**JOBSHEET 4 DOUBLE LINK LIST**

**PRAKTIKUM STRUKTUR DATA**



**OLEH :**

**Muhammad Zhidny Ilman**

**22343062**

**PRODI INFORMATIKA**

**JURUSAN ELEKTRONIKA  
FAKULTAS TEKNIK**

**UNIVERSITAS NEGERI PADANG**  
**2022**

1. Analisis dan buatlahpenjelasan 4 program diatasdengan format berikut. File dikumpulkandalam format .PDF

(Catatan: Tidakperlumenjelaskansetiap baris program, cukuphalutamadaripemahamanandaterhadap program ataukumpulandari baris program).

Berilah minimal 5 penjelasanuntuk masing-masing program (total 20). Perhatikancontohdibawah!

|  |  |  |  |
| --- | --- | --- | --- |
| **NOMOR PROGRAM** | **BARIS PROGRAM** | **PETIKAN SOURCE CODE** | **PENJELASAN** |
| **1** | 5-10 | struct Node  {  int data;  struct Node \*next; // Pointer to next node  struct Node \*prev; // Pointer to previous node  }; | Deklarasistrukturbarudengannama node ( simpul ). Next dan Prevadalahvariabel pointer yang akandigunakanuntukmengarahkankesimpulsebelumatausetelahsebuahsimpulbarudibuat |
| **2** | 12-26 | void push(struct Node\*\* head\_ref, int new\_data)  {  /\* 1. allocate node \*/  struct Node\* new\_node = (struct Node\*)malloc(sizeof(struct Node));  /\* 2. put in the data \*/  new\_node->data = new\_data;  /\* 3. Make next of new node as head and previous as NULL \*/  new\_node->next = (\*head\_ref);  new\_node->prev = NULL;  /\* 4. change prev of head node to new node \*/  if ((\*head\_ref) != NULL)  (\*head\_ref)->prev = new\_node;  /\* 5. move the head to point to the new node \*/  (\*head\_ref) = new\_node;  } | Deklarasi***void***dengannama push ( memasukkanelemenstruktur ).  1. allocate node  Mengalokasimemoriuntuk node  2. put in the data  Perintahuntukmelakukansebuah Tindakan atauaksi yang diminta oleh user  3. make next of new node as head  Menambah node dan membuat tail dari single lins  4. change prev of head node to new  Menggantilanjutandarikepala node menjadibaru  5. move the head to point to the new node  Menunjukkan pada node pertamadidalam linked list |
| **3** | 27-41 | 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;  }  } | Fungsiiniberfungsiuntukmencetakisidari linked list yang diberikan, dengan traversing ataupenjelajahandariawalhinggaakhir dan dariakhirhinggaawal.  Pertama, fungsimencetak***"Traversal in forward direction"***, dan kemudianmenggunakan while ***loop***untukmelakukanpenjelajahandariawalkeakhir linked list. Selama traversing, data darisetiap node dicetakdenganmenggunakan***printf()*** function, dan node berikutnyadiaksesmenggunakan pointer ***node->next***. Jika sudahmencapai node terakhir, pointer last diisidengan node terakhirtersebut.  Karena linked list dapatberisibanyak node, makafungsiprintListdapatdipanggiluntukmengecekapakah linked list yang dibuatsudahberisi data yang diharapkanatautidak. |
| **4** | 37-39 | while (last != NULL) {  printf(" %d ", last->data);  last = last->prev; | Code iniadalahimplementasidari loop while yang digunakanuntukmencetaksemua data dalam linked list dariakhirkeawal (dari tail ke head). |
| **5** | 25-55 | int main()  {  /\* Start with the empty list \*/  struct Node\* head = NULL;  push(&head, 6);  push(&head, 5);  push(&head, 2);  printf("Nama : Muhammad Zhidny Ilman\n");  printf("Nim : 22343062\n\n");  printf("Created DLL is: ");  printList(head);  getchar();  return 0;  } | implementasidari main program yang menggunakan linked list untukmembuat double linked list denganbeberapa node, mencetak data yang ada pada linked list kelayarmenggunakanfungsiprintList(), dan mencetaknama dan NIM penulis pada layar. |
|  |  | **PERCOBAAN 1** |  |
| **6** | 12-30 | void push(struct Node\*\* head\_ref, int new\_data)  {  /\* 1. allocate node \*/  struct Node\* new\_node = (struct Node\*)malloc(sizeof(struct Node));  /\* 2. put in the data \*/  new\_node->data = new\_data;  /\* 3. Make next of new node as head and previous as NULL \*/  new\_node->next = (\*head\_ref);  new\_node->prev = NULL;  /\* 4. change prev of head node to new node \*/  if ((\*head\_ref) != NULL)  (\*head\_ref)->prev = new\_node;  /\* 5. move the head to point to the new node \*/  (\*head\_ref) = new\_node;  } | * Membuat node barudenganmenggunakanmalloc() untukmengalokasikanruangmemori yang baru. * Menempatkan data barukedalam node tersebutdenganmenggunakannew\_data. * Mengatur next dari node barusebagaihead\_ref, dan previous sebagai NULL. * Mengubahprevdarihead\_refmenjadi node barujikahead\_reftidak NULL. * Memindahkanhead\_refke node baru, sehingga node barusekarangmenjadi head.   Dengandemikian, setiap kali fungsi***push()***dipanggil, node baruakanditambahkankeawal linked list dan menjadi head baru. |
| **7** | 31 | void insertAfter(struct Node\* prev\_node, int new\_data) | insertAfteradalahfungsi yang digunakanuntukmenambahkansebuah node barusetelah node tertentu( yaitu***prev\_node*** ) dalamsebuah double linked list |
| **8** | 34-37 | if (prev\_node == NULL) {  printf("the given previous node cannot be NULL");  return; | Kondisi yang digunakanfungsi***InsertAfter***untukmengecekapakah***prev\_node***diberikan oleh penggunatidak***NULL***ataukosong |
| **9** | 61-64 | printf("\nTraversal in reverse direction \n");  while (last != NULL) {  printf(" %d ", last->data);  last = last->prev; | Code initermasukdaribagianfungsi***printListReverse*** yang digunakanuntukmencetakisidarisebuah double linked list secaraterbaik, yaitudariakhirkeawal |
| **10** | 70-74 | struct Node\* head = NULL;  push(&head, 6);  push(&head, 5);  push(&head, 2);  insertAfter(head->next->prev, 5); | Inimerupakancontohpemanggilan function / fungsi pada program yang menggunakan linked list, sebagaicontoh :   * Sebuahvariabel head daritipe struct Node dideklarasikan dan diinisialisasidengan NULL. * Kemudian, fungsi push dipanggiltiga kali untukmenambahkan node barukedalam linked list, dengan masing-masing nilai data 6, 5, dan * Setelahitu, fungsiinsertAfterdipanggiluntukmenambahkan node barudengannilai data 5 setelah node keduadalam linked list. Dalamhalini, nilai parameter prev\_nodediisidengan head->next->prev, yang samadengan NULL karena node pertamadalam linked list tidakmemiliki node sebelumnya. * Secarakeseluruhan, program inimenunjukkancontohbagaimanamembuat linked list denganmenambahkanbeberapa node, dan juga contohbagaimanamenambahkan node barusetelah node yang telahadadalam linked list. |
|  |  | **PERCOBAAN 2** |  |
| **11** | 21-26 | /\* 4. change prev of head node to new node \*/  if ((\*head\_ref) != NULL)  (\*head\_ref)->prev = new\_node;  /\* 5. move the head to point to the new node \*/  (\*head\_ref) = new\_node;  } | Beberapabagiandari function / fungsi***push*** pada program yang menggunakan double linked list, fungsi***push*** juga digunakanuntukmenambahkan node barukeawal linked list |
| **12** | 27 | void append(struct Node\*\* head\_ref, int new\_data) | Program yang menggunakan double linked list dan befungsiuntukmenambahkan node bar uke akhir linked list, fungsi***append***menerima dua parameter, yaitualamatdari pointer ke ***head***node data yang akandimasukkankedalam node baru |
| **13** | 30-31 | struct Node\* new\_node = (struct Node\*)malloc(sizeof(struct Node));  struct Node\* last = \*head\_ref; | Tujuandari code iniadalahuntukmenambahkan node bar uke linked list, sebagaicontoh :   * Pada baris pertama, dilakukanalokasimemoriuntukmembuat node baru pada linked list. Fungsi***malloc***akanmengalokasikanblokmemori yang baruuntukmenampung node baru yang dibuat. ***sizeof(struct Node)***menunjukkanukuranmemori yang dialokasikanuntukmenyimpaninformasidari node baru. * Pada baris kedua, ***last*** diaturuntukmenunjukke***head***dari linked list yang dilewatkansebagaiargumendalam parameter fungsi***push.*** Pointer ***last*** akandigunakanuntukmenelusuri linked list hinggamencapai node terakhir pada linked list. |
| **14** | 45-51 | while (last->next != NULL)  last = last->next;  /\* 6. Change the next of last node \*/  last->next = new\_node;  /\* 7. Make last node as previous of new node \*/  new\_node->prev = last;  return; | Bergunauntukmenambhkan node barukeakhir linked list pada program yang menggunakan double linked list |
| **15** | 70-82 | /\* Start with the empty list \*/  struct Node\* head = NULL;  // Insert 6. So linked list becomes 6->NULL  append(&head, 8);  // Insert 7 at the beginning. So  // linked list becomes 7->6->NULL  push(&head, 9);  // Insert 1 at the beginning. So  // linked list becomes 1->7->6->NULL  push(&head, 2);  // Insert 4 at the end. So linked  // list becomes 1->7->6->4->NULL  append(&head, 5); | Menggunakan double linked list untukmenambahkan node baru pada linked list, yang pada awalnya linked list kosongdan  ***head***diaturke***NULL,***kemudiandilakukanbeberapaoperasipenambahan node baru pada linked list mengunakanfungsi ***append***  dan  ***push***  Linked list tersebutakanmemiliki node dengannilai 1 > 7 > 9 > 6 > 5 |
|  |  | **PERCOBAAN 3** |  |
| **16** | 22 | void insertBefore(struct Node\*\* head\_ref, struct Node\* next\_node, int new\_data) | InsertBeforemerupakansebuahfungsiuntukmenyisipkansebuah node barudengannilaitertentusebelum node diberikut***next\_node*** pada linked list yang diberikan***head\_ref*** |
| **17** | 30 | struct Node\* new\_node = (struct Node\*)malloc(sizeof(struct Node)); | Melakukanalokasimemoridinamismenggunakan function ***malloc***untukmembuatsebuah node baru pada double linked list, variabel***new\_node***menyimpanalamatmemori yang dialokasikan dan akandiisidengfannilai data dan pointer next dan prevsesuaidengan input |
| **18** | 47-54 | 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; | Fungsi***printList***digunakanuntukmenampilkanisiataunilaidarisetiap node pada doubly linked list, dimulaidari head hingga tail.   * fungsiinimembuatvariabel***last***bertipe***struct Node\****, yang akandigunakanuntukmenyimpanalamat node terakhir pada doubly linked list. * Kemudian, fungsiinimelakukaniterasidari head hingga tail menggunakan while loop. Setiap node dicetakdatanyakelayarmenggunakanfungsi***printf***, kemudianvariabel***last***diisidenganalamat node saatini dan node saatinidipindahkanke node berikutnyadenganmengakses pointer ***next***. * Setelahiterasiselesai, fungsiakanmencetakpesan "Traversal in forward direction" sebagaiindikasibahwa proses cetak telah selesa |
| **19** | 56-59 | printf("\nTraversal in reverse direction \n");  while (last != NULL) {  printf(" %d ", last->data);  last = last->prev; | * Fungsimencetakisi linked list secaraiteratifdariawaldenganmenggunakanvariabel***node***sebagai pointer ke node pertama pada linked list dan melakukaniterasiterus-menerussampaimenemukan node yang bernilai NULL (akhirdari linked list). * Kemudian, pada setiapiterasi, fungsimencetaknilai data dari node yang ditunjuk oleh ***node*** dan memindahkan***node***ke***node***berikutnyadengan***node = node->nex***t. * Setelahmencetakisi linked list secaraiteratifdariawalkeakhir, fungsikemudianmencetakisi linked list daribelakangkedepansecaraiteratif. Untukmencetakdaribelakangkedepan, fungsimenggunakanvariabel***last*** yang menunjukke node terakhir pada linked list dan melakukaniterasimundurdenganmemindahkan***last***ke***node***sebelumnyadengan***last = last->prev*** dan mencetaknilai data dari node yang ditunjuk oleh ***last*** pada setiap iterasi. |
| **20** | 67-71 | struct Node\* head = NULL;  push(&head, 7);  push(&head, 1);  push(&head, 4);  insertBefore(&head, head->next, 8); | Kode tersebutmenginisialisasisebuah pointer ***head*** yang menunjuk pada ***NULL***, kemudian 3 node barudimasukkanke linked list menggunakanfungsi***push()***, sehingga linked list-nyamenjadi: 4 <-> 1 <-> 7.  Kemudian, fungsi***insertBefore()***dipanggildengan argument ***&head***sebagai head node, ***head->next***sebagai node setelah node yang baruingindimasukkan, dan 8 sebagai data dari node yang ingindimasukkan.  Fungsi***insertBefore()***akanmemasukkan node barusebelum node yang ditunjuk oleh next\_node, sehinggasetelahpemanggilanfungsi, linked list-nyamenjadi: 4 <-> 8 <-> 1 <-> 7. |
|  |  | **PERCOBAAN 4** |  |