

AOL DATA STRUCTURE

Pilar Nalendra Sarwanto – 2702362604 – LD07

Code Explanation

// LIBRARY

```
#include <stdio.h> // Library input output
#include <stdlib.h> // Library alokasi memori
#include <stdbool.h> // Library fungsi boolean
#include <string.h> // Library modifikasi bentuk string
```

// CONST CHAR: Digunakan untuk inisialisasi size dari array

```
#define ALPHABET_SIZE 26
#define MAX_WORD 100
```

// VARIABEL GLOBAL

```
int globalNum = 1;
char globalPrefix[50];
```

// STRUKTUR DATA TRIENODE: fondasi trie || representasi isi dari setiap node dalam trie

```
typedef struct TrieNode {
    struct TrieNode* children[ALPHABET_SIZE]; // Array anak-anak node || node-node yang
    terhubung
    bool terminal; // Akhir node dari sebuah slangword
    char meaning[100]; // Menyimpan makna slang word
} TrieNode;
```

// MEMBUAT NODE BARU

```
TrieNode* createNode() {  
    TrieNode* newNode = (TrieNode*)malloc(sizeof(TrieNode)); // Alokasi memori  
    if (newNode) { // Inisialisasi nilai default  
        newNode->terminal = false; // Memastikan bahwa newNode yang terbentuk bukan  
        merupakan terminal  
        int i;  
        for (i = 0; i < ALPHABET_SIZE; ++i)  
            newNode->children[i] = NULL;  
        newNode->meaning[0] = '\0';  
    }  
    return newNode; // Mengembalikan newNode  
}
```

// MEMASUKAN MAKNA SLANGWORD KE DALAM TRIE

```
void insert(TrieNode* root, const char* word, const char* meaning) {  
    TrieNode* curr = root; //Inisialisasi curr sebagai root (memudahkan dan meminimalisir  
    kesalahan)  
    int len = strlen(word);  
  
    int i;  
    for (i = 0; word[i] != '\0'; i++) {  
        int index = word[i] - 'a'; // Menghitung indeks untuk setiap huruf dalam slang word  
        if (!curr->children[index])  
            curr->children[index] = createNode(); // Jika node belum ada, buat node baru  
        curr = curr->children[index];  
    }  
    curr->terminal = true; // Tandai node terakhir sebagai terminal node  
    strcpy(curr->meaning, meaning); // Simpan makna dari slang word  
}
```

// MENCARI SLANGWORD DALAM TRIE

```
bool search(TrieNode* root, const char* word, char* meaning) {  
    TrieNode* curr = root;  
    int i;  
    for (i = 0; word[i] != '\0'; i++) {  
        int index = word[i] - 'a'; // Menghitung indeks untuk setiap huruf dalam slang word  
        if (!curr->children[index])  
            return false; // Jika node tidak ditemukan, slang word tidak ada dalam trie  
        curr = curr->children[index];  
    }  
    if (curr && curr->terminal) {  
        strcpy(meaning, curr->meaning); // Mengambil makna dari node akhir  
        return true; // Return true: slang word ditemukan  
    }  
    return false; // Return false: slang word tidak ditemukan  
}
```

// MENANDAKAN SEMUA SLANG WORD DENGAN AWAL TERTENTU SEHINGGA NANTINYA AKAN DI PRINT

```
void printWordsWithPrefixUtil(TrieNode* node, char* buffer, int depth, char** words, int* count,  
const char* prefix) {  
    if (node->terminal) {  
        buffer[depth] = '\0'; // Mengakhiri string buffer  
        words[*count] = strdup(buffer); // Salin buffer ke array words  
        (*count)++;  
    }  
  
    int i;  
    for (i = 0; i < ALPHABET_SIZE; i++) {  
        if (node->children[i]) {  
            buffer[depth] = i + 'a'; // Tambahkan huruf ke buffer
```

```

        printWordsWithPrefixUtil(node->children[i], buffer, depth + 1, words, count, prefix);
// Rekursif ke fungsi printWordsWithPrefixUtil
    }
}
}

```

// MENCETAK SEMUA SLANG WORD DENGAN PREFIX TERTENTU

```

void printWordsWithPrefix(TrieNode* root, const char* prefix) {
    TrieNode* temp = root;
    int length = strlen(prefix);
    int level;

    for (level = 0; level < length; level++) {
        int index = prefix[level] - 'a';
        if (!temp->children[index]) { // Periksa apakah setiap karakter ada dalam trie
            printf("\nThere is no prefix \"%s\" in the dictionary.", prefix);
            return;
        }
        temp = temp->children[index]; // Lanjut ke node berikutnya
    }

    char* words[MAX_WORD];
    int count = 0;
    char buffer[1000]; // Asumsikan panjang maksimum kata 1000
    strcpy(buffer, prefix); // Salin prefixnya ke buffer
    printWordsWithPrefixUtil(temp, buffer, length, words, &count, prefix);

    if (count == 0) { // Jika countnya 0, maka tak ada prefix tersebut dalam dictionary
        printf("\nThere is no prefix \"%s\" in the dictionary.", prefix);
        return;
    }
}

```

```
}
```

// Jika tidak ada kata slang yang sesuai, cetak pesan bahwa tidak ada kata slang dengan prefix tersebut

```
printf("\nWords starts with \"%s\":\n", prefix);
```

```
int i;
```

```
for (i = 0; i < count; i++) {
```

```
    printf("%d. %s\n", i + 1, words[i]);
```

```
    free(words[i]); // Free memori yang dialokasikan untuk setiap kata
```

```
}
```

```
}
```

// SAMBUNGAN DARI VIEWALL

```
void viewAllHelper(TrieNode* node, char* buffer, int depth, int* number) {
```

```
    if (node->terminal) { // Cek jika merupakan terminal node
```

```
        buffer[depth] = '\0';
```

```
        printf("%d. %s\n", (*number)++, buffer); // Print complete wordnya
```

```
    }
```

```
    int i;
```

```
    for (i = 0; i < ALPHABET_SIZE; i++) { // Iterasi rekursif berdasarkan semua anak node A - Z
```

```
        if (node->children[i]) {
```

```
            buffer[depth] = i + 'a';
```

```
            viewAllHelper(node->children[i], buffer, depth + 1, number);
```

```
        }
```

```
    }
```

```
}
```

```
// PRINT SEMUA SLANG WORD YANG SUDAH DIINPUT
```

```
void viewAll(TrieNode* root) {  
    bool isEmpty = true; // Boolean untuk cek apakah trie kosong  
    int i;  
    for (i = 0; i < ALPHABET_SIZE; i++) {  
        if (root->children[i] != NULL) {  
            isEmpty = false; // Jika trie kosong maka isEmpty = false  
            break;  
        }  
    }  
  
    if (isEmpty) { // Jika isEmpty, cetak pesan bahwa tidak ada kata slang dalam trie  
        printf("\nThere is no slang word yet in the dictionary.");  
    }  
  
    else { // Jika trie tidak kosong, panggil viewAllHelper untuk mencetak semua kata slang dalam trie  
        printf("\nList of all slang words in the dictionary:\n");  
        int number = 1;  
        char buffer[MAX_WORD];  
        viewAllHelper(root, buffer, 0, &number);  
    }  
}
```

```

int main() {
    TrieNode* root = createNode(); // Inisialisasi root node

    // Variabel-variabel yang digunakan untuk menunjang algoritma program
    int choice; int countSpace = 0;
    char newWord[50], meaning[100]; char word[50];
    bool alreadyPrinted = false;

    do { // SWITCH CASE TITLE
        printf("1. Release a new slang word\n");
        printf("2. Search a slang word\n");
        printf("3. View all slang words starting with a certain prefix word\n");
        printf("4. View all slang words\n");
        printf("5. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) { // SWITCH CASE NUMBERING
            case 1: // INPUT SLANGWORD DAN MEANING
                printf("Input a new slang word [Must be more than 1 characters and contains no space]: ");
                scanf("%s", newWord);

                // Membersihkan newline character dari input buffer
                getchar();

                while (strlen(newWord) <= 1 || strchr(newWord, ' ') != NULL) {
                    printf("Input a new slang word [Must be more than 1 characters and contains no space]: ");
                    scanf("%s", newWord);
                    while (getchar() != '\n'); // Membersihkan input buffer
                }
            }
        }
    } while (choice != 5);
}

```

words]: ");

// Input untuk meaning

printf("Input a new slang word description [Must be more than 2

scanf("%[^\n]", meaning);

// Loop sampai input untuk meaning valid

while (1) {

// Hitung jumlah spasi di meaning

int i;

for (i = 0; meaning[i] != '\0'; i++) {

if (meaning[i] == ' ')

countSpace++;

}

// Jika jumlah spasi lebih dari 1, maka input valid

if (countSpace >= 2)

break;

// Membersihkan input buffer

while (getchar() != '\n');

// Jika tidak, minta input baru

2 words]: ");

printf("Input a new slang word description [Must be more than

scanf("%[^\n]", meaning);

countSpace = 0; // Reset countSpace

}

// Menghapus newline character jika ada di meaning

if (strlen(meaning) > 0 && meaning[strlen(meaning) - 1] == '\n')

meaning[strlen(meaning) - 1] = '\0';

dalam trie

```
insert(root, newWord, meaning); // Masukkan slang word baru ke
```

```
printf("\n");  
printf("Successfully released new slang word.\n");  
printf("Press enter to continue... ");  
getchar(); getchar();  
printf("\n");  
break;
```

case 2: // CARI SLANGWORDNYA

```
printf("Input a slang word to be searched [Must be more than 1  
characters and contains no space]: ");  
  
char searchWord[50];  
scanf("%s", searchWord); // Input kata yang ingin dicari  
  
while (strlen(searchWord) <= 1 || strchr(searchWord, ' ') != NULL) {  
    printf("Input a slang word to be searched [Must be more than 1  
characters and contains no space]: ");  
    scanf("%s", searchWord);  
    while (getchar() != '\n'); // Membersihkan input buffer  
}  
  
if (search(root, searchWord, meaning)){  
    printf("\nSlang Word : %s\n", searchWord); // Jika slang word  
ditemukan, cetak maknanya  
    printf("Description : %s\n\n", meaning);  
  
    printf("Press enter to continue... ");  
    getchar(); getchar();
```

```

printf("\n");

        }

        else{ // Slangword tidak ditemukan
            printf("\nThere is no word \"%s\" in the dictionary.\n",
searchWord);

            printf("Press enter to continue... ");

            getchar(); getchar();

            printf("\n");

            }

        break;

case 3: // CARI BERDASARKAN AWALAN KATA SECARA PREFIX

    printf("Input a prefix to be searched: ");

        char prefix[50];

        scanf("%s", prefix); // Input starting word

        strcpy(globalPrefix, prefix);

        printWordsWithPrefix(root, prefix); // Cetak semua slang words dengan
prefix tertentu

        printf("\nPress enter to continue... "); globalNum = 1; alreadyPrinted =
false; // Reset globalNum dan alreadyPrinted

        getchar(); getchar();

        printf("\n");

        break;

case 4: // CARI ATAU MELIHAT SEMUA SLANGWORD YANG SUDAH DIINPUT

    viewAll(root); // Panggil fungsi viewAll

        printf("\nPress enter to continue... ");

        getchar(); getchar();

        printf("\n");

    break;

```

```

case 5: // EXIT PROGRAM

    printf("Thank you... Have a nice day :)\n");

    exit(0);

default: // DEFAULT (SELAIN 1,2,3,4,5)

    printf("Please choose a valid number!\n\n");

}

} while (choice != 5);

return 0; // AKHIR KODE
}

```

Input Example

Notes: Kode di bawah yang akan saya berikan berbahasa Indonesia. Saya tidak tau kenapa tapi saya bisa memastikan bahwa SS yang akan saya upload merupakan file asli dari kodingan yang saya berikan dan file asli dari kodingan yang saya jelaskan di atas. Saya sudah coba untuk mengubah dan otak atik terminal saya namun tetap jadinya berbahasa Indonesia ntah mengapa

Case 1:

```

1. Masukkan kata slang baru
2. Cari sebuah kata slang
3. Lihat semua kata slang yang dimulai dengan sebuah kata tertentu
4. Lihat semua kata slang
5. Keluar
Masukkan pilihan Anda: 1
Masukkan kata slang baru [Harus lebih dari 1 karakter dan tidak mengandung spasi]: daijou
Masukkan deskripsi kata slang baru [Harus lebih dari 2 kata]: super idol desaurung

Kata slang baru berhasil dimasukkan.
Tekan enter untuk melanjutkan...

```

Case 1 berhasil menginput slangword yang diinginkan beserta dengan maknanya. Jika slangword dan makna tidak sesuai dengan ketentuan maka akan meminta input ulang seperti ini:

```

1. Release a new slang word
2. Search a slang word
3. View all slang words starting with a certain prefix word
4. View all slang words
5. Exit
Enter your choice: 1
Input a new slang word [Must be more than 1 characters and contains no space]: d
Input a new slang word [Must be more than 1 characters and contains no space]: daijou
Input a new slang word description [Must be more than 2 words]: super idol
Input a new slang word description [Must be more than 2 words]: super idol desaurung

Successfully released new slang word.
Press enter to continue...

```

Case 2:

```
1. Masukkan kata slang baru
2. Cari sebuah kata slang
3. Lihat semua kata slang yang dimulai dengan sebuah kata tertentu
4. Lihat semua kata slang
5. Keluar
Masukkan pilihan Anda: 2
Masukkan kata slang yang akan dicari [Harus lebih dari 1 karakter dan tidak mengandung spasi]: daijou

Kata Slang : daijou
Deskripsi : super idol desaurung

Tekan enter untuk melanjutkan...
```

Case 2 berhasil tereksekusi, dimana saya mencari kata yang sudah saya input yakni “daijou” dan keluar hasilnya. Jika tidak ada maka akan tampil seperti ini:

```
1. Release a new slang word
2. Search a slang word
3. View all slang words starting with a certain prefix word
4. View all slang words
5. Exit
Enter your choice: 2
Input a slang word to be searched [Must be more than 1 characters and contains no space]: pu

There is no word "pu" in the dictionary.
Press enter to continue...
```

Case 3:

```
1. Masukkan kata slang baru
2. Cari sebuah kata slang
3. Lihat semua kata slang yang dimulai dengan sebuah kata tertentu
4. Lihat semua kata slang
5. Keluar
Masukkan pilihan Anda: 1
Masukkan kata slang baru [Harus lebih dari 1 karakter dan tidak mengandung spasi]: dadan
Masukkan deskripsi kata slang baru [Harus lebih dari 2 kata]: temennya si rudi

Kata slang baru berhasil dimasukkan.
Tekan enter untuk melanjutkan...

1. Masukkan kata slang baru
2. Cari sebuah kata slang
3. Lihat semua kata slang yang dimulai dengan sebuah kata tertentu
4. Lihat semua kata slang
5. Keluar
Masukkan pilihan Anda: 1
Masukkan kata slang baru [Harus lebih dari 1 karakter dan tidak mengandung spasi]: dagang
Masukkan deskripsi kata slang baru [Harus lebih dari 2 kata]: dagang di pasar

Kata slang baru berhasil dimasukkan.
Tekan enter untuk melanjutkan...
```

Pada case 3, saya menginput dulu 2 lagi kata yang berawalan “da” yakni “dadan” dan juga “dagang”. Dan jika saya mencari berdasarkan prefix dengan awalan “da” maka outputnya:

```
1. Masukkan kata slang baru
2. Cari sebuah kata slang
3. Lihat semua kata slang yang dimulai dengan sebuah kata tertentu
4. Lihat semua kata slang
5. Keluar
Masukkan pilihan Anda: 3
Masukkan awalan yang akan dicari: da

Kata-kata dimulai dengan "da":
1. dadan
2. dagang
3. daijou

Tekan enter untuk melanjutkan...
```

Case 4:

```
1. Masukkan kata slang baru
2. Cari sebuah kata slang
3. Lihat semua kata slang yang dimulai dengan sebuah kata tertentu
4. Lihat semua kata slang
5. Keluar
Masukkan pilihan Anda: 1
Masukkan kata slang baru [Harus lebih dari 1 karakter dan tidak mengandung spasi]: rizz
Masukkan deskripsi kata slang baru [Harus lebih dari 2 kata]: grizzly adalah beruang

Kata slang baru berhasil dimasukkan.
Tekan enter untuk melanjutkan...

1. Masukkan kata slang baru
2. Cari sebuah kata slang
3. Lihat semua kata slang yang dimulai dengan sebuah kata tertentu
4. Lihat semua kata slang
5. Keluar
Masukkan pilihan Anda: 1
Masukkan kata slang baru [Harus lebih dari 1 karakter dan tidak mengandung spasi]: sohib
Masukkan deskripsi kata slang baru [Harus lebih dari 2 kata]: adalah temannya tanu

Kata slang baru berhasil dimasukkan.
Tekan enter untuk melanjutkan...

1. Masukkan kata slang baru
2. Cari sebuah kata slang
3. Lihat semua kata slang yang dimulai dengan sebuah kata tertentu
4. Lihat semua kata slang
5. Keluar
Masukkan pilihan Anda: 1
Masukkan kata slang baru [Harus lebih dari 1 karakter dan tidak mengandung spasi]: tanu
Masukkan deskripsi kata slang baru [Harus lebih dari 2 kata]: adalah temannya sohib
```

Saya memasukan lagi 3 kata, sehingga jika saya menggunakan case 4 akan mengeluarkan output sebagai berikut:

1. Masukkan kata slang baru
2. Cari sebuah kata slang
3. Lihat semua kata slang yang dimulai dengan sebuah kata tertentu
4. Lihat semua kata slang
5. Keluar

Masukkan pilihan Anda: 4

Daftar semua kata slang dalam kamus:

1. dadan
2. dagang
3. daijou
4. rizz
5. sohib
6. tanu

Tekan enter untuk melanjutkan...

Case 5:

1. Masukkan kata slang baru
2. Cari sebuah kata slang
3. Lihat semua kata slang yang dimulai dengan sebuah kata tertentu
4. Lihat semua kata slang
5. Keluar

Masukkan pilihan Anda: 5

Terima kasih... Semoga harimu menyenangkan :)

Process exited after 388.4 seconds with return value 0
Press any key to continue . . .

Case 5 juga berhasil tereksekusi dengan baik, menandakan program berhasil berjalan dari case 1 dan ditutup pada case 5