 LEMBAR ASISTENSI

PRAKTIKUM STRUKTUR DATA

LABORATORIUM TEKNIK KOMPUTER

JURUSAN TEKNIK ELEKTRO FAKULTAS TEKNIK

UNIVERSITAS LAMPUNG

Judul Praktikum : JUDUL

Praktikan (NPM) : Rahman Hidayat (2415061073)

Asisten (NPM) : Nama Asisten Praktikum (NPM)

Nama Asistensi Praktikum (NPM)

Nama Asistensi Praktikum (NPM)

Nama Asistensi Praktikum (NPM)

Nama Asistensi Praktikum (NPM)

Kelas : PSTI C

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Catatan** | **Tanggal** | **Paraf** |
|  |  |  |  |

Bandar Lampung, 2025

Nama Asisten Praktikum

NPM. 2315

1. JUDUL PERCOBAAN

VARIABEL STRUKTUR DATA

1. TUJUAN PERCOBAAN

Adapun tujuan dari percobaan ini adalah sebagai berikut :

1. Dapat membuat dan memahami penggunaan array.
2. Dapat membuat dan memahami penggunaan linked list.
3. Dapat membuat dan memahami penggunaan vector.
4. TEORI DASAR

3.1 Sub bab

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus vitae augue ultrices, iaculis ligula quis, dictum turpis. Duis sed eros tellus. Morbi convallis efficitur porttitor. Praesent id felis et lorem egestas congue. Duis sed eros tellus. Morbi convallis efficitur porttitor. Praesent id felis et lorem egestas congue (Rhalin, 2025)



*Gambar 3.1 \_\_\_\_\_\_\_\_*

Loren ipsu

3.2 sub bab

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus vitae augue ultrices, iaculis ligula quis, dictum turpis. Duis sed eros tellus. Morbi convallis efficitur porttitor. Praesent id felis et lorem egestas congue. …….



*Gambar 3.2 \_\_\_\_\_\_\_\_*

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus vitae augue ultrices, iaculis ligula quis, dictum turpis. Duis sed eros tellus. Morbi convallis efficitur porttitor. Praesent id felis et lorem egestas congue. (Bagas & Favian, 2025)

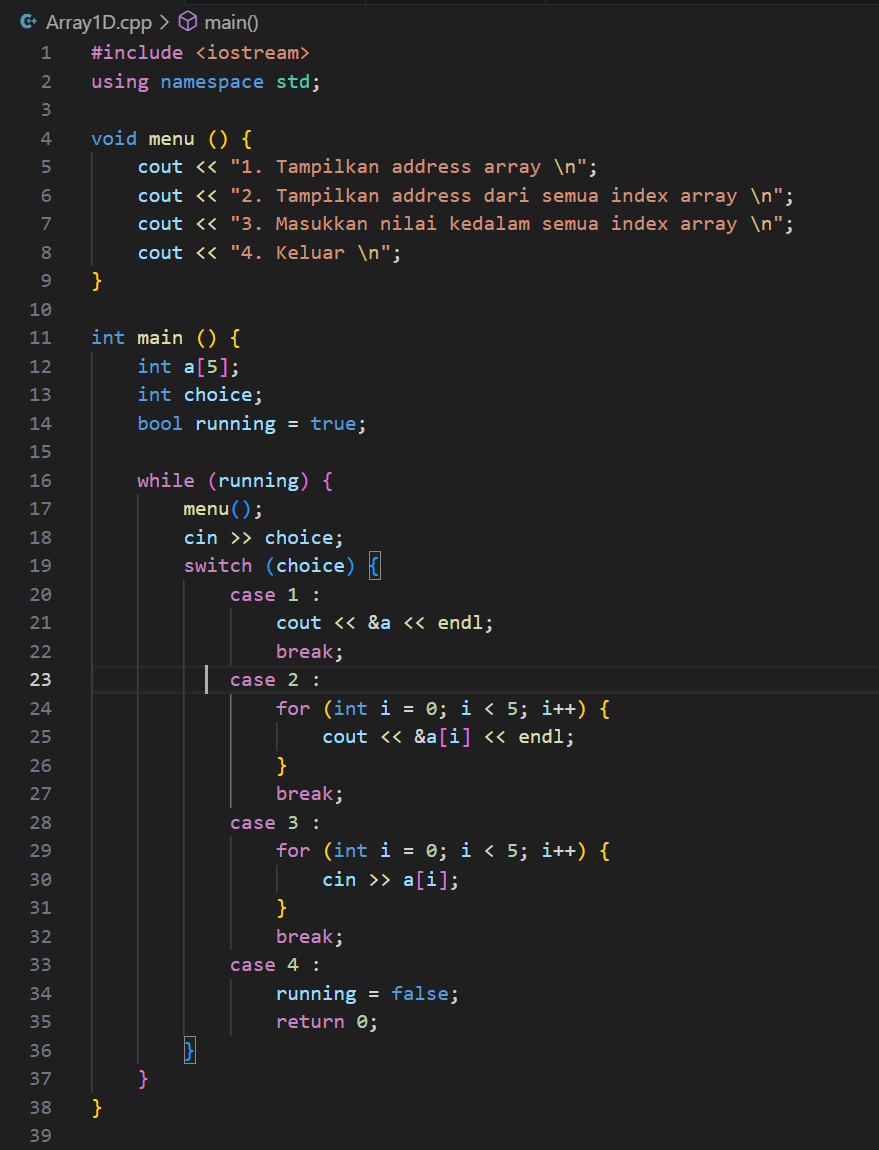
**BUAT TEODAS 3,5 LEMBAR, SERTAKAN SITASI**

1. PROSEDUR PERCOBAAN

Adapun prosedur dari percobaan ini adalah sebagai berikut:

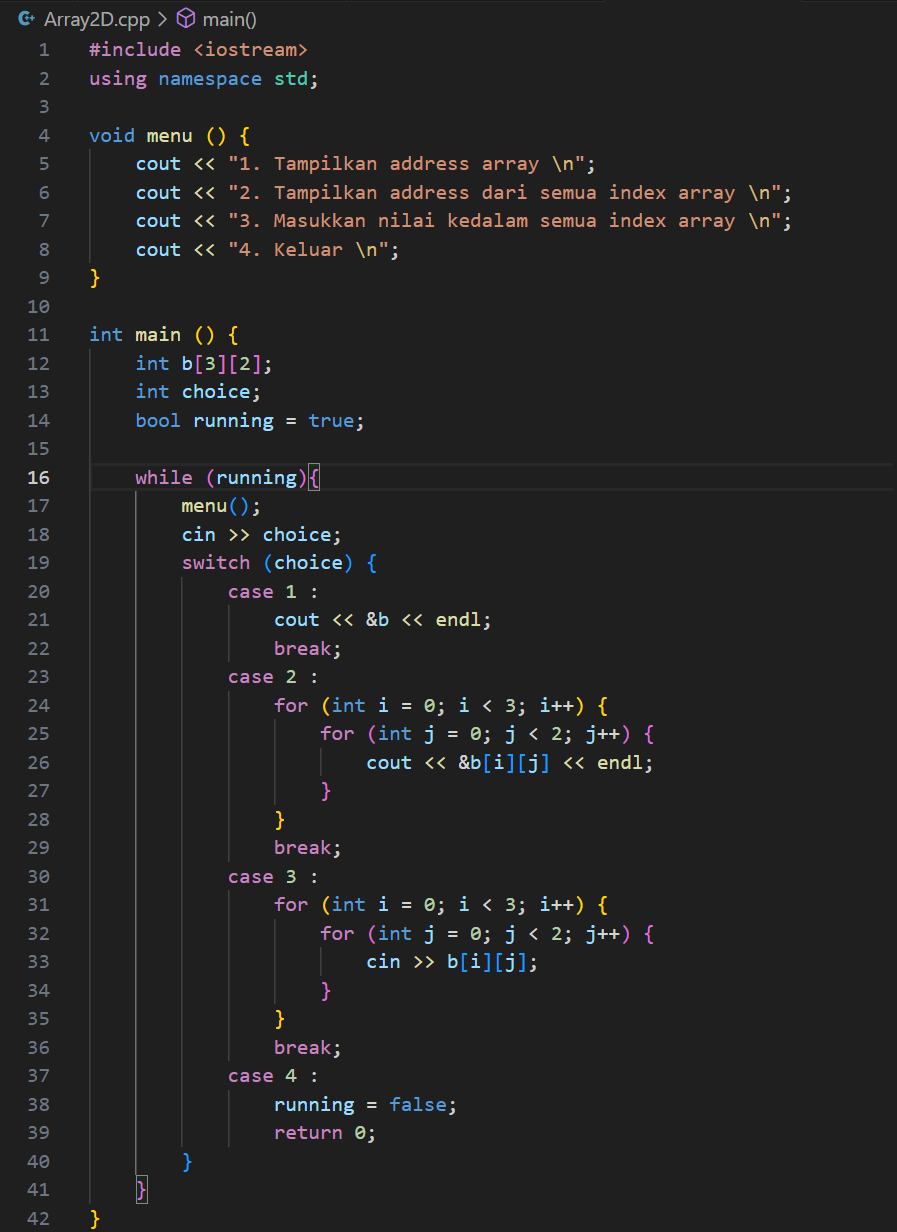
* 1. Percobaan 1: Array

4.1.1 Percobaan 1-1: Array 1D



*Gambar 4.1.1 Source Code Percobaan 1-1: Array 1D*

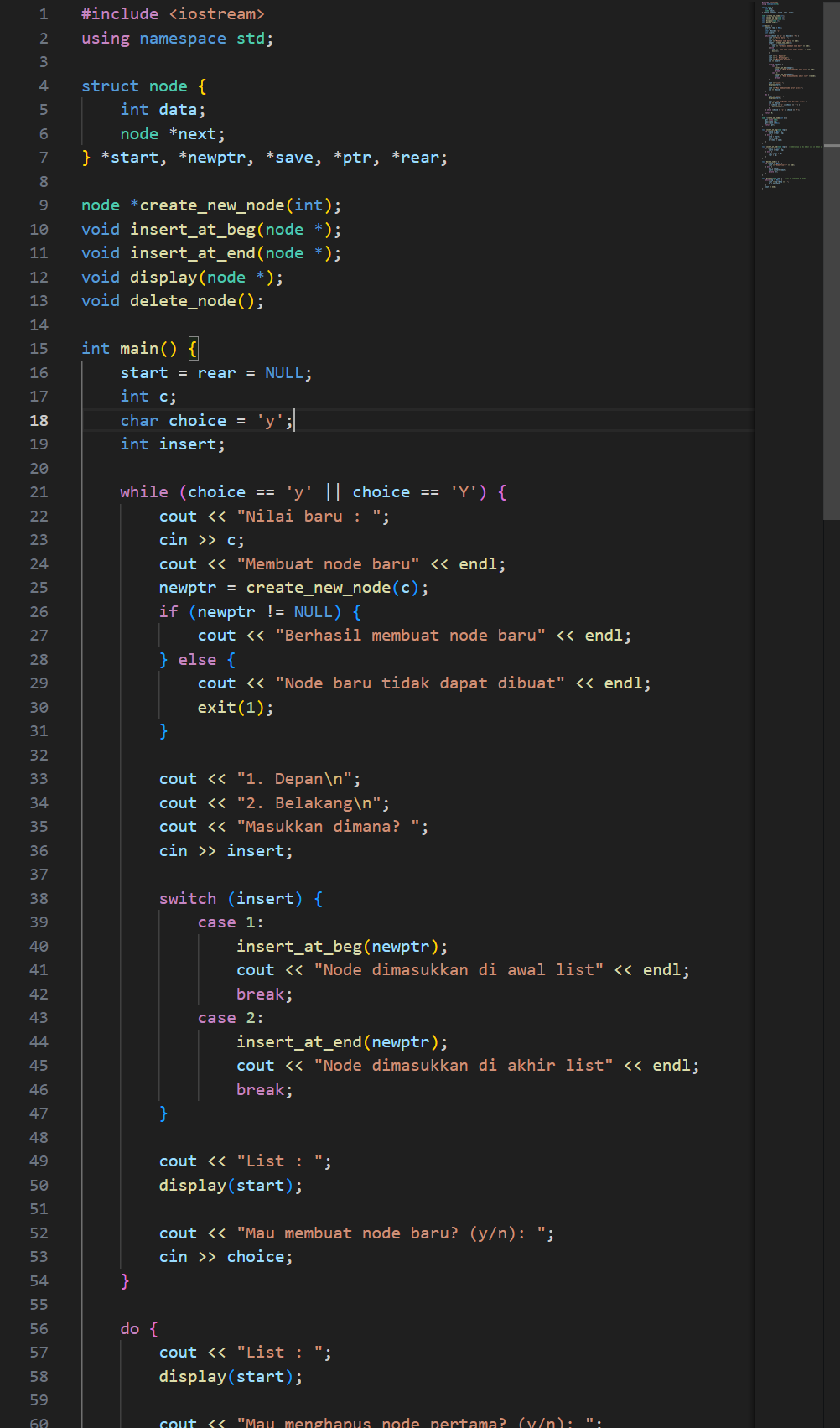
4.1.2 Percobaan 1-2 : Array 2D

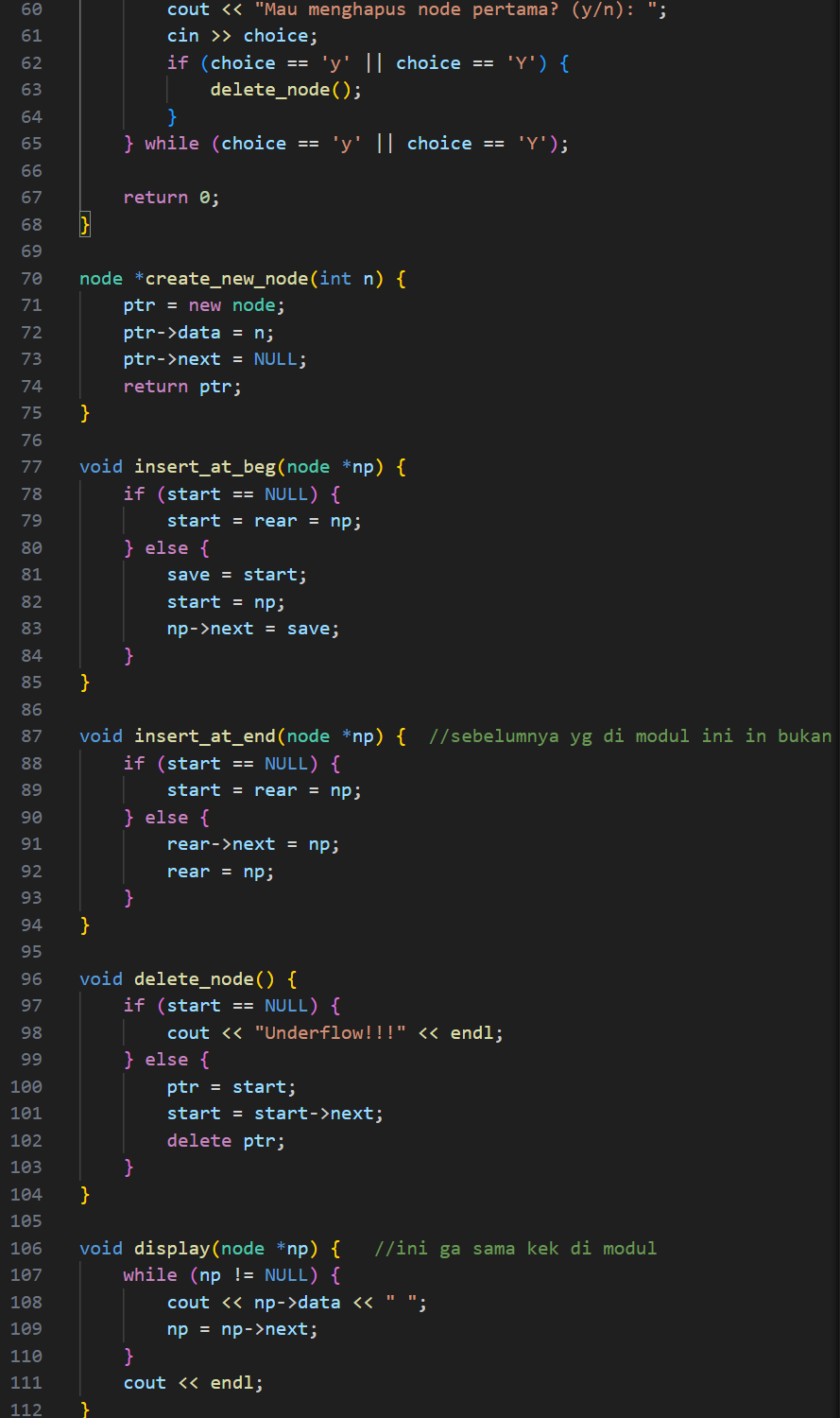


*Gambar 4.1.2 Source Code Percobaan 1-2: Array 2D*

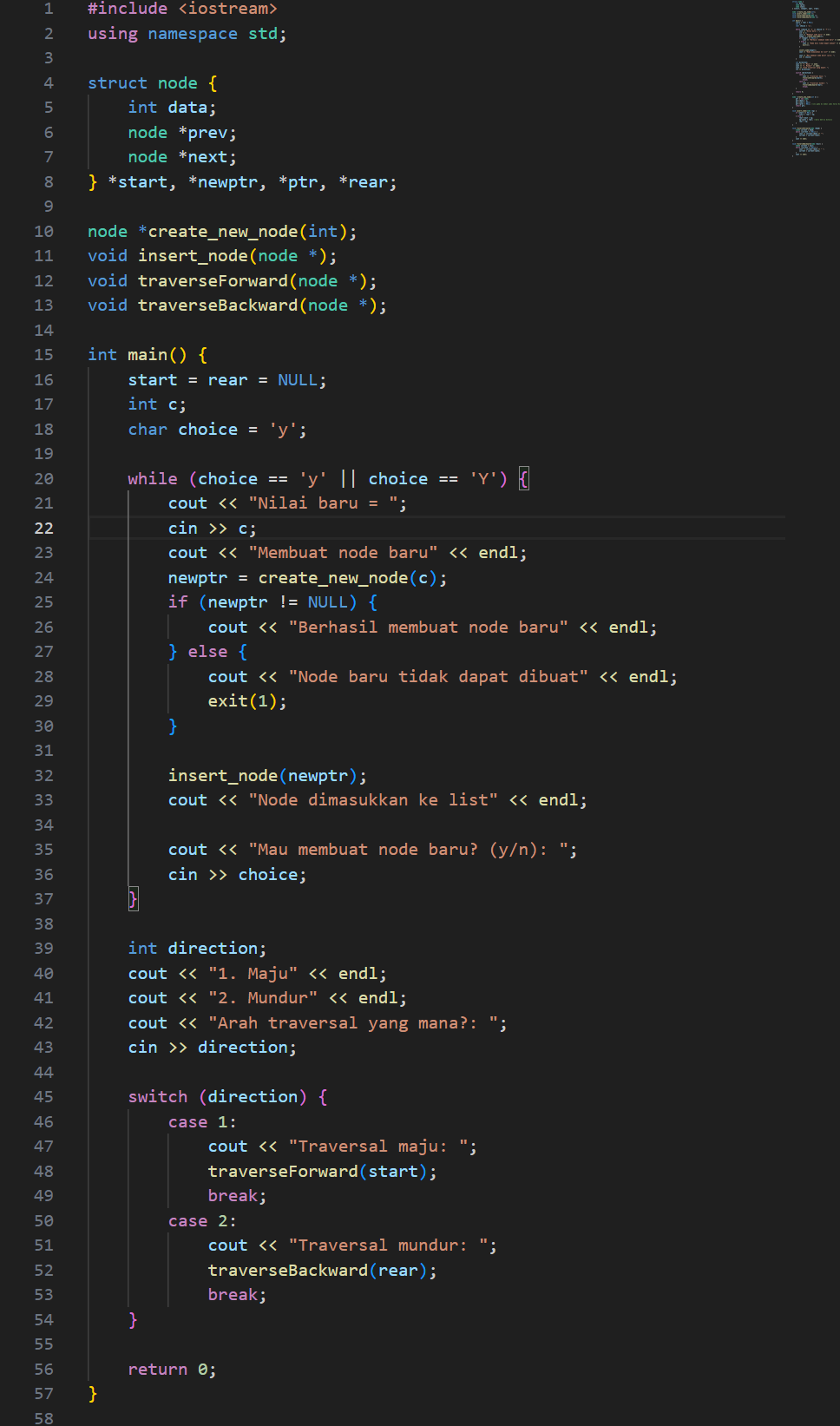
* 1. Percobaan 2 : Linked list

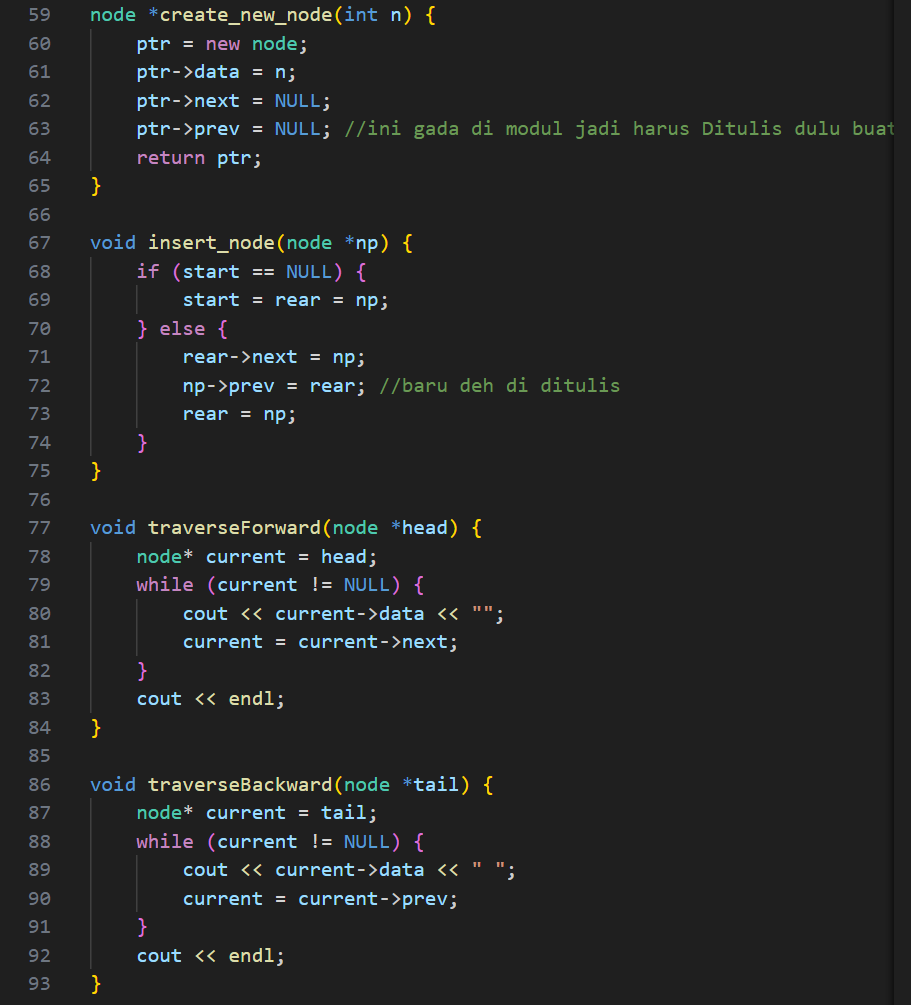
4.2.1 Percobaan 2-1 : Singly Linked List





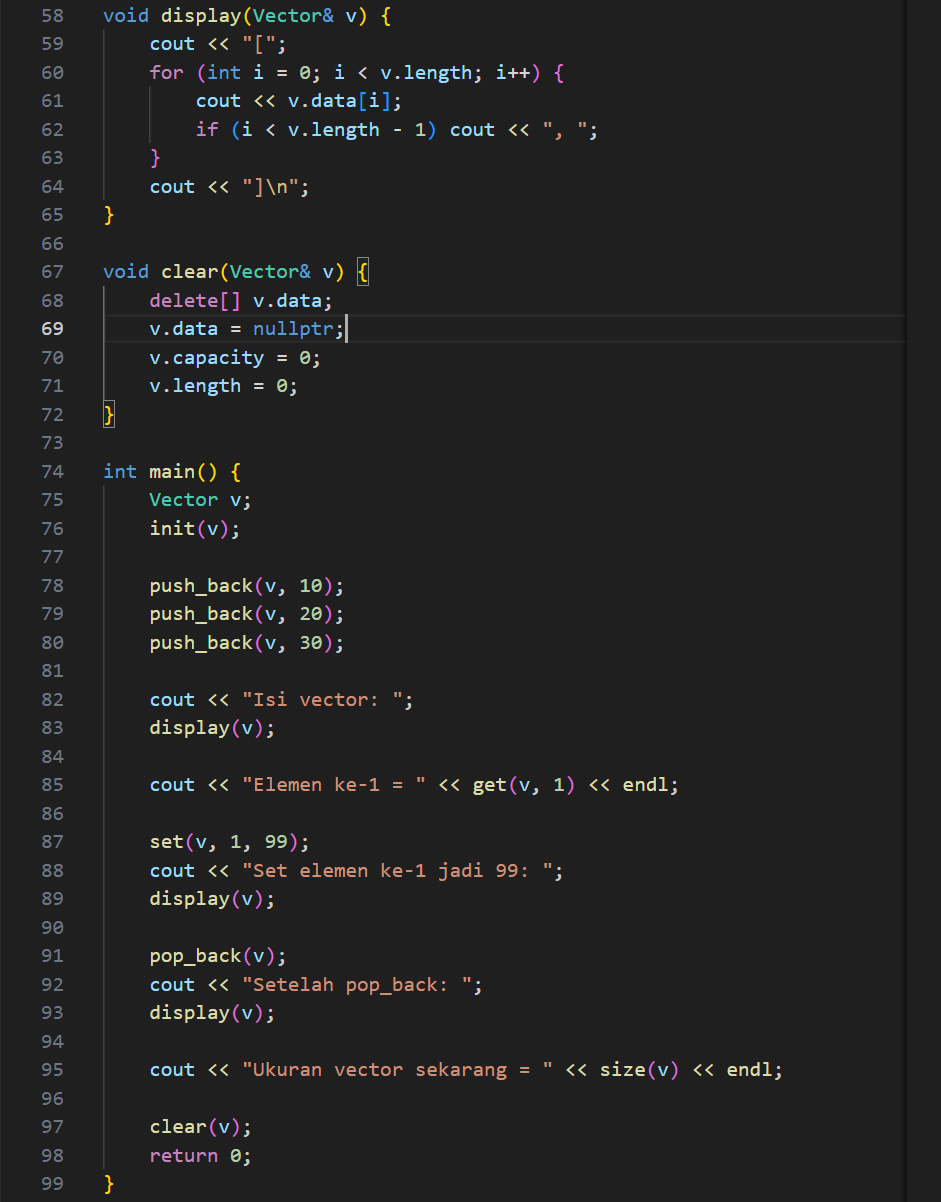
*Gambar 4.2.1 Source Code Percobaan 2-1 : Singly Linked List*

4.2.2 Percobaan 2-2 : Double Linked List



*Gambar 4.2.2 Souce Code Percobaan 2-2 : Double Linked List*

4.3 Percobaan 3 : Vector



*Gambar 4.3 Source Code Percobaan 3 : Vector*

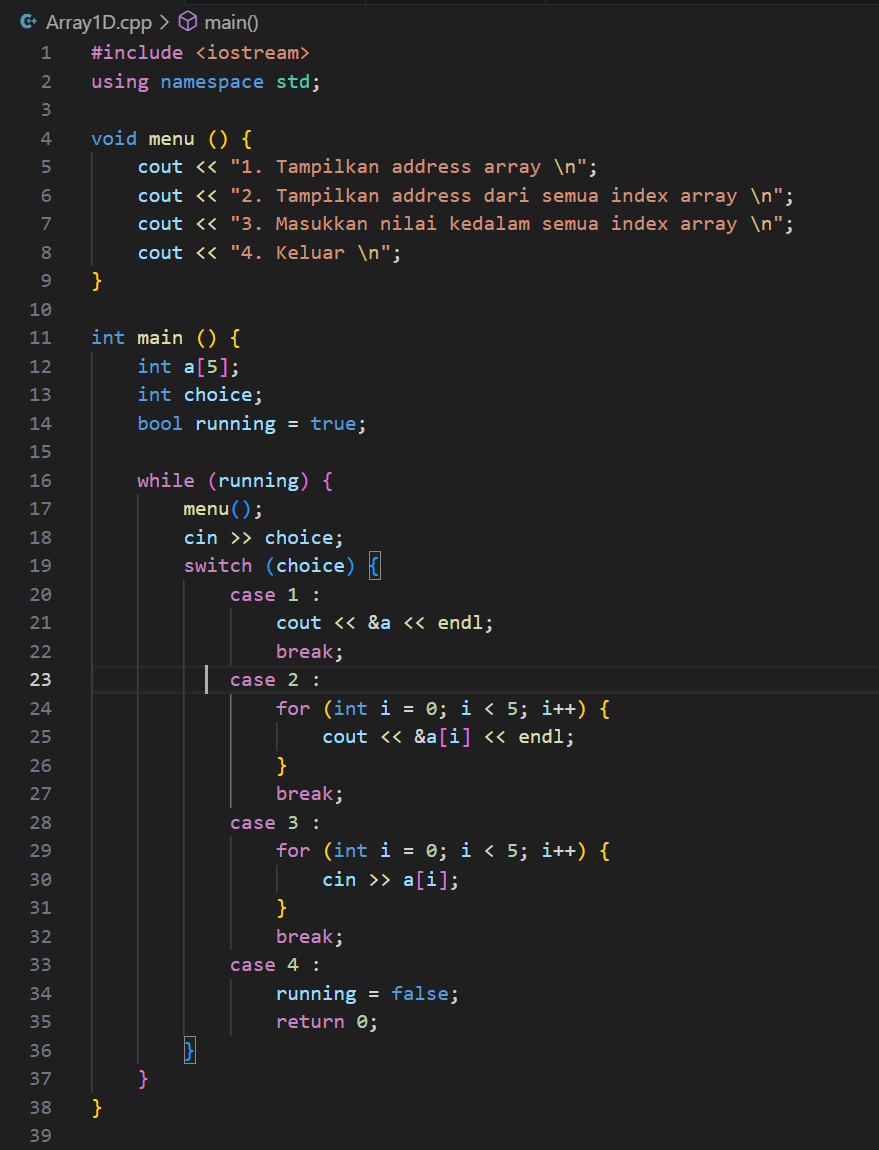
1. PEMBAHASAN

Adapun pembahasan dari percobaan ini adalah sebagai berikut:

* 1. Percobaan 1 : Array

5.1.1 Percobaan 1-1 : Array 1D

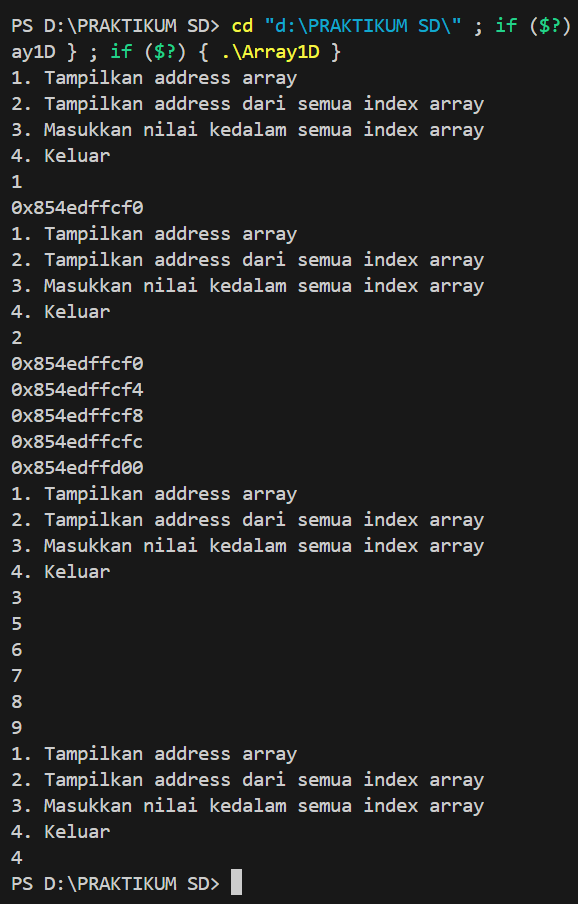
5.1.1.a Source Code Percobaan 1-1 : Array 1D



*Gambar 5.1.1.a Source Code Percobaan ­­­­1-1 : Array 1D*

Berdasarkan gambar 5.1.1.a, pada baris 1 terdapat penggunaan library #include iostream yang digunakan agar program bisa menggunakan perintah input atau output dengan bentuk cin dan cout untuk membaca dan menulis data. Pada baris 2 terdapat using namespace std; yang bertujuan untuk menggunakan fungsi dan objek library tanpa perlu menambahkan "std::". Pada baris 4 terdapat fungsi menu yang akan menampilkan pilihan berbentuk teks dilayar yang didefinisikan oleh cout dari baris ke 5-8. Pilihan pertama digunakan untuk menampilkan alamat array, pilihan kedua untuk menampilkan semua alamat dari semua indeks array, pilihan ketiga digunakan untuk menginputkan suatu nilai ke dalam semua indeks array, dan pilihan keempat digunakan untuk keluar dari program. Selanjutnya masuk pada baris ke 11 terdapat int main yang mendefinisikan program Utama dimulai. Pada baris 12 sampai 14 berisi beberapa deklarasi yaitu pada baris 12 terdapat deklarasi array berkapasitas 5 elemen, lalu pada baris 13 terdapat variable choice yang akan menyimpan pilihan yang diinputkan serta akan menjalankan program sesuai pilihan pengguna, dan pada baris 14 terdapat bool running = true; sebagai deklarasi dari running yang akan selalu bernilai true. Pada baris 16 sampai 34 terdapat nested loop. Pada baris 16 terdapat while (running) yang akan terus berjalan karena running selalu bernilai true jadi perulangan akan terus bernilai true, pada baris 17 terdapat fungsi menu yang digunakan untuk menampilkan pilihan dari 1 sampai 4 yang kemudian pada baris 8 program akan meminta inputan dari user yang akan diterima menggunakan variable choice. Pada baris 19 terdapat switch (choice) yang berfungsi untuk menampilkan output sesuai dengan angka yang diinputkan oleh user. Jika user memilih no 1, maka program akan menampilkan alamat memori dari array. selanjutnya apabila user memilih no 2, maka program akan melakukan perulangan for sebanyak 5 kali untuk menampilkan alamat dari setiap indeks yang ada dalam array. Jika user memilih no 3, maka program akan melakukan perulangan sebanyak 5 kali ntuk menerima inputan nilai dari indeks 0 sampai indeks 4 (5 elemen). Lalu Ketika user memilih no 4, maka program akan tertutup dikarenakan kondisi perulangan telah berubah menjadi false. Terakhir pada baris 35 terdapat return 0 yang menandakan program telah berjalan tanpa adanya kesalahan.

5.1.1.b Output Percobaan 1-1 : Array 1D

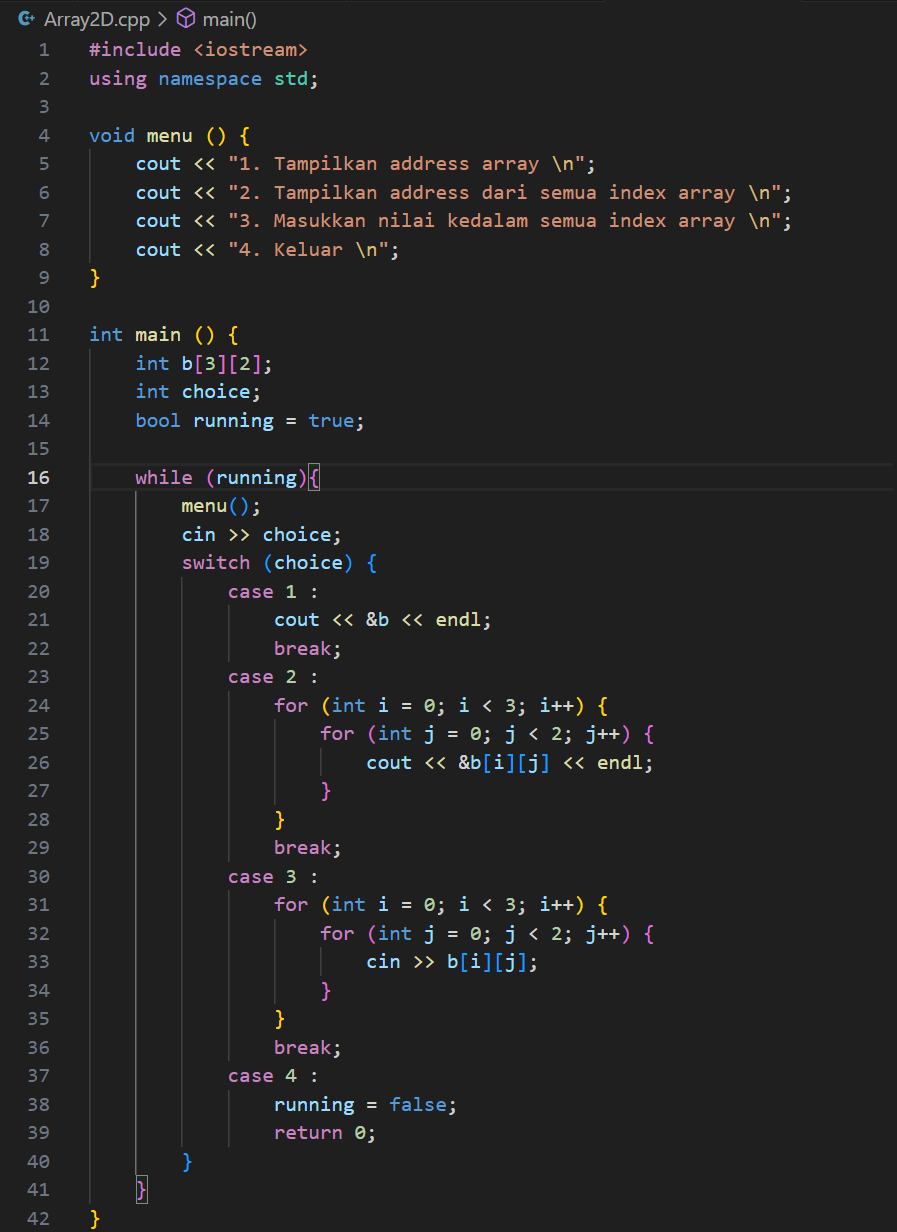


*Gambar 5.1.1.b Output Percobaan 1-1 : Array 1D*

Berdasarkan gambar 5.1.1.b, Pada output terlihat bahwa pilihan 1 sampai 4 muncul berulang kali setelah pengguna memasukkan pilihan, hal ini dikarenakan loop while (running) selalu memanggil fungsi menu sebab nilainya selalu true. Pada baris 5 ketika pengguna memilih 1, maka program menampilkan alamat dari memori array­­­­ yaitu 0x854edffcf0 yang merupakan output dari kode program baris 21 yaitu cout << &a. Pada baris 11 ketika pengguna memilih 2, maka program akan melakukan perulangan for sebanyak 5 kali untuk menampilkan alamat setiap elemen array yang mana output ini merupakan hasil dari kode program pada baris 24 dan 25. Pada baris 21 ketika pengguna memilih 3, maka program akan meminta user untuk menginputkan 5 angka untuk mengisi array a dikarenakan kapasitas yang ditentukan adalah 5, output ini merupakan hasil dari kode program pada baris 29 dan 30. Terakhir, pada baris 31 ketika pengguna memilih 4, maka perulangan akan berhenti dan kode program tertutup dikarenakan pada kondisi 4 running berubah menjadi false, maka perulangan juga akan bernilai false, output ini merupakan hasil dari kode program pada baris 34.

5.1.2 Percobaan 1-2 : Array 2D

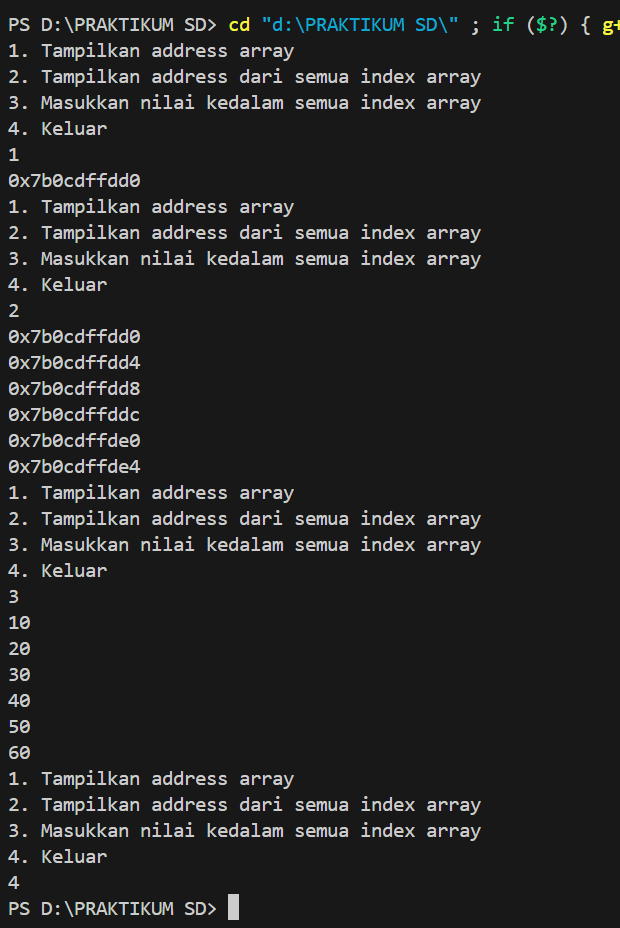
5.1.2.a Source Code Percobaan 1-2 : Array 2D



*Gambar 5.1.2.a Source Code Percobaan 1-2 : Array 2D*

Berdasarkan gambar 5.1.2.a, pada baris 1 terdapat penggunaan library #include iostream yang digunakan agar program bisa menggunakan perintah input atau output dengan bentuk cin dan cout untuk membaca dan menulis data. pada baris 2 terdapat using namespace std; yang bertujuan untuk menggunakan fungsi dan objek library tanpa perlu menambahkan "std::". Pada baris 4 terdapat fungsi menu yang akan menampilkan pilihan berbentuk teks dilayar yang didefinisikan oleh cout dari baris ke 5-8. Pilihan pertama digunakan untuk menampilkan alamat array, pilihan kedua untuk menampilkan semua alamat dari semua indeks array, pilihan ketiga digunakan untuk menginputkan suatu nilai ke dalam semua indeks array, dan pilihan keempat digunakan untuk keluar dari program. Selanjutnya masuk pada baris ke 11 terdapat int main yang mendefinisikan program Utama dimulai. Pada baris 12 sampai 14 berisi beberapa deklarasi yaitu pada baris 12 terdapat deklarasi array berkapasitas 3 baris dan 2 kolom, lalu pada baris 13 terdapat variable choice yang akan menyimpan pilihan yang diinputkan serta akan menjalankan program sesuai pilihan pengguna, dan pada baris 14 terdapat bool running = true sebagai deklarasi dari running yang akan selalu bernilai true. Pada baris 16 sampai 38 terdapat nested loop. Pada baris 16 terdapat while (running) yang akan terus berjalan karena running selalu bernilai true jadi perulangan akan terus bernilai true, pada baris 17 terdapat fungsi menu yang digunakan untuk menampilkan pilihan dari 1 sampai 4 yang kemudian pada baris 18 program akan meminta inputan dari user yang akan diterima menggunakan variable choice. Pada baris 19 terdapat switch (choice) yang berfungsi untuk menampilkan output sesuai dengan angka yang diinputkan oleh user. Jika user memilih nomor 1, maka program akan menampilkan alamat memori dari array. selanjutnya apabila user memilih nomor 2, maka program akan melakukan perulangan for sebanyak 6 kali (3 baris x 2 kolom) untuk menampilkan alamat dari setiap indeks yang ada dalam array. Jika user memilih nomor 3, maka program akan melakukan perulangan sebanyak 6 kali (3 baris 2 kolom) untuk menerima inputan nilai. Lalu Ketika user memilih nomor 4, maka program akan tertutup dikarenakan kondisi perulangan telah berubah menjadi false. Terakhir pada baris 39 terdapat return 0 yang menandakan program telah berjalan tanpa adanya kesalahan.

5.1.2.b Output Percobaan 1-2 : Array 2D

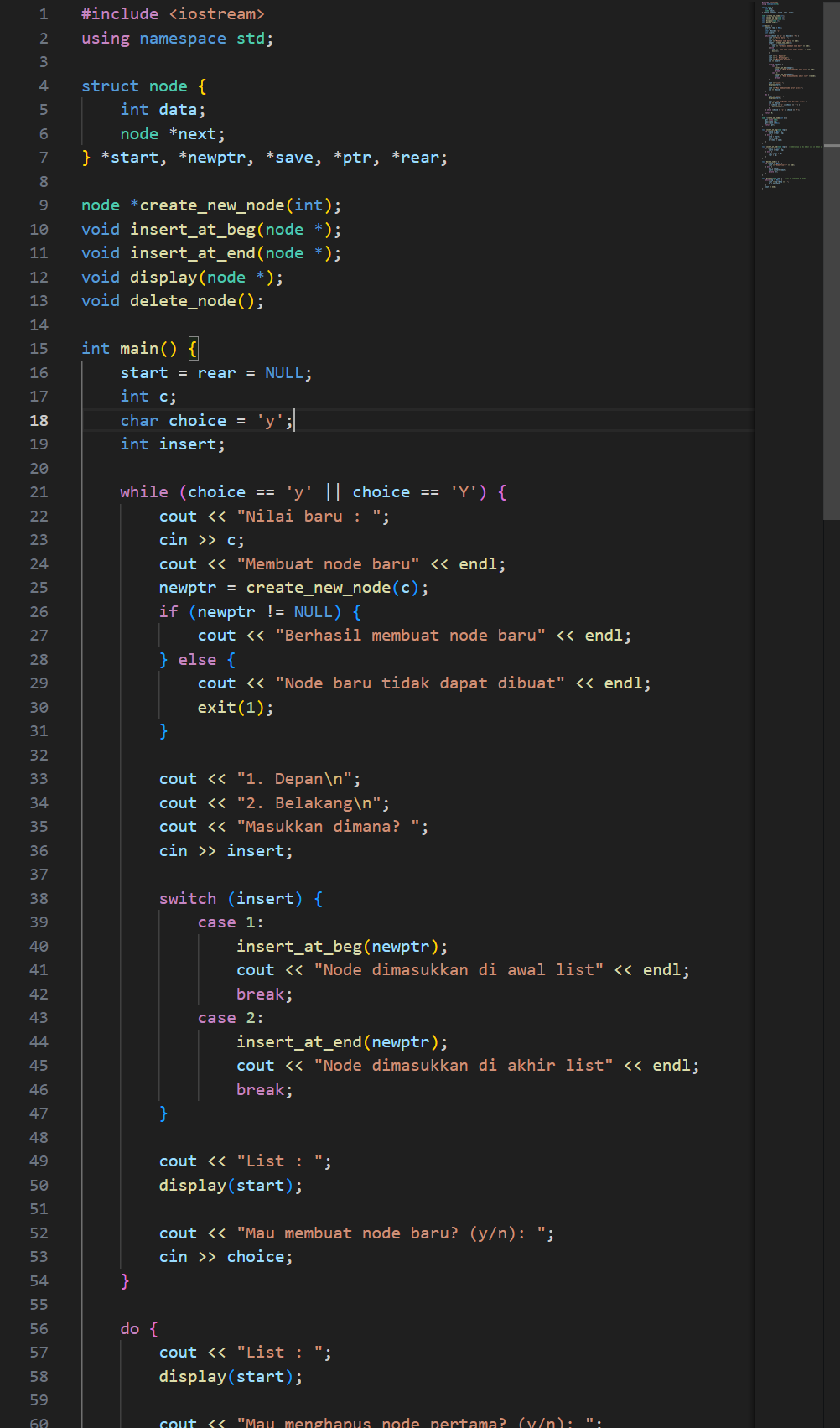


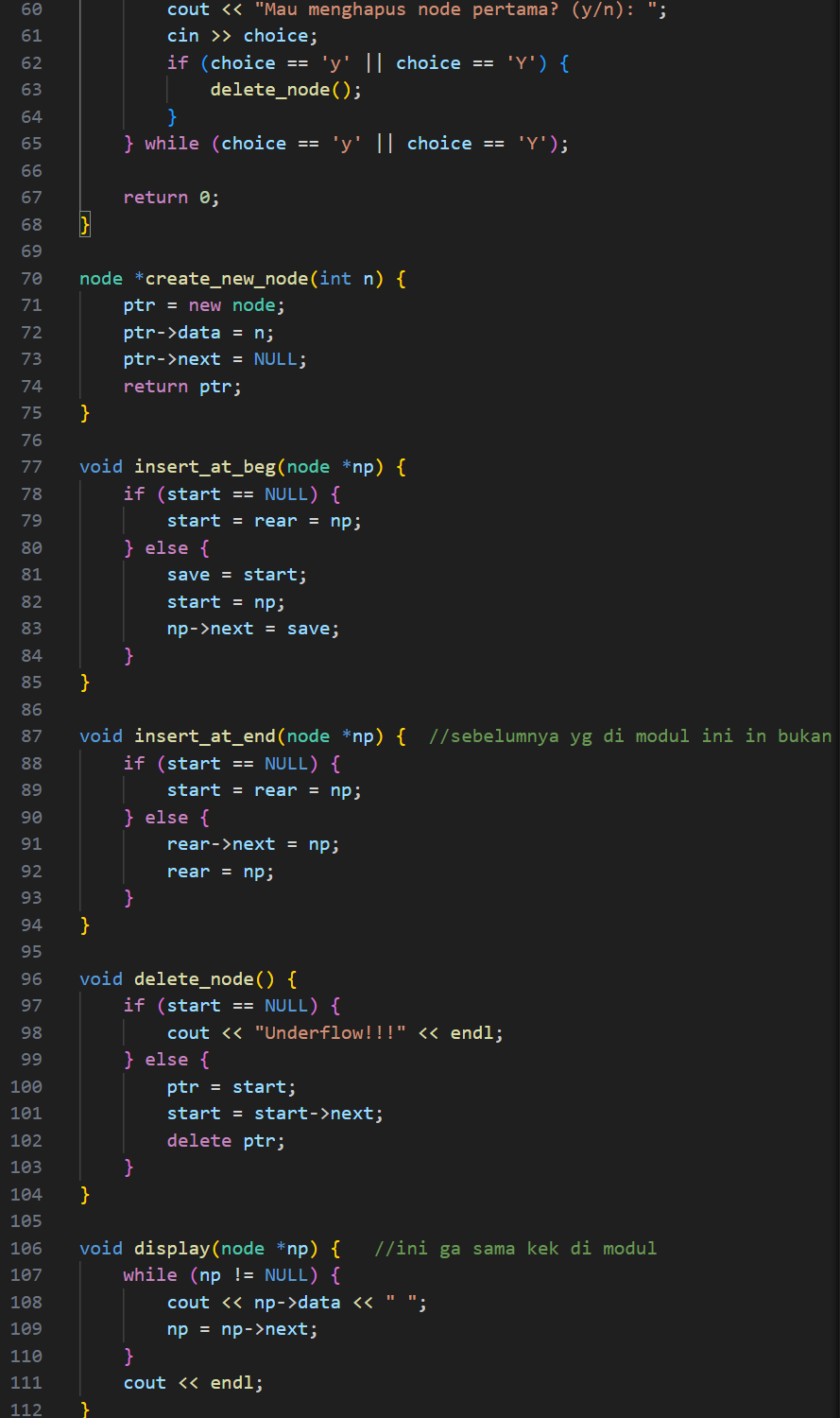
*Gambar 5.1.2.b Output Percobaan 1-2 : Array 2D*

Berdasarkan gambar 5.1.2.b, Pada output terlihat bahwa pilihan 1 sampai 4 muncul berulang kali setelah pengguna memasukkan pilihan, hal ini dikarenakan loop while (running) selalu memanggil fungsi menu sebab nilainya selalu true. Pada baris 5 ketika pengguna memilih 1, maka program menampilkan alamat dari memori array­­­­ yaitu 0x7b0cdffdd0 yang merupakan output dari kode program baris 21 yaitu cout << &b. Pada baris 11 ketika pengguna memilih 2, maka program akan melakukan perulangan for sebanyak 6 kali untuk menampilkan alamat setiap elemen array dari 3 baris dan 2 kolom yang mana output ini merupakan hasil dari kode program pada baris 24 sampai 26. Pada baris 22 ketika pengguna memilih 3, maka program akan meminta pengguna untuk menginputkan 6 angka untuk mengisi array dengan 3 baris dan 2 kolom dikarenakan kapasitas yang ditentukan adalah 3 baris dan 2 kolom, output ini merupakan hasil dari kode program pada baris 31 sampai 33. Terakhir, pada baris 33 ketika pengguna memilih 4, maka perulangan akan berhenti dan kode program tertutup dikarenakan pada kondisi 4 running berubah menjadi false, maka perulangan juga akan bernilai false, output ini merupakan hasil dari kode program pada baris 38.

* 1. Percobaan 2 : Linked List

5.2.1 Percobaan 2-1 : Singly Linked List

5.2.1.a Source Code Percobaan 2-1 : Singly Linked List

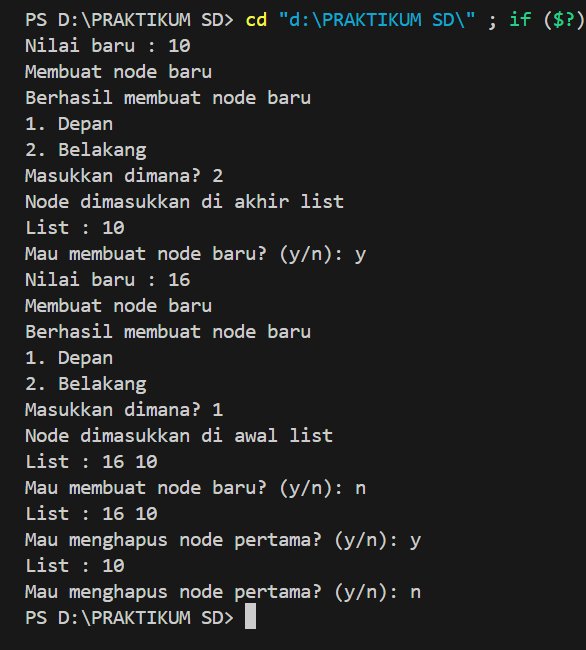


*Gambar 5.2.1.a Source Code Percobaan 2-1 : Singly Linked List*

Berdasarkan gambar 5.2.1.a, pada baris 1 terdapat penggunaan library #include iostream yang digunakan agar program bisa menggunakan perintah input atau output dengan bentuk cin dan cout untuk membaca dan menulis data. pada baris 2 terdapat using namespace std; yang bertujuan untuk menggunakan fungsi dan objek library tanpa perlu menambahkan "std::". Pada baris 2 terdapat using namespace std; yang berfungsi agar pemanggilan fungsi dan objek dari library standar lebih ringkas tanpa menuliskan std:: di depannya. Pada baris 4 terdapat struct node { yang digunakan untuk membuat elemen dari linked list. Elemen ini memiliki dua bagian, yaitu int data pada baris 5 untuk menyimpan nilai dari pengguna, dan node \*next pada baris 6 yang berfungsi sebagai pointer menuju node berikutnya. Pada baris 7 terdapat beberapa pointer, yaitu \*start sebagai penunjuk node pertama, \*newptr untuk menyimpan node baru yang dibuat, \*save sebagai penyimpanan sementara (misalnya saat traversal), \*ptr untuk mengecek node dari awal hingga akhir, dan \*rear sebagai penunjuk node terakhir agar penambahan data di akhir list lebih efisien. Pada baris 9 sampai 13 terdapat deklarasi beberapa fungsi utama, yaitu node \*create\_new\_node(int); untuk membuat node baru, insert\_at\_end(node \*) untuk menambahkan node di bagian akhir, display(node \*) untuk menampilkan semua isi node dalam linked list, dan delete\_node() untuk menghapus node dari list. Pada baris 15 terdapat fungsi main yang menandakan program utama dimulai, dengan pointer start dan rear yang diinisialisasi dengan NULL, menandakan list masih kosong. Pada baris 17 terdapat variabel int c untuk menyimpan pilihan menu dari pengguna, pada baris 18 terdapat variabel char choice dengan nilai awal 'y' sebagai kontrol perulangan agar program tetap berjalan, dan pada baris 19 terdapat variabel int insert untuk menampung data sebelum disimpan ke dalam node baru. Pada baris 21 sampai 30 terdapat perulangan while (choice == 'y' || choice == 'Y') yang membuat program tetap berjalan selama pengguna mengetikkan y atau Y. Pada baris 22 program menampilkan teks “Nilai baru”, lalu pada baris 23 pengguna memasukkan data yang disimpan ke variabel c. Pada baris 24 muncul pesan “Membuat node baru” dan pada baris 25 fungsi create\_new\_node(c) dipanggil untuk membuat node baru. Jika node berhasil dibuat (newptr tidak bernilai NULL), maka ditampilkan pesan “Berhasil membuat node baru”, tetapi jika gagal, ditampilkan pesan “Node baru tidak dapat dibuat” dan program dihentikan dengan exit(1). Setelah node dibuat, pada baris 33 sampai 35 program menampilkan menu dengan dua pilihan yaitu “1. Depan” dan “2. Belakang”, lalu pada baris 36 program meminta pengguna untuk memasukkan pilihan dengan teks “Masukkan dimana?” dan menyimpannya di variabel insert. Pada baris 38 digunakan struktur switch untuk menentukan posisi penyisipan node. Jika memilih 1, maka pada baris 40 fungsi insert\_at\_beg(newptr) dijalankan dan ditampilkan pesan “Node dimasukkan di awal list” pada baris 41. Jika memilih 2, maka pada baris 44 fungsi insert\_at\_end(newptr) dipanggil dan ditampilkan pesan “Node dimasukkan di akhir list” pada baris 45.

Pada baris 49 fungsi display(start) dipanggil untuk menampilkan seluruh isi list dari node pertama hingga terakhir. Kemudian pada baris 52 program menanyakan kembali dengan teks “Mau membuat node baru? (y/n)” dan hasilnya disimpan di variabel choice pada baris 53. Jika pengguna memilih y, maka perulangan akan diulang untuk menambah node baru, sedangkan jika memilih n, maka program keluar dari perulangan. Setelah itu, pada baris 56 program masuk ke perulangan do-while yang menampilkan isi list pada baris 57 dan menanyakan apakah pengguna ingin menghapus node pertama pada baris 60. Jawaban disimpan pada variabel choice, jika y maka pada baris 62–63 fungsi delete\_node() dijalankan untuk menghapus node paling depan. Perulangan ini akan terus berjalan selama pengguna memilih y karena terdapat kondisi while pada baris 65. Pada baris 70 terdapat fungsi create\_new\_node(int n) untuk membuat node baru. Pada baris 71 pointer ptr dialokasikan untuk node baru, pada baris 72 nilai n disimpan ke dalam data, pada baris 73 pointer next diatur NULL, dan pada baris 74 node baru dikembalikan lewat return ptr. Pada baris 77 terdapat fungsi insert\_at\_beg(node \*np) untuk menambahkan node di awal list. Pada baris 78–79 dilakukan pengecekan, jika start == NULL berarti list kosong, maka node baru langsung menjadi node pertama sekaligus terakhir dengan start dan rear menunjuk ke np. Jika list tidak kosong (baris 80–83), pointer save menyimpan alamat node pertama, lalu start digeser menunjuk ke node baru, dan node baru dihubungkan ke node lama dengan np->next = save. Pada baris 87 terdapat fungsi insert\_at\_end(node \*np) untuk menambahkan node di akhir list. Pada baris 88–89 jika start == NULL, node baru otomatis menjadi node pertama sekaligus terakhir. Namun jika list sudah berisi node (baris 91–92), node baru disambungkan ke node terakhir dengan rear->next = np, lalu pointer rear diperbarui untuk menunjuk node baru tersebut. Pada baris 96 terdapat fungsi delete\_node() untuk menghapus node pertama dari list. Pada baris 97–98 dilakukan pengecekan, jika start == NULL berarti list kosong dan ditampilkan pesan “Underflow!!!”. Namun jika tidak kosong (baris 99–102), pointer ptr menunjuk node pertama, lalu start digeser ke node berikutnya, dan node lama yang ditunjuk oleh ptr dihapus dengan delete ptr. Pada baris 106 terdapat fungsi display(node \*np) untuk menampilkan semua isi node dalam list. Pointer np digunakan sebagai penunjuk awal (biasanya start). Pada baris 107 dilakukan perulangan while (np != NULL) yang akan berjalan selama masih ada node. Pada baris 108 setiap data node dicetak, lalu pada baris 109 pointer np digeser ke node berikutnya. Proses ini berulang sampai node terakhir. Setelah selesai, pada baris 111 ditambahkan baris baru dengan cout << endl; agar tampilan output lebih rapi.

5.2.1.b Output Percobaan 2-1 : Singly Linked List

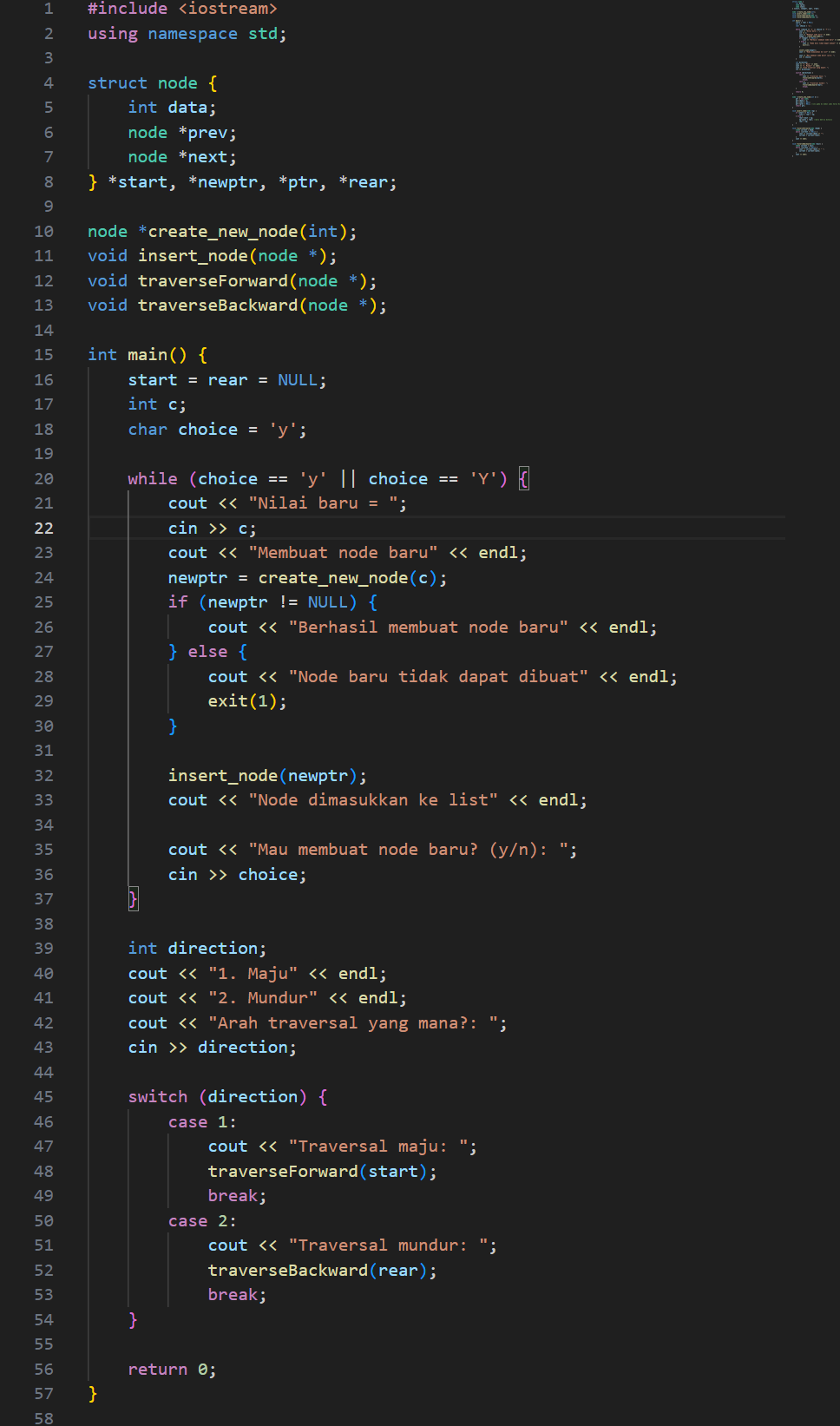


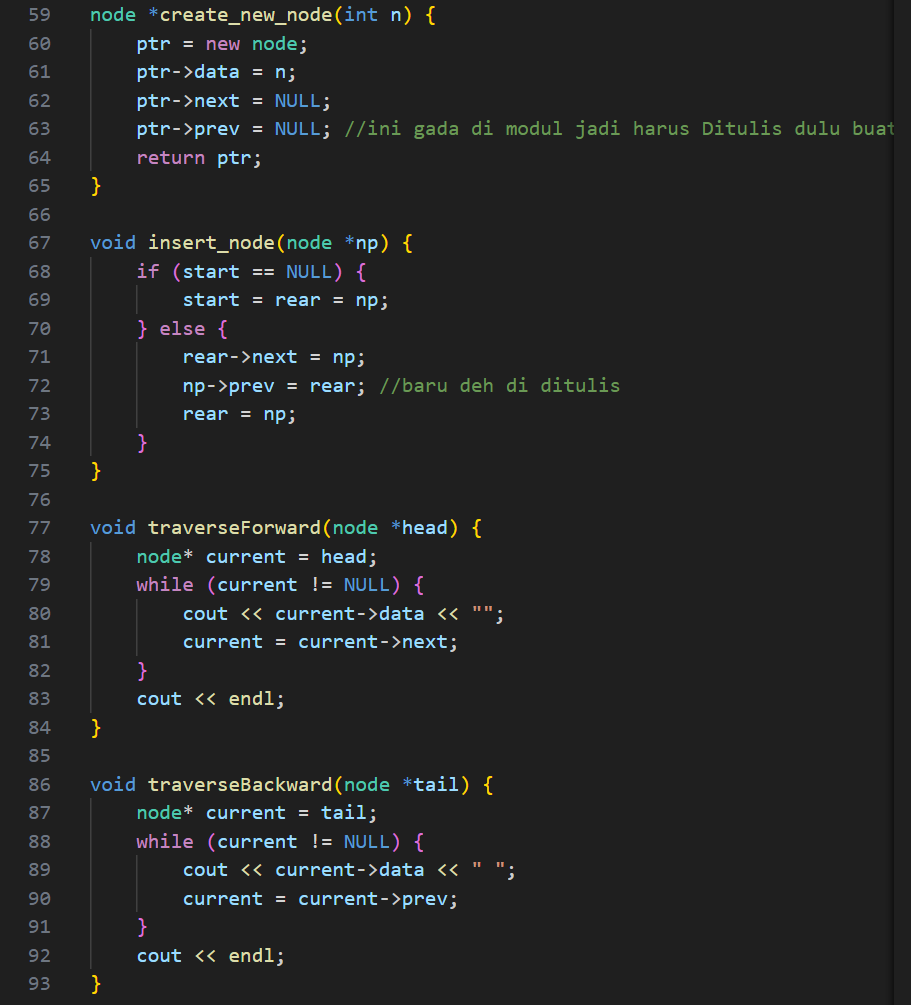
*Gambar 5.2.1.b Output Percobaan 2-1 : Singly Linked List*

Berdasarkan gambar 5.2.1.b, pada baris 1 terdapat output "Nilai baru:" yang muncul karena perintah cout pada kode program baris 22, selanjutnya pengguna memasukkan angka 10 lalu program menampilkan pesan “Membuat node baru” pada baris 2 dan “Berhasil membuat node baru” di baris 3, output ini merupakan hasil dari fungsi create new node yang telah dideklarasikan pada kode program baris ke 9 yang kemudian dipanggil pada kode program baris 25 untuk kemudian disimpan dalam variable c. Kemudian, pada baris 4 dan 5 program memberikan pilihan posisi penyisipannya yaitu depan atau belakang, lalu di baris 6 ketika pengguna memilih 2, maka pada baris 7 akan keluar output "node dimasukkan di akhir list", output ini merupakan hasil dari kode program baris 44 dan 45, kemudian program akan mencetak "list : 10" pada baris 8. Pada baris 9 program akan menanyakan apakah ingin membuat node baru lagi, pengguna menjawab y, sehingga perulangan berjalan Kembali, baris ini merupakan hasil dari kode program baris 52 dan 53. Selanjutnya pengguna Kembali memasukkan nilai yakni 16, dan node baru ditambahkan, kemudian program Kembali memberikan pilihan posisi yakni depan atau belakang. Saat pengguna memilih 1, maka program akan menjalankan kode yan ada paa baris 40 dan 41 yaitu fungsi "insert at beg" sehingga node 16 ditambahkan di awal list, yang kemudian menghasilkan tampilan “List : 16 10" yang merupakan hasil dari kode program pada baris 49 dan 50. Selanjutnya program kembali menanyakan apakah ingin menambah node baru, tetapi pengguna menjawab n, sehingga perulangan berhenti. Program lalu masuk ke bagian do-while yang menampilkan isi list kembali sebagai “List : 16 10” dan menanyakan apakah pengguna ingin menghapus node pertama. Karena pengguna menjawab y, maka fungsi delete\_node() dijalankan dan dan program menghapus node pertama yakni 16, kemudian program mencetak node yang tersisa yaitu 10, lalu Kembali menanyakan "mau menghapus node pertama?", namun dikarenakan pengguna menjawab n maka program akan tertutup karena perulangannya sudah dihentikan.

5.2.2 Percobaan 2-2 : Double Linked List

5.2.2.a Source Code Percobaan 2-2 : Double Linked List



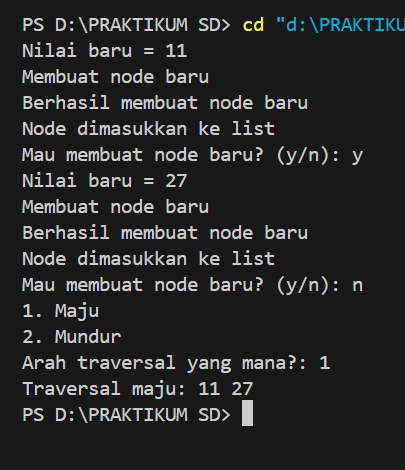


*Gambar 5.2.2.a Source Code Percobaan 2-2 : Double Linked List*

Berdasarkan gambar 5.2.2.a, pada baris 1 terdapat penggunaan library #include <iostream> yang digunakan agar program bisa menggunakan perintah input atau output dengan cin dan cout. Pada baris 2 terdapat using namespace std; yang bertujuan agar fungsi dan objek dari library standar dapat digunakan tanpa harus menambahkan std::.Pada baris 4 terdapat struct node yang berfungsi untuk membuat elemen dari linked list. Elemen ini memiliki tiga bagian, yaitu int data pada baris 5 untuk menyimpan nilai, node \*prev pada baris 6 sebagai pointer menuju node sebelumnya, dan node \*next pada baris 7 sebagai pointer menuju node berikutnya. Selanjutnya, pada baris 8 terdapat deklarasi beberapa pointer, yaitu \*start sebagai penunjuk node pertama, \*rear sebagai penunjuk node terakhir, \*newptr untuk menyimpan node baru, dan \*ptr sebagai pointer serbaguna untuk traversal. Pada baris 10 sampai 13 terdapat deklarasi fungsi-fungsi utama, yaitu create\_new\_node(int) untuk membuat node baru dari input pengguna, insert\_node(node\*) untuk menambahkan node ke dalam list, traverseForward(node\*) untuk menampilkan isi list dari depan ke belakang, dan traverseBackward(node\*) untuk menampilkan isi list dari belakang ke depan. Pada baris 15 terdapat fungsi main() yang menandakan program utama dimulai. Pada baris 16 pointer start dan rear diinisialisasi dengan NULL, yang berarti list masih kosong. Pada baris 17 terdapat variabel int c untuk menyimpan pilihan menu, sedangkan pada baris 18 terdapat variabel char choice bernilai awal 'y' yang digunakan sebagai kontrol agar perulangan dapat terus berjalan. Pada baris 20 sampai 29 terdapat perulangan while (choice == 'y' || choice == 'Y') yang membuat program tetap berjalan selama pengguna memasukkan y atau Y. Pada baris 21 program menampilkan teks “Nilai baru”, lalu pada baris 22 nilai yang dimasukkan pengguna disimpan ke dalam variabel c. Pada baris 23 ditampilkan pesan “Membuat node baru”, dan pada baris 24 fungsi create\_new\_node(c) dipanggil untuk membuat node baru. Jika node berhasil dibuat (newptr tidak bernilai NULL), maka ditampilkan pesan “Berhasil membuat node baru”. Jika gagal, akan muncul pesan “Node baru tidak dapat dibuat” dan program dihentikan dengan exit(1).

Pada baris 32 fungsi insert\_node(newptr) dipanggil untuk memasukkan node baru ke dalam list, kemudian pada baris 33 ditampilkan pesan “Node dimasukkan ke list”. Selanjutnya pada baris 35 program menanyakan kembali apakah pengguna ingin membuat node baru dengan menampilkan teks “Mau membuat node baru? (y/n):” dan hasil input disimpan pada variabel choice di baris 36. Jika pengguna menjawab y, perulangan akan kembali dijalankan. Setelah proses penambahan selesai, program masuk ke bagian traversal. Pada baris 39 sampai 43 ditampilkan menu pilihan “1. Maju” dan “2. Mundur”, kemudian pada baris 40 input dari pengguna disimpan ke dalam variabel direction. Pada baris 45 digunakan switch(direction) untuk menentukan arah traversal. Jika memilih 1, maka pada baris 47 ditampilkan teks “Traversal maju:” dan fungsi traverseForward(start) dipanggil pada baris 48 untuk menampilkan list dari node pertama hingga terakhir. Jika memilih 2, maka pada baris 51 ditampilkan teks “Traversal mundur:” dan fungsi traverseBackward(rear) dipanggil untuk menampilkan list dari node terakhir hingga pertama. Fungsi create\_new\_node(int n) terdapat pada baris 59. Pada baris 60 pointer ptr dialokasikan untuk node baru, pada baris 61 nilai n disimpan ke dalam data, pada baris 62–63 pointer next dan prev diinisialisasi dengan NULL, lalu node baru dikembalikan dengan return ptr. Fungsi insert\_node(node \*np) terdapat pada baris 67. Jika list masih kosong, maka pada baris 69 node baru dijadikan node pertama sekaligus terakhir dengan start = rear = np;. Jika sudah ada isi, maka pada baris 71–73 node baru ditempatkan di akhir list dengan rear->next = np, np->prev = rear, lalu pointer rear diperbarui menunjuk ke node baru. Fungsi traverseForward(node \*head) terdapat pada baris 70. Pointer current digunakan untuk menelusuri list dari depan hingga NULL menggunakan perulangan while pada baris 79. Pada baris 80 nilai data ditampilkan, lalu pada baris 81 pointer digeser ke node berikutnya. Setelah mencapai akhir, pada baris 83 ditambahkan baris baru dengan endl. Sebaliknya, fungsi traverseBackward(node \*tail) pada baris 86 menelusuri list dari belakang. Pointer current diawali pada tail, kemudian pada baris 89 data ditampilkan, baris 90 pointer digeser ke node sebelumnya, hingga selesai dan pada baris 92 ditambahkan baris baru.

5.2.2.b Output Percobaan 2-2 : Double Linked List



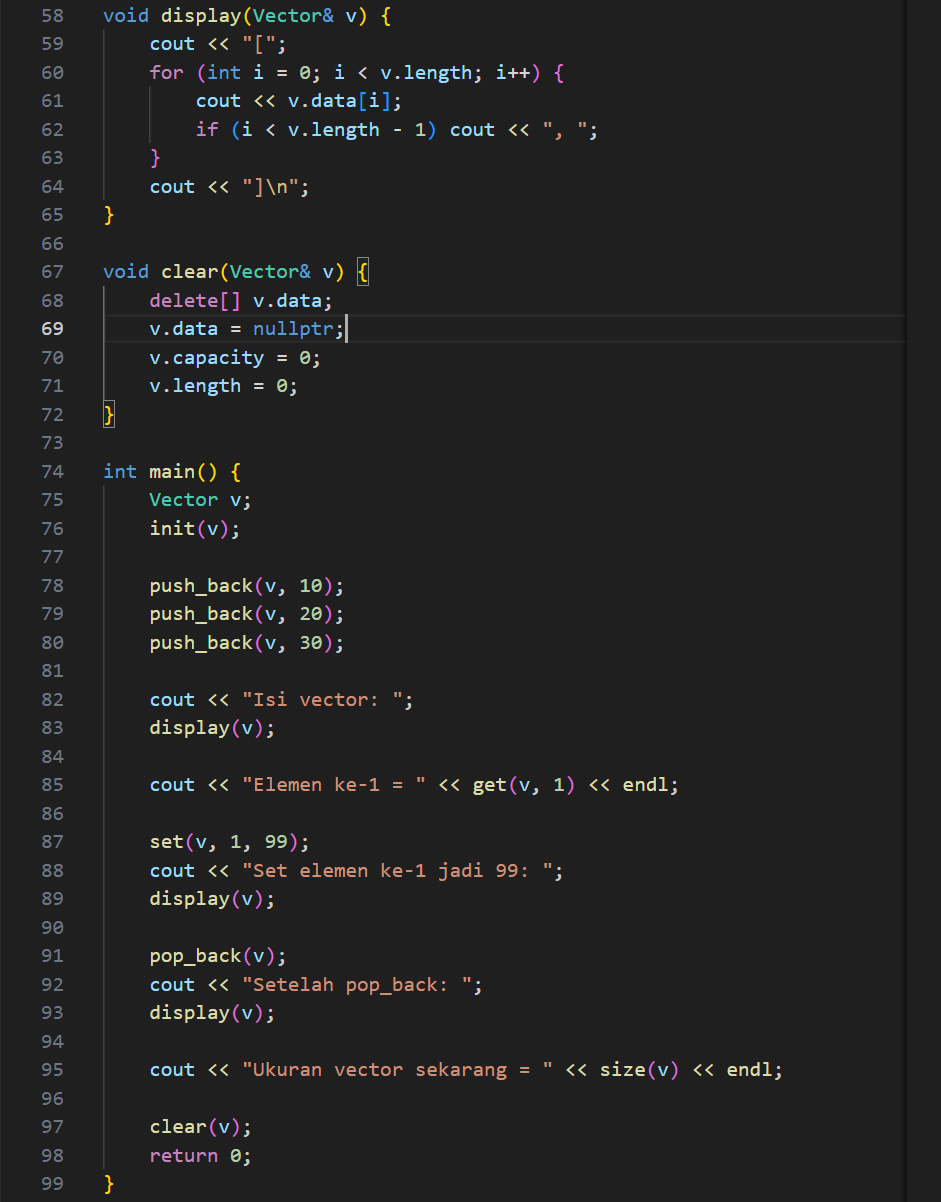
*Gambar 5.2.2.b Output Percobaan 2-2 : Double Linked List*

Berdasarkan gambar 5.2.2.b, Pada baris 1 terdapat output "Nilai baru =" yang muncul karena perintah cout pada kode program baris 21. Selanjutnya pengguna memasukkan angka 11 melalui perintah cin pada baris 22. Setelah itu, pada baris 2 ditampilkan pesan "Membuat node baru" yang merupakan hasil dari perintah cout pada baris 23. Kemudian program memanggil fungsi create\_new\_node(c) pada baris 24. Karena node berhasil dibuat, maka pada baris 3 muncul pesan "Berhasil membuat node baru" yang merupakan hasil dari kode program pada baris 26. Setelah node berhasil dibuat, fungsi insert\_node(newptr) pada baris 32 dijalankan untuk memasukkan node baru ke dalam list, lalu pada baris 4 ditampilkan pesan "Node dimasukkan ke list" yang berasal dari perintah cout pada baris 33. Pada baris 5 program menanyakan "Mau membuat node baru? (y/n)" yang merupakan hasil dari kode program baris 35. Ketika pengguna menjawab y, maka perulangan while pada baris 20 dijalankan kembali. Pada proses berikutnya, pengguna memasukkan nilai 27, kemudian program kembali menampilkan pesan "Membuat node baru" dan "Berhasil membuat node baru", yang merupakan hasil dari kode program pada baris 21 dan 23, karena node berhasil dibuat. Setelah itu fungsi insert\_node(newptr) pada baris 32 dipanggil lagi untuk menambahkan node ke bagian akhir list, sehingga muncul pesan "Node dimasukkan ke list" dari kode program baris 33. Pada tahap ini isi list adalah 11 dan 27. Ketika pertanyaan pada baris 35 ditampilkan lagi, pengguna menjawab n sehingga perulangan berhenti. Selanjutnya program menampilkan menu "1. Maju" dan "2. Mundur" yang merupakan hasil dari perintah cout pada kode program baris 40 dan 41. Pada baris selanjutnya terdapat output "Arah traversal yang mana?" lalu pengguna memasukkan pilihan. Saat pengguna memilih 1, maka switch pada kode program baris 45 akan menjalankan case 1, sehingga program menampilkan output "Traversal maju:" pada baris 47 dan memanggil fungsi traverseForward(start) pada baris 48. Fungsi ini membaca isi list dari node pertama hingga node terakhir, sehingga hasil yang ditampilkan adalah "Traversal maju: 11 27".

* 1. Percobaan 3 : Vector

5.3.a Source Code Percobaan 3 : Vector



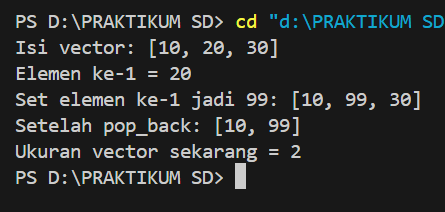


*Gambar 5.3.a Source Code Percobaan 3 : Vector*

Berdasarkan gambar 5.3.a, ada baris 1, terdapat #include <iostream> supaya kita bisa pakai cin dan cout sebagai input dan output yang akan ditampilkan di layar. Pada baris 2, terdapat using namespace std; yang berfungsi supaya kita tidak perlu menggunakan td:: disetiap pemanggilan fungsi standar C++ sehingga penulisannya lebih ringkas.Pada baris 4 terdapat definisi sebuah struktur Bernama vector yang mana didalamnya terdapata 3 struktur yang ada di baris 5 sampai baris 7. Pada baris 5 terdapat pointer bertipe integer yang menunjuk ke vector yang akan menyimpan elemen elemen yang diinputkan oleh user. pada baris 6 terdapat deklarasi capacity yang mana ini berfungsi untuk menyimpan seberapa besar kapasitas atau seeberapa banyak elemen yang dapat ditampung sebelum diperbesar. Pada baris 7 terdapat deklarasi length yang mana ini berfungsi untuk mendeklarasikan berapa banyak elemen yang telah terisi ke dalam kapasitas yang telah ditentukan. Pada baris 10 terdapat fungsi inisialisasi vector yang didalamnya berisi deklarasi beberapa fungsi dari baris ke 11 sampai 13. Pada baris 11 terdapat fungsi untuk menetukan kapasitas untuk dapat menampung sebanyak 2 elemen, lalu pada baris 12 terdapat inisialisasi length bernilai 0 untuk memberi tahu bahwa belum ada elemen yang terisi dalam kapasitas, dan pada baris 13 terdapat program yang berfungsi untuk mengalokasikan memori vector sesuai dengan kapasitas yang ditentukan. Pada baris 16 sampai 23 terdapat fungsi untuk mengubah ukuran vector Ketika telah melebihi kapasitas. pada baris 17 terdapat pointer newData yang berfungsi untuk membuat vector baru dengan ukuran yang diperbesar. Pada baris 18 dan 19 terdapat perulangan yang berfungsi untuk menyalin semua elemen lama untuk dipindahkan pada vector baru. Pada baris 21 terdapat fungsi untuk menghapus vector lama agar tidak terjadi memory leak. Pada baris 22 terdapat fungsi untuk mengganti pointer lama ke vector baru yang lebih besar. Lalu pada baris 23 terdapat fungsi untuk memperbarui informasi kapasitas vector sesuai ukuran baru. pada baris 26 terdapat fungsi yang digunakan untuk menambahkan elemen ke dalam vector. Pada baris 27 dan 28 terdapat kode yang berfungsi untuk memriksaapakah kapasitas penyimpanan masih tersedia. Jika jumlah elemen yang terisi sama dengan kapasitas maksimum, maka vector akan menggandakan kapasitas menjadi 2 kali lipat. Pada baris 30 terdapat kode yang berfungsi untuk menempatkan nilai baru di indeks terakhir, setelah elemen yang sudah ada sebelumnya. Pada baris 31 terdapat kode yang berfungsi untuk menambahkan nilai length sebanyak satu. Baris 34 berfungsi untuk mendefinisikan fungsi pop back yang digunakan untuk menghapus elemen terakhir dari vector. Pada baris 35 dan 36 terdapat kode yang berfungsi untuk mengecek apakah vector memiliki elemen, jika vector memiliki elemen (v.length > 0) maka program akan mengurangi atau menghapus elemen yang terletak di belakang atau elemen terakhir. Selanjutnya baris 40 terdapat fungsi get yang berfungsi untuk mengembalikan nilai bertipe int dari elemen vector pada indeks yang diinginkan. Pada baris 41 dan 42 terdapat kode yang berfungsi untuk melakukan pengecekan apakah indeks yang diminta lebih besar atau sama dengan 0, serta lebih kecil dari panjang vector. Jika kondisi tersebut benar atau terpenuhi, maka nilai pada posisi indeks tersebut dikembalikan sebagai hasil fungsi.

Pada baris 44, Jika kondisi indeks tidak valid, maka fungsi mengembalikan nilai -1 sebagai tanda bahwa terjadi kesalahan akses data. Pada baris 48 sampai 50 terdapat fungsi set yang menerima sebuah parameter vector, sebuah indeks dan sebuah nilai baru. bagiain ini berfungsi untuk mengecek apakah indeks yang diberikan lebih besar atau sama dengan 0, serta lebih kecil dari panjang vector. Apabila kondisi tersebut valid maka nilai pada indeks tersebut akan diganti dengan nilai baru yang diinputkan. Pada baris 54 dan 55 terdapat fungsi size yang berfungsi untuk memberikan info terkait jumlah elemen yang tersimpan dalam vector atau length. Pada baris 58 terdapat fungsi display yang digunakan untuk menampilkan seluruh elemen yang ada di dalam vector. Pada baris 59 dan 64 terdapat perintah untuk mencetah kurung siku di awal dan akhir dari elemen yang nanti akan ditampilkan. Pada baris 60 sampai 62 terdapat perulangan for yang berjalan dari indeks 0 sampai v.length-1. Dalam perulangan, setiap elemen vector akan ditampilkan, selain itu jika indeks belum mencapai elemen terakhir maka akan dintambahkan tanda koma dan spasi untuk memisahkakn masing masing elemen vector. Pada baris 67 sampai 71 terdapat fungsi clear yang digunakan untuk menghapus seluruh isi vector dan mengembalikannya ke kondisi awal. Baris 68 berfungsi untuk membebaskan memori yang sebelumnya digunakan oleh vector. Lalu, baris 69 berfungsi untuk mengatur pointer agar tidak lagi menunjuk ke memori manapun. Baris 70 dan 71 berfungsi untuk mengatur kapasitas serta jumlah elemen yang ada di dalam vector agar menjadi 0. Pada baris 74, terdapat fungsi untama yang menjadi titik awal eksekusi. Selanjutnya pada baris 75 dan 76 terdapat kode untuk membuat variable v bertipe vector, lalu memanggil fungsi init untuk menginisialisasi vector dengan kapasitas awal 2 dan jumlah elemen 0. Pada baris 78 sampai 80 terdapat fungsi push back yang digunakan untuk menambahkan tiga buah elemen ke dalam vector, yaitu 10,20, dan 30. Selanjutnya pada baris 82 dan 83 berfungsi untuk mencetak "isi vector:" kemudian memanggil fungsi display untuk menampilkan isi vectot yaitu [10,20,30]. Baris 82 terdapat fungsi get yan digunakan untuk mengambil elemen pada indeks ke-1, yaitu 20. Pada baris 87 sampai 89 terdapat fungsi set untuk mengubah elemen pada indeks ke 1 menjadi 99, lalu menampilkan hasil vector yang telah diperbarui yakni [10,99,30] dengan fungsi display. Selanjutnya pada baris 91 sampai 93 terdapat fungsi pop back yang berfungsi untuk menghapus elemen pada indeks terakhir, lalu menampilkan hasilnya dengan fungsi display yaitu [10, 99]. pada baris 95 terdapat fungsi size yang digunakan untuk menampilkan jumlah elemen yang ada saat ini. Selanjutnya pada baris 97 terdapat fungsi clear yang digunakan untuk menghapus seluruh isi vector dan mengembalikan kondisi seperti semula. Terakhir pada baris 98, terdapat kode return 0 yag berfungsi untuk mengembalikan nilai 0 yang menandakan bahwa program selesai dijalankan tanpa error.

5.3.b Output Percobaan 3 : Vector



*Gambar 5.3.b Output Percobaan 3: Vector*

Berdasarkan gambar 5.3.b, Pada baris 1 terdapat cetakan "Isi vector: [10, 20, 30]" yang merupakan output dari fungsi push back yang di deklarasikan pada program baris 26 yang kemudian dipanggil sebanyak 3 kali pada program baris 78 sampai 80. Selanjutnya isi dari vector ditampilkan dengan fungsi display yang ada pada program baris 83 yang sebelumnya telah dideklarasikan pada baris 58 sampai 64. Pada baris 2 terdapat cetakan "Elemen ke-1 = 20" yang merupakan output dari fungsi get yang telah dideklarasikan pada program baris 40 yang kemudian dipanggil untuk menampilkan elemen pada indeks ke 1 yaitu 20. Selanjutnya pada baris 3 terdapat cetakan "Set elemen ke-1 jadi 99: [10, 99, 30]" yang merupakan output dari fungsi set yang telah dideklarasikan pada program baris 48 yang kemudian dipanggil pada baris 87 untuk mengganti nilai elemen pada indeks 1 dari yang semula 20 menjadi 99, lalu hasil set ditampilkan dengan pemanggilan fungsi display. Pada baris 4 terdapat cetakan "Setelah pop\_back: [10, 99]" yang merupakan output dari fungsi pop back yang telah dideklarasikan pada program baris 34 dan kemudian dipanggil pada program baris 91 untuk menghapus elemen pada urutan paling akhir, lalu elemen yang tersisa akan Kembali ditampilkan dengan memanggil fungsi display. Terakhir, pada baris 5 terdapat cetakan "Ukuran vector sekarang = 2" yang merupakan output dari fungsi size yang telah dideklarasikan pada program baris 54 yang kemudian dipanggil pada baris 95 untuk menampilkan length atau jumlah elemen yang ada pada vector.

1. KESIMPULAN

Adapun kesimpulan dari percobaan ini adalah sebagai berikut:

1. Berdasarkan percobaan 5.1.1 Array 1D, dapat disimpulkan bahwa\_\_\_\_\_\_\_\_\_.
2. Berdasarkan percobaan \_\_\_\_\_\_\_\_\_\_\_\_\_, dapat disimpulkan bahwa\_\_\_\_\_\_\_\_\_.

**BUATLAH 1 KESIMPULAN UNTUK SETIAP PERCOBAAN ATAU SOURCE CODE**

**DAFTAR PUSTAKA**

Bagas Pangestu & Favian Rizki. 2025. *Struktur Data dan Algoritma*. Bandung: Informatika Media. (contoh)

Radhitya Agrayasa Rhalin. 2025. *Implementasi Struktur Data dalam C++*. Yogyakarta: Andromeda Press. (contoh)

**TULISKAN MINIMAL 3 DAFTAR PUSTAKA DISINI SESUAI FORMAT.**

**TUGAS AKHIR**

1. Flowchart

2. Source Code

3. Output

4. Link GitHub