

No. Program	Baris Program	Petikan Source Code	Penjelasan
1	5-10	c struct Node { int data; struct Node *next; struct Node *prev; };	Deklarasi struktur baru dengan nama node (simpul). Next dan prev adalah variable pointer yang akan digunakan untuk mengarahkan ke simpul sebelum atau setelah sebuah simpul baru dibuat.
1	12-14	c void push(struct Node** head_ref, int new_data) { struct Node* new_node = (struct Node*)malloc(sizeof(struct Node)); new_node->data = new_data; new_node-> next = (*head_ref); new_node->prev = NULL; if ((*head_ref) != NULL) (*head_ref)->prev = new_node; (*head_ref) = new_node; }	Fungsi push digunakan untuk menyisipkan node baru di depan linked list ganda berantai. Ini dilakukan dengan mengalokasikan memori untuk node baru, mengatur data di dalamnya, menetapkan node berikutnya sebagai kepala (head) dari linked list, dan memperbarui referensi ke node sebelumnya menjadi NULL.
1	16-21	c void printList(struct Node* node) { struct Node* last; printf("\nTraversal in forward direction \n"); while (node != NULL) { printf(" %d ", node->data); last = node; node = node-> next; } printf("\nTraversal in reverse direction \n"); while (last != NULL) { printf(" %d ", last->data); last = last->prev; } }	Fungsi printList digunakan untuk mencetak linked list ganda berantai ke layar, terlebih dahulu dalam urutan maju dan kemudian dalam urutan mundur.
1	23-29	c int main() { struct Node* head = NULL; push(&head, 6); push(&head, 5); push(&head, 2); printf("Created DLL is: "); printList(head); getchar(); return 0; }	Pada fungsi main, beberapa node baru disisipkan di depan linked list, kemudian linked list tersebut dicetak untuk memverifikasi keberhasilan penyisipan.

2	5-10	<pre>c struct Node { int data; struct Node *next; struct Node *prev; };</pre>	Deklarasi struktur baru dengan nama node (simpul). Next dan prev adalah variable pointer yang akan digunakan untuk mengarahkan ke simpul sebelum atau setelah sebuah simpul baru dibuat.
2	12-20	<pre>c void push(Node** head_ref, int new_data) { Node* new_node = new Node(); new_node-&gt;data = new_data; new_node-&gt; next = (*head_ref); new_node-&gt;prev = NULL; if ((*head_ref) != NULL) (*head_ref)-&gt;prev = new_node; (*head_ref) = new_node; }</pre>	Fungsi push digunakan untuk menyisipkan node baru di depan linked list ganda berantai. Ini dilakukan dengan mengalokasikan memori untuk node baru, mengatur data di dalamnya, menetapkan node berikutnya sebagai kepala (head) dari linked list, dan memperbarui referensi ke node sebelumnya menjadi NULL.
2	22-34	<pre>c void insertAfter(struct Node* prev_node, int new_data) { if (prev_node == NULL) { printf("the given previous node cannot be NULL"); return; } struct Node* new_node = (struct Node*)malloc(sizeof(struct Node)); new_node-&gt;data = new_data; new_node-&gt; next = prev_node-&gt;next; prev_node-&gt;next = new_node; new_node-&gt; prev = prev_node; if (new_node-&gt;next != NULL) new_node-&gt;next-&gt;prev = new_node; }</pre>	Fungsi insertAfter digunakan untuk menyisipkan node baru setelah node tertentu dalam linked list. Ini dilakukan dengan mengalokasikan memori untuk node baru, mengatur data di dalamnya, menetapkan node berikutnya dari node sebelumnya sebagai node berikutnya dari node baru, dan memperbarui pointer next dan prev dari node sebelumnya dan node berikutnya sesuai dengan node baru.

2	36-42	<pre>c int main() {     struct Node* head = NULL;     push(&amp;head, 6);     push(&amp;head, 5);     push(&amp;head, 2);     insertAfter(head-&gt;next, 5);     printf("Created DLL is: ");     printList(head); getchar();     return 0; }</pre>	Dalam fungsi main, setelah beberapa node baru disisipkan di depan linked list, sebuah node baru juga disisipkan setelah node kedua, kemudian linked list dicetak untuk memverifikasi hasilnya.
3	5-10	<pre>c struct Node { int data;     struct Node *next; struct     Node *prev; }</pre>	Deklarasi struktur baru dengan nama node (simpul). Next dan prev adalah variable pointer yang akan digunakan untuk mengarahkan ke simpul sebelum atau setelah sebuah simpul baru dibuat.
3	12-21	<pre>c void push(Node**     head_ref, int new_data) {     Node* new_node = new     Node(); new_node-&gt;data =     new_data; new_node-     &gt;next = (*head_ref);     new_node-&gt;prev = NULL;     if ((*head_ref) != NULL)         (*head_ref)-&gt;prev =         new_node; (*head_ref) =         new_node; }</pre>	Fungsi push digunakan untuk menyisipkan node baru di depan linked list ganda berantai. Ini dilakukan dengan mengalokasikan memori untuk node baru, mengatur data di dalamnya, menetapkan node berikutnya sebagai kepala (head) dari linked list, dan memperbarui referensi ke node sebelumnya menjadi NULL.
3	23-47	<pre>c void append(struct     Node** head_ref, int     new_data) { struct Node*     new_node = (struct     Node*)malloc(sizeof(struct     Node)); struct Node* last =     *head_ref; new_node-     &gt;data = new_data;     new_node-&gt;next = NULL;     if (*head_ref == NULL) {         new_node-&gt;prev = NULL;         *head_ref = new_node;</pre>	Fungsi append digunakan untuk menyisipkan node baru di akhir linked list. Ini dilakukan dengan

		return; } while (last->next != NULL) last = last->next; last->next = new_node; new_node->prev = last; }	
4	1-4	c // A linked list node struct Node { int data; struct Node* next; struct Node* prev; };	Struktur Node digunakan untuk merepresentasikan simpul dalam linked list ganda.
4	16-15	c void push(struct Node** head_ref, int new_data) { struct Node* new_node = (struct Node*)malloc(sizeof(struct Node)); new_node->data = new_data; new_node->next = (*head_ref); new_node->prev = NULL; if ((*head_ref) != NULL) (*head_ref)->prev = new_node; (*head_ref) = new_node; }	Fungsi push digunakan untuk menyisipkan node baru di depan linked list ganda berantai.
4	17-29	c void insertBefore(struct Node** head_ref, struct Node* next_node, int new_data) { if (next_node == NULL) { printf("the given next node cannot be NULL"); return; } struct Node* new_node = (struct Node*)malloc(sizeof(struct Node)); new_node->data = new_data; new_node->prev = next_node->prev; new_node->next = next_node; if (new_node->prev != NULL) new_node->prev->next = new_node; else (*head_ref) = new_node; next_node->prev = new_node; }	Fungsi insertBefore digunakan untuk menyisipkan node baru sebelum node tertentu dalam linked list ganda.
4	31-48	int main() { struct Node* head = NULL; push(&head, 7); push(&head, 1); push(&head, 4); insertBefore(&head, head->next, 8); printf("Created DLL is: "); printList(head); getchar(); return 0; }	Fungsi main yang menunjukkan penggunaan operasi penyisipan di depan dan sebelum node tertentu dalam linked list ganda.


Nama : M.Gilang Mulya Putra

NIM : 23343073

Prodi : Informatika