

# **LAPORAN TUGAS BESAR STRUKTUR DATA DAN ALGORITMA**

**Program Aplikasi Simulasi Huffman Code**



Oleh:

Icha Cahya Wulan 181524013

Naufal Rajabi 181524025

1A/D4 TEKNIK INFORMATIKA

JURUSAN TEKNIK KOMPUTER DAN INFORMATIKA

POLITEKNIK NEGERI BANDUNG

2018/2019

## 1. Deskripsi Aplikasi

Aplikasi kami menerima tiga macam masukan untuk dapat dikonversikan ke dalam huffman code yaitu: karakter serta frekuensinya, kalimat, dan file teks.

Kemampuan aplikasi yang kami buat, yaitu:

- a. Dapat menampilkan kondisi awal dari queue (tempat awal mengumpulkan huffman node)
  - Setiap huruf dan frekuensinya yang diurutkan *ascending* berdasarkan frekuensi yang terkecil
- b. Dapat menampilkan proses pembentukan tree
  - Setiap langkah pembentukan tree ditampilkan hingga terbentuk root dari treenya
- c. Dapat menampilkan tabel hasil konversi setiap karakter
  - Tabel disusun berdasarkan kode ASCII secara ascending
  - Tabel menampilkan tiap karakter dan kode hasil konversinya
- d. Dapat menampilkan hierarki tree
  - Tree ditampilkan tiap level per level
- e. Dapat menampilkan string dalam bentuk kode yang sudah dikonversi\*
- f. Dapat mengembalikan kode yang telah dikonversi ke dalam string asal\*

Keterangan:

\* *hanya untuk masukan berupa kalimat dan file teks*

## 2. Rancangan

### a. Perancangan Data

Berikut adalah struktur data yang kami definisikan. Terdiri dari tiga struktur yang berbeda. Yaitu `huffman_node_t`, `priority_queue_t`, dan `codewords_t` yang masing-masing memiliki fungsi penyimpanan yang berebeda.

#### i. Record: `huffman_node_t`

```
<
    letter      : char
    frequency   : int
    left        : pointer of huffman_node_t
    right       : pointer of huffman_node_t
    next        : pointer of huffman_node_t
>
```

Struktur ini berfungsi untuk menyimpan huruf, jumlah kemunculan huruf, penunjuk hierarki di bawahnya (anak), dan penunjuk node selanjutnya dalam queue. Nantinya queue akan hanya berisi satu node yaitu root dari huffman tree.

#### ii. Record: `priority_queue_t`

```
<
    front : pointer of huffman_node_t
    size  : int
>
```

Struktur ini berguna untuk menunjuk pada queue yang berisi huffman tree pada `queue.front`, dan menyimpan ukuran queue yang berguna dalam proses pembentukan tree.

#### iii. Record : `codewords_t`

```
<
    code_length : size_t
    bit_code    : uint32_t
>
```

Struktur ini berfungsi sebagai tabel konversi. Struktur ini kami dapat dari salah satu referensi. Nantinya nilai yang berada di field struktur ini akan di proses untuk mencetak codewords dengan operasi tertentu.

b. Perancangan Modul

i. Daftar Modul

Modul pada mainProgram.c

Tipe Modul	Tipe Kembalian	Nama Modul	Parameter	Fungsi
Function	int	choose_read_type	-	Menampilkan pilihan menu dan mengembalikan pilihan user
Procedure	void	ask_for_exit	-	Bertanya kepada user apakah ingin mensimulasikan kembali atau tidak. Jika ya, maka akan kembali ke menu utama, jika tidak maka program akan berakhir.
Procedure	void	run_huffman_read_char	-	Menjalankan simulasi kompresi huffman untuk input berupa karakter beserta frekuensinya.
Procedure	void	run_huffman_read_string	-	Menjalankan simulasi kompresi huffman untuk input berupa teks yang akan dikodifikasikan/dikompresikan
Procedure	void	run_huffman_read_file	-	Menjalankan simulasi kompresi huffman untuk input berupa nama file yang akan dibaca.
Procedure	void	exit_huffman	-	Menampilkan pesan keluar dan keluar dari program.

Modul pada header dan body function ADT Huffman Code

Tipe Modul	Tipe Kembalian	Nama Modul	Parameter	Fungsi
Procedure	void	printHierarchy	huffman_node_t *current, int level	Mencetak hierarki tree yang telah terbentuk
Procedure	void	init_prior_queue	priority_queue_t *queue	Menginisiasi antrian huffman code
Function	int	is_prior_queue_empty	priority_queue_t queue	Memeriksa adanya node dalam antrian
Procedure	Void	enqueue_prior	priority_queue_t *queue, huffman_node_t	Memasukkan node ke antrian sesuai dengan urutan prioritas

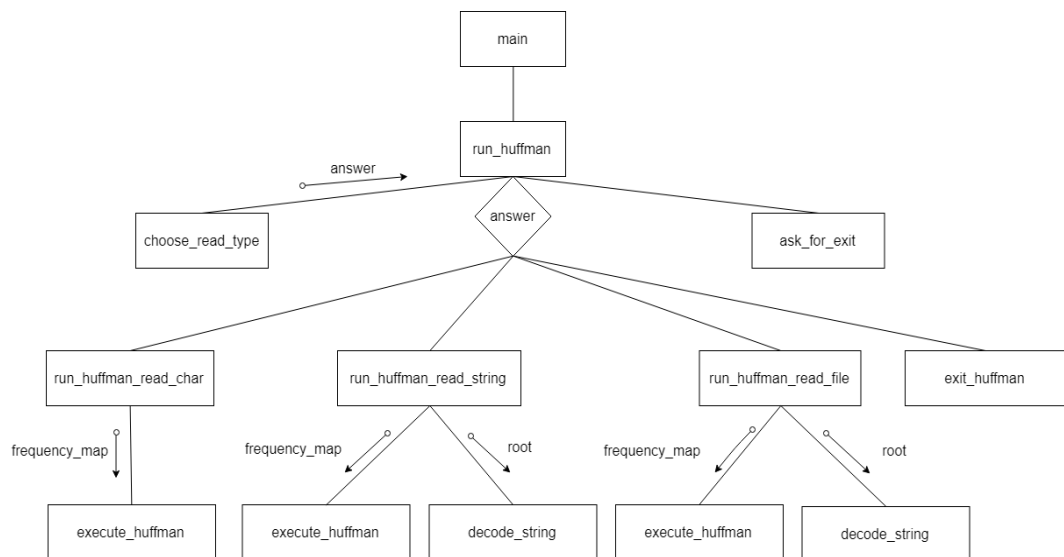
			e_t *new_node	
Function	huffman_node_t *	dequeue_prior	priority_queue_t *queue	Mengeluarkan node dari antrian
Function	huffman_node_t *	make_huffman_node	char letter, int frequency, huffman_node_t *left, huffman_node_t *right	Membuat huffman node
Function	int	prior_queue_size	priority_queue_t queue	Menghitung besar antrian
Procedure	void	print_prior_queue	priority_queue_t queue	Mencetak antrian
Procedure	void	is_leaf	huffman_node_t *node	Memeriksa apakah node merupakan daun atau bukan
Function	huffman_node_t*	make_huffman_tree	*queue	Membuat huffman tree dari kumpulan huffman node (forest)
Function	huffman_node_t*	execute_huffman	int frequency_map [MAX_ASCII_CHARACTER], codewords_t *table, codewords_t code	Menjalankan pembentukan huffman tree dari map frequency yang ada, dilanjut inisiasi forest sampai dengan print huffman tree
Procedure	void	compute_code_table	huffman_node_t *node, codewords_t *table, codewords_t code	Komputasi bit code
Procedure	void	code_print	codewords_t *code	Mencetak bit code huffman ke layar
Procedure	void	write_code_to_file	codewords_t *code	Menulis code biner ke dalam file
Procedure	void	decode_string	huffman_node_t *root	Mendekompresi string hasil kompresi

Procedure	<b>void</b>	<b>print_code_table</b>	<b>codewords_t</b> <i>*table</i>	Mencetak tabel konversi karakter
-----------	-------------	-------------------------	-------------------------------------	----------------------------------

## ii. Interaksi Antarmodul

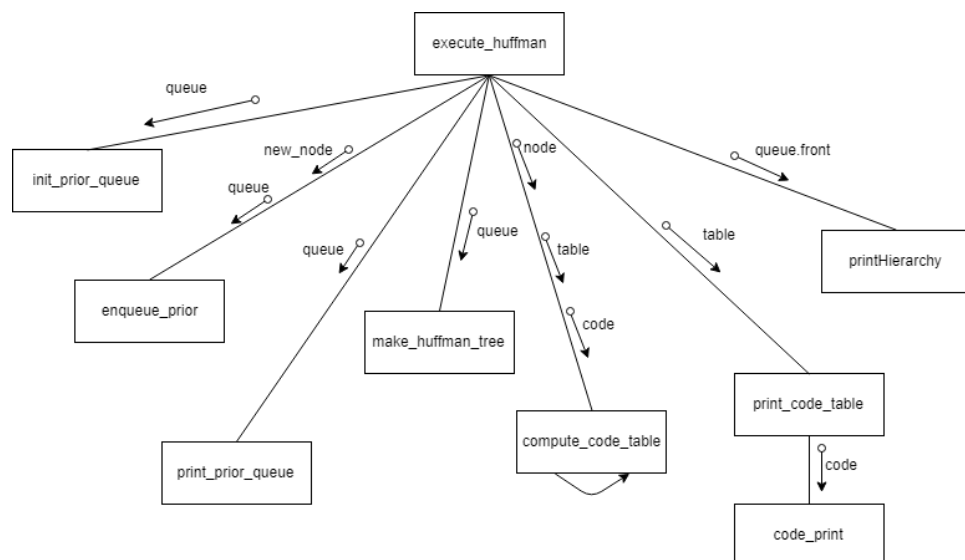
Interaksi antar modul kami ilustrasikan dalam beberapa bentuk structure chart berikut. Structure chart kami bagi menjadi dua yaitu yang ada pada mainProgram.c dan yang menjadi bagian dari ADT Huffman Code itu sendiri.

- Structure pada mainProgram.c

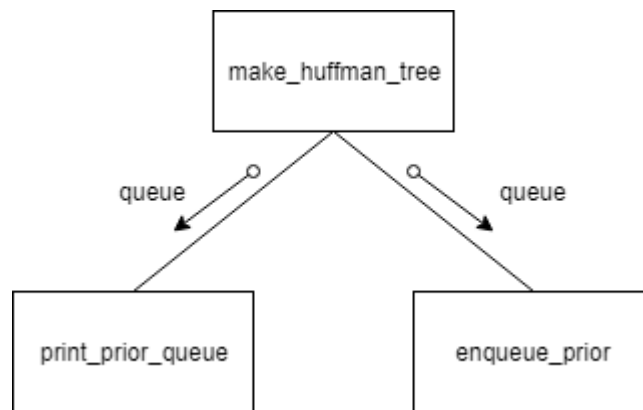


Gambar 1. Structure chart mainProgram.c

- Structure chart modul yang ada pada body function ADT Huffman Code (huffmanCode.c)

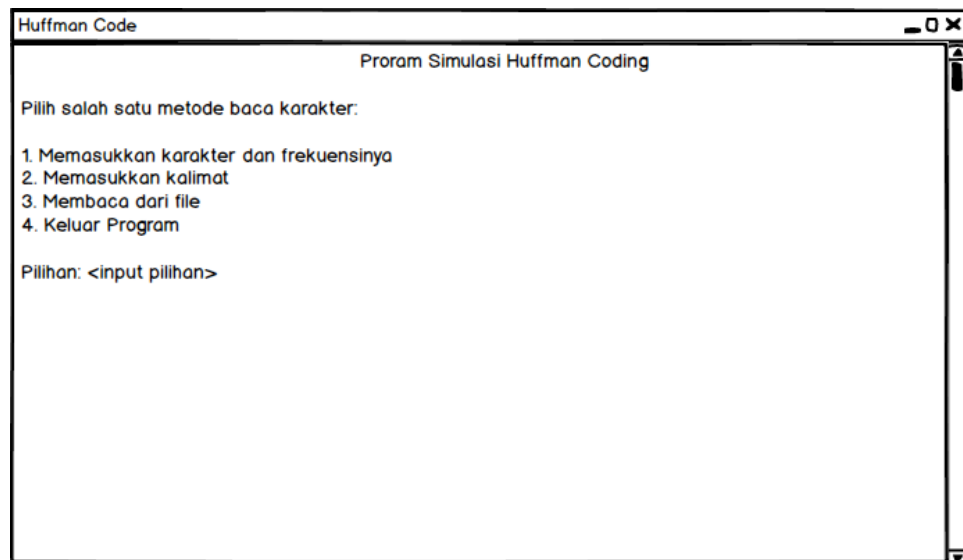


Gambar 2. Structure chart modul execute\_huffman

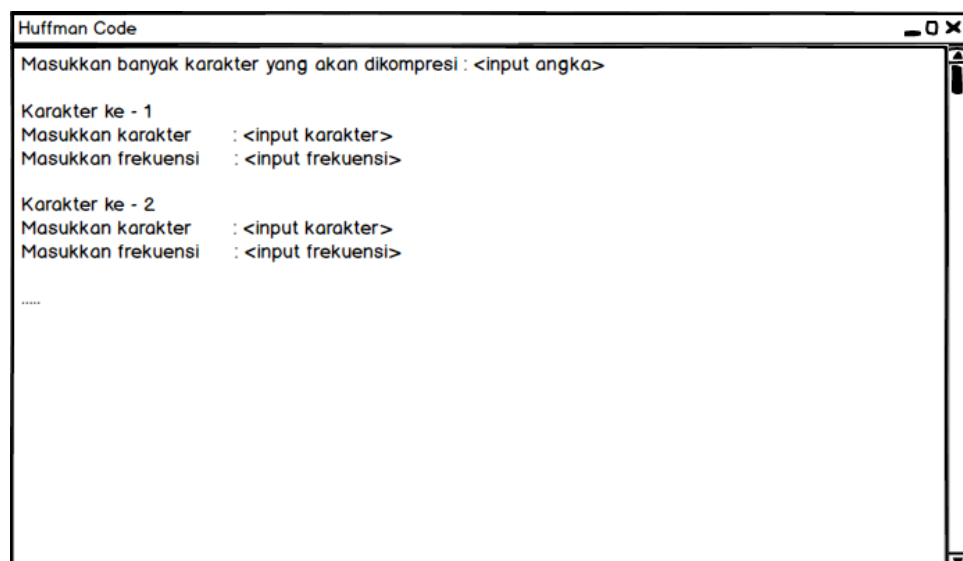


Gambar 3. Structure chart modul make\_huffman\_tree

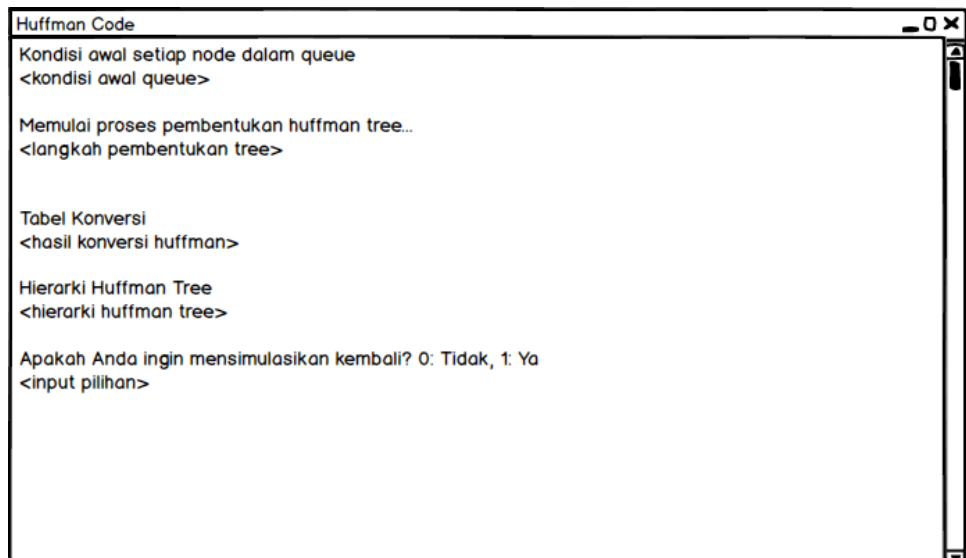
c. Perancangan Interaksi dengan User



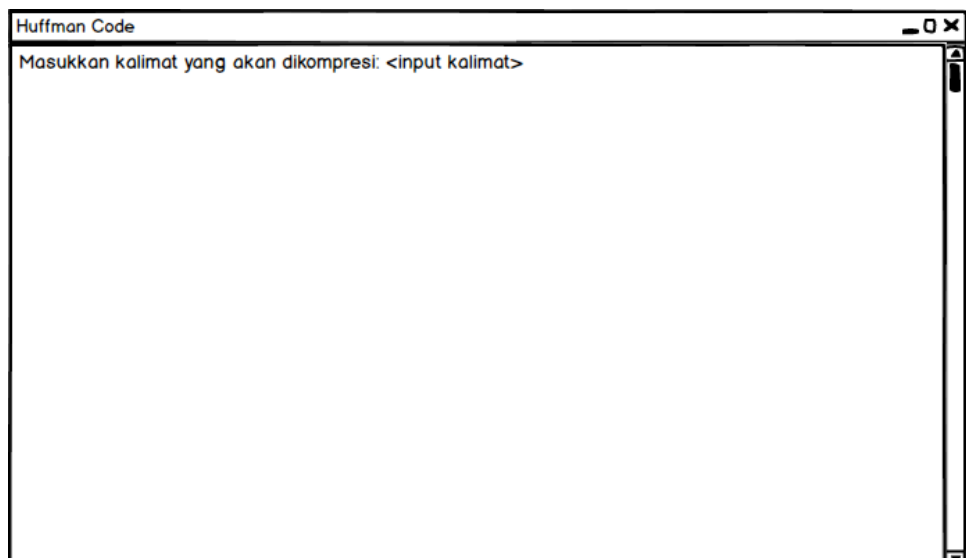
Gambar 4. Tampilan menu utama



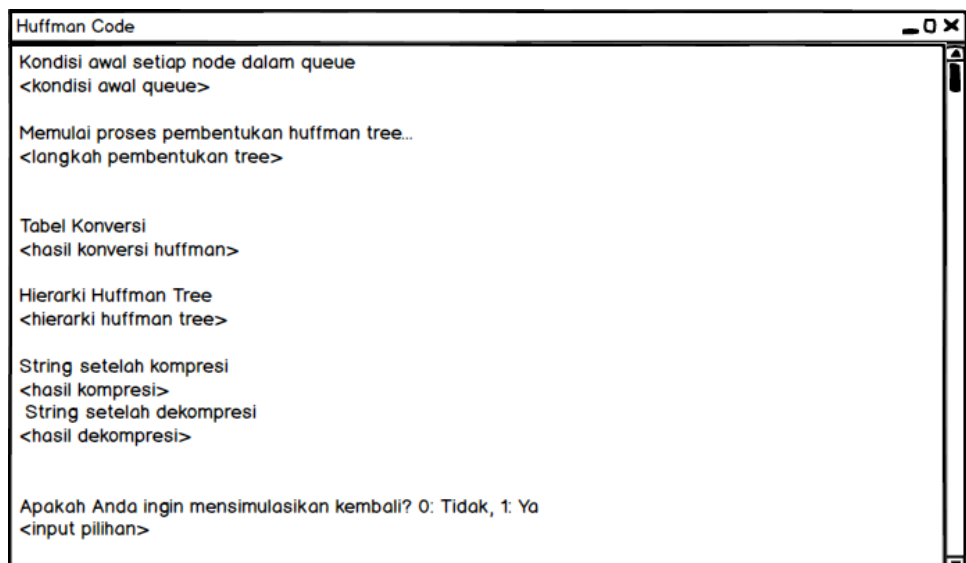
Gambar 5. Tampilan input menu 1



Gambar 6. Tampilan proses pada menu 1

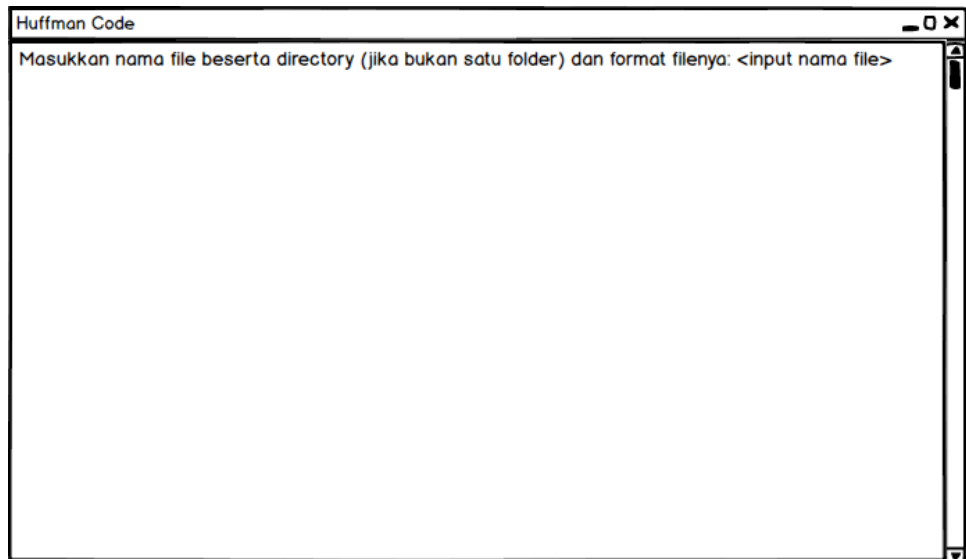


Gambar 7. Tampilan input menu 2

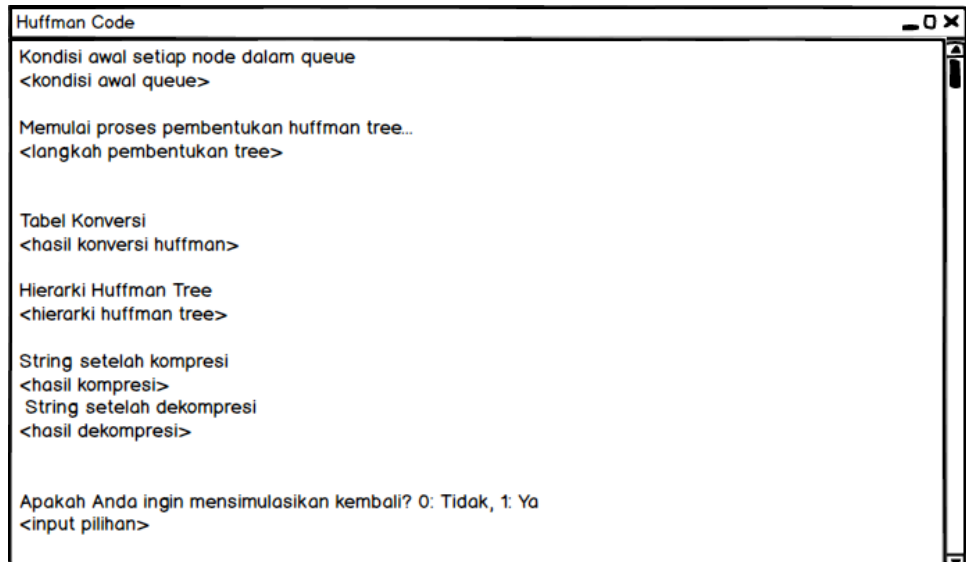




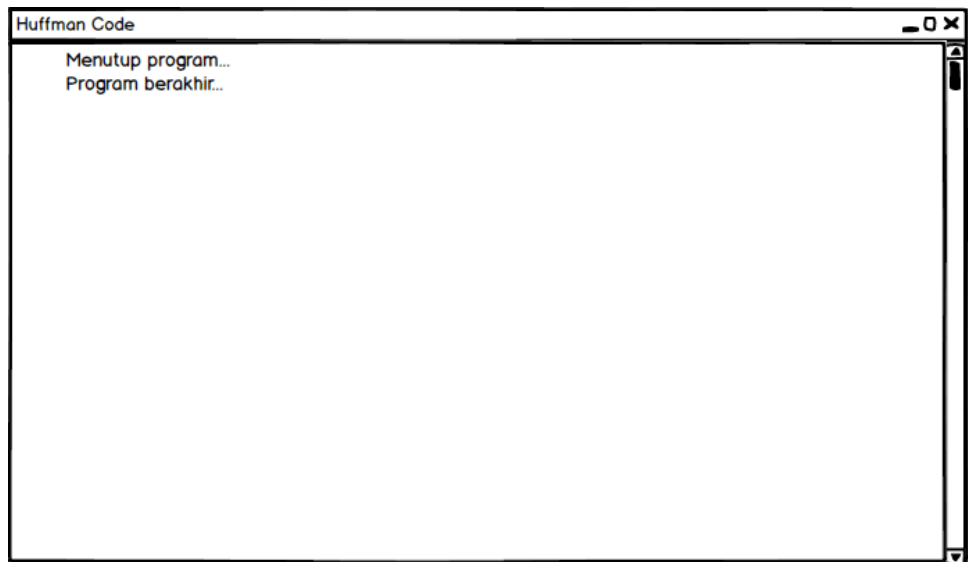
Gambar 8. Tampilan proses menu 2



Gambar 9. Tampilan input menu 3



Gambar 10. Tampilan proses menu 3



Gambar 11. Tampilan keluar program

### 3. Kendala dan Lesson Learned

#### 3.1 Kendala

Selama proses pengerjaan kami memiliki kendala dan sedikit bingung dalam menyimpan tabel konversi kode Huffman dari hasil traversing Huffman Tree. Awalnya kami mencoba menyimpannya ke dalam array of char, tetapi bahasa C sangat terbatas dalam manajemen string. Lalu kami pun sempat berpikir untuk menyimpannya dalam linked list dengan info setiap nodenya adalah 0 atau 1. Namun, ide ini pun masih terlalu banyak memakan memori dalam worst case-nya.

Akhirnya kami memakai source code dari salah satu referensi, sehingga kami dapat membuat sekaligus mencetak tabel konversi yang dibuat melalui traversing pada huffman tree yang telah terbentuk.

#### 3.2 Lesson Learned

Dari topik tugas besar yang kami ambil ini kami mengambil beberapa pelajaran penting dan beberapa hal baru. Berikut adalah *lesson learned* yang kami dapatkan melalui pengerjaan tugas besar ini.

- a. Mengetahui lebih lanjut mengenai penggunaan pointer terutama dalam konteks struktur data tree.
- b. Melatih kemampuan membuat program dengan mendefinisikan ADT untuk kasus yang akan dihadapi.
- c. Melatih kemampuan *clean-code* ketika menulis program.
- d. Melatih kemampuan bekerja secara kolaboratif dalam bersama-sama menyelesaikan masalah yang dihadapi.
- e. Melatih kemampuan manajemen dalam pembagian tugas dan tanggung jawab.

#### 4. Referensi

Berikut adalah referensi yang mendukung dalam pembuatan tugas besar kami.

<https://courses.cs.washington.edu/courses/cse143x/15au/lectures/huffman/huffman.pdf>

<https://www.codelike.in/c/linked-list/singly-linked-list-insert-data-in-ascending-order>

<http://www.cplusplus.com/reference/>

[https://www.tutorialspoint.com/cprogramming/c\\_operators.htm](https://www.tutorialspoint.com/cprogramming/c_operators.htm)

[https://www.tutorialspoint.com/cprogramming/c\\_bitwise\\_operators.htm](https://www.tutorialspoint.com/cprogramming/c_bitwise_operators.htm)

<https://stackoverflow.com/questions/47421732/how-can-i-generate-a-binary-code-table-of-a-huffman-tree>

<https://stackoverflow.com/questions/4815251/how-do-i-clear-the-whole-contents-of-a-file-in-c>