1 12-14	ct Node { int data; Node *next; struct *prev; };	Penjelasan Deklarasi struktur
1 12-14 c void head struct (struct Node Node Node new_>next new_if ((*heat new_new_new_new_new_new_if (remains the new_new_new_new_new_new_new_new_new_new_		_ = = = = = = = = = = = = = = = = = = =
1 12-14 c void head struct (struct Node Node Node new_>next new_if ((*head new_new_new_new_new_new_new_new_new_new_	*prev; };	baru dengan nama
1 16-21 c voic Node new_ snew_ if ((*heat new_ new_ new_ new_ new_ new_ new_ new_		node (simpul). Next
1 16-21 c voic Node new_ snew_ if ((*heat new_ new_ new_ new_ new_ new_ new_ new_		dan prev adalah
1 16-21 c voic Node new_ snew_ if ((*heat new_ new_ new_ new_ new_ new_ new_ new_		variable pointer yang
1 16-21 c voic Node new_ snew_ if ((*heat new_ new_ new_ new_ new_ new_ new_ new_		akan digunakan untuk
1 16-21 c voic Node new_ snew_ if ((*heat new_ new_ new_ new_ new_ new_ new_ new_		mengarahkan ke
1 16-21 c voic Node new_ snew_ if ((*heat new_ new_ new_ new_ new_ new_ new_ new_		simpul sebelum atau
1 16-21 c voic Node new_ snew_ if ((*heat new_ new_ new_ new_ new_ new_ new_ new_		setelah sebuah simpul
1 16-21 c voic Node new_ snew_ if ((*heat new_ new_ new_ new_ new_ new_ new_ new_		baru dibuat.
1 16-21 c voic Node new_ snew_ if ((*heat new_ new_ new_ new_ new_ new_ new_ new_	push(struct Node**	Fungsi push
1 16-21 c voic Node new_ >next new_ if ((**heat new_ new_ new_ new_ new_ new_ while printf last = >next in rev while printf last = 1 23-29 c int to the control of the cont	ref, int new_data) {	digunakan untuk
1 16-21 c void Node new_ if ((*heat new_ new_ new_ new_ new_ new_ new_ new_	: Node* new_node =	menyisipkan node
1 16-21 c voice Node new_ if ((*heat new_ new_ new_ new_ new_ new_ new_ new_	-	baru di depan linked
1 16-21 c voice Node new_ if ((*heat new_ new_ new_ new_ new_ new_ new_ new_	*)malloc(sizeof(struct	list ganda berantai. Ini
1 16-21 c void Node Node Printf forwa while printf last = >next in rew while printf last = 1 23-29 c int i)); new_node->data =	dilakukan dengan
	data; new_node-	mengalokasikan
1 16-21 c voice Node Node printf forward while printf last = >next in revented by the printf last = 1 23-29 c int is set in the printf last = 1 c int is set in the printf las	= (*head_ref);	memori untuk node
1 16-21 c voice Node Node Prints forward while prints last = >next in revenue while prints last = 1 23-29 c int s	node->prev = NULL;	baru, mengatur data
1 16-21 c void Node Node Printf forwa while printf last = >next in rev while printf last = 1 23-29 c int i	ead_ref) != NULL)	di dalamnya,
1 16-21 c void Node Node printf forwa while printf last = >next in rev while printf last = 1 23-29 c int i	d_ref)->prev =	menetapkan node
1 16-21 c voice Node Node Node prints forward while prints last = >next in revenue while prints last = 1 23-29 c int s	node; (*head_ref) =	berikutnya sebagai
1 16-21 c voice Node Node Node prints forward while prints last = >next in revenue while prints last = 1 23-29 c int s	node; }	kepala (head) dari
Node Node printf forwa while printf last = >next in rev while printf last = 1 23-29 c int i	, ,	linked list, dan
Node Node printf forwa while printf last = >next in rev while printf last = 1 23-29 c int i		memperbarui
Node Node printf forwa while printf last = >next in rev while printf last = 1 23-29 c int i		referensi ke node
Node Node printf forwa while printf last = >next in rev while printf last = 1 23-29 c int i		sebelumnya menjadi
Node Node printf forwa while printf last = >next in rev while printf last = 1 23-29 c int i		NULL.
Node printf forward while printf last = >next in review while printf last = 1 23-29 c int is second to the printf last = 1 c int is sec	printList(struct	Fungsi printList
printf forwa while printf last = >next in rev while printf last = 1 23-29 c int i	* node) { struct	digunakan untuk
forware while printf last = >next in rev while printf last = 1 23-29 c int to the control of the	* last;	mencetak linked list
while printf last = >next in rev while printf last = 1 23-29 c int is seen to be a	("\nTraversal in	ganda berantai ke
printf last = >next in rev while printf last = 1	rd direction \n");	layar, terlebih dahulu
last = >next in rev while printf last = 1 23-29 c int i	(node != NULL) {	dalam urutan maju
>next in rev while printf last = 1 23-29 c int i	(" %d ", node->data);	dan kemudian dalam
in rev while printf last = 1 23-29 c int i	node; node = node-	urutan mundur.
while printf last = 1 23-29 c int i	; } printf("\nTraversal	
printf last = 1 23-29 c int i	erse direction \n");	
1 23-29 c int i	(last != NULL) {	
1 23-29 c int i	(" %d ", last->data);	
	last->prev; } }	
la a a al	nain() { struct Node*	Pada fungsi main,
nead	= NULL; push(&head,	beberapa node baru
	sh(&head, 5);	disisipkan di depan
	&head, 2);	linked list, kemudian
	("Created DLL is: ");	linked list tersebut
	ist(head); getchar();	dicetak untuk
retur		memverifikasi
	. -	keberhasilan
		penyisipan.
printl printl	("Created DLL is: "); .ist(head); getchar();	linked list tersebut dicetak untuk memverifikasi

	F 40	and the state of t	Billion
2	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.
2	12-20	c void push(Node** head_ref, int new_data) { Node* new_node = new 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.
2	22-34	c void insertAfter(struct Node* prev_node, int new_data) { if (prev_node	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.

	26.42		
2	36-42	c int main() { struct Node* head = NULL; push(&head, 6); push(&head, 5); push(&head, 2); insertAfter(head->next, 5); printf("Created DLL is: "); printList(head); getchar(); return 0; }	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	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.
3	12-21	c void push(Node** head_ref, int new_data) { Node* new_node = new 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.
3	23-47	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- >data = new_data; new_node->next = NULL; if (*head_ref == NULL) { new_node->prev = NULL; *head_ref = new_node;	Fungsi append digunakan untuk menyisipkan node baru di akhir linked list. Ini dilakukan dengan

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

Nama : M.Gilang Mulya Putra

NIM: 23343073

Prodi : Informatika