

1. Jelaskan perbedaan dari masing – masing algoritma sort berikut!

**a. Linear search**

Linear search merupakan algoritma untuk searching. Linear search menggunakan teknik pencarian di mana kita mulai menelusuri semua data satu per satu, dari awal sampai akhir data dan memeriksa setiap elemennya hingga menemukan elemen yang diinginkan. Algoritma linear search ini adalah algoritma searching yang paling sederhana dan disebut juga dengan bruteforce, karena dicoba satu satu sampai menemukan data yang diinginkan.

Algoritma ini mempunyai **best case time complexitynya adalah  $O(1)$**  jika data yang dicari adalah data pertama dan algoritma ini mempunyai **worst case time complexitynya adalah  $O(n)$**  di mana  $n$  adalah banyaknya data.

**b. Binary search**

Binary search merupakan algoritma untuk searching yang menggunakan divide and conquer. Cara kerja binary search ini adalah dengan cara berulang kali membagi separuh dari jumlah data yang dicari sehingga memperkecil lokasi pencarian menjadi satu data. Tetapi binary search ini bisa berjalan dan berfungsi jika array tersebut udah tersortir. Misalnya kita mempunyai array  $=\{2, 5, 10, 3, 1, 7, 8\}$  dan ingin mencari angka 3, lalu kita sortir array tersebut menjadi  $\{1, 2, 3, 5, 7, 8, 10\}$ .

1. Iterasi Pertama -> kita memilih mid nya yaitu 5, lalu kita lihat apakah 5 lebih besar dari 3? Iya lalu kita ambil array sebelah kiri dari 5 dan membuang sebelah kanan array dari 5. Sehingga array menjadi  $\{1, 2, 3, 5\}$ .
2. Iterasi Kedua -> kita memilih mid nya yaitu 3, ternyata 3 adalah yang dicari maka binary search ini akan mereturn value index dari 3 yaitu index ke 2.

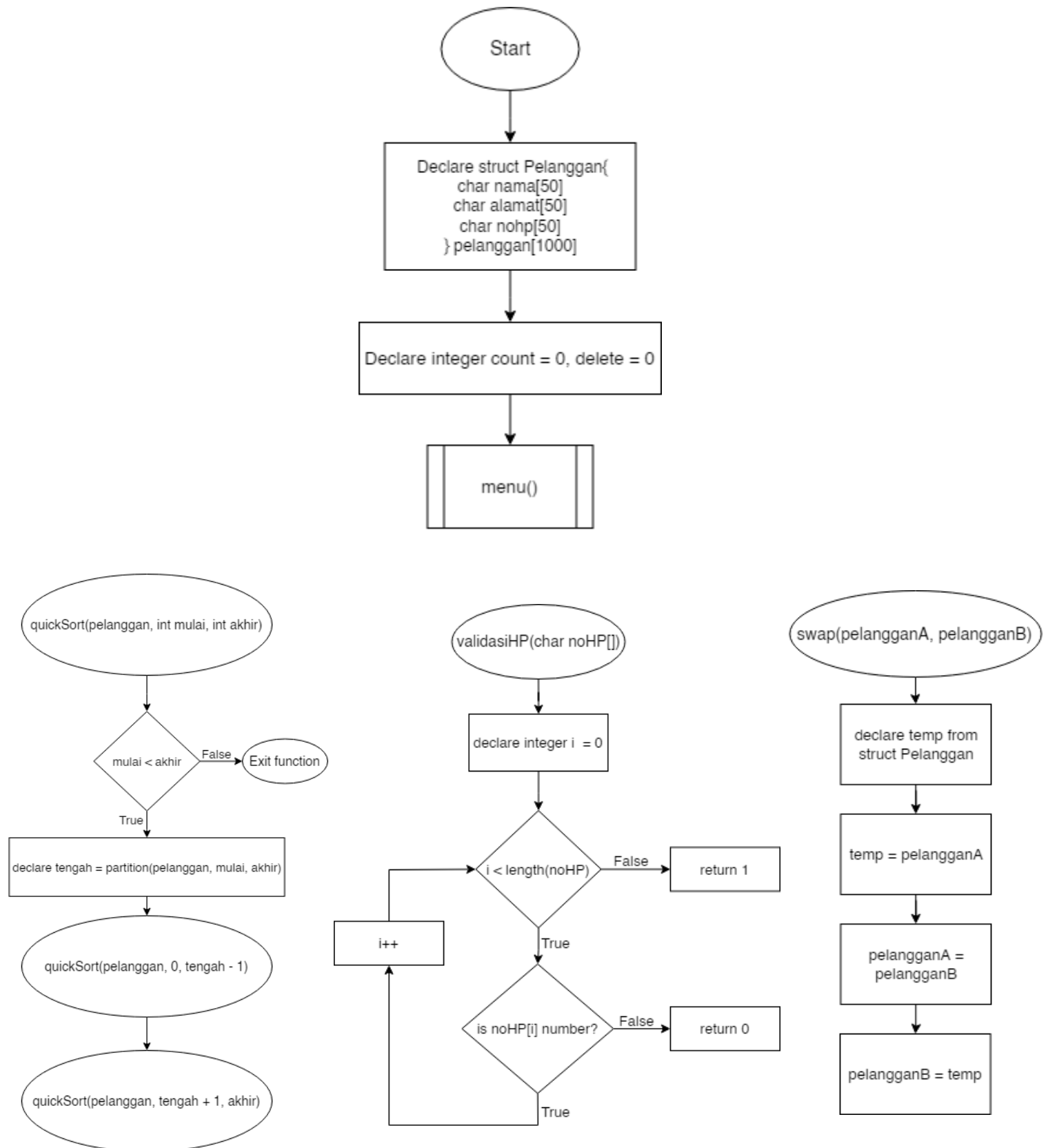
Algoritma ini mempunyai **best case time complexitynya adalah  $O(1)$**  dan mempunyai **average dan worst case complexitynya adalah  $O(\log n)$**  dimana  $n$  adalah banyaknya data. Hal ini membuat binary search adalah algoritma searching yang optimal untuk dipakai karena kecilnya time complexity tersebut.

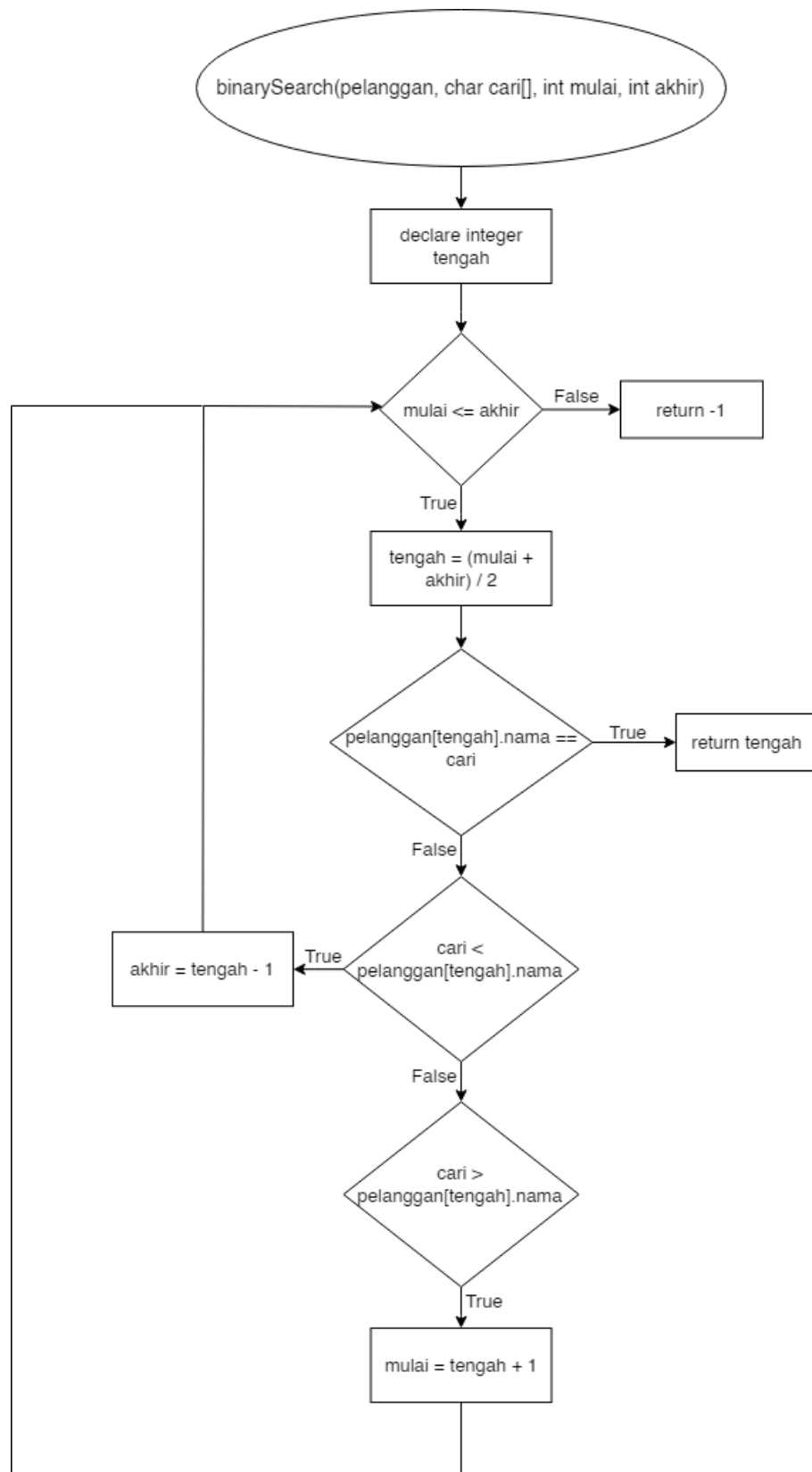
**c. Interpolation search**

Interpolation search merupakan algoritma untuk searching yang merupakan pengembangan dari binary search. Jika binary search akan selalu memeriksa nilai tengah dari setiap array, sedangkan interpolation search dapat pergi ke lokasi yang berbeda berdasarkan key yang didapat. Jika nilai key lebih dekat ke array yang terakhir, maka teknik interpolation search akan memulai pencarian dari array yang terakhir. Nilai mid ini mempunyai rumus seperti berikut :

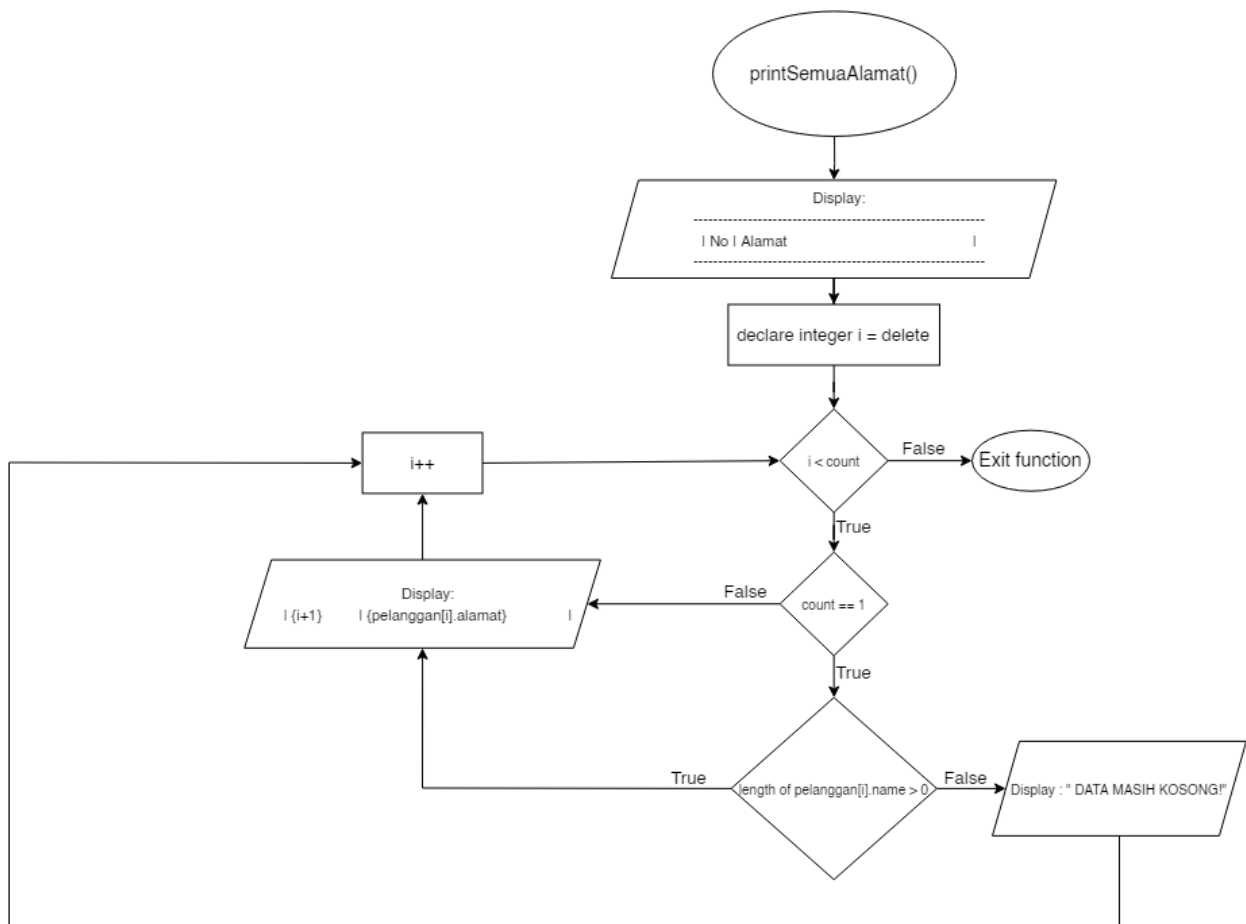
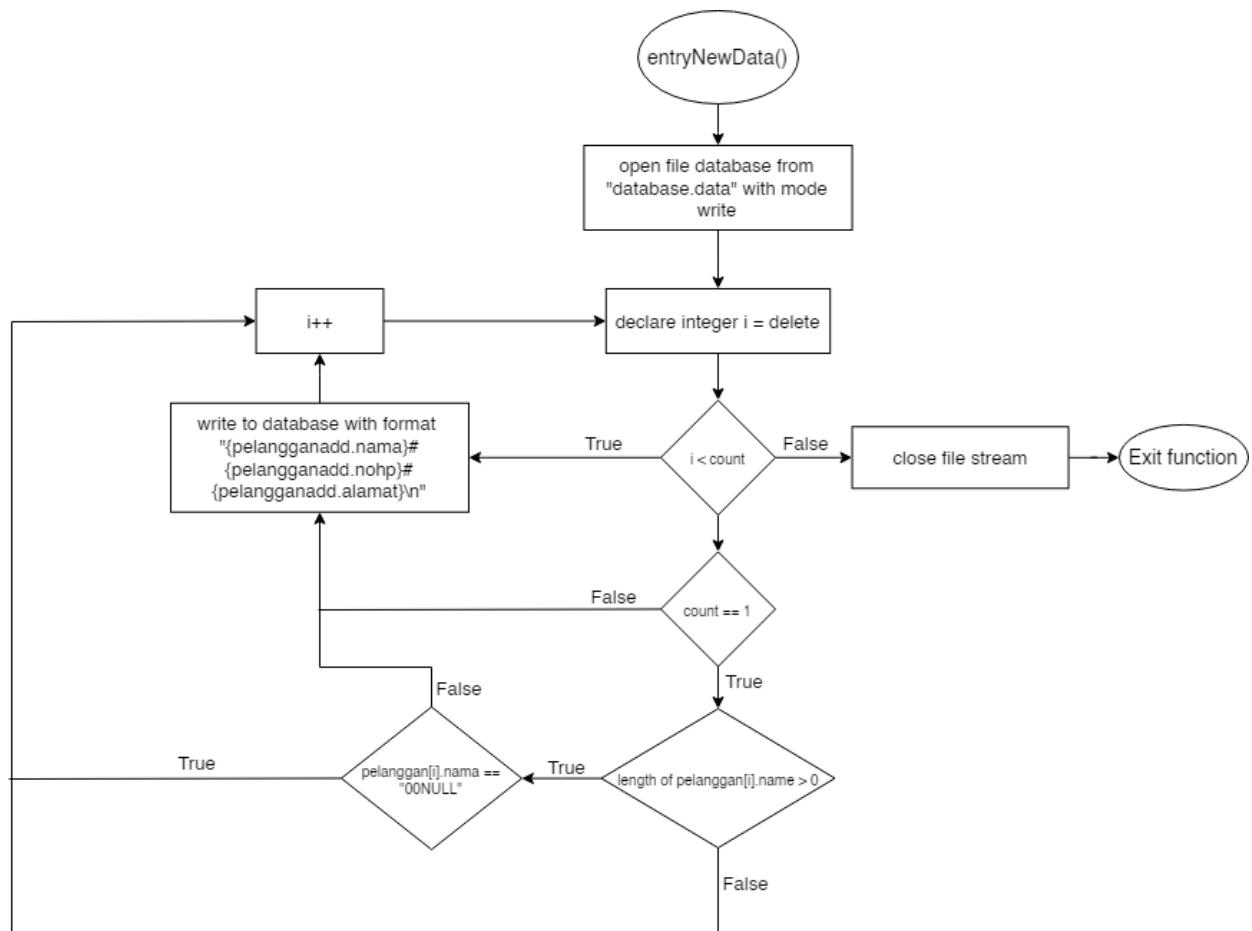
$$\text{Mid} = \frac{\text{kunci} - \text{data}[\text{min}]}{\text{data}[\text{max}] - \text{data}[\text{min}]} \times (\text{max} - \text{min}) + \text{min}$$

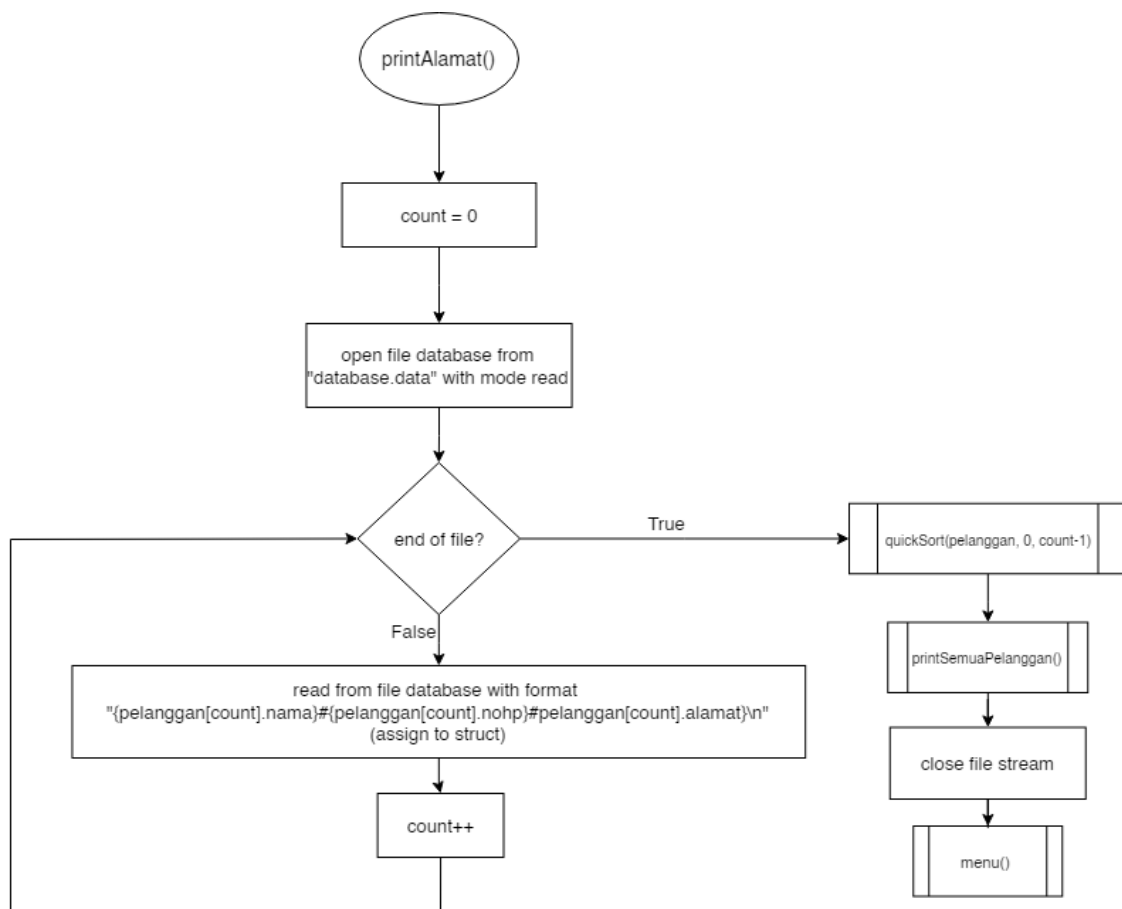
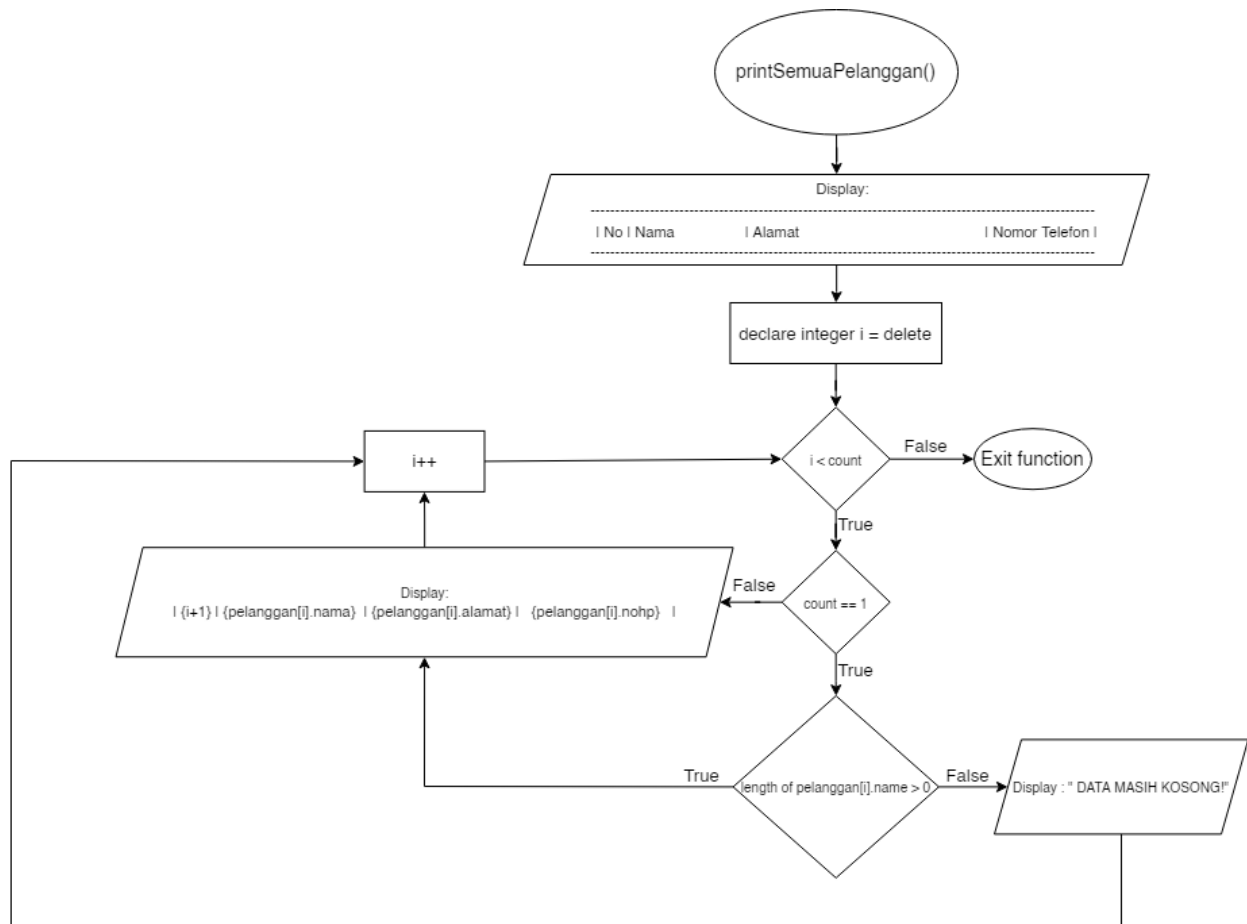
2. Desain algoritma dalam bentuk flowchart

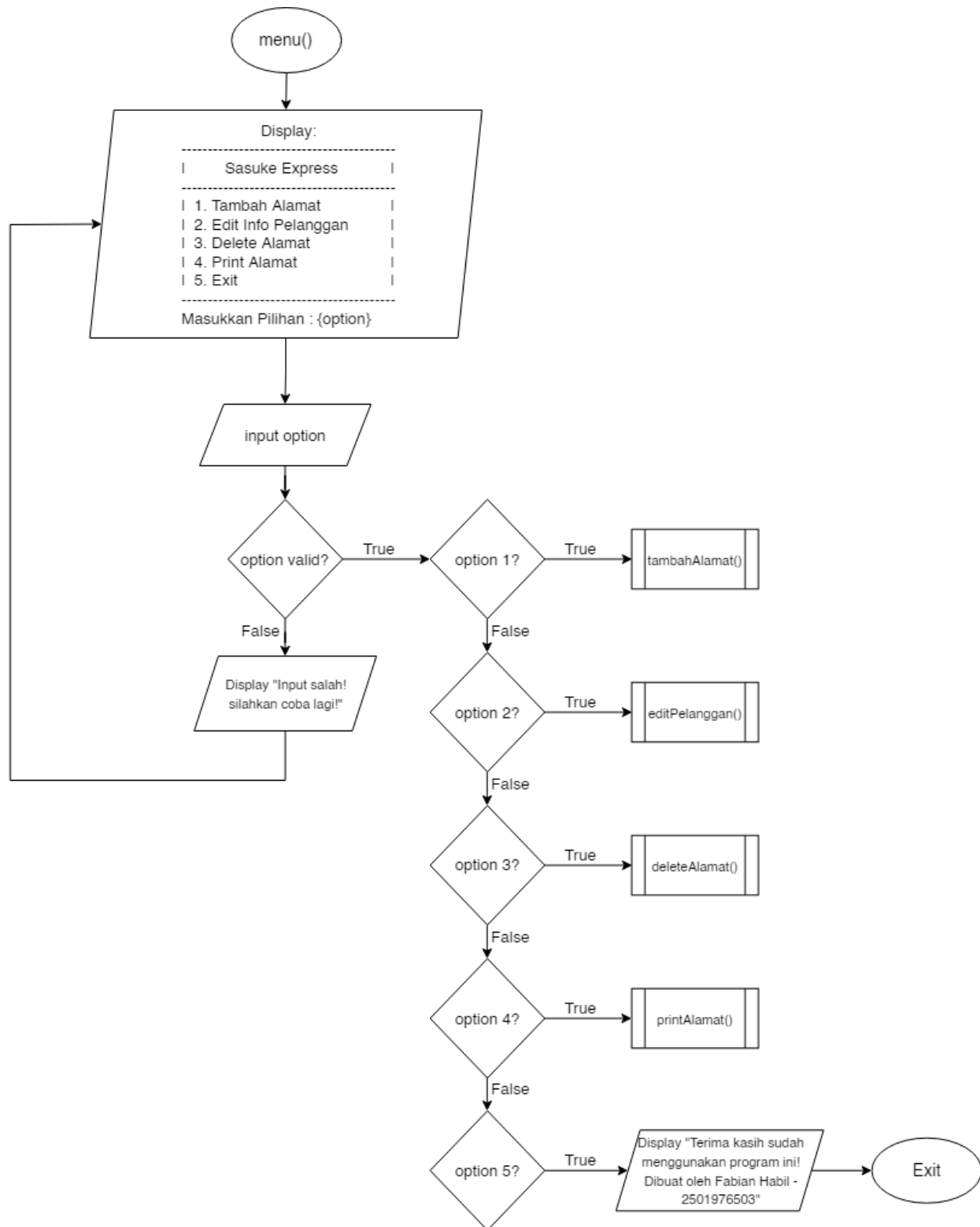




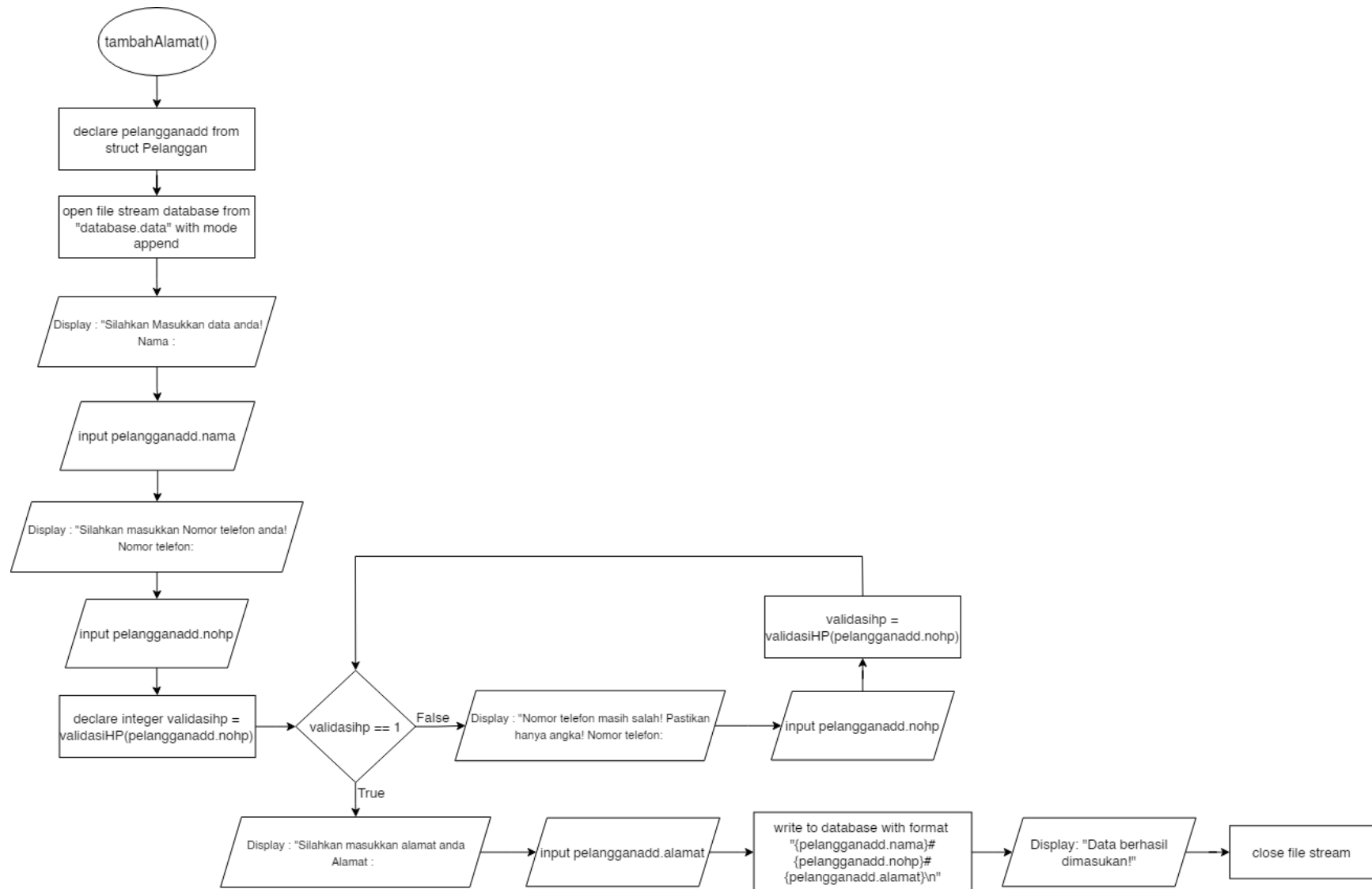




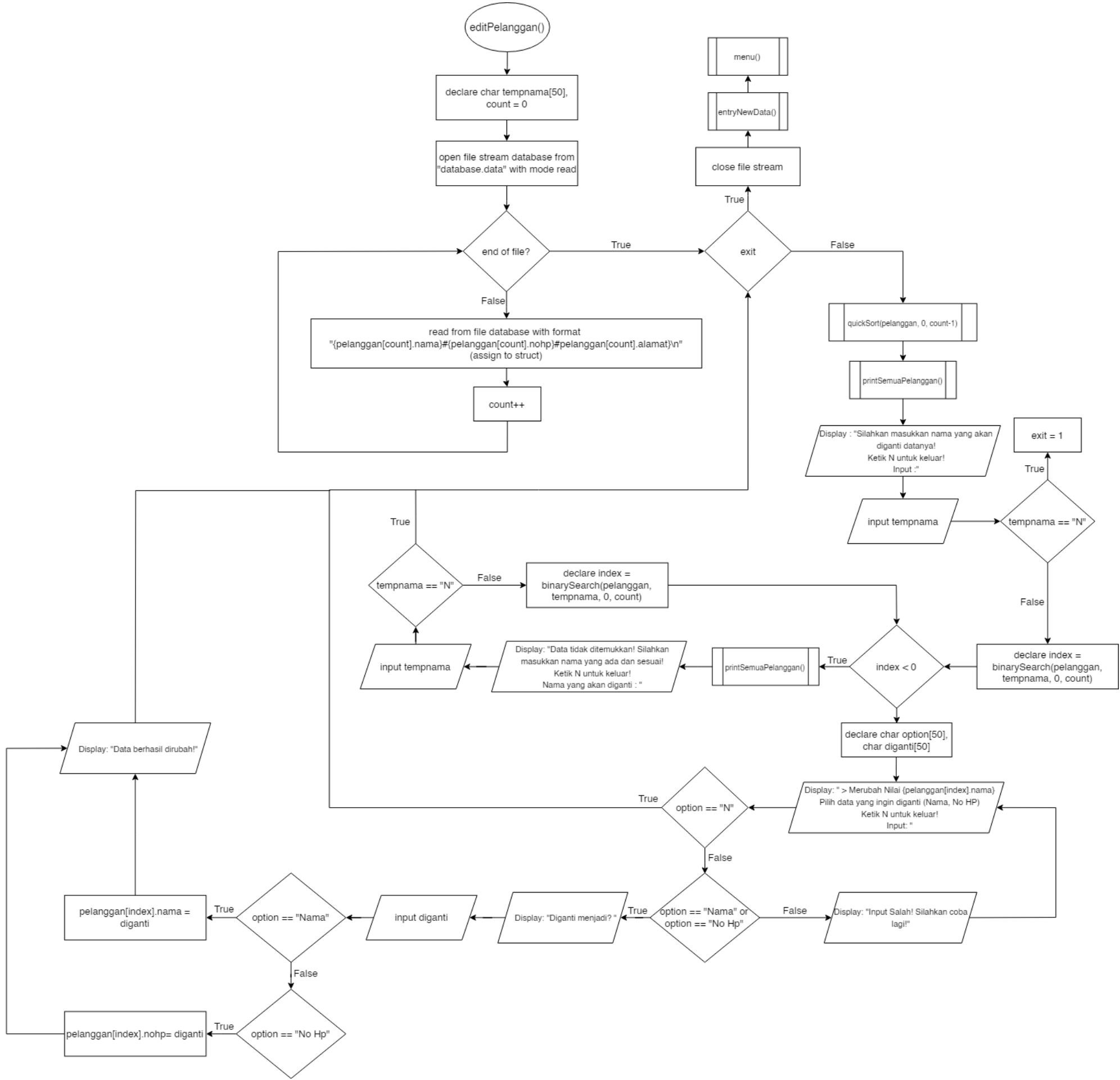




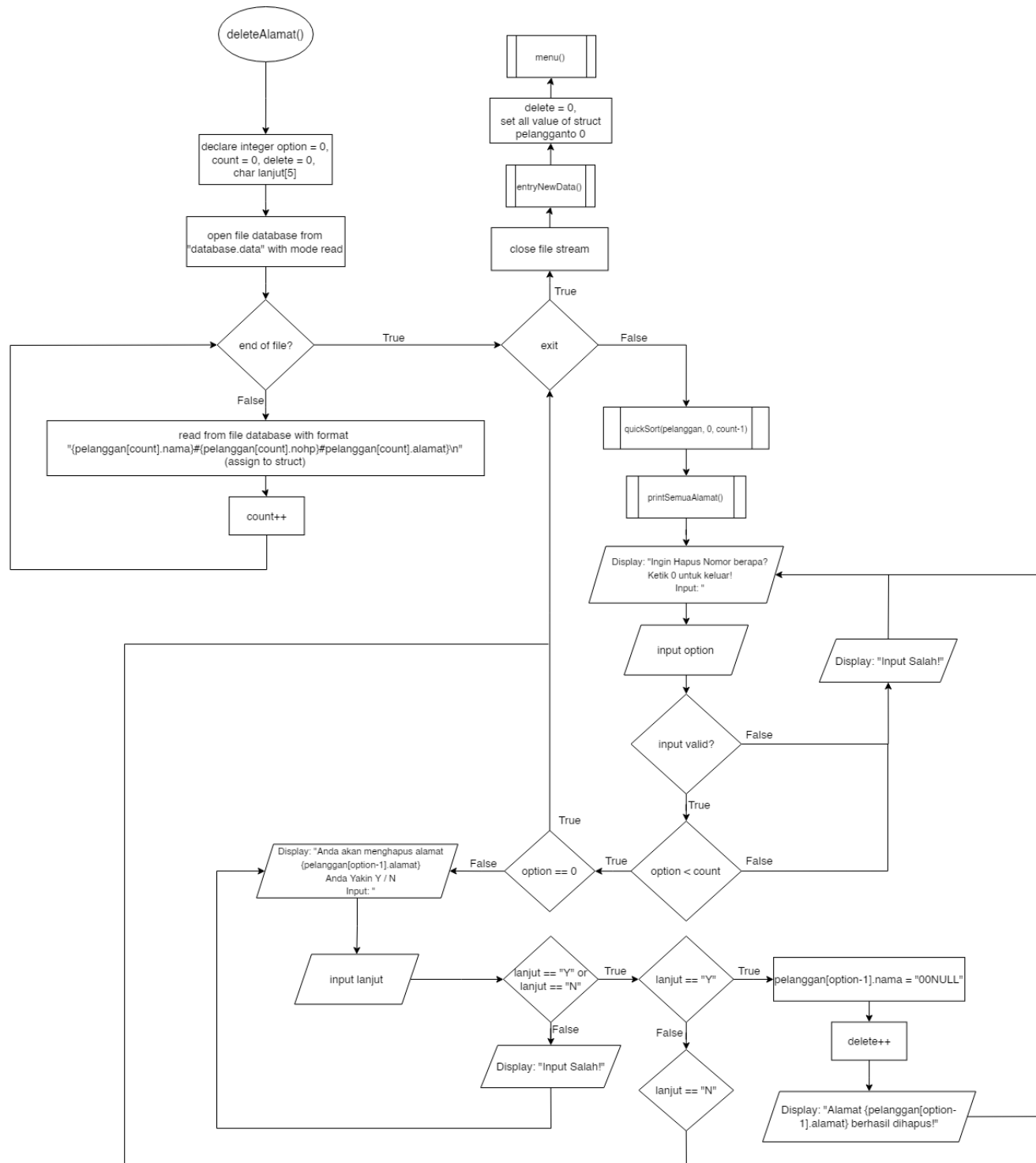
Algorithm & Programming – SasukeExpress Program – Fabian Habil Ramdhan – 2501976503 – LB75







# Algorithm & Programming – SasukeExpress Program – Fabian Habil Ramdhan – 2501976503 – LB75



## 3. Implementasi menggunakan bahasa C

```

1. #include <stdio.h>
2. #include <stdlib.h>
3. #include <string.h>
4.
5. // 2501976503 - Fabian Habil Ramdhan - LB75 - SasukeExpress Assignment
6. // Algoritma Searching -> Binary Search
7. // Algoritma Sorting -> Quick Sort
8.
9. // Deklarasi Struct dan Variabel Global.
10. // Struct untuk menyimpan data pelanggan dengan length 1000 data.
11. typedef struct pelanggan{
12.     char nama[50];
13.     char alamat[50];
14.     char nohp[50];
15. }Pelanggan;
16. Pelanggan pelanggan[1000];
17.
18. // Variabel count untuk menghitung sekarang ada berapa jumlah data di database.
19. int count = 0;
20. // Variabel delete untuk menghitung ada berapa banyak data yang dihapus.
21. int delete = 0;
22.
23. // Fungsi dan Prosedur prototype.
24. void menu();
25. void menuOption(unsigned short option);
26. void enterToContinue();
27. int validasiHP(char noHP[]);
28. int binarySearch(Pelanggan pelanggan[], char cari[], int mulai, int akhir);
29. void quickSort(Pelanggan pelanggan[], int mulai, int akhir);
30. int partition(Pelanggan pelanggan[], int mulai, int akhir);
31. void swap(Pelanggan *pelangganA, Pelanggan *pelangganB);
32. void entryNewData();
33. void printSemuaPelanggan();
34. void printSemuaAlamat();
35. void tambahAlamat();
36. void editPelanggan();
37. void deleteAlamat();
38. void printAlamat();
39.
40. // Fungsi main untuk menjalankan program.
41. int main(){
42.     system("cls");
43.
44.     /* Open file stream saat pertama kali menjalankan program, jika tidak ada file,
       maka akan dibuat
45.     dan jika masih error, mengeluarkan file tidak ditemukan.*/
46.     FILE *fp = fopen("database.data", "a");
47.     if(fp == NULL) printf("File tidak ditemukan!\n");
48.     fclose(fp);
49.     menu();
50. }
51.
52. // Prosedur untuk menampilkan utama dan memilih fitur yang ada.
53. void menu(){
54.     unsigned short exit = 0;
55.     unsigned short option = 0;
56.     unsigned short valid = 0;
57.     do{

```

```

58.      // Output Menu
59.      puts("-----");
60.      puts("|           Sasuke Express           |");
61.      puts("-----");
62.      puts("| 1. Tambah Alamat                    |");
63.      puts("| 2. Edit Info Pelanggan              |");
64.      puts("| 3. Delete Alamat                   |");
65.      puts("| 4. Print Alamat                    |");
66.      puts("| 5. Exit                           |");
67.      puts("-----");
68.      printf("Masukkan Pilihan: ");
69.
70.      /* Mendapatkan Input dari user, valid akan menjadi 1 jika input user benar
71.      namun jika input user salah, valid akan menjadi 0 */
72.      valid = scanf("%hu", &option);
73.      getchar();
74.
75.      // Jika input benar, maka akan dicek.
76.      if(valid){
77.          // Jika diantara 1 dan 4 akan dihandle oleh procedure menuOption
78.          if(option >= 1 && option <=4){
79.              system("cls");
80.              menuOption(option);
81.          }
82.          // Jika option 5, maka akan keluar dari program.
83.          else if(option == 5){
84.              system("cls");
85.              puts("Terima kasih sudah menggunakan program ini!");
86.              puts("Dibuat oleh Fabian Habil - 2501976503");
87.              exit = 1;
88.          }
89.          // Jika input angka tetai masih salah, akan dikasih tau masih salah dan
90.          dipaksa untuk input lagi
91.          else{
92.              system("cls");
93.              puts("Input salah! silahkan coba lagi!");
94.          }
95.          else{
96.              system("cls");
97.              puts("Input salah! silahkan coba lagi!");
98.          }
99.      }while(!exit);
100.     }
101.
102.     // Prosedur untuk handle input dan switch ke fungsi menu sesuai input user.
103.     void menuOption(unsigned short option){
104.         switch(option){
105.             case 1:
106.                 tambahAlamat();
107.                 break;
108.             case 2:
109.                 editPelanggan();
110.                 break;
111.             case 3:
112.                 deleteAlamat();
113.                 break;
114.             case 4:
115.                 printAlamat();
116.                 break;
117.         }

```

```

118.     }
119.
120.     // Prosedur untuk membuat prompt agar user menekan enter terlebih dahulu lalu
    melanjutkan program.
121.     void enterToContinue(){
122.         printf("Tekan Enter untuk melanjutkan.");
123.         getchar();
124.     }
125.
126.     /* Fungsi untuk validasi Nomor hp, kita akan loop dari awal string sampai akhir
127.     dan cek jika ada selain angka akan mereturn value 0 (false), namun jika tidak
    ditemukan
128.     selain angka, akan mereturn value 1 (true) */
129.     int validasiHP(char noHP[]){
130.         for(int i = 0; i < strlen(noHP); i++){
131.             if(noHP[i] >= '0' && noHP[i] <= '9'){
132.                 continue;
133.             }
134.             else{
135.                 return 0;
136.             }
137.         }
138.         return 1;
139.     }
140.
141.     // Fungsi untuk mencari data, menggunakan Binary Search, jika tidak ditemukan
    maka akan return -1
142.     int binarySearch(Pelanggan pelanggan[], char cari[], int mulai, int akhir){
143.         int tengah;
144.         while(mulai <= akhir){
145.             tengah = (mulai + akhir) / 2;
146.             if(strcmp(cari, pelanggan[tengah].nama) == 0){
147.                 return tengah;
148.             }
149.             if(strcmp(cari, pelanggan[tengah].nama) < 0){
150.                 akhir = tengah - 1;
151.             }
152.             if(strcmp(cari, pelanggan[tengah].nama) > 0){
153.                 mulai = tengah + 1;
154.             }
155.         }
156.         return -1;
157.     }
158.
159.     // Fungsi untuk swap struct
160.     void swap(Pelanggan *pelangganA, Pelanggan *pelangganB){
161.         Pelanggan temp = *pelangganA;
162.         *pelangganA = *pelangganB;
163.         *pelangganB = temp;
164.     }
165.
166.     // Fungsi untuk partisi dan sortir untuk quicksort
167.     int partition(Pelanggan pelanggan[], int mulai, int akhir){
168.         Pelanggan pivot = pelanggan[akhir];
169.         int i = mulai - 1;
170.
171.         // Sorting berdasarkan nama
172.         for(int j = mulai; j <= akhir - 1; j++){
173.             if(strcmp(pelanggan[j].nama, pivot.nama) < 0){
174.                 i++;
175.                 /* Jika dibandingkan nama index kiri lebih besar dari index kanan

```

```

176.         sesuai ascii, maka diswap. */
177.         swap(&pelanggan[j], &pelanggan[i]);
178.     }
179. }
180. // Swap partisi yang belum disort dengan yang sudah disort.
181. swap(&pelanggan[i+1], &pelanggan[akhir]);
182. return (i + 1);
183. }
184.
185. // Prosedur untuk quicksort dengan memanggil fungsi partition untuk nilai tengah
dan sortir.
186. void quickSort(Pelanggan pelanggan[], int mulai, int akhir){
187.     if (mulai < akhir){
188.         int tengah = partition(pelanggan, mulai , akhir);
189.         quickSort(pelanggan, 0, tengah - 1);
190.         quickSort(pelanggan, tengah + 1, akhir);
191.     }
192. }
193.
194. // Prosedur untuk Menambahkan Alamat.
195. void tambahAlamat(){
196.     // Deklarasi struct pelangganadd untuk pelanggan yang akan ditambah ke
database.
197.     Pelanggan pelangganadd;
198.
199.     // Open stream file menggunakan mode append.
200.     FILE *fp = fopen("database.data", "a");
201.
202.     // Input nama pelanggan.
203.     printf("Silahkan masukkan data anda!\n");
204.     printf("Nama: ");
205.     scanf("%s", pelangganadd.nama);
206.     getchar();
207.
208.     // Input Nomor telepon pelanggan.
209.     printf("Silahkan masukkan Nomor telepon anda!\n");
210.     printf("Nomor telepon: ");
211.     scanf("%s", pelangganadd.nohp);
212.     getchar();
213.
214.     /* Melakukan validasi, jika input user ada huruf dan bukan angka, akan
return 0
215.     Namun jika input sudah benar angka semua, akan return 1 */
216.
217.     int validasihp = validasiHP(pelangganadd.nohp);
218.
219.     // Ketika validasihp masih salah, user akan diminta input terus sampai
benar.
220.     while(!validasihp){
221.         printf("Nomor telepon masih salah! Pastikan hanya angka!\n");
222.         printf("Nomor telepon: ");
223.         scanf("%s", pelangganadd.nohp);
224.         getchar();
225.
226.         // validasihp lagi dari input yang baru.
227.         validasihp = validasiHP(pelangganadd.nohp);
228.     }
229.
230.     // Input alamat pelanggan.
231.     printf("Silahkan masukkan alamat anda!\n");
232.     printf("Alamat: ");

```

```

233.         scanf("%[^\\n]", pelangganadd.alamat);
234.         getchar();
235.
236.         // Untuk memasukkan data yang sudah diinput user ke file "database.data"
237.         fprintf(fp, "%s%s%s\\n", pelangganadd.nama, pelangganadd.noHp,
    pelangganadd.alamat);
238.
239.         /* Untuk memberi tahu bahwa data berhasil dimasukkan ke user,
240.            dan ada prompt enter untuk melanjutkan. */
241.         printf("Data berhasil dimasukkan!\\n\\n");
242.         enterToContinue();
243.         system("cls");
244.
245.         // Menutup file stream.
246.         fclose(fp);
247.     }
248.
249.     // Prosedur untuk edit data pelanggan
250.     void editPelanggan(){
251.         char tempnama[50];
252.
253.         // Open stream file menggunakan mode read dan mendeklarasikan ulang count
    menjadi 0
254.         FILE *fp = fopen("database.data", "r");
255.
256.         // Mendeklarasikan ulang variabel count menjadi 0, karena akan membaca
    kembali dari file.
257.         count = 0;
258.
259.         // Scan dari file dan memasukkan ke struct.
260.         while(!feof(fp)){
261.             fscanf(fp, "%[^#]#[^#]#[^\\n]\\n", pelanggan[count].nama,
    pelanggan[count].noHp, pelanggan[count].alamat);
262.             count++;
263.         }
264.
265.         do{
266.             // Sort pelanggan berdasarkan nama agar bisa melakukan binary search
267.             quickSort(pelanggan, 0, count - 1);
268.
269.             // Print semua pelanggan menggunakan procedure printSemuaPelanggan
270.             printSemuaPelanggan();
271.
272.             // Input nama yang akan diedit, lalu dicari menggunakan binary search
273.             printf("Silahkan masukkan nama yang akan diganti datanya!\\n");
274.             printf("Ketik N untuk keluar!\\n");
275.             printf("Input: ");
276.             scanf("%[^\\n]", tempnama);
277.             getchar();
278.
279.             // Mendapatkan data tersebut di index berapa menggunakan binary search.
280.             int index = binarySearch(pelanggan, tempnama, 0, count);
281.
282.             // Jika user input N maka akan keluar dari menu edit pelanggan.
283.             if(strcmp(tempnama, "N") == 0) break;
284.
285.             /* Jika index nya di bawah 0, maka function binarySearch tidak menemukan
    data di struct
286.             maka akan terus meminta input dari user. */
287.             while(index < 0){
288.                 system("cls");

```

```

289.
290.          // Print semua pelanggan menggunakan procedure printSemuaPelanggan
291.          printSemuaPelanggan();
292.
293.          // Input kembali nama yang akan dicari menggunakan binary search.
294.          printf("Data tidak ditemukan! Silahkan masukkan nama yang ada dan
sesuai!\n");
295.          printf("Ketik N untuk keluar!\n");
296.          printf("Nama yang akan diganti: ");
297.          scanf(" %[^\\n]", tempnama);
298.          getchar();
299.
300.          /* Jika input N, maka akan keluar dari pemilihan data yang akan
dirubah
301.          dan akan kembali input data yang akan di ubah */
302.          if(strcmp(tempnama, "N") == 0){
303.              system("cls");
304.              break;
305.          }
306.
307.          // Mendapatkan data tersebut di index berapa menggunakan binary
search.
308.          index = binarySearch(pelanggan, tempnama, 0, count);
309.          }
310.          system("cls");
311.
312.          /* Jika index nya di atas sama dengan 0, maka function
binarySearch menemukan data di struct
313.          maka akan melanjutkan meminta input apa yang akan diganti, namun jika
tidak sesuai dengan
314.          perintah, akan terus mengulang. */
315.          while(index >= 0){
316.              printf("> Merubah Nilai %s\\n\\n", pelanggan[index].nama);
317.              char option[50];
318.              char diganti[50];
319.
320.              // Input apakah akan mengganti nama atau no hp, jika input N akan
keluar.
321.              // Jika masih salah, akan dipaksa untuk input sampai benar.
322.              printf("Pilih Data apa yang ingin diganti (Nama, No HP)\\n");
323.              printf("Ketik N untuk keluar!\n");
324.              printf("Input: ");
325.              scanf(" %[^\\n]", option);
326.              getchar();
327.
328.              // Jika input N, maka akan keluar.
329.              if(strcmp(option, "N") == 0) {
330.                  system("cls");
331.                  break;
332.              }
333.
334.              // Jika input nama atau no hp
335.              if(strcmp(option, "Nama") == 0 || strcmp(option, "No HP") == 0){
336.
337.                  /* Input user dimasukan ke variabel diganti, nantinya akan
338.                  dicopy sesuai dengan input user */
339.                  printf("Diganti menjadi? ");
340.                  scanf(" %[^\\n]", diganti);
341.                  getchar();
342.
343.                  /* Jika input nama, copy string dari variabel diganti ke

```



```

344.         struct pelanggan nama sesuai index */
345.         if(strcmp(option, "Nama") == 0){
346.             strcpy(pelanggan[index].nama, diganti);
347.         }
348.
349.         /* Jika input no hp, copy string dari variabel diganti ke
350.         struct pelanggan no hp sesuai index */
351.         else if(strcmp(option, "No HP") == 0){
352.             strcpy(pelanggan[index].nohp, diganti);
353.         }
354.
355.         /* Untuk memberi tahu bahwa data berhasil diedit
356.         dan ada prompt enter untuk melanjutkan lalu break dari loop. */
357.         printf("Data berhasil dirubah!\n");
358.         enterToContinue();
359.         system("cls");
360.         break;
361.     }
362.     else{
363.         // Jika input salah, maka akan dioutput kepada user.
364.         system("cls");
365.         printf("Input salah! Silahkan coba lagi!\n\n");
366.     }
367. };
368. }while(1);
369. system("cls");
370.
371. // Menutup file stream.
372. fclose(fp);
373.
374. // Write / timpa semua nilai yang ada di struct (sudah diganti) ke database.
375. entryNewData();
376. }
377.
378. // Prosedur Untuk delete alamat yang sudah sampai
379. void deleteAlamat(){
380.     int option = 0;
381.     int valid = 1;
382.
383.     // Open file stream menggunakan mode read.
384.     FILE *fp = fopen("database.data", "r");
385.
386.     // Mendeklarasikan ulang variabel delete dan count menjadi 0, karena akan
387.     membaca kembali dari file.
388.     delete = 0;
389.     count = 0;
390.     do{
391.         // Membaca ulang data dari file.
392.         while(!feof(fp)){
393.             fscanf(fp, " %[^#]#[^#]#[^#\n]\n", pelanggan[count].nama,
394.                 pelanggan[count].nohp, pelanggan[count].alamat);
395.             count++;
396.         }
397.
398.         // Sort pelanggan berdasarkan nama agar bisa melakukan binary search
399.         quickSort(pelanggan, 0, count - 1);
400.
401.         // Output semua alamat agar user tau akan memilih yang mana.
402.         printSemuaAlamat();

```

```

403.          // Jika input tidak valid atau user memilih lebih dari range data, maka
         akan diberi tahu input salah.
404.          if(!valid || option > count) printf("Input Salah!\n");
405.          printf("Ingin Hapus Nomor berapa?\n");
406.          printf("Ketik 0 untuk keluar!\n");
407.          printf("Input: ");
408.
409.          /* Mendapatkan Input dari user, valid akan menjadi 1 jika input user
         benar
410.          namun jika input user salah, valid akan menjadi 0 */
411.          // Input data mana yang akan dihapus menggunakan number yang dioutput di
         kode.
412.          valid = scanf("%d", &option);
413.          getchar();
414.
415.          // Jika input benar, maka akan dicek.
416.          if(valid){
417.
418.              // Jika input 0, maka akan keluar dari procedure.
419.              if(option == 0){
420.                  break;
421.                  system("cls");
422.              }
423.              else{
424.                  // Jika input lebih besar dari data yang ada, akan meminta
         kembali input.
425.                  if(option > count){
426.                      system("cls");
427.                  }
428.                  else{
429.                      /* Jika input Y, maka akan didelete, ide nya adalah
         mengganti value namanya menjadi "00NULL"
430.                      kenapa "00NULL" karena string tersebut jika disort akan
         muncul paling awal sehingga memudahkan kita nantinya
431.                      untuk menulis kembali data yang ada di struct ke file kita.
         Lalu kita increment variabel delete tadi.*/
432.
433.                      /* Membuat loop jika user input Y maka akan delete, jika N
         untuk keluar. Namun jika selain keduanya, akan terus
434.                      meminta input dari user*/
435.                      while(1){
436.                          // Memberi prompt untuk user, Y untuk hapus N untuk
         cancel.
437.                          printf("\nAnda akan menghapus alamat %s\n",
         pelanggan[option-1].alamat);
438.                          printf("Anda Yakin? Y / N\n");
439.                          printf("Input: ");
440.                          char lanjut[5];
441.                          scanf("%s", lanjut);
442.                          getchar();
443.                          if(strcmp(lanjut, "Y") == 0){
444.                              strcpy(pelanggan[option-1].nama, "00NULL");
445.                              delete++;
446.                              printf("Alamat %s berhasil dihapus!\n",
         pelanggan[option-1].alamat);
447.                              break;
448.                          }
449.                          else if(strcmp(lanjut, "N") == 0){
450.                              break;
451.                          }
452.                          else{

```

```

453.                                     system("cls");
454.                                     printf("Input Salah!\n");
455.                                     }
456.                                     }
457.                                     // Jika input N, maka akan keluar dari loop.
458.                                     // Prompt Enter to Continue.
459.                                     enterToContinue();
460.                                     system("cls");
461.                                     }
462.                                     }
463.                                     }
464.                                     else{
465.                                     // Memberi tahu user jika input salah.
466.                                     printf("Input Salah!\n");
467.                                     system("cls");
468.                                     }
469.                                     }while(1);
470.
471.                                     // Prompt Enter to Continue.
472.                                     enterToContinue();
473.                                     system("cls");
474.
475.                                     /* Kita akan memasukkan data yang baru dari struct yang ada dan sudah
dihapus ke file
476.                                     yang ada,*/
477.                                     fclose(fp);
478.
479.                                     /* Write / timpa semua nilai yang ada di struct (sudah diganti) ke database.
480.                                     lalu kita declare ulang delete menjadi 0*/
481.                                     entryNewData();
482.                                     delete = 0;
483.
484.                                     // Set semua struct menjadi 0 menggunakan memset.
485.                                     memset(pelanggan , 0, sizeof(pelanggan));
486.                                     }
487.
488.                                     // Prosedur untuk menampilkan semua alamat
489.                                     void printAlamat(){
490.                                     // Open file stream menggunakan mode read.
491.                                     FILE *fp = fopen("database.data", "r");
492.
493.                                     // Mendeklarasikan ulang variabel count menjadi 0, karena akan membaca
kembali dari file.
494.                                     count = 0;
495.                                     while(!feof(fp)){
496.                                     fscanf(fp, " %[^#]#[^#]#[^\\n]\\n", pelanggan[count].nama,
pelanggan[count].nohp, pelanggan[count].alamat);
497.                                     count++;
498.                                     }
499.
500.                                     // Sort pelanggan berdasarkan nama agar outputnya enak dilihat.
501.                                     quickSort(pelanggan, 0, count - 1);
502.
503.                                     // Print semua pelanggan dan enter to continue setelahnya.
504.                                     printSemuaPelanggan();
505.                                     enterToContinue();
506.                                     system("cls");
507.
508.                                     // Menutup file stream.
509.                                     fclose(fp);
510.                                     }

```

```

511.
512.     // Prosedur untuk output semua data pelanggan yang ada di database.
513.     void printSemuaPelanggan(){
514.         printf("-----");
515.         printf("-----");
516.         printf("\n");
517.         printf("| %-2s | %-20s | %-50s | %-15s |\n", "No", "Nama", "Alamat", "Nomor
    Telefon");
518.         printf("-----");
519.         printf("-----");
520.         printf("\n");
521.
522.         // Output sebanyak data yang ada sampai variabel count
523.         for(int i = delete; i < count; i++){
524.             /* Terdapat edge case, jika di file tidak ada data sama sekali ternyata
    mendapatkan count = 1 karena "/"0"
525.             dan akan tetap membaca adanya data. Hal ini Membuat UX menjadi kurang
    karena tidak ada data di database tetapi
526.             tetap mengoutput tabel kosong. Maka kita akan cek, jika count nya 1,
    kita cek apakah ada value menggunakan strlen
527.             jika tidak ada data akan print "DATA MASIH KOSONG", namun jika ada data,
    akan print sebagai mana mestinya.*/
528.             if(count == 1){
529.                 if(strlen(pelanggan[i].nama) > 0){
530.                     printf("| %-2d | %-20s | %-50s | %-15s |\n", i+1,
    pelanggan[i].nama, pelanggan[i].alamat, pelanggan[i].nohp);
531.                 }
532.                 else{
533.                     printf("\t\t\t\t\tDATA MASIH KOSONG!\n");
534.                 }
535.             }
536.             else{
537.                 printf("| %-2d | %-20s | %-50s | %-15s |\n", i+1, pelanggan[i].nama,
    pelanggan[i].alamat, pelanggan[i].nohp);
538.             }
539.         }
540.         printf("-----");
541.         printf("-----");
542.         printf("\n");
543.         printf("Total data yang tersedia %d\n", count);
544.         printf("\n");
545.     }
546.
547.     // Prosedur untuk output semua alamat yang ada di database.
548.     void printSemuaAlamat(){
549.         printf("-----");
550.         printf("\n");
551.         printf("| %-2s | %-50s |\n", "No", "Alamat");
552.         printf("-----");
553.         printf("\n");
554.
555.         // Output sebanyak data yang ada sampai variabel count
556.         for(int i = delete; i < count; i++){
557.             if(count == 1){
558.                 if(strlen(pelanggan[i].nama) > 0){
559.                     printf("| %-2d | %-50s |\n", i+1, pelanggan[i].alamat);
560.                 }
561.                 else{

```

```

562.             printf("\t\t DATA MASIH KOSONG!\n");
563.         }
564.     }
565.     else{
566.         printf("| %-2d | %-50s |\n", i+1, pelanggan[i].alamat);
567.     }
568. }
569. printf("-----\n");
570. printf("Total data yang tersedia %d\n", count);
571. printf("\n");
572. }
573.
574. // Setelah edit data, kita akan menulis kembali data tersebut di database.
575. void entryNewData(){
576.     // Open file stream menggunakan mode writing (menimpa seluruh data).
577.     FILE *fp = fopen("database.data", "w");
578.
579.     /* Loop dari delete sampai count, kenapa dari delete? karena jika kita
580.     mempunyai 5 data dan
581.     delete 2 data, kita set data nama yang di delete menjadi "00NULL" dan disort,
582.     "00NULL" ini akan muncul
583.     di awal karena disort. Maka kita data yang akan ditulis kembali dari index
584.     2, karena index 0 dan 1 akan dihapus dan
585.     mempunyai value "00NULL" maka karena dari itu kita memulai loop dari delete
586.     sampai count.*/
587.     for(int i = delete; i < count; i++){
588.         /* Terdapat edge case, jika di file tidak ada data sama sekali ternyata
589.         mendapatkan count = 1 karena "/0"
590.         dan akan tetap menulis di file database ## tetapi tidak ada data. Hal
591.         ini membuat program kita crash. Maka
592.         cara mengatasinya adalah jika count nya 1, kita cek apakah ada value
593.         nama menggunakan strlen, jika ada data
594.         tidak melakukan apa apa, namun jika tidak ada data. maka kita continue
595.         sehingga loop selesai dan tidak akan
596.         fprintf yang di bawahnya.*/
597.
598.         /* Namun jika count 1 dan ada datanya, bisa saja data hanya ada satu dan
599.         kita akan hapus,
600.         maka kita cek apakah itu akan dihapus atau tidak, jika iya, kita
601.         continue dan tidak akan kita tulis di file.
602.         Jika tidak, tidak melakukan apa apa.*/
603.         if(count == 1){
604.             if(strlen(pelanggan[i].nama) > 0){
605.                 if(strcmp(pelanggan[i].nama, "00NULL") == 0) continue;
606.                 else;
607.             }
608.             else continue;
609.         }
610.         fprintf(fp, "%s#%s#%s\n", pelanggan[i].nama, pelanggan[i].nohp,
611.             pelanggan[i].alamat);
612.     }
613.     fclose(fp);
614. }

```

Source code bisa didapatkan di

<https://github.com/fabianhabil/semester1/blob/main/express/express.c>

1. Jika Database masih kosong

Masih ada bug di bagian total data yang tersedia tetap menampilkan satu meskipun data masih kosong, hal ini disebabkan karena file yang kosong masih mempunyai “\0” dan tetap terkena loop kita yang mendeteksi adanya data atau tidak, namun saya sudah berhasil untuk handle apakah masih kosong di tabelnya.

Ke Menu 2 (Edit Pelanggan)

```
-----
|              Sasuke Express              |
|-----|
| 1. Tambah Alamat                        |
| 2. Edit Info Pelanggan                  |
| 3. Delete Alamat                        |
| 4. Print Alamat                        |
| 5. Exit                                |
|-----|
Masukkan Pilihan: 2
```

```
-----
| No | Nama                | Alamat                | Nomor Telefon |
|-----|
|                                     DATA MASIH KOSONG!                                     |
|-----|
Total data yang tersedia 1
Silahkan masukkan nama yang akan diganti datanya!
Ketik N untuk keluar!
Input:
```

Ke Menu 3 (Delete Alamat)

```
-----
|              Sasuke Express              |
|-----|
| 1. Tambah Alamat                        |
| 2. Edit Info Pelanggan                  |
| 3. Delete Alamat                        |
| 4. Print Alamat                        |
| 5. Exit                                |
|-----|
Masukkan Pilihan: 3
```

```
-----
| No | Alamat                |
|-----|
|                                     DATA MASIH KOSONG!                                     |
|-----|
Total data yang tersedia 1
Ingin Hapus Nomor berapa?
Ketik 0 untuk keluar!
Input:
```

Ke Menu 4 (Print Alamat)

```
-----  
|              Sasuke Express              |  
-----  
| 1. Tambah Alamat                        |  
| 2. Edit Info Pelanggan                  |  
| 3. Delete Alamat                        |  
| 4. Print Alamat                         |  
| 5. Exit                                |  
-----  
Masukkan Pilihan: 4
```

```
-----  
| No | Nama                | Alamat                | Nomor Telefon |  
-----  
DATA MASIH KOSONG!  
-----  
Total data yang tersedia 1  
Tekan Enter untuk melanjutkan..
```

2. Menambahkan Alamat

```
-----  
|              Sasuke Express              |  
-----  
| 1. Tambah Alamat                        |  
| 2. Edit Info Pelanggan                  |  
| 3. Delete Alamat                        |  
| 4. Print Alamat                         |  
| 5. Exit                                |  
-----  
Masukkan Pilihan: 1
```

```
Silahkan masukkan data anda!  
Nama: Fabian Habil  
Silahkan masukkan Nomor telepon anda!  
Nomor telepon: 081817asd  
Nomor telepon masih salah! Pastikan hanya angka!  
Nomor telepon: asdasd  
Nomor telepon masih salah! Pastikan hanya angka!  
Nomor telepon: 081718545  
Silahkan masukkan alamat anda!  
Alamat: Jalan Bandung  
Data berhasil dimasukkan!  
  
Tekan Enter untuk melanjutkan..
```

```

-----
| No | Nama                | Alamat                | Nomor Telefon |
-----
| 1  | Fabian Habil        | Jalan Bandung        | 081718545    |
-----
Total data yang tersedia 1
Tekan Enter untuk melanjutkan..

```

### 3. Edit Pelanggan

```

-----
|                Sasuke Express                |
-----
| 1. Tambah Alamat                             |
| 2. Edit Info Pelanggan                       |
| 3. Delete Alamat                            |
| 4. Print Alamat                             |
| 5. Exit                                     |
-----
Masukkan Pilihan: 2

```

```

-----
| No | Nama                | Alamat                | Nomor Telefon |
-----
| 1  | Fabian Habil        | Jalan Bandung        | 081718545    |
| 2  | Habil               | Jakarta Selatan      | 08112345     |
-----
Total data yang tersedia 2

Silahkan masukkan nama yang akan diganti datanya!
Ketik N untuk keluar!
Input: Habil_

```

```

> Merubah Nilai Habil

Pilih Data apa yang ingin diganti (Nama, No HP)
Ketik N untuk keluar!
Input: No HP
Diganti menjadi? 1234567
Data berhasil dirubah!
Tekan Enter untuk melanjutkan.._

```

```

-----
| No | Nama                | Alamat                | Nomor Telefon |
-----
| 1  | Fabian Habil        | Jalan Bandung        | 081718545    |
| 2  | Habil               | Jakarta Selatan      | 1234567      |
-----
Total data yang tersedia 2

Silahkan masukkan nama yang akan diganti datanya!
Ketik N untuk keluar!
Input:

```



```
-----
|           Sasuke Express           |
|-----|
| 1. Tambah Alamat                  |
| 2. Edit Info Pelanggan            |
| 3. Delete Alamat                  |
| 4. Print Alamat                   |
| 5. Exit                           |
|-----|
Masukkan Pilihan: 4
```

```
-----
| No | Nama                | Alamat                | Nomor Telefon |
|-----|-----|-----|-----|
| 1  | Fabian Habil        | Jalan Bandung         | 081718545     |
| 2  | Habil               | Jakarta Selatan       | 1234567       |
|-----|-----|-----|-----|
Total data yang tersedia 2
Tekan Enter untuk melanjutkan..
```

#### 4. Delete Alamat

```
-----
|           Sasuke Express           |
|-----|
| 1. Tambah Alamat                  |
| 2. Edit Info Pelanggan            |
| 3. Delete Alamat                  |
| 4. Print Alamat                   |
| 5. Exit                           |
|-----|
Masukkan Pilihan: 3
```

```
-----
| No | Alamat                |
|-----|-----|
| 1  | Jalan Bandung         |
| 2  | Jakarta Selatan       |
|-----|-----|
Total data yang tersedia 2

Ingin Hapus Nomor berapa?
Ketik 0 untuk keluar!
Input: 1_
```

```

-----
| No | Alamat |
-----
| 1 | Jalan Bandung |
| 2 | Jakarta Selatan |
-----

```

Total data yang tersedia 2

Ingin Hapus Nomor berapa?

Ketik 0 untuk keluar!

Input: 1

Anda akan menghapus alamat Jalan Bandung

Anda Yakin? Y / N

Input: Y

Alamat Jalan Bandung berhasil dihapus!

Tekan Enter untuk melanjutkan..

```

-----
| No | Alamat |
-----
| 2 | Jakarta Selatan |
-----

```

Total data yang tersedia 1

Ingin Hapus Nomor berapa?

Ketik 0 untuk keluar!

Input: ASDASDASDAS

```

-----
| No | Alamat |
-----
| 2 | Jakarta Selatan |
-----

```

Total data yang tersedia 1

Input Salah!

Ingin Hapus Nomor berapa?

Ketik 0 untuk keluar!

Input: 2

Anda akan menghapus alamat Jakarta Selatan

Anda Yakin? Y / N

Input: Y

Alamat Jakarta Selatan berhasil dihapus!

Tekan Enter untuk melanjutkan..

```

-----
| No | Alamat |
-----

```

Total data yang tersedia 0

Ingin Hapus Nomor berapa?

Ketik 0 untuk keluar!

Input:

5. Jika ada banyak data

```
-----  
|           Sasuke Express           |  
-----  
| 1. Tambah Alamat                  |  
| 2. Edit Info Pelanggan            |  
| 3. Delete Alamat                  |  
| 4. Print Alamat                   |  
| 5. Exit                           |  
-----  
Masukkan Pilihan: 4
```

```
-----  
| No | Nama                | Alamat                                | Nomor Telefon |  
-----  
| 1  | Bianha              | Tebet Dalam 1                        | 0812381293    |  
| 2  | Fabian Habil        | Bandung                              | 0817185213    |  
| 3  | Muhamad Arteezy     | Vancouver                            | 8781273821    |  
| 4  | Sasuke Express      | JALAN KONOHA NO 5, KEC LEAF VILLAGE | 08772145123    |  
-----  
Total data yang tersedia 4  
Tekan Enter untuk melanjutkan..
```