



FIKOM



MODUL PRAKTIKUM STRUKTUR DATA

20
24
20
25

Tim Penyusun

- Lutfi Budi Ilmawan,
S.Kom., M.Cs., MTA.
- Amaliah Faradibah,
S.Kom., M.Kom., MTA.,
MCF
- Ramdaniah, S.Kom.,
M.T., MTA.
- Syariful Mujaddid,
S.Kom., M.T.
- Tim Asisten Laboratorium

KATA PENGANTAR

Puji syukur ke hadirat Tuhan Yang Maha Kuasa, yang telah memberikan rahmat-Nya sehingga Modul Praktikum **Struktur Data** untuk mahasiswa/i Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Muslim Indonesia ini dapat diselesaikan dengan sebaik-baiknya.

Modul praktikum ini dibuat sebagai pedoman dalam melakukan kegiatan praktikum **Struktur Data** yang merupakan kegiatan penunjang mata kuliah pada Program Studi Teknik Informatika. Modul praktikum ini diharapkan dapat membantu mahasiswa/i dalam mempersiapkan dan melaksanakan praktikum dengan lebih baik, terarah, dan terencana. Pada setiap topik telah ditetapkan capaian pembelajaran mata kuliah pelaksanaan praktikum dan semua kegiatan yang harus dilakukan oleh mahasiswa/i serta teori singkat untuk memperdalam pemahaman mahasiswa/i mengenai materi yang dibahas.

Penyusun menyakini bahwa dalam pembuatan Modul Praktikum **Struktur Data** ini masih jauh dari sempurna. Oleh karena itu penyusun mengharapkan kritik dan saran yang membangun guna penyempurnaan modul praktikum ini dimasa yang akan datang.

Akhir kata, penyusun mengucapkan banyak terima kasih kepada semua pihak yang telah membantu baik secara langsung maupun tidak langsung.

Makassar, September 2024

Tim Penyusun

TATA TERTIB PELAKSANAAN PRAKTIKUM

Tata Tertib Pelaksanaan Praktikum pada Laboratorium Terpadu Fakultas Ilmu Komputer UMI adalah sebagai berikut:

1. Seluruh Pengguna laboratorium harus dalam keadaan sehat tidak menunjukkan gejala sakit (batuk, hidung tersumbat, dan suhu badan diatas 37°C).
2. Praktikan hanya diizinkan melaksanakan praktikum apabila :
 - a. Pria
 - Berpakaian rapi memakai kemeja putih polos;
 - Menggunakan celana kain berwarna hitam bukan dari bahan jeans/semi jeans;
 - Rambut rapi dan tidak panjang;
 - b. Wanita
 - Berpakaian rapi memakai kemeja tunik putih polos (tidak transparan)
 - Memakai Jilbab Segitiga Hitam (bukan pasmina) dan menutupi dada.
 - Menggunakan Rok Panjang berwarna hitam yang tidak terbelah dan tidak span serta bukan dari bahan jeans/semi jeans;
 - Memakai kaos kaki dengan tinggi minimal 10 cm di atas mata kaki;
3. Ketika memasuki dan selama berada dalam ruangan, praktikan diwajibkan :
 - Tenang, tertib, dan sopan;
 - Tidak mengganggu praktikan lain yang sedang melaksanakan praktikum;
 - Tidak diperbolehkan merokok, membawa makanan / minuman senjata tajam dan senjata api ke dalam ruangan praktikum;
 - Tidak diperbolehkan membawa *handphone* ke meja praktikum dan *handphone* dalam mode senyap;
 - Tidak diperbolehkan membawa media penyimpanan eksternal atau *flashdisk* ke meja praktikum tanpa seizin Dosen Pengampu atau Asisten;
4. Dilarang membawa, mengambil, serta memindahkan perangkat yang digunakan pada saat praktikum tanpa instruksi dari Dosen Pengampu atau Asisten.
5. Toleransi keterlambatan praktikan maksimal 5 menit.
6. Praktikan berada diarea laboratorium dengan mengikuti jadwal yang telah ditentukan oleh Kepala Laboratorium.

7. Penggunaan fasilitas Laboratorium menyesuaikan dengan kapasitas ruang Laboratorium.
8. Segala pelanggaran yang dilakukan oleh praktikan akan berakibat pada penutupan dan penghentian penggunaan seluruh fasilitas laboratorium dan ditindak sesuai dengan aturan yang berlaku.

SANKSI-SANKSI

Sanksi terhadap pelanggaran **TATA TERTIB**:

Dosen Pengampu dan Asisten laboratorium berhak menjatuhkan sanksi, sesuai dengan aturan yang berlaku di Laboratorium Terpadu Fakultas Ilmu Komputer UMI apabila :

1. Praktikan merusak peralatan praktikum (*Personal Computer*) secara sengaja, maka praktikan bertanggung jawab untuk mengganti kerusakan tersebut.
2. Praktikan tidak mematuhi dan mentaati aturan praktikum maka tidak diperkenankan mengikuti praktikum.

Pelanggaran point lainnya dikenakan sanksi teguran, dikeluarkan/dicoret namanya dalam kegiatan praktikum (mengulang mata kuliah sesuai dengan semester berjalan) sampai sanksi akademik.



Kepala Laboratorium Terpadu,

Ir. Abdul Rachman Manga', S.Kom., M.T., MTA., MCF

DAFTAR ISI

KATA PENGANTAR	2
TATA TERTIB PELAKSANAAN PRAKTIKUM.....	3
DAFTAR ISI.....	5
MODUL 1 – STACK DAN QUEUE.....	6
MODUL 2 – SINGLE LINKED LIST	10
MODUL 3 – DOUBLE LINKED LIST	14
MODUL 4 – BINARY SEARCH TREE.....	19

MODUL 1 – STACK DAN QUEUE**A. Capaian Pembelajaran Mata Kuliah (CPMK)**

Mahasiswa mampu mengimplementasikan Stack dan Queue menggunakan Struktur Data Sederhana.

B. Sub Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu mengimplementasikan Stack menggunakan struktur data Array.
2. Mahasiswa mampu mengimplementasikan Queue menggunakan struktur data Array.

C. Teori Dasar**a. Stack**

Struktur Stack :

```
struct Stack{  
    int elemen[MAX_STACK];  
    int top;  
};
```

Pembuatan Stack baru:

```
void createStack(Stack &stack){  
    stack.top = -1;  
}
```

Pengecekan jika stack kosong:

```
bool isEmpty(Stack stack){  
    return stack.top == -1;  
}
```

Pengecekan jika stack penuh:

```
bool isFull(Stack stack){  
    return stack.top == MAX_STACK-1;  
}
```

Proses Push:

```
void push(Stack &stack, int i){  
    stack.top++;  
    stack.elemen[stack.top] = i;  
}
```

Proses pop:

```
void pop(Stack &stack, int &i){  
    i = stack.elemen[stack.top];  
    stack.top--;  
}
```

b. Queue**Struktur Queue :**

```
struct Queue{
    int elemen[1000];
    int rear;
    int front;
    int jml;
};
```

Pembuatan Queue baru:

```
void createQ(Queue &q){
    q.front = 0;
    q.rear = -1;
    q.jml = 0;
}
```

Pengecekan jika queue penuh:

```
bool isFull(Queue q){
    return q.jml == MAX_QUEUE;
}
```

Pengecekan jika queue kosong:

```
bool isEmpty(Queue q){
    return q.jml == 0;
}
```

Proses enqueue:

```
void enQ(Queue &q, int i){
    if(isFull(q)) return;

    q.rear++;
    q.jml++;
    q.elemen[q.rear] = i;
}
```

Proses dequeue:

```
void deQ(Queue &q, int &i){
    if(isEmpty(q)) return;

    i = q.elemen[q.front];
    q.front++;
    q.jml--;
}
```

D. Kegiatan Praktikum**1. Instrument**

- a) Perangkat komputer
- b) Sistem Operasi Windows

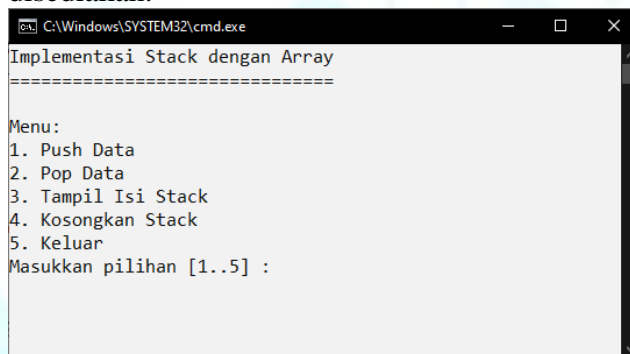
- c) C++ Compiler (MinGW)
- d) Text Editor (Geany, Notepad++, Visual Studio Code, SublimeText, dll)

2. Prosedur

- a) Baca dan pahami semua tahapan praktikum dengan cermat.
- b) Gunakan fasilitas yang disediakan dengan penuh rasa tanggung jawab.
- c) Rapihkan kembali setelah menggunakan komputer (mouse, keyboard, kursi, dll)
- d) Perhatikan sikap anda untuk tidak mengganggu rekan praktikan lain.
- e) Pastikan diri anda tidak menyentuh sumber listrik.

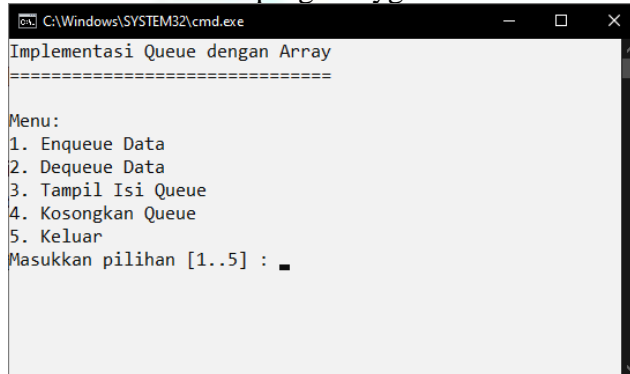
3. Studi Kasus

- a. Buat program untuk implementasi stack menggunakan struktur data array dengan kapasitas $n=5$. Output dapat dilihat pada Gambar di bawah. Demo program dapat diakses di [sini](#). Untuk lebih jelasnya, silakan buka demo program yg telah disediakan.



```
C:\Windows\SYSTEM32\cmd.exe
Implementasi Stack dengan Array
=====
Menu:
1. Push Data
2. Pop Data
3. Tampil Isi Stack
4. Kosongkan Stack
5. Keluar
Masukkan pilihan [1..5] :
```

- b. Buat program untuk implementasi queue menggunakan struktur data array dengan kapasitas $n=5$. Output dapat dilihat pada Gambar di bawah. Untuk lebih jelasnya, silakan buka demo program yg telah disediakan.



```
C:\Windows\SYSTEM32\cmd.exe
Implementasi Queue dengan Array
=====
Menu:
1. Enqueue Data
2. Dequeue Data
3. Tampil Isi Queue
4. Kosongkan Queue
5. Keluar
Masukkan pilihan [1..5] :
```


LEMBAR EVALUASI PRAKTIKUM

Evaluasi Praktikum 1 :

No	Indikator	Skor Penilaian				
		Sangat Kurang (E) ≤ 40	Kurang (D) 41-55	Cukup (C) 55-65	Baik (B) 66-85	Sangat Baik (A) ≥ 86
1.						
2.						
3.						
4.						

Catatan Asisten :

Dosen : _____

Asisten 1 : _____

Asisten 2 : _____

TINGKAT KEBERHASILAN

MODUL 2 – SINGLE LINKED LIST

A. Capaian Pembelajaran Mata Kuliah (CPMK)

Mahasiswa mampu mengimplementasikan struktur data dinamis single linked list.

B. Sub Capaian Pembelajaran Mata Kuliah (CPMK)

Mahasiswa mampu mengimplementasikan struktur data single linked list, dengan operasi meliputi:

1. Pembentukan node awal
2. Penambahan node pada posisi depan, tengah, dan belakang.
3. Penghapusan node pada posisi depan, tengah, dan belakang.

C. Teori Dasar

1. Struktur Node :

```
struct Node{
    int data;
    Node *next;
};
```

```
Node *n=NULL, *head=NULL, *tail=NULL, *x=NULL;
```

2. Pembuatan Node Awal:

```
void buatNodeAwal(int i){
    n = new Node;
    n->data = i;
    tail = n;
    head = n;
    tail->next = NULL;
}
```

3. Penambahan node pada posisi belakang:

```
void tambahDiBelakang(int i){
    n = new Node;
    n->data = i;
    tail->next = n;
    tail = n;
    tail->next = NULL;
}
```

4. Penambahan node pada posisi depan:

```
void tambahDiDepan(int i){
    n = new Node;
    n->data = i;
    n->next = head;
    head = n;
}
```

5. Penambahan node pada posisi tengah:

```
void tambahDiTengah(int i, int j){
    x = head;
    while(x->data != j) x = x->next;
    n = new Node;
    n->data = i;
    n->next = x->next;
    x->next = n;
}
```

6. Penghapusan node pada posisi depan:

```
void hapusDiDepan(){
    x = head;
    head = head->next;
    delete(x);
}
```

7. Penghapusan node pada posisi belakang:

```
void hapusDiBelakang(){
    x = head;
    while(x->next != tail) x=x->next;
    tail = x;
    delete(x->next);
    tail->next = NULL;
}
```

8. Penghapusan node pada posisi tengah:

```
void hapusDiTengah(int i){
    Node *temp=NULL;
    x = head;
    while(x->data != i){
        temp = x;
        x = x->next;
    }
    temp->next = x->next;
    delete(x);
}
```

9. Tampil data:

```
void tambahDiTengah(int i, int j){
    x = head;
    while(x->data != j) x = x->next;
    n = new Node;
    n->data = i;
    n->next = x->next;
    x->next = n;
}
```

10. Penambahan node pada posisi tengah:

```

void tampilData() {
    x=head;
    while (x!=NULL) {
        cout << x->data << " ";
        x= x->next;
    }
}

```

D. Kegiatan Praktikum**1. Instrument**

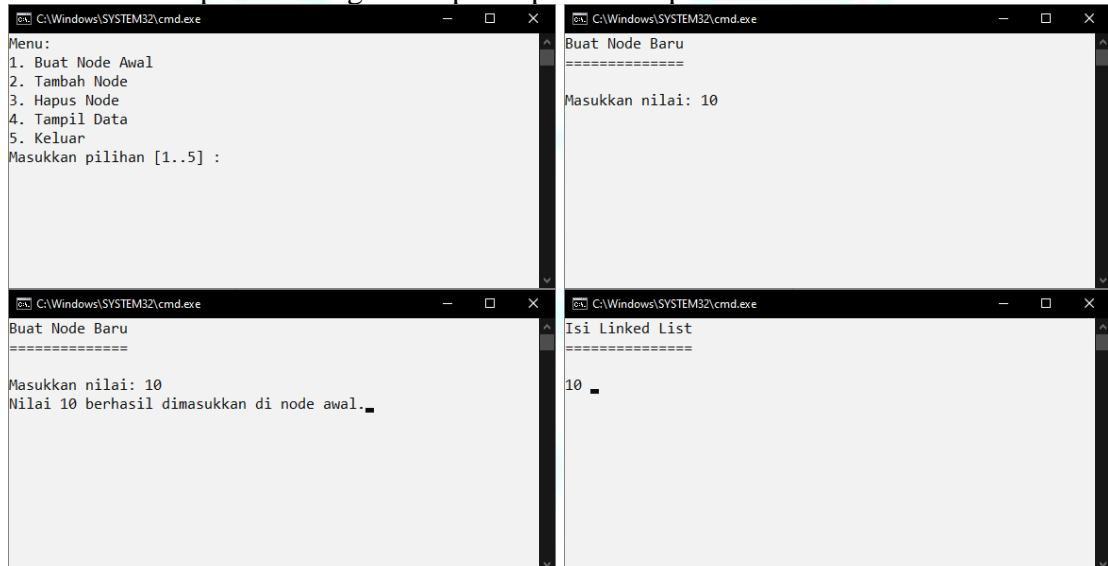
- Perangkat komputer
- Sistem Operasi Windows
- C++ Compiler (MinGW)
- Text Editor (Geany, Notepad++, Visual Studio Code, SublimeText, dll)

2. Prosedur

- Baca dan pahami semua tahapan praktikum dengan cermat.
- Gunakan fasilitas yang disediakan dengan penuh rasa tanggung jawab.
- Rapihkan kembali setelah menggunakan komputer (mouse, keyboard, kursi, dll)
- Perhatikan sikap anda untuk tidak mengganggu rekan praktikan lain.
- Pastikan diri anda tidak menyentuh sumber listrik.

3. Studi Kasus

Buat program untuk implementasi single linked list. Data yang dimasukkan dalam linked list berupa data integer. Output dapat dilihat pada Gambar di bawah.



Untuk lebih jelasnya, silakan buka demo program yg telah disediakan.

LEMBAR EVALUASI PRAKTIKUM

Evaluasi Praktikum 2 :

No	Indikator	Skor Penilaian				
		Sangat Kurang (E) ≤ 40	Kurang (D) 41-55	Cukup (C) 55-65	Baik (B) 66-85	Sangat Baik (A) ≥ 86
1.						
2.						
3.						
4.						

Catatan Asisten :

Dosen : _____

Asisten 1 : _____

Asisten 2 : _____

TINGKAT KEBERHASILAN

MODUL 3 – DOUBLE LINKED LIST

A. Capaian Pembelajaran Mata Kuliah (CPMK)

Mahasiswa mampu mengimplementasikan struktur data dinamis double linked list.

B. Sub Capaian Pembelajaran Mata Kuliah (CPMK)

Mahasiswa mampu mengimplementasikan struktur data single linked list, dengan operasi meliputi:

1. Pembentukan node awal
2. Penambahan node pada posisi depan, tengah, dan belakang.
3. Penghapusan node pada posisi depan, tengah, dan belakang.
4. Pengaksesan data dari depan (head ke tail)
5. Pengaksesan data dari belakang (tail ke head)

C. Teori Dasar

1. Struktur Node :

```
struct Node{
    int data;
    Node *next;
    Node *prev;
};
```

```
Node *n=NULL, *head=NULL, *tail=NULL, *x=NULL;
```

2. Pembuatan Node Awal:

```
void buatNodeBaru(int i){
    n = new Node;
    n->data = i;
    n->prev = NULL;
    head = n;
    tail = n;
    tail->next = NULL;
}
```

3. Penambahan node pada posisi belakang:

```
void tambahDiBelakang(int i){
    n = new Node;
    n->data = i;
    n->prev = tail;
    tail->next = n;
    tail = n;
    tail->next = NULL;
}
```

4. Penambahan node pada posisi depan:

```
void tambahDiDepan(int i){
    n = new Node;
    n->data = i;
    n->next = head;
    head->prev = n;
    n->prev = NULL;
    head = n;
}
```

5. Penambahan node pada posisi tengah:

```
void tambahDiTengah(int i, int j){
    x = head;
    while(x->data != j) x=x->next;
    n = new Node;
    n->data = i;
    n->next = x->next;
    x->next = n;
    n->prev = x;
    x = n->next;
    x->prev = n;
}
```

6. Penghapusan node pada posisi depan:

```
void hapusDiDepan() {
    x = head;
    head = x->next;
    head->prev = NULL;
    delete(x);
    x = NULL;
}
```

7. Penghapusan node pada posisi belakang:

```
void hapusDiBelakang() {
    x = tail;
    tail = tail->prev;
    tail->next = NULL;
    delete(x);
    x = NULL;
}
```

8. Penghapusan node pada posisi tengah:

```
void hapusDiTengah(int i) {
    n = NULL;
    x = head;
    while(x->data != i) {
        n = x;
        x = x->next;
    }
}
```



```

n->next = x->next;
x = x->next;
delete(x->prev);
x->prev = n;

}

```

9. Pengaksesan data dari depan:

```

void tampilDariDepan() {
    x = head;
    while(x != NULL) {
        cout << x->data << " ";
        x = x->next;
    }
}

```

10. Pengaksesan data dari belakang:

```

void tampilDariBelakang() {
    x = tail;
    while(x != NULL) {
        cout << x->data << " ";
        x = x->prev;
    }
}

```

D. Kegiatan Praktikum

1. Instrument

- Perangkat komputer
- Sistem Operasi Windows
- C++ Compiler (MinGW)
- Text Editor (Geany, Notepad++, Visual Studio Code, SublimeText, dll)

2. Prosedur

- Baca dan pahami semua tahapan praktikum dengan cermat.
- Gunakan fasilitas yang disediakan dengan penuh rasa tanggung jawab.
- Rapihkan kembali setelah menggunakan komputer (mouse, keyboard, kursi, dll)
- Perhatikan sikap anda untuk tidak mengganggu rekan praktikan lain.
- Pastikan diri anda tidak menyentuh sumber listrik.

3. Studi Kasus

Buat program untuk implementasi double linked list. Data yang dimasukkan dalam linked list berupa data integer. Output dapat dilihat pada Gambar di bawah.

```

C:\Windows\SYSTEM32\cmd.exe
Menu:
1. Buat Node Awal
2. Tambah Node
3. Hapus Node
4. Tampil Data
5. Keluar
Masukkan pilihan [1..5] : 1

C:\Windows\SYSTEM32\cmd.exe
Buat Node Baru
=====
Masukkan nilai: 30
Nilai 30 berhasil dimasukkan di node awal.

C:\Windows\SYSTEM32\cmd.exe
Menu:
1. Buat Node Awal
2. Tambah Node
3. Hapus Node
4. Tampil Data
5. Keluar
Masukkan pilihan [1..5] : 2

C:\Windows\SYSTEM32\cmd.exe
Pilihan:
1. Tambah Node di Depan
2. Tambah Node di Belakang
3. Tambah Node di Tengah
4. Batal
Masukkan pilihan [1..4] : 1

C:\Windows\SYSTEM32\cmd.exe
Tambah Node di Depan
=====
Masukkan nilai: 45
Node dengan nilai 45 berhasil ditambahkan pada posisi depan.

C:\Windows\SYSTEM32\cmd.exe
Menu:
1. Buat Node Awal
2. Tambah Node
3. Hapus Node
4. Tampil Data
5. Keluar
Masukkan pilihan [1..5] : 4

C:\Windows\SYSTEM32\cmd.exe
Pilihan:
1. Tampil dari depan
2. Tampil dari belakang
3. Batal
Masukkan pilihan [1..3] :

C:\Windows\SYSTEM32\cmd.exe
Isi linked List
=====
30 45
    
```

Untuk lebih jelasnya, silakan buka demo program yg telah disediakan.

LEMBAR EVALUASI PRAKTIKUM

Evaluasi Praktikum 3 :

No	Indikator	Skor Penilaian				
		Sangat Kurang (E) ≤ 40	Kurang (D) 41-55	Cukup (C) 55-65	Baik (B) 66-85	Sangat Baik (A) ≥ 86
1.						
2.						
3.						
4.						

Catatan Asisten :

Dosen : _____

Asisten 1 : _____

Asisten 2 : _____

TINGKAT KEBERHASILAN

MODUL 4 – BINARY SEARCH TREE

A. Capaian Pembelajaran Mata Kuliah (CPMK)

Mahasiswa mampu mengimplementasikan struktur data dinamis non-linear Binary Search Tree.

B. Sub Capaian Pembelajaran Mata Kuliah (CPMK)

Mahasiswa mampu mengimplementasikan binary search tree, dengan operasi meliputi:

1. Penambahan Node
2. Binary Tree Traversal
 - a. Level Order
 - b. Preorder
 - c. Inorder
 - d. Postorder
3. Pencarian data pada Node
4. Penghapusan Node

C. Teori Dasar

1. Struktur Node :

```
struct Node{
    int data;
    Node *left;
    Node *right;
};
```

```
Node *root=NULL, *n=NULL, *temp=NULL;
```

2. Pembuatan Node Baru:

```
Node *createNode(int data){
    n = new Node;
    n->data = data;
    n->left = NULL;
    n->right = NULL;
    return n;
}
```

3. Proses Insert node pada BST:

```
void insert(Node *&root, int data){
    if (root==NULL)
        root = createNode(data);
    else if (data <= root->data)
        insert(root->left, data);
    else if (data > root->data)
        insert(root->right, data);
}
```

4. Binary Tree Traversal menggunakan Level Order:

```
void levelOrder(Node *root){
    if (root==NULL) return;

    queue<Node*> q;
    q.push(root);
    while(!q.empty()){
        cout << q.front()->data << " ";
        if(q.front()->left != NULL)
            q.push(q.front()->left);
        if(q.front()->right != NULL)
            q.push(q.front()->right);
        q.pop();
    }
}
```

5. Binary Tree Traversal menggunakan Preorder:

```
void preOrder(Node *root){
    if (root==NULL) return;

    cout << root->data << " ";
    preOrder(root->left);
    preOrder(root->right);
}
```

6. Binary Tree Traversal menggunakan Inorder:

```
void inOrder(Node *root){
    if (root==NULL) return;

    inOrder(root->left);
    cout << root->data << " ";
    inOrder(root->right);
}
```

7. Binary Tree Traversal menggunakan Postorder:

```
void postOrder(Node *root){
    if (root==NULL) return;

    postOrder(root->left);
    postOrder(root->right);
    cout << root->data << " ";
}
```

8. Pencarian data pada BST:

```
bool cari(Node *root, int data){
    if (root==NULL)
        return false;
    else if(data < root->data)
        return cari(root->left, data);
}
```

```

    else if(data > root->data)
        return cari(root->right, data);
    else
        return true;
}

```

9. Mencari alamat memori node yang memiliki nilai minimum:

```

Node *cariMin(Node *root){
    if (root==NULL)
        return NULL;
    while(root->left != NULL)
        root = root->left;
    return root;
}

```

10. Penghapusan node pada BST:

```

void hapus(Node *&root, int data){
    if (root==NULL)
        return;
    else if(data < root->data)
        return hapus(root->left, data);
    else if(data > root->data)
        return hapus(root->right, data);
    else{
        // kasus I
        if (root->left == NULL && root->right == NULL){
            delete(root);
            root=NULL;
        }
        // kasus II
        else if (root->left == NULL){
            temp = root;
            root = root->right;
            delete(temp);
            temp=NULL;
        }
        else if (root->right == NULL){
            temp = root;
            root = root->left;
            delete(temp);
            temp=NULL;
        }
        // kasus III
        else{
            temp = cariMin(root->right);
            root->data = temp->data;
            hapus(root->right, root->data);
        }
    }
}

```

}

D. Kegiatan Praktikum**1. Instrument**

- Perangkat komputer
- Sistem Operasi Windows
- C++ Compiler (MinGW)
- Text Editor (Geany, Notepad++, Visual Studio Code, SublimeText, dll)

2. Prosedur

- Baca dan pahami semua tahapan praktikum dengan cermat.
- Gunakan fasilitas yang disediakan dengan penuh rasa tanggung jawab.
- Rapihkan kembali setelah menggunakan komputer (mouse, keyboard, kursi, dll)
- Perhatikan sikap anda untuk tidak mengganggu rekan praktikan lain.
- Pastikan diri anda tidak menyentuh sumber listrik.

3. Studi Kasus

Buat program untuk implementasi Binary search tree meliputi operasi: insert, search, delete, dan binary tree traversal (level order, preorder, inorder dan postorder). Output dapat dilihat pada Gambar di bawah.

```

C:\Windows\SYSTEM32\cmd.exe
Menu:
1. Buat Node Awal
2. Tambah Node
3. Hapus Node
4. Tampil Data
5. Keluar
Masukkan pilihan [1..5] : 1

C:\Windows\SYSTEM32\cmd.exe
Buat Node Baru
=====
Masukkan nilai: 30
Nilai 30 berhasil dimasukkan di node awal.

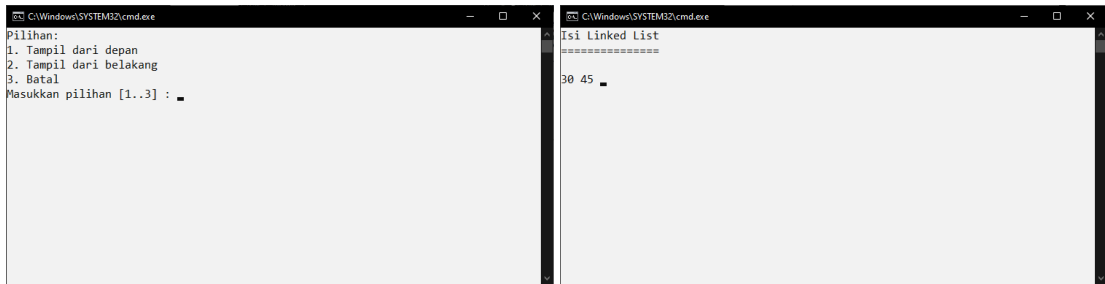
C:\Windows\SYSTEM32\cmd.exe
Menu:
1. Buat Node Awal
2. Tambah Node
3. Hapus Node
4. Tampil Data
5. Keluar
Masukkan pilihan [1..5] : 2

C:\Windows\SYSTEM32\cmd.exe
Pilihan:
1. Tambah Node di Depan
2. Tambah Node di Belakang
3. Tambah Node di Tengah
4. Batal
Masukkan pilihan [1..4] : 1

C:\Windows\SYSTEM32\cmd.exe
Tambah Node di Depan
=====
Masukkan nilai: 45
Node dengan nilai 45 berhasil ditambahkan pada posisi depan.

C:\Windows\SYSTEM32\cmd.exe
Menu:
1. Buat Node Awal
2. Tambah Node
3. Hapus Node
4. Tampil Data
5. Keluar
Masukkan pilihan [1..5] : 4

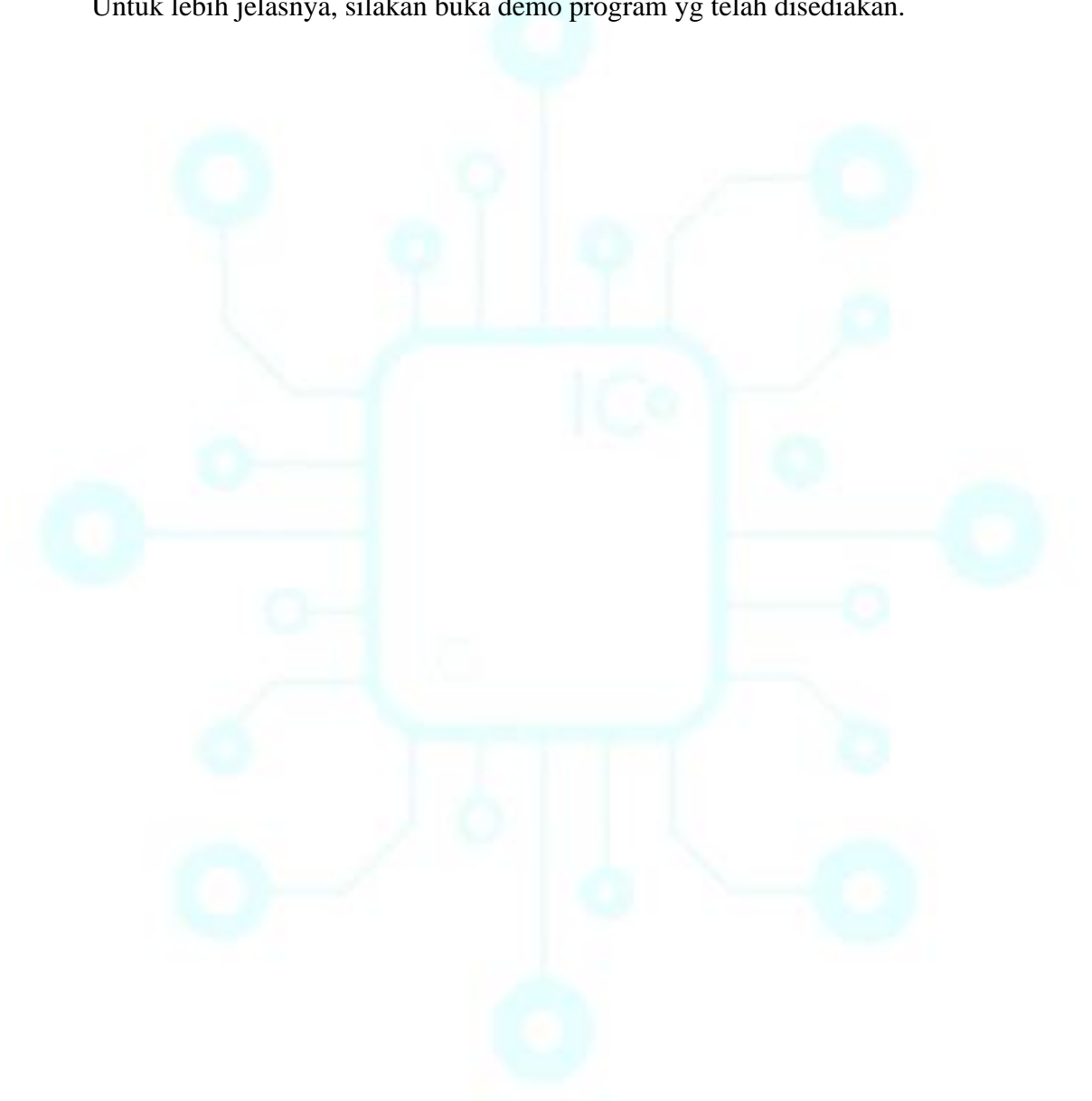
```

```
C:\Windows\SYSTEM32\cmd.exe
Pilihan:
1. Tampil dari depan
2. Tampil dari belakang
3. Batal
Masukkan pilihan [1..3] :

C:\Windows\SYSTEM32\cmd.exe
Isi Linked List
=====
30 45
```

Untuk lebih jelasnya, silakan buka demo program yg telah disediakan.



LEMBAR EVALUASI PRAKTIKUM

Evaluasi Praktikum 4 :

No	Indikator	Skor Penilaian				
		Sangat Kurang (E) ≤ 40	Kurang (D) 41-55	Cukup (C) 55-65	Baik (B) 66-85	Sangat Baik (A) ≥ 86
1.						
2.						
3.						
4.						

Catatan Asisten :

Dosen : _____

Asisten 1 : _____

Asisten 2 : _____

TINGKAT KEBERHASILAN