**LANJUTAN PRAKTEK**

**21 SAMPAI 29**

**Praktek 22**

A screen shot of a computer program

AI-generated content may be incorrect.

A computer screen shot of text

AI-generated content may be incorrect.

**Hasil ouputnya**

A screenshot of a computer program

AI-generated content may be incorrect.

**Penjelasannya :**

**Baris 1:  
Komentar judul "Praktek 22: Membuat Single Linked List", menunjukkan fokus praktik adalah implementasi struktur data linked list.**

**Baris 3–4:  
Mendefinisikan fungsi buat\_node(data) yang membuat satu node dalam bentuk dictionary dengan kunci 'data' dan 'next'.**

**Baris 6:  
Mendefinisikan fungsi tambah\_node(head, data) untuk menambahkan node baru ke akhir linked list.**

**Baris 7:  
Memanggil fungsi buat\_node untuk membuat node baru.**

**Baris 8–9:  
Jika head bernilai None, maka linked list masih kosong, dan node baru akan langsung menjadi head.**

**Baris 10:  
Jika tidak kosong, proses iterasi dilakukan mulai dari head.**

**Baris 11–13:  
Melakukan perulangan hingga node terakhir ditemukan (yaitu node dengan 'next' bernilai None).**

**Baris 14:  
Node terakhir diarahkan ke node baru sebagai sambungan.**

**Baris 15:  
Mengembalikan head sebagai referensi awal linked list.**

**Baris 17:  
Mendefinisikan fungsi cetak\_linked\_list(head) untuk menampilkan isi linked list.**

**Baris 18:  
Memulai iterasi dari head.**

**Baris 19:  
Mencetak label “Head → ” untuk memulai tampilan.**

**Baris 20–22:  
Selama current tidak None, mencetak data tiap node, lalu berpindah ke node berikutnya.**

**Baris 23:  
Mencetak “NULL” untuk menunjukkan akhir dari linked list.**

**Baris 25:  
Inisialisasi linked list dengan head = None, artinya list masih kosong.**

**Baris 27–29:  
Menambahkan tiga node ke dalam linked list dengan nilai 10, 11, dan 12.**

**Baris 31:  
Mencetak label “Linked List: ” sebagai keterangan hasil akhir.**

**Baris 32:  
Memanggil fungsi cetak\_linked\_list untuk menampilkan seluruh isi linked list.**

**Praktek 23**

**A screen shot of a computer program

AI-generated content may be incorrect.**

**A screen shot of a computer program

AI-generated content may be incorrect.**

**A screen shot of a computer code

AI-generated content may be incorrect.**

**Hasil ouputnya**

A black screen with white text

AI-generated content may be incorrect.

**Penjelasannya :**

**Baris 1:  
Judul praktik ditulis sebagai “Praktek 23: Traversal Linked-List” — membahas penelusuran node dalam struktur data linked list.**

**Baris 2–4:  
Mendefinisikan fungsi buat\_node(data) yang membuat node dengan nilai data dan referensi ke node berikutnya (next) diatur ke None.**

**Baris 6–13:  
Mendefinisikan fungsi tambah\_node(head, data) untuk menambahkan node baru ke akhir linked list.**

* **Baris 7: Membuat node baru dari data.**
* **Baris 8–9: Jika head kosong, node baru jadi head.**
* **Baris 10–12: Jika tidak kosong, lakukan iterasi sampai node terakhir, lalu sambungkan node baru.**

**Baris 15–21:  
Mendefinisikan fungsi traversal\_to\_display(head) untuk menampilkan isi linked list:**

* **Baris 16: Menampilkan "Head →".**
* **Baris 17–20: Iterasi setiap node dan cetak nilainya sampai mencapai NULL.**

**Baris 23–28:  
Mendefinisikan fungsi traversal\_to\_count\_nodes(head) untuk menghitung jumlah node:**

* **Baris 24: Variabel count diset awal 0.**
* **Baris 25–27: Iterasi tiap node dan tambahkan 1 ke count.**
* **Baris 28: Mengembalikan jumlah node.**

**Baris 30–35:  
Mendefinisikan fungsi traversal\_to\_get\_tail(head) untuk mendapatkan nilai node terakhir (tail):**

* **Baris 31: Jika head kosong, kembali None.**
* **Baris 32–34: Iterasi sampai node terakhir ditemukan.**
* **Baris 35: Mengembalikan node terakhir.**

**Baris 37:  
Komentar bahwa bagian ini adalah contoh penerapan fungsi-fungsi sebelumnya.**

**Baris 38:  
Inisialisasi awal head sebagai None.**

**Baris 40–43:  
Menambahkan empat node ke linked list dengan nilai 10, 15, 17, dan 19.**

**Baris 45:  
Mencetak label "Isi Linked-List".**

**Baris 46:  
Memanggil fungsi traversal\_to\_display(head) untuk menampilkan semua node.**

**Baris 48:  
Mencetak jumlah node menggunakan traversal\_to\_count\_nodes(head).**

**Baris 50–51:  
Menampilkan nilai dari HEAD node (node pertama).**

**Baris 53–54:  
Menampilkan nilai dari TAIL node (node terakhir) menggunakan traversal\_to\_get\_tail(head).**

**Praktek 24**

**A screen shot of a computer program

AI-generated content may be incorrect.**

**A screen shot of a computer

AI-generated content may be incorrect.**

**Hasil ouputnya**

**A screen shot of a computer

AI-generated content may be incorrect.**

**Penjelasannya :**

**Baris 1:  
Judul praktik “Praktek 24: Menyisipkan Node di awal Linked-List”.**

**Baris 2–5:  
Mendefinisikan fungsi sisip\_depan(head, data) untuk menyisipkan node baru di awal:**

* **Baris 3: Membuat node baru dengan data, dan next menunjuk ke head (node sebelumnya).**
* **Baris 4: Node baru dikembalikan sebagai head yang baru.**

**Baris 7–13:  
Mendefinisikan fungsi cetak\_linked\_list(head) untuk menampilkan isi linked list:**

* **Baris 8: current diset ke head.**
* **Baris 9: Mencetak label "Head →".**
* **Baris 10–12: Menelusuri node satu per satu dan mencetak nilainya hingga akhir (NULL).**

**Baris 15:  
Komentar bahwa bagian ini adalah pembuatan linked list awal.**

**Baris 16:  
Inisialisasi head dengan None.**

**Baris 17–19:  
Menambahkan tiga node ke awal linked list secara bertahap:**

* **Menyisipkan 30 → 20 → 10**
* **Karena disisipkan di depan, urutan akhir: 10 → 20 → 30**

**Baris 21:  
Mencetak informasi bahwa isi linked list akan ditampilkan sebelum penyisipan baru.**

**Baris 22:  
Menampilkan isi linked list awal.**

**Baris 24:  
Menentukan data 99 yang akan disisipkan di depan.**

**Baris 25:  
Menyisipkan data 99 ke awal linked list.**

**Baris 27:  
Menampilkan informasi nilai yang disisipkan.**

**Baris 29:  
Menampilkan informasi bahwa isi linked list akan ditampilkan setelah penyisipan.**

**Baris 30:  
Menampilkan isi linked list terbaru setelah penambahan node di depan.**

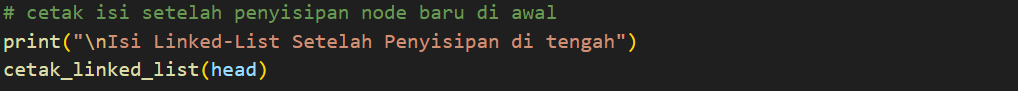
**Praktek 25**

**A screen shot of a computer program

AI-generated content may be incorrect.**

**A screen shot of a computer program

AI-generated content may be incorrect.**

****

**Hasil ouputnya**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**Penjelasannya :**

**Bagian 1 – Fungsi untuk membuat node baru**

**Baris 1:  
Judul praktik “Praktek 25: Menyisipkan node di posisi tertentu”.**

**Baris 2–5:  
Fungsi sisip\_depan(head, data) membuat node baru dan meletakkannya di depan (awalan).**

* **Baris 3: Node baru dibuat dengan data, dan next menunjuk ke head.**
* **Baris 4: Mengembalikan node baru sebagai head.**

**Baris 7–8:  
Fungsi sisip\_tengah(head, data, posisi) untuk menyisipkan node di tengah.**

* **Membuat node baru.**

**Baris 9–10:  
Jika posisi 0, maka gunakan sisip\_depan() karena dianggap sebagai awal.**

**Baris 11–14:  
Traversal untuk menuju ke posisi sebelum lokasi penyisipan.**

* **index digunakan sebagai penghitung langkah.**
* **Melintasi hingga mencapai posisi − 1.**

**Baris 15–16:  
Jika current bernilai None, berarti posisi terlalu jauh atau melebihi panjang linked list → kembalikan head tanpa perubahan.**

**Baris 17–19:  
Sisipkan node baru di antara node saat ini dan node setelahnya.**

* **new\_node['next'] diarahkan ke node setelah current.**
* **current['next'] diarahkan ke new\_node.**

**Baris 21–26:  
Fungsi cetak\_linked\_list(head) untuk menampilkan isi linked list dari awal ke akhir hingga NULL.**

**Baris 28:  
Inisialisasi linked list kosong (head = None).**

**Baris 29–32:  
Menambahkan node di depan: 30, 20, 10 → hasil awal: 10 → 20 → 30.**

**Baris 34–35:  
Menampilkan isi linked list sebelum penyisipan di tengah.**

**Baris 37–39:  
Menyisipkan nilai 99 pada posisi ke-2 (setelah node ke-1, sebelum node ke-2).**

**Baris 41:  
Menampilkan nilai dan posisi yang disisipkan.**

**Baris 43:  
Menampilkan isi linked list setelah penyisipan node di tengah.**

**Praktek 26**

**A screen shot of a computer program

AI-generated content may be incorrect.**

**A screen shot of a computer program

AI-generated content may be incorrect.**

**A black screen with text

AI-generated content may be incorrect.**

**Hasil ouputnya**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Penjelasannya :**

**Baris 1:  
Judul praktik: “Praktek 26: Penghapusan node di awal Linked-List”.**

**Baris 2–5:  
Fungsi sisip\_depan(head, data) untuk membuat node baru dan menyisipkannya di awal.**

* **Baris 3: Node baru dibuat dengan data dan menunjuk ke head.**
* **Baris 4: Mengembalikan node baru.**

**Baris 6–7:  
Fungsi sisip\_diimana(head, data, posisi) untuk menyisipkan node pada posisi tertentu.**

* **Node dibuat dengan menunjuk ke None.**

**Baris 8–10:  
Jika posisi 0, maka langsung gunakan fungsi sisip\_depan().**

**Baris 11–16:  
Traversal untuk mencari posisi sebelum tempat penyisipan.**

* **Gunakan variabel index sebagai penghitung.**
* **Berjalan hingga position - 1.**

**Baris 17–18:  
Jika node tujuan tidak ditemukan (posisi lebih panjang dari panjang linked-list), tampilkan peringatan dan kembalikan head tanpa perubahan.**

**Baris 19–21:  
Proses penyisipan node ke posisi tengah.**

* **Node baru disisipkan setelah node yang ditemukan.**

**Baris 23–28:  
Fungsi hapus\_head(head) untuk menghapus node pertama.**

* **Jika linked-list kosong, tampilkan pesan.**
* **Jika tidak kosong, tampilkan data node yang dihapus dan kembalikan head berikutnya.**

**Bagian 3 – Fungsi menampilkan isi linked-list**

**Baris 30–35:  
Fungsi cetak\_linked\_list(head) untuk menampilkan semua node dalam linked-list.**

* **Dimulai dari head, ditampilkan satu per satu hingga NULL.**

**Baris 37:  
Inisialisasi linked-list kosong.**

**Baris 38–42:  
Menambahkan beberapa node ke dalam linked list menggunakan sisip\_depan().**

* **Hasil akhir: 20 → 30 → 40 → 50 (karena sisip depan).**

**Baris 44–45:  
Menampilkan isi linked-list sebelum penghapusan.**

**Baris 47:  
Menghapus head dengan hapus\_head(head).**

**Baris 49–50:  
Menampilkan isi linked-list setelah node pertama (head) dihapus.**

**Praktek 27**

**A computer screen shot of code

AI-generated content may be incorrect.**

**A screen shot of a computer program

AI-generated content may be incorrect.**

**Hasil ouputnya**

**A black screen with white text

AI-generated content may be incorrect.**

**Penjelasannya :**

**Bagian 1 – Fungsi membuat node baru**

**Baris 2–5:  
Fungsi sisip\_depan(head, data) untuk menyisipkan node baru di depan.**

* **new\_node menyimpan data dan menunjuk ke head.**
* **Node baru dikembalikan.**

**Baris 7–25:  
Fungsi hapus\_tail(head) untuk menghapus node terakhir:**

* **Baris 8–10: Jika linked list kosong (head == None), tampilkan pesan bahwa tidak ada yang bisa dihapus.**
* **Baris 12–14: Jika hanya ada satu node (tail = head), tampilkan pesan bahwa node dihapus dan linked list kosong.**
* **Baris 16–21:  
  Traversal mencari node sebelum tail.**
  + **Saat node berikutnya (current['next']['next']) sudah None, berarti node saat ini adalah sebelum tail.**
  + **Tampilkan data node tail yang dihapus dan setel pointer ke None (menghapus tail).**
* **Baris 22: Kembalikan head yang sudah diperbarui.**

**Baris 27–33:  
Fungsi cetak\_linked\_list(head) untuk mencetak isi linked list:**

* **Mulai dari head, cetak setiap data hingga node terakhir, lalu tampilkan NULL.**

**Baris 35:  
Inisialisasi linked list kosong (head = None).**

**Baris 36–40:  
Menyisipkan node secara berurutan ke depan:**

* **Head akhir: 70 → 60 → 50 → 20 → 30 (30 adalah tail karena dimasukkan pertama).**

**Baris 42–43:  
Menampilkan isi linked list sebelum penghapusan tail.**

**Baris 45:  
Memanggil hapus\_tail(head) untuk menghapus node terakhir (30).**

**Baris 47–48:  
Menampilkan isi linked list setelah tail dihapus.**

**Praktek 28**

**A computer screen shot of a program code

AI-generated content may be incorrect.**

**A screen shot of a computer program

AI-generated content may be incorrect.**

**Hasil ouputnya**

**A screen shot of a computer

AI-generated content may be incorrect.**

**Penjelasannya :**

**Baris 1  
Komentar bahwa ini adalah praktek menghapus node di posisi tengah pada linked list.**

**Baris 2  
Mendefinisikan fungsi sisip\_depanh() untuk menyisipkan elemen di depan linked list.**

**Baris 3  
Membuat node baru berupa dictionary yang berisi data dan next, lalu mengembalikan node tersebut sebagai head baru.**

**Baris 5  
Mendefinisikan fungsi hapus\_tengah() untuk menghapus node dari linked list di posisi tertentu.**

**Baris 6  
Memeriksa apakah head kosong. Jika ya, tidak ada yang bisa dihapus.**

**Baris 7  
Menampilkan pesan bahwa linked list kosong.**

**Baris 8  
Mengembalikan nilai head karena tidak ada perubahan.**

**Baris 10  
Jika position adalah 0, maka node paling depan (head) akan dihapus.**

**Baris 11  
Menampilkan pesan bahwa node pertama dihapus.**

**Baris 12  
Mengembalikan node setelah head sebagai head baru.**

**Baris 14  
Menginisialisasi variabel current dengan head.**

**Baris 15  
Jika node awal kosong (seharusnya redundant karena sudah dicek sebelumnya), tampilkan pesan dan return.**

**Baris 18  
Memeriksa apakah posisi < 0. Jika ya, berarti tidak valid.**

**Baris 19  
Menampilkan pesan bahwa posisi tidak valid.**

**Baris 20  
Mengembalikan head tanpa perubahan.**

**Baris 22  
Memeriksa lagi apakah posisi = 0. (Redundant karena sudah ditangani sebelumnya.)**

**Baris 23  
Menampilkan pesan bahwa node dengan data tertentu dihapus dari posisi 0.**

**Baris 24  
Menghapus node pertama dengan menggeser head ke node berikutnya.**

**Baris 26  
Menginisialisasi pointer current kembali ke head.**

**Baris 27  
Inisialisasi indeks untuk tracking posisi.**

**Baris 29  
Melakukan loop untuk mencari node sebelum posisi target (position - 1).**

**Baris 30  
Memindahkan pointer ke node berikutnya.**

**Baris 31  
Menambah nilai indeks.**

**Baris 33  
Memeriksa apakah posisi melebihi panjang list (yaitu current atau current['next'] kosong).**

**Baris 34  
Jika ya, tampilkan pesan bahwa posisi melebihi panjang linked list.**

**Baris 35  
Kembalikan head karena tidak ada yang dihapus.**

**Baris 37  
Menampilkan pesan bahwa node pada posisi tertentu akan dihapus.**

**Baris 38  
Menghapus node dengan mengubah referensi next dari node sebelumnya.**

**Baris 39  
Mengembalikan head setelah penghapusan.**

**Baris 41  
Mendefinisikan fungsi cetak\_linked\_list() untuk mencetak isi linked list.**

**Baris 42  
Mengatur pointer current ke head list.**

**Baris 43  
Melakukan perulangan untuk mencetak isi node sampai None.**

**Baris 44  
Mencetak data dari node dan panah ke node berikutnya.**

**Baris 45  
Pindah ke node berikutnya.**

**Baris 46  
Mencetak "None" sebagai penutup linked list.**

**Baris 48  
Mulai penerapan: membuat linked list awal.**

**Baris 49–51  
Menyisipkan data 30, 20, dan 10 ke linked list di depan (maka list = 10 → 20 → 30).**

**Baris 53  
Mencetak isi linked list sebelum penghapusan.**

**Baris 54  
Memanggil fungsi cetak\_linked\_list().**

**Baris 55  
Mencetak baris kosong untuk jeda.**

**Baris 57  
Memanggil fungsi hapus\_tengah() untuk menghapus node di posisi ke-1.**

**Baris 59  
Mencetak isi linked list setelah node di posisi tengah dihapus.**

**Baris 60  
Memanggil kembali fungsi cetak\_linked\_list() untuk melihat hasil akhir.**

**Praktek 29**

**A screen shot of a computer program

AI-generated content may be incorrect.**

**A computer screen shot of a program code

AI-generated content may be incorrect.**

**Hasil ouputnya**

**A screen shot of a computer program

AI-generated content may be incorrect.**

**Penjelasannya :**

**Baris 1  
Komentar bahwa ini adalah Praktek 29 untuk membuat double linked list dari depan (awal).**

**Baris 2  
Komentar bahwa akan dibuat node baru.**

**Baris 3  
Mendefinisikan fungsi buat\_node\_double(data) untuk membuat node double linked-list.**

**Baris 4  
Fungsi mengembalikan dictionary berisi data, prev = None, dan next = None.**

**Baris 6  
Komentar bahwa fungsi selanjutnya akan menambahkan node baru di awal double linked list.**

**Baris 7  
Mendefinisikan fungsi tambah\_node\_depan(head, data).**

**Baris 8  
Membuat node baru dengan buat\_node\_double(data).**

**Baris 9  
Menetapkan new\_node['next'] menunjuk ke head (karena ditambahkan di awal).**

**Baris 10  
Menetapkan new\_node['prev'] tetap None (karena jadi node pertama).**

**Baris 11  
Jika head tidak kosong, maka node lama harus menunjuk balik ke new\_node.**

**Baris 12  
Mengatur head['prev'] = new\_node untuk menghubungkan balik.**

**Baris 14  
Mengembalikan node baru sebagai head yang baru.**

**Baris 16  
Komentar bahwa fungsi berikut akan mencetak isi double linked list dengan traversal maju (next).**

**Baris 17  
Mendefinisikan fungsi cetak\_dl(head).**

**Baris 18  
Mencetak “HEAD” sebagai awal pencetakan.**

**Baris 19  
Menginisialisasi pointer current ke head.**

**Baris 20  
Melakukan loop selama current tidak kosong.**

**Baris 21  
Mencetak data node sekarang dan panah ganda (<>).**

**Baris 22  
Memindahkan current ke node berikutnya (next).**

**Baris 23  
Setelah loop selesai, mencetak NULL sebagai akhir dari list.**

**Baris 25  
Komentar bahwa bagian ini adalah penerapan fungsi.**

**Baris 26  
Inisialisasi head menjadi kosong (None).**

**Baris 28  
Menambahkan node bernilai 16 di depan. Sekarang list: 16**

**Baris 29  
Menambahkan node bernilai 19 di depan. Sekarang list: 19 → 16**

**Baris 30  
Komentar bahwa double linked list akan dicetak sebelum penyisipan di awal.**

**Baris 31  
Mencetak judul tampilan.**

**Baris 32  
Memanggil cetak\_dl(head) untuk menampilkan isi list.**

**Baris 34  
Komentar bahwa akan ditambahkan lagi node di depan dengan data 22 dan 99.**

**Baris 35  
Menambahkan node 22 di depan. Sekarang list: 22 → 19 → 16**

**Baris 36  
Menambahkan node 99 di depan. Sekarang list: 99 → 22 → 19 → 16**

**Baris 37  
Mencetak judul list setelah penyisipan awal.**

**Baris 38  
Memanggil kembali cetak\_dl(head) untuk menampilkan isi double linked list yang sudah diperbarui.**