

**LAPORAN PRAKTIKUM
STRUKTUR DATA**

**MODUL 4 & 5
SINGLY LINKED LIST**



Disusun Oleh :

NAMA : Muhammad Azzam Satria

NIM : 103112400112

Dosen

FAHRUDIN MUKTI WIBOWO S. Kom., M. Eng

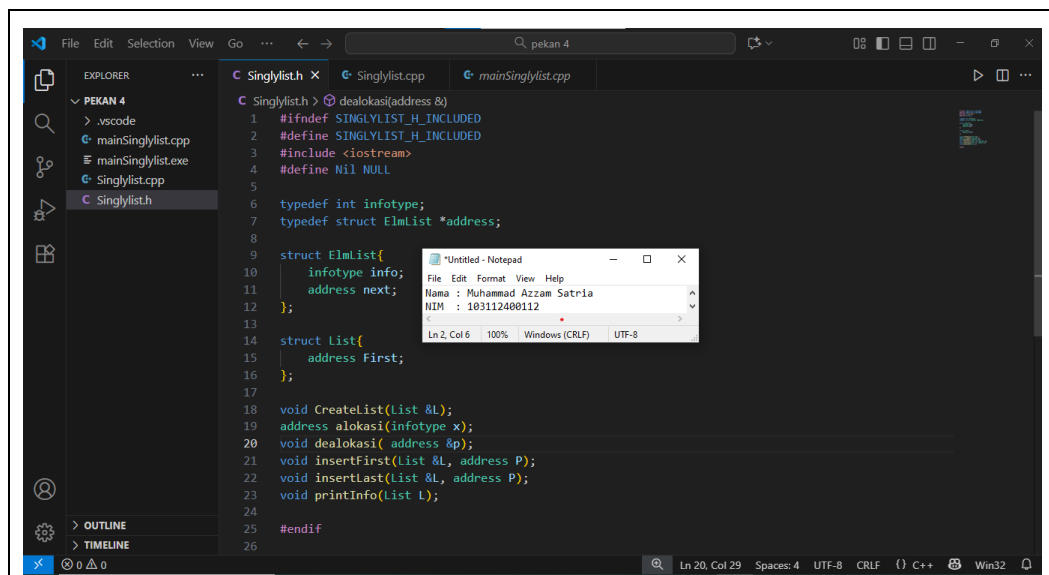
**PROGRAM STUDI STRUKTUR DATA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025**

A. Dasar Teori

Linked list merupakan salah satu struktur data yang terdiri dari sekumpulan elemen dengan tipe data yang sama, dimana setiap elemen saling terhubung melalui pointer. Pointer memiliki fungsi sebagai penyimpan alamat memori ke elemen berikutnya agar setiap elemen dapat saling berkaitan secara dinamis didalam memori. Salah satu jenis linked list adalah singly linked list yang merupakan struktur data berbentuk daftar berantai dimana setiap simpulnya saling terhubung satu sama lain dengan menggunakan pointer. Setiap simpul didalamnya memiliki dua bagian utama, yaitu bagian yang berfungsi untuk menyimpan data dan pointer yang mengarah ke simpul berikutnya. Simpul pada linked list hanya memiliki satu arah saja, dengan simpul pertama (head) menuju ke simpul akhir (tail).

Jika dibandingkan dengan array, linked list memiliki keunggulan yaitu lebih fleksibel karena ukurannya dapat disesuaikan tanpa harus tahu kapasitas dari awal. Dengan keunggulan seperti itu, singly linked list dapat membantu dalam pengolahan data yang jumlahnya dapat berubah-ubah. Akan tetapi singly linked list juga memiliki kelemahan, yaitu tidak bisa langsung mengakses elemen tertentu karena setiap elemen harus ditelusuri secara berurutan dari awal sehingga membutuhkan proses yang tidak cepat.

B. Guided 1



```
1  Singlylist.h > dealokasi(address &B)
2  #ifndef SINGLYLIST_H_INCLUDED
3  #define SINGLYLIST_H_INCLUDED
4  #include <iostream>
5  #define Nil NULL
6
7  typedef int infotype;
8  typedef struct Elmlist *address;
9
10 struct Elmlist{
11     infotype info;
12     address next;
13 };
14
15 struct List{
16     address First;
17 };
18
19 void Createlist(List &L);
20 address alokasi(infotype x);
21 void dealokasi( address &p);
22 void insertFirst(List &L, address P);
23 void insertLast(List &L, address P);
24 void printInfo(List L);
25
26 #endif
```

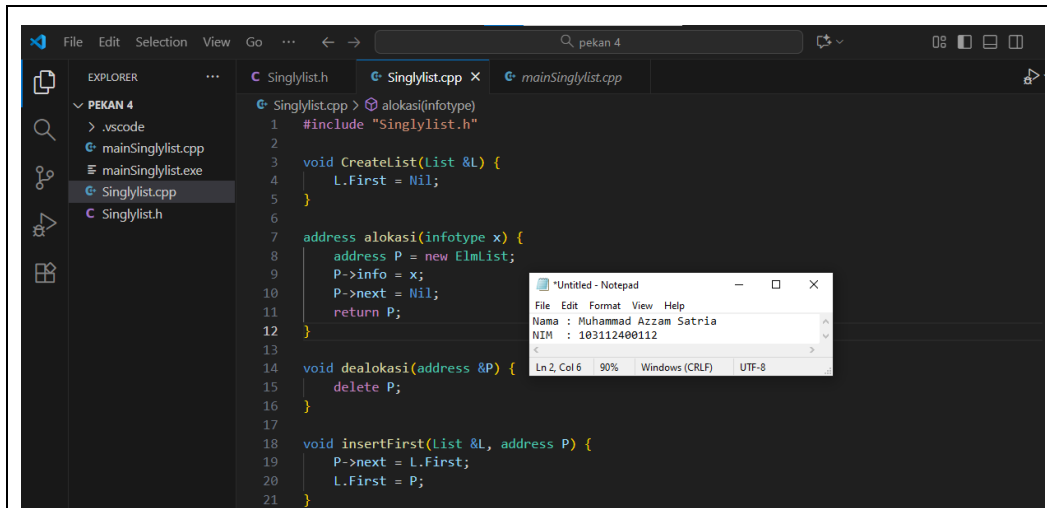
Untitled - Notepad

File Edit Format View Help

Nama : Muhammad Azzam Satria

NIM : 103112400112

Ln 2, Col 6 100% Windows (CRLF) UTF-8



File Edit Selection View Go ... Search: pekan 4

EXPLORER

- PEKAN 4
 - .vscode
 - mainSinglylist.cpp
 - mainSinglylist.exe
 - Singlylist.cpp
 - Singlylist.h

Singlylist.cpp

```
1 #include "Singlylist.h"
2
3 void CreateList(List &L) {
4     L.First = Nil;
5 }
6
7 address alokasi(infotype x) {
8     address P = new ElmList;
9     P->info = x;
10    P->next = Nil;
11    return P;
12 }
13
14 void dealokasi(address &P) {
15     delete P;
16 }
17
18 void insertFirst(List &L, address P) {
19     P->next = L.First;
20     L.First = P;
21 }
```

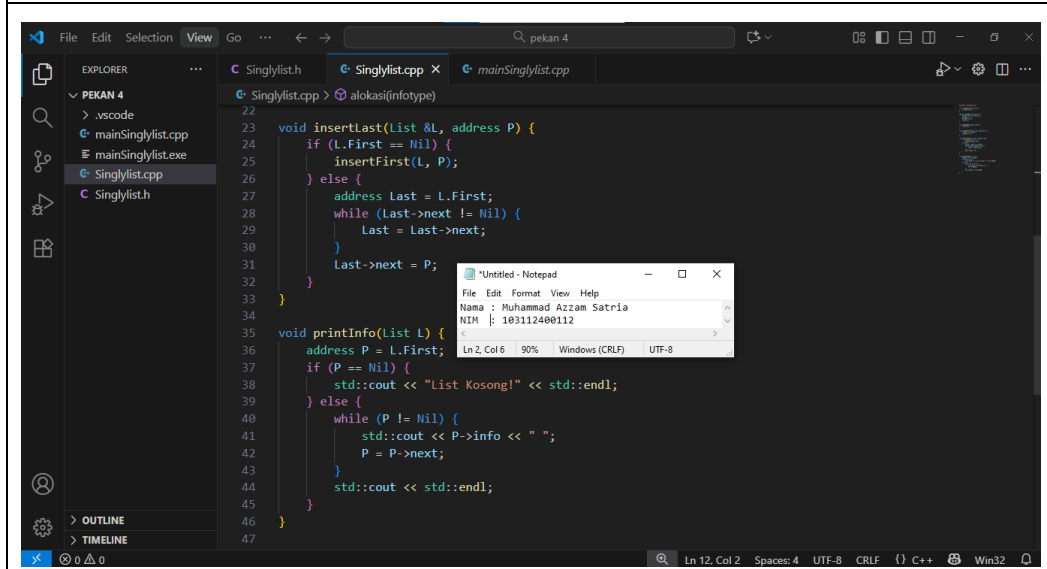
Untitled - Notepad

File Edit Format View Help

Nama : Muhammad Azzam Satria

NIM : 103112400112

Ln 2, Col 6 90% Windows (CRLF) UTF-8



File Edit Selection View Go ... Search: pekan 4

EXPLORER

- PEKAN 4
 - .vscode
 - mainSinglylist.cpp
 - mainSinglylist.exe
 - Singlylist.cpp
 - Singlylist.h

Singlylist.cpp

```
22
23 void insertLast(List &L, address P) {
24     if (L.First == Nil) {
25         insertFirst(L, P);
26     } else {
27         address Last = L.First;
28         while (Last->next != Nil) {
29             Last = Last->next;
30         }
31         Last->next = P;
32     }
33 }
34
35 void printInfo(List L) {
36     address P = L.First;
37     if (P == Nil) {
38         std::cout << "List Kosong!" << std::endl;
39     } else {
40         while (P != Nil) {
41             std::cout << P->info << " ";
42             P = P->next;
43         }
44         std::cout << std::endl;
45     }
46 }
47 }
```

Untitled - Notepad

File Edit Format View Help

Nama : Muhammad Azzam Satria

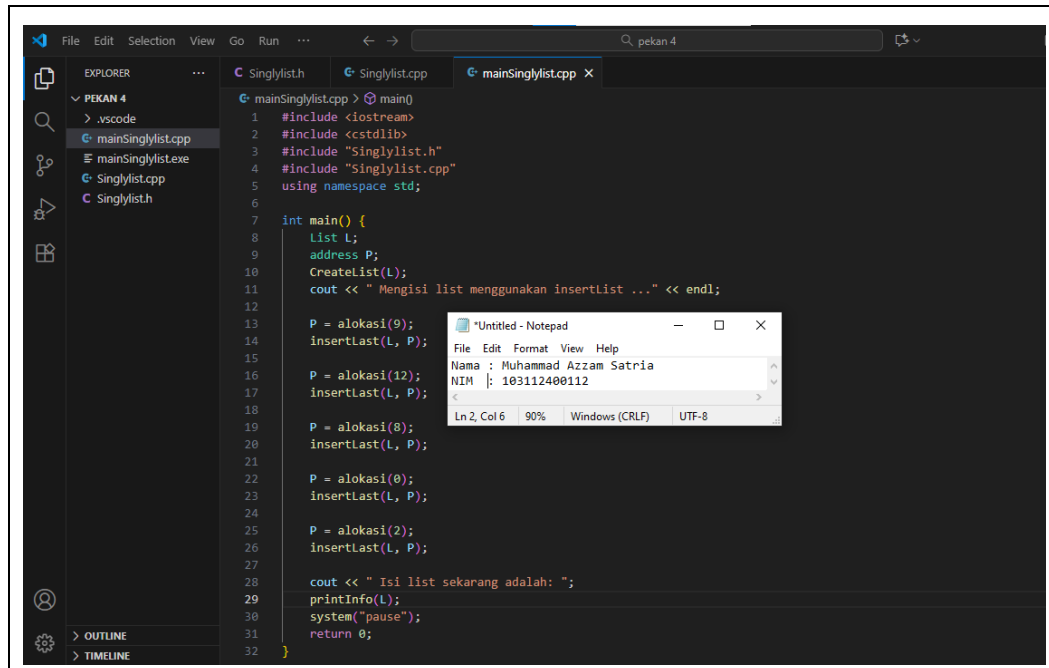
NIM : 103112400112

Ln 2, Col 6 90% Windows (CRLF) UTF-8

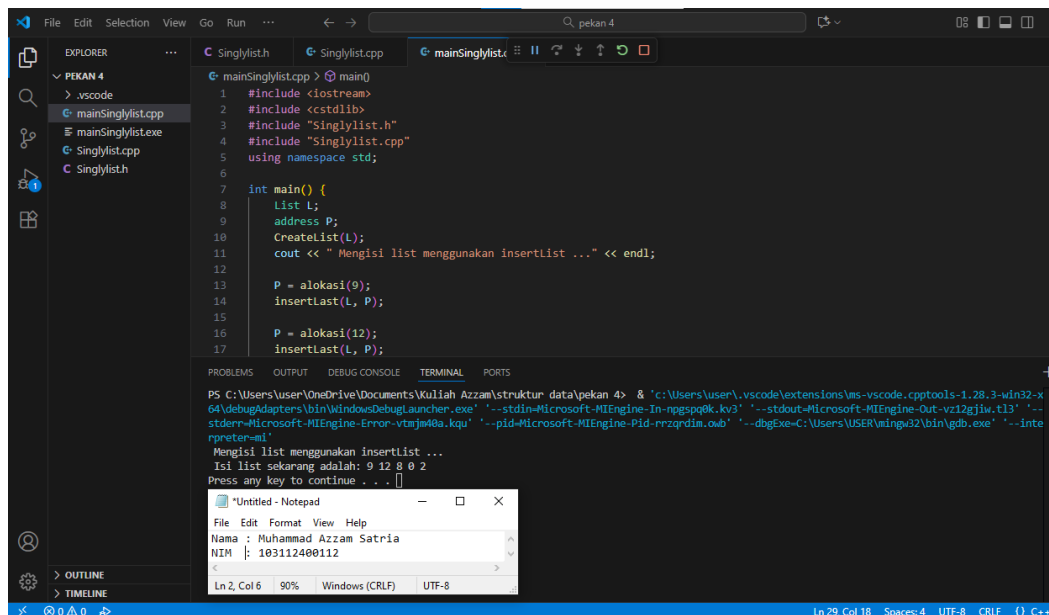
OUTLINE

TIMELINE

Ln 12, Col 2 Spaces: 4 UTF-8 CRLF {} C++ Win32



Screenshots Output



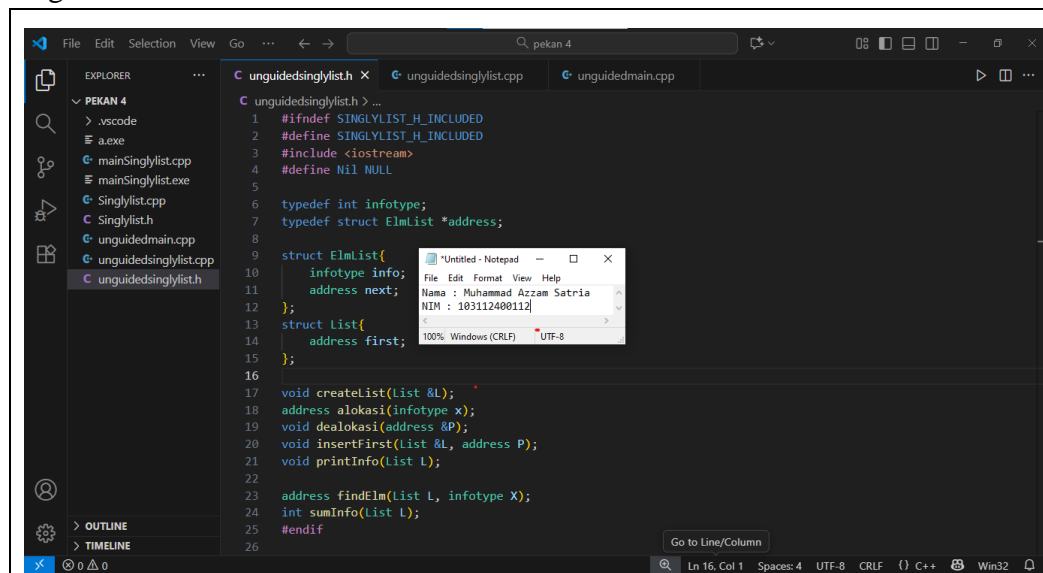
Deskripsi:

Struktur program ini dibagi menjadi tiga file yaitu Singlylist.h, Silylist.cpp dan mainSinglylist.cpp. Singlylist,h digunakan untuk mendeklarasikan struktur

data dan fungsi dasar dari singly linked list. Struktur ElmList memiliki dua elemen utama yaitu info untuk menyimpan data integer dan next sebagai penunjuk ke elemen berikutnya, sedangkan pada struktur List hanya menyimpan satu pointer First yang menunjukkan ke node pertama dalam list.

Pada file SinglyList.cpp berisi implementasi dari fungsi dasar seperti createList, alokasi, dealokasi, insertFirst, insertLast dan printInfo. Fungsi createList digunakan untuk meninisialisasi list agar kosong, sedangkan fungsi alokasi sebagai pembuat node baru menggunakan operator new. File mainSinglylist.cpp berisi program utama yang menguji operasi insert dan print list dengan cara menambahkan data menggunakan insertLast, lalu menampilkan output seluruh isi list dengan menggunakan printInfo

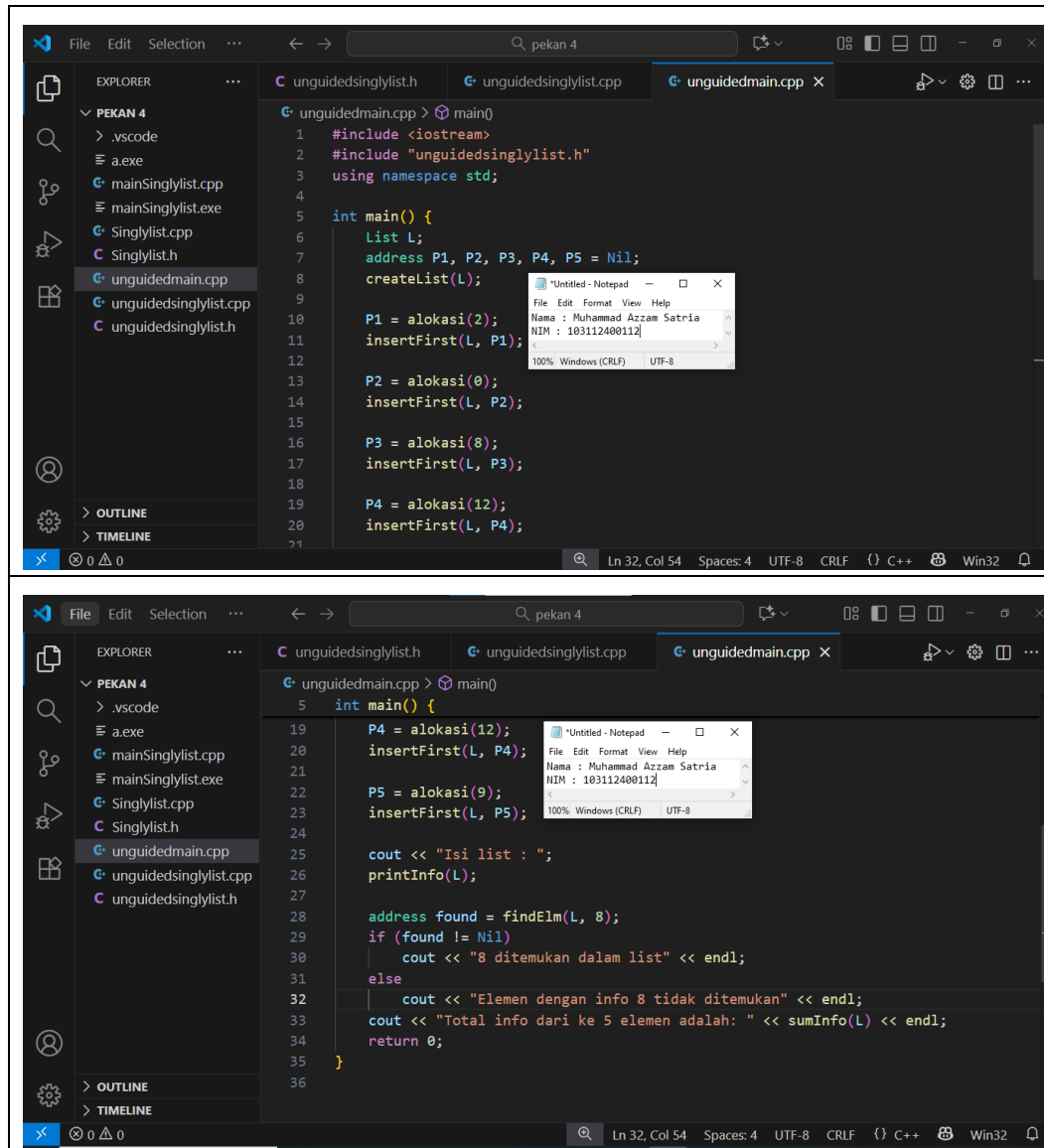
C. Unguided 1



```
1 #ifndef SINGLYLIST_H_INCLUDED
2 #define SINGLYLIST_H_INCLUDED
3 #include <iostream>
4 #define Nil NULL
5
6 typedef int infotype;
7 typedef struct ElmList *address;
8
9 struct ElmList{
10     infotype info;
11     address next;
12 };
13 struct List{
14     address first;
15 };
16
17 void createlist(List &L);
18 address alokasi(infotype x);
19 void dealokasi(address &P);
20 void insertfirst(List &L, address P);
21 void printInfo(List L);
22
23 address findElm(List L, infotype X);
24 int sumInfo(List L);
25 #endif
```

```
1 #include "unguidedsinglylist.h"
2
3 void createList(List &L){
4     L.first = Nil;
5 }
6 address alokasi(infotype x){
7     address P = new ElmList;
8     P->info = x;
9     P->next = Nil;
10    return P;
11 }
12 void dealokasi(address &P){
13     delete P;
14     P = Nil;
15 }
16
17 void insertFirst(List &L, address P){
18     P->next = L.first;
19     L.first = P;
20 }
21 void printInfo(List L){
22     address P = L.first;
23     if (P == Nil){
24         std::cout << "List kosong!" << std::endl;
25     }else{
26         while (P != Nil){
```

```
21 void printInfo(List L){
22     address P = L.first;
23     while (P != Nil){
24         std::cout << P->info << " ";
25         P = P->next;
26     }
27     std::cout << std::endl;
28 }
29
30 address findElm(List L, infotype X){
31     address P = L.first;
32     while (P != Nil && P->info != X){
33         P = P->next;
34     }
35     return P;
36 }
37
38 int sumInfo(List L){
39     int total = 0;
40     address P = L.first;
41     while (P != Nil){
42         total += P->info;
43         P = P->next;
44     }
45     return total;
46 }
```



Screenshots Output 1

The screenshot shows a C++ IDE with a terminal window. The terminal output is as follows:

```
PS C:\Users\user\OneDrive\Documents\Kuliah Azzam\struktur data\pekan 4> g++ unguidedmain.cpp unguidedsinglylist.cpp
PS C:\Users\user\OneDrive\Documents\Kuliah Azzam\struktur data\pekan 4> ./a.exe
Isi list : 9 12 8 0 2
PS C:\Users\user\OneDrive\Documents\Kuliah Azzam\struktur data\pekan 4> 
```

Overlaid on the terminal is a Notepad window titled "Untitled - Note...". It contains the following text:

```
File Edit Format View Help
Nama : Muhammad Azzam Satria
NIM : 103112400112
```

The status bar at the bottom of the Notepad window shows "100 Windows (CRLF) UTF-8".

Screenshot Output 2

The screenshot shows a C++ IDE with a terminal window. The terminal output is as follows:

```
PS C:\Users\user\OneDrive\Documents\Kuliah Azzam\struktur data\pekan 4> g++ unguidedmain.cpp unguidedsinglylist.cpp
PS C:\Users\user\OneDrive\Documents\Kuliah Azzam\struktur data\pekan 4> ./a.exe
8 ditemukan dalam list
PS C:\Users\user\OneDrive\Documents\Kuliah Azzam\struktur data\pekan 4> 
```

Overlaid on the terminal is a Notepad window titled "Untitled - Notepad". It contains the following text:

```
File Edit Format View Help
Nama : Muhammad Azzam Satria
NIM : 103112400112
```

The status bar at the bottom of the Notepad window shows "100% Windows (CRLF) UTF-8". The IDE's status bar at the bottom shows "Ln 26, Col 20 Spaces: 4 UTF-8 CRLF {} C++ Win32".

Screenshot Output 3

The screenshot shows a C++ IDE with a terminal window. The terminal output is as follows:

```
PS C:\Users\user\OneDrive\Documents\Kuliah Azzam\struktur data\pekan 4> g++ unguidedmain.cpp unguidedsinglylist.cpp
PS C:\Users\user\OneDrive\Documents\Kuliah Azzam\struktur data\pekan 4> ./a.exe
Total info dari ke 5 elemen adalah: 31
PS C:\Users\user\OneDrive\Documents\Kuliah Azzam\struktur data\pekan 4> 
```

Overlaid on the terminal is a Notepad window titled "Untitled - Notepad". It contains the following text:

```
File Edit Format View Help
Nama : Muhammad Azzam Satria
NIM : 103112400112
```

The status bar at the bottom of the Notepad window shows "100% Windows (CRLF) UTF-8". The IDE's status bar at the bottom shows "Ln 32, Col 66 Spaces: 4 UTF-8 CRLF {} C++ Win32".

Deskripsi:

Struktur program ini terbagi ke dalam 3 file berbeda yaitu `unguidedsinglylist.h`, `unguidedsinglylist.cpp` dan `unguidedmain.cpp`. Pada file `unguidedsinglylist.h` digunakan sebagai mendeklarasikan struktur data dan

fungsi dasar dari singly linked list. Didalamnya terdapat struktur ElmList yang memiliki 2 elemen utama yaitu info untuk menyimpan data dengan tipe integer dan next untuk penunjuk ke elemen berikutnya. Selain Elm List terdapat juga struktur List yang berisi satu pointer first sebagai penunjuk ke node pertama pada list.

File `unguidedsinglylist.cpp` didalamnya terdapat implementasi fungsi-fungsi dasar yaitu `createList`, alokasi, dealokasi, `insertFirst`, `printInfo`, `findElm` dan `sumInfo`. Selanjutnya ada file `unguidedmain.cpp` sebagai program utama yang menjalankan seluruh fungsi dengan menambahkan data ke dalam list, menampilkan isi list, mencari data dengan nilai 8 dan menghitung total jumlah seluruh nilai yang ada di dalam list. Setelah program dirunning maka akan menampilkan output dari data yang telah ditambahkan pada program.

D. Kesimpulan

Berdasarkan pembahasan pada modul 4 dan 5 disimpulkan bahwa singly linked list merupakan salah satu struktur data dinamis yang menggunakan pointer untuk menghubungkan setiap elemen data. Berbeda dengan array yang bersifat statis, singly linked list lebih fleksibel dalam pengelolaan data karena dapat menambah dan mengurangi data sesuai kebutuhan. Fungsi-fungsi dasar seperti `a`, `insertFirst`, `findElm` dan `sumInfo` merupakan proses pembuatan, penelusuran, pencarian dan penghitungan data yang dilakukan oleh pointer. Selain itu penggunaan pointer dapat membuat program menjadi lebih efisien dalam penggunaan memori.

Latihan yang ada pada modul membantu saya memahami bagaimana program dibuat dengan konsep ADT, kemudian bagaimana cara menambahkan node, menampilkan isi list, mencari nilai tertentu dan menghitung total data dalam list. Proses menyusun program juga membuat saya lebih mengerti mengenai cara penulisan kode yang rapi dan modular agar mudah dipahami dan dikembangkan Kembali.

E. Referensi

Setiyawan, R. D., Hermawan, D., Abdillah, A. F., Mujayanah, A., & Vindua, R. (2024). Penggunaan struktur data stack dalam pemrograman C++ dengan

pendekatan array dan linked list. *JUTECH: Journal Education and Technology*, 5(2), 484–498.

Mbejo, M. T., Nopa, L. A., Putri, J. S., & Risky, M. (2025). Analisis struktur data linked list dalam pengolahan data mahasiswa. *Jurnal Sains Informatika Terapan*, 4(2), 441–444.