection dan atribut carName pada sResult untuk memastikan bahwa rekor mobil yang dikembalikan bukan garbage value, melainkan berada pada array selection. Apabila syarat terpenuhi, found akan diberikan nilai true. Berikutnya, dilihat apakah nilai found true atau false. Apabila nilai found true, atribut-atribut sResult akan ditampilkan pada aplikasi. Namun sebaliknya, string “Not found” akan ditampilkan jika syarat tidak dipenuhi. Case 2 akan dihentikan dengan perintah break dan program akan kembali ke menu.

F. Shopping Cart

```c
void rentinput(struct car selection[]){
    bool inArray = false;
    struct carOrder rent;
    FILE *fp = fopen("./isiorderan.txt", "a");
    printf("Name: ");
    scanf("%[^\n]", rent.userName);
    getchar();
    printf("Car: ");
    scanf("%[^\n]", rent.carName);
    getchar();
    for (int i=0; i<6; i++){
        if(strcmp(selection[i].carName, rent.carName) == 0){
            inArray = true;
        }
    }
    if(inArray != true){
        puts("Car not available");
        fclose(fp);
        return;
    }
    printf("Pickup date (DD-MM-YYYY): ");
    scanf("%d-%d-%d", &rent.pickUp.d, &rent.pickUp.m, &rent.pickUp.y);
    printf("Dropoff date (DD-MM-YYYY) (Max. 7 days after pickup): ");
    scanf("%d-%d-%d", &rent.dropOff.d, &rent.dropOff.m, &rent.dropOff.y);
    printf("\n");
    fprintf(fp, "%s#%s#%d-%d-%d#%d-%d-%d\n", rent.userName, rent.carName,\
    rent.pickUp.d, rent.pickUp.m, rent.pickUp.y, rent.dropOff.d, rent.dropOff.m,\
    rent.dropOff.y);
    fclose(fp);
}
```

Pada shopping cart, terdapat beberapa opsi untuk mengaksesnya. Namun, sebelum melakukan perubahan padanya, shopping cart perlu diisi terlebih dahulu. Hal ini dapat ditempuh dengan memilih input 4 pada menu. Case 4 akan dijalankan dimana fungsi void rentinput akan dipanggil dengan parameter array of structs selection.

Dalam fungsi tersebut, akan dibuka file isiorderan.txt dengan mode append, kemudian pengguna akan diminta untuk mengisi nama, mobil, tanggal sewa, dan tanggal kembali. Seluruh atribut tersebut akan disetor dalam struct carOrder rent. Selain itu, juga dibuat loop untuk mengecek apakah mobil yang diinginkan terdapat pada array. Apabila syarat terpenuhi bool inArray akan bernilai true. Akan tetapi, apabila inArray bernilai false, teks “Car not available” akan dimunculkan, file akan ditutup, dan fungsi akan kembali ke main. Jika proses pengisian data sukses, seluruh atribut struct carOrder rent akan dicetak txt file, lalu program akan kembali ke menu.

```
void DisplayOrder(struct car selection[]){
        border();
        long int price;
        bool leap;
        int days;
        struct carOrder order;
        FILE *fptr = fopen(".\\isiorderan.txt","r");
        while(fscanf(fptr,"%[^#]#%[^#]#%d-%d-%d#%d-%d-%d\n", order.userName, order.carName,\
        &order.pickUp.d, &order.pickUp.m, &order.pickUp.y, &order.dropOff.d, &order.dropOff.m,\
        &order.dropOff.y) != EOF){
                for(int i=0; i<6; i++){
                        if(strcmp(selection[i].carName, order.carName) == 0){
                                price = selection[i].price;
                                if (order.pickUp.y % 4 == 0){
                                        leap = true;
                                }
                                if (order.pickUp.m == 2 && order.dropOff.m % 2 != 0 && leap){
                                        days = (29 % order.pickUp.d + order.dropOff.d + 1);
                                }
                                else if (order.pickUp.m == order.dropOff.m){
                                        days = (order.dropOff.d - order.pickUp.d + 1);
                                }
                                else{
                                        if (1 <= order.pickUp.m && order.pickUp.m <= 7){
                                                if(order.pickUp.m % 2 == 0 && order.dropOff.m % 2 != 0){
                                                        days = (30 % order.pickUp.d + order.dropOff.d + 1);
                                                }
                                                else if (order.pickUp.m % 2 != 0 && order.dropOff.m % 2 == 0){
```

Sekarang, pengguna sudah bisa menampilkan ordernya dengan menginput 3 pada menu paling depan, lalu menginput angka 2. Fungsi void DisplayOrder akan dipanggil. Dalam fungsi tersebut, isiorderan.txt akan dibuka dengan mode read untuk membaca seluruh isi file. Selama isi file yang dibaca belum menyampai akhir file, program akan mencari harga per hari yang sesuai dengan mobil sewaan yang dipilih dan mengkalkulasi total biaya yang perlu dibayar oleh nama pemesan terkait.

```
void updateOrder(){
        border();
        char x[200];
        printf("Search username: ");
        scanf("%[^\n]", x);
        getchar();
        FILE *fp = fopen ("./isiorderan.txt", "r");
        FILE *fp2 = fopen ("./tampungansementara.txt", "w");
        struct carOrder car;
        while (fscanf(fp, "%[^#]#%[^#]#%d-%d-%d#%d-%d-%d\n",car.userName, car.carName, &car.pickUp.d, &car.pickUp.m, &car.pickUp.y, &car.dropOff.d, &ca
                if (strcmp(x, car.userName) == 0){
                        printf("Enter new pickup date: ");
                        scanf("%d-%d-%d", &car.pickUp.d, &car.pickUp.m, &car.pickUp.y);
                        printf("Enter new dropoff date: ");
                        scanf("%d-%d-%d", &car.dropOff.d, &car.dropOff.m, &car.dropOff.y);
                }
                fprintf(fp2, "%s#%s#%d-%d-%d#%d-%d-%d\n", car.userName, car.carName, car.pickUp.d, car.pickUp.m, car.pickUp.y, car.dropOff.d, car.dropO
        }
        fclose(fp);
        fclose(fp2);


        fp = fopen ("./isiorderan.txt", "w");
        fp2 = fopen ("./tampungansementara.txt", "r");

        while (fscanf(fp2, "%[^#]#%[^#]#%d-%d-%d#%d-%d-%d\n",car.userName, car.carName, &car.pickUp.d, &car.pickUp.m, &car.pickUp.y, &car.dropOff.d, &c
                fprintf(fp, "%s#%s#%d-%d-%d#%d-%d-%d\n", car.userName, car.carName, car.pickUp.d, car.pickUp.m, car.pickUp.y, car.dropOff.d, car.dropOf
        }
        fclose(fp);
        fclose(fp2);
}
```

Di samping itu, program juga akan melihat apakah tanggal sewa dan kembali valid atau tidak. Jika hari sewa melebihi 7 hari, price akan bernilai 0 dan sebuah peringatan ditampilkan di aplikasi. Apabila hal tersebut terjadi, pengguna dapat melakukan update tanggal dengan menginput 1 pada menu case 3.

Dalam fungsi void updateOrder, pengguna akan diminta untuk mengisi usernamenya terlebih dahulu. Berikutnya, file isiorderan.txt dan tampungansementara.txt akan dibuka masing-masing dengan mode read dan write. Program akan membaca seluruh isi isiorderan.txt dan mencetaknya pada tampungansementara.txt. Apabila string username sesuai dengan username yang terdaftar, pengguna diminta menginput tanggal sewa dan tanggal kembali baru. Atribut tanggal sewa dan tanggal kembali yang awalnya diisi hasil baca dari isiorderan.txt akan diganti dengan hasil input baru sehingga hal itulah yang akan dicetak pada tampungansementara.txt. Kedua file akan ditutup dulu.

Lalu dengan pointer yang sama dibuka lagi kedua file tersebut, kali ini dengan mode yang ditukar. Sementara isi tampungansementara.txt dibaca, fungsi fprintf akan mencetak hasil baca ke isiorderan.txt. Fungsi akan berakhir dengan kedua file ditutup dan kembali ke menu case 3.

```
void dlete (){
        FILE *fp = fopen ("./isiorderan.txt", "w");
        fclose(fp);
}
```

Apabila pengguna ingin menghapus seluruh rekor sewa yang ada, dapat menginput 5 pada menu paling awal dimana pengguna akan ditanya apakah mereka ingin menghapus seluruh rekor. Jika mereka mengisi N, program akan kembali ke menu paling depan. Namun, jika mereka isi Y, program akan memanggil fungsi void dlete. Di dalam fungsi ini, isiorderan.txt akan dibuka dengan mode write, secara langsung seluruh isi file akan dihapus. Namun, karena tidak ada data yang mau dicetak ke file dan file ditutup, file isiorderan.txt tidak akan ada isi.

G. Sort Cars

Dalam program “Renture”, terdapat dua opsi sorting, namun keduanya menggunakan algoritma yang sama, yakni merge sort. Berdasarkan artikel Abidin, dinyatakan sebagai berikut.

> Metode ini menggunakan konsep devide and conquer yang membagi data S dalam dua kelompok yaitu S1 dan S2 yang tidak beririsan (disjoint). Proses pembagian data

dilakukan secara rekursif sampai data tidak dapat dibagi lagi atau dengan kata lain data dalam sub bagian menjadi tunggal. Setelah data tidak dapat dibagi lagi, proses penggabungan (merging) dilakukan antara sub-sub bagian dengan memperhatikan urutan data yang diinginkan (2016: 2).

Apabila input choice bernilai 6, daftar mobil akan diurutkan secara menurun dan hasil dapat dilihat dengan menuju opsi Display All Selections. Demikian juga dengan opsi Sort by Price yang akan mengurutkan daftar mobil mulai dari harga termurah dan dapat diakses dengan menginput angka 7 pada menu.

```
void splitPrice(struct car selection[], int l, int r){
        if (l >= r){
                return;
        }
        // when the low limit exceeds or is equal to the high limit, \
        the function will return the accumulated results
        int mid = (l + r) / 2;
        // divide and conquer
        splitPrice(selection, l, mid);
        splitPrice(selection, mid+1, r);
        // recursions used to repeatedly divide the array, \
        then returning the elements of the given array in batches
        mergePrice(selection, l, mid, r);
        // the function will run with the final values of l, mid, r \
        before the if condition is fulfilled (0 0 1) or (6 6 7)
}
```

```
void splitName(struct car selection[], int l, int r){
        if (l >= r){
                return;
        }
        // when the low limit exceeds or is equal to the high limit, \
        the function will return the accumulated results
        int mid = (l + r) / 2;
        // divide and conquer
        splitName(selection, l, mid);
        splitName(selection, mid+1, r);
        // recursions used to repeatedly divide the array, \
        then returning the elements of the given array in batches
        mergeName(selection, l, mid, r);
        // the function will run with the final values of l, mid, r \
        before the if condition is fulfilled (0 0 1) or (6 6 7)
}
```

Keduanya melalui proses yang sangat identik, namun ada perbedaan sedikit dalam menata elemen array. Pertama, array dipotong-potong menjadi segmen lebih kecil melalui fungsi void splitName apabila Sort by Name atau fungsi splitPrice apabila Sort by Price.

Dengan rekursi, mulai dari segmen array paling kecil dimasukkan ke dalam fungsi mergeName atau mergePrice dimana segmen-segmen tersebut akan dipisahkan ke dalam dua bagian lalu mereka dibandingkan dan ditata kembali ke array awal.

```
void mergePrice(struct car selection[], int l, int mid, int r){
        int Lsize = mid - l + 1;
        int Rsize = r - mid;

        struct car Larray [Lsize];
        struct car Rarray [Rsize];

        for (int i=0; i<Lsize; i++){
                Larray[i] = selection[i + l];
                // the array index must be added by l \
                to adapt to new inputs from the recursion
        }

        for (int i=0; i<Rsize; i++){
                Rarray[i] = selection[i + mid + 1];
        }
```

```
void mergeName(struct car selection[], int l, int mid, int r){
        int Lsize = mid - l + 1;
        int Rsize = r - mid;

        struct car Larray [Lsize];
        struct car Rarray [Rsize];

        for (int i=0; i<Lsize; i++){
                Larray[i] = selection[i + l];
                // the array index must be added by l \
                to adapt to new inputs from the recursion
        }

        for (int i=0; i<Rsize; i++){
                Rarray[i] = selection[i + mid + 1];
        }
```

H. Feedback

```c
void DisplayFeedback(){
        border();
        float rating;
        char comment[151];
        FILE *fptr = fopen(".\\feedback.txt","r");
        float sum = 0;
        int count = 0;
        while(fscanf(fptr, "%f#%[^\n]\n", &rating, comment) != EOF){
                count++;
                sum += rating;
                printf("Rating: %.1f\n", rating);
                printf("Comment: %s\n\n", comment);
        }
        printf("The average rating: %.1f / 5.0\n", sum/count);
        fclose(fptr);
}
```

Ketika pengguna memberi input 8, aplikasi akan menuju ke fitur Display Feedback. Dalam fitur ini, pengguna dapat melihat komen-komen dari pengguna lain. Dalam program, fungsi void DisplayFeedback akan dijalankan. Dalam fungsi ini, program akan membaca data rating dan comment dari feedback.txt. Hal ini dapat ditempuh melalui fopen dengan mode r beserta while loop dengan argument fscanf hingga akhir file txt. Selama looping, rating dan comment dicetak ke aplikasi serta seluruh rating dijumlahkan dan disimpan dalam variable sum serta dibuat counter untuk menghitung banyak rating yang ada. Setelah keluar loop, rata-rata rating akan dicetak dengan satu angka di belakang koma dan fopen dihentikan. Switch case akan dihentikan break dan program akan kembali ke menu.

```
void AddFeedback(){
	border();
	float rating;
	char comment[151], x[151];
	FILE *fptr = fopen(".\\feedback.txt","a");
	printf("Rating ( / 5.0 ): ");
	scanf("%f", &rating);
	getchar();
	printf("Comment (max. 150 char): ");
	scanf("%[^\n]", x);
	getchar();
	strcpy(comment, x);
	if(strlen(comment) > 100){
		puts("Don't get too carried away there, buddy");
	}
	fprintf(fptr, "%.1f#%s\n", rating, comment);
	fclose(fptr);
}
```

Apabila input bernilai 9, pengguna dapat menambah input rating dan komen. Pertama, program akan mengakses AddFeedback yang membuka feedback.txt kali ini dengan mode append. Menurut Pann (2019), append adalah "Perintah dalam aplikasi pengolahan file untuk menambahkan sebuah record baru pada suatu file atau menggabungkan file baru dengan file lama." Keunggulan append daripada write terdapat pada cara mereka menambah isi file. Buku *Algoritma dan Pemrograman* menyatakan, "Dengan metode write, isi file yang lama akan dihapus dan diganti dengan isi yang baru. Adapun dengan cara append, isi file yang lama akan ditambahkan dengan isi file yang baru" (2020: 211).

Pengguna akan diminta untuk mengisi rating dan comment maksimal 150 karakter. Setiap fungsi scanf perlu diberi getchar agar standar input berupa new line tidak mempengaruhi fungsi scanf lainnya. Kemudian, comment yang awalnya disimpan dalam variable x akan dipindah ke variable comment. Apabila comment melebihi 100 karakter, pengguna akan dapat peringatan setelah mengirimnya. Terakhir, rating dan comment akan dicetak ke file txt dan fopen dihentikan.

BAB III
PENUTUP

Dengan penutup laporan dokumentasi ini, penulis memiliki saran agar bagi pemrogram selanjutnya agar lain kali melakukan penelitian pada setiap fungsi terlebih dahulu sebelum menggabungkannya dalam program akhir. Selain itu, bagi pengguna, penulis menyarankan untuk mendalami tentang pemrograman supaya memperoleh kejelasan dalam jalannya aplikasi "Renture" serta diharapkan pengguna menyampaikan kritik dan saran konstruktif guna perbaikan program oleh para *developer*.

DAFTAR PUSTAKA

Alza. 2020. *Algoritma binary search*, (Online), (https://koding.alza.web.id/algoritma-binary-search/), diakses 6 September 2021

Andre. 2018. *Tutorial Belajar C Part 20: Pengertian dan Contoh Kode Program Tipe Data String,* (Online), (https://www.duniailkom.com/tutorial-belajar-c-pengertian-dan-contoh-kode-program-tipe-data-string/), diakses 5 September 2021

Abidin, Taufik Fuadi dan Irvanizam Zamanhuri. 2016. *Metode Pengurutan Merge Sort,* (Online), (http://www.informatika.unsyiah.ac.id/tfa/ds/mergesort.pdf), diakses 6 September 2021

Dimas, Setiawan. 2020. *Contoh Program Switch Case Pada C++,* (Online), (https://kelasprogrammer.com/contoh-program-switch-case-pada-c/), diakses 6 September 2021

Fajar. 2018. *Pengertian Function*, (Online), (https://www.belajarcpp.com/tutorial/cpp/function/), diakses 5 September 2021

Karimishaq. 2019. *Header dan Library,* (Online), (https://www.codelogi.com/2019/06/header.html), diakses 5 September 2021

Kusuma, Purba Daru. 2020. *Algoritma dan Pemrograman.* Yogyakarta: Deepublish

Muhardian, Ahmad. 2021. *Belajar Pemrograman C #13: Mengenal Tipe Data Struct,* (Online), (https://www.petanikode.com/c-struct/), diakses 5 September 2021

Muhardian, Ahmad. 2021. *Perbedaan Perulangan While dan Do/While,* (Online), (https://www.petanikode.com/perbedaan-perulangan-while-dan-do-while/), diakses 6 September 2021

Pann. 2019. *append – (Teknologi Informasi),* (Online), (https://glosarium.org/arti-append-di-komputer/), diakses 6 September 2021

Pramuja, Alex Putra Bagus. 2019. *STRING PADA BAHASA C,* (Online), (https://redaksi.pens.ac.id/2019/10/31/string-pada-bahasa-c/), diakses 6 September 2021

Wildan. 2018. *cstdio (stdio.h),* (Online), (https://www.belajarcpp.com/referensi/cstdio/), diakses 5 September 2021

This page is intentionally left blank.