

## Diskusi.1

[Selesai](#)

**Jatuh tempo:** Minggu, 19 Oktober 2025, 23:59

menampilkan balasan dalam bentuk bertingkat

[Setelan ▾](#)

Sudah mencapai batas waktu untuk mengirim ke forum ini sehingga Anda tidak dapat lagi mengirim ke forum ini.

### Diskusi.1

Rabu, 4 Juni 2025, 08:52

Silahkan teman-teman mahasiswa diskusikan pertanyaan dibawah ini :

Linked List terdiri dari node-node yang saling terhubung melalui pointer. Menurut Anda, apa keunggulan utama Linked List dibandingkan dengan Array, dan pada kondisi seperti apa Linked List lebih efisien digunakan?

[Tautan permanen](#)



#### Re: Diskusi.1

oleh [ARYADUTA PRATAMA PUTERA 056027427](#) - Rabu, 15 Oktober 2025, 16:54

menurut saya, keunggulan utama Linked List dibandingkan Array adalah pada fleksibilitasnya dalam menambah atau menghapus data tanpa perlu menggeser elemen lain. Kalau di array, ukuran datanya tetap, jadi kalau mau menambah elemen baru sering harus buat array baru.

Linked list lebih efisien digunakan saat kita sering melakukan insert atau delete data di tengah atau awal struktur, karena cukup ubah pointer antar node saja tanpa memindahkan seluruh data. Jadi cocok digunakan untuk aplikasi yang datanya sering berubah-ubah.

[Tautan permanen](#) [Tampilkan induk](#)



#### Re: Diskusi.1

oleh [F. LIA DWI CAHYANTI 04003643](#) - Jumat, 17 Oktober 2025, 19:42

Terimakasih atas jawabannya . Linked List memang sangat ideal untuk aplikasi dengan frekuensi perubahan data yang tinggi.

[Tautan permanen](#) [Tampilkan induk](#)



#### Re: Diskusi.1

oleh [MUKLIS ARIEF 055224365](#) - Rabu, 15 Oktober 2025, 21:40

Selamat pagi/siang/sore/malam saya izin berdiskusi ibu dan rekan rekan sekalian

Keunggulan utama Linked List dibandingkan Array terletak pada fleksibilitas memorinya. berdasarkan pada materi inisisasi 2 saya akan mengumpamakan Array itu seperti kotak obat harian yang punya sekat-sekat permanen. saya sudah tentukan dari awal hanya ada 7 sekat (untuk 7 hari). Kalau tiba-tiba saya butuh menyimpan obat untuk hari ke-8, saya tidak bisa secara langsung menambahkan sekat baru. Kamu harus membeli kotak obat yang lebih besar dan memindahkan semua obat dari kotak lama ke kotak baru. Jelas ini tidak efisien Proses "menggeser" elemen saat menambah atau menghapus data di tengah-tengah Array juga sama merepotkannya

Linked List itu lebih mirip rangkaian gerbong kereta api. Setiap gerbong (node) tahu gerbong mana yang ada di belakangnya (pointer). Kalau mau menyisipkan gerbong baru di tengah, kamu cukup melepas sambungan antara dua gerbong, lalu menyambungkan gerbong baru di antara keduanya. tidak perlu menggeser semua gerbong yang ada di belakangnya. Jauh lebih praktis begitulah kira kira pengumpamaannya

[Tautan permanen](#) [Tampilkan induk](#)

#### Re: Diskusi.1

oleh [F. LIA DWI CAHYANTI 04003643](#) - Jumat, 17 Oktober 2025, 19:43

terimakasih atas jawabannya . Perumpamaan Anda sangat tepat dan mudah dipahami.

[Tautan permanen](#) [Tampilkan induk](#)

#### Re: Diskusi.1

oleh [053908901 MAHFUL FADILLAH](#) - Kamis, 16 Oktober 2025, 20:29

assalamualaikum wr. wb

izin menjawab diskusi pada kesempatan kali ini

Keunggulan utama Linked List dibandingkan dengan Array adalah pada fleksibilitas alokasi memori dan efisiensi operasi perubahan data.

##### 1. Keunggulan Utama Linked List

unggu karena sifatnya yang dinamis. Tidak seperti Array yang ukurannya harus ditentukan di awal (statis) dan memerlukan blok memori yang berdekatan (contiguous), Linked List hanya mengalokasikan memori seperlunya, satu per satu, saat runtime (ketika node baru ditambahkan). Memori yang digunakannya pun bersifat tersebar (non-contiguous), memungkinkan pemanfaatan memori secara optimal dan menghindari pemborosan ruang (pre-allocation).

Namun, keunggulan terbesar Linked List adalah pada operasi penyisipan (insertion) dan penghapusan (deletion) data. Operasi ini dilakukan dengan sangat efisien, hanya dengan mengubah beberapa pointer, mencapai kompleksitas waktu  $O(1)$  jika posisi node (misalnya awal atau akhir) sudah diketahui. Sebaliknya, pada Array, operasi penyisipan atau penghapusan di tengah atau di awal memerlukan pergeseran semua elemen setelahnya, menjadikannya tidak efisien dengan kompleksitas waktu  $O(n)$ .

##### 2. Kondisi Linked List Lebih Efisien

Linked List menjadi pilihan yang jauh lebih efisien ketika:

- Struktur Data Sering Berubah Ukuran: Saat ukuran data set tidak diketahui di awal atau diperkirakan akan sering bertambah dan berkurang secara signifikan, Linked List adalah solusi ideal karena sifat alokasi memorinya yang dinamis.

- Operasi Penyisipan dan Penghapusan Dominan: Linked List sangat efisien digunakan dalam implementasi struktur data seperti Stack (tumpukan) dan Queue (antrian), serta manajemen memori sistem operasi, di mana elemen sering ditambahkan di satu ujung dan dihapus di ujung lainnya.

- Memori yang Digunakan Terbatas atau Terfragmentasi: Dalam sistem yang memiliki kendala memori atau di mana memori sudah terfragmentasi, Linked List dapat memanfaatkan blok memori kecil yang tersebar, yang tidak mungkin dilakukan oleh Array yang membutuhkan blok besar dan berdekatan.

Namun perlu diingat, Linked List memiliki kelemahan utama dalam akses acak (random access). Untuk mengakses elemen ke-\$i\$, kita harus menelusuri (traverse) dari awal, yang memakan waktu  $\$O(n)$ , berbeda dengan Array yang dapat mengaksesnya secara langsung dalam  $\$O(1)$ .

[Tautan permanen](#) [Tampilkan induk](#)

Hide sidebar

### Re: Diskusi.1

oleh [F. LIA DWI CAHYANTI 04003643](#) - Jumat, 17 Oktober 2025, 19:44

Terima kasih atas jawabannya yang sangat lengkap!

[Tautan permanen](#) [Tampilkan induk](#)

### Re: Diskusi.1

oleh [IQBAL FAHRIANSAH 054872216](#) - Jumat, 17 Oktober 2025, 20:07

Izin Menanggapi Diskusi Sesi 1, menurut saya, Keunggulan utama linked list dibandingkan array terletak pada fleksibilitas dan efisiensi dalam pengelolaan memori. Linked list memiliki ukuran yang fleksibel, sehingga elemen dapat ditambah atau dihapus tanpa harus menentukan ukuran awal seperti pada array yang ukurannya tetap. Operasi tambah atau hapus elemen lebih cepat dan mudah pada linked list karena hanya perlu mengubah pointer, tanpa harus menggeser elemen lain seperti pada array. Linked list hanya menggunakan memori sesuai elemen yang ada, sementara array mengalokasikan memori penuh meski tidak seluruhnya dipakai itulah yang menjadikan linked list lebih efisien saat digunakan.

Referensi Sumber Materi :

- [perbedaan.co.id tentang kelebihan dan kekurangan linked list vs array.](#)
- [Modul MSIM4202 Struktur Data](#)

[Tautan permanen](#) [Tampilkan induk](#)

### Re: Diskusi.1

oleh [F. LIA DWI CAHYANTI 04003643](#) - Minggu, 19 Oktober 2025, 20:44

jawaban Anda menangkap esensi dari perbedaan fundamental kedua struktur data tersebut, menjadikannya masukan yang berkualitas dalam diskusi ini. Kerja bagus! 🌟

[Tautan permanen](#) [Tampilkan induk](#)

### Re: Diskusi.1

oleh [FANDI ALFARIS 053127685](#) - Sabtu, 18 Oktober 2025, 08:19

Assalamu'alaikum wr wb Izin menjawab diskusi

Linked List terdiri dari kumpulan node yang saling terhubung melalui pointer atau referensi.

Keunggulan utamanya dibandingkan Array adalah:

1. Ukuran yang Dinamis

-Linked List dapat bertambah atau berkurang ukurannya secara fleksibel tanpa perlu menentukan ukuran di awal.

-Array memiliki ukuran tetap, sehingga jika kapasitas penuh, perlu membuat array baru dan menyalin data lama.

2. Efisiensi dalam Penyisipan dan Penghapusan Data

-Menambah atau menghapus elemen pada Linked List cukup dengan mengubah pointer antar node, tanpa harus memindahkan elemen lain.

-Sebaliknya, pada Array, proses insert atau delete sering membutuhkan penggeseran elemen, yang memakan waktu lebih lama.

### 3. Pemanfaatan Memori yang Lebih Fleksibel

Linked List mengalokasikan memori sesuai kebutuhan, tidak seperti Array yang harus memesan blok memori besar di awal.

Linked List lebih efisien digunakan pada kondisi berikut:

-Ketika jumlah data tidak diketahui atau sering berubah-ubah.

-Saat operasi penambahan dan penghapusan elemen (terutama di awal atau tengah struktur) sering dilakukan.

-Ketika akses data secara acak tidak terlalu dibutuhkan, karena Linked List hanya bisa diakses secara berurutan (sekuensial).

kesimpulan

Linked List unggul dalam fleksibilitas ukuran dan efisiensi manipulasi data, sedangkan Array unggul dalam kecepatan akses langsung berdasarkan indeks

[Tautan permanen](#) [Tampilkan induk](#)

#### Re: Diskusi.1

oleh [F. LIA DWI CAHYANTI 04003643](#) - Minggu, 19 Oktober 2025, 20:45

Wa'alaikumussalam wr. wb.

Terima kasih atas jawaban diskusi Anda yang sangat lengkap dan terstruktur!

[Tautan permanen](#) [Tampilkan induk](#)

#### Re: Diskusi.1

oleh [AYU MAYA CHARINA KRISNAWANTI PRAYITNO 055448096](#) - Sabtu, 18 Oktober 2025, 19:19

Assalamualaikum, selamat malam ibu mohon izin Ayu Maya Charina KP 055448096 dari Prodi Sistem Informasi. mohon maaf sebelumnya jika ada kesalahan. izin menyampaikan pendapat saya pada diskusi pertama ini.

##### 1. Apa keunggulan Linked List dibandingkan Array?

Linked List pengimplementasian list dalam bahasa pemrograman java yang merupakan rangkaian daftar dari data yang dibaca komputer. Tertuang dalam BMP Struktur Data UT halaman 1.25.

Keunggulannya sangat efisien dalam menghapus elemen dan memiliki ukuran yang dinamis. dalam artian linked list dapat kita tambahkan atau kurangkan ukurannya tanpa harus melihat ukuran disaat awal. dibandingkan dengan array yang ukurannya tetap setelah dibuat. serta ketika menghapus dan menyisipkan elemen. serta tidak akan ada memori yang terbuang dengan percuma. karena linked list hanyamembutuhkan memori untuk data yang tersimpan dan pointernya.

Kapan Linked list lebih efisien digunakan?

ketika kita tidak mengetahui berapa banyak elemen yang akan kita gunakan karena ukurannya akan bertambah sesuai dengan apa yang kita sisipkan.

Terimakasih banyak ibu, mohon maaf apabila ada kesalahan

Sumber refensi :

BMP Struktur data Universitas Terbuka

[Materi Inisiasi 2](#) Struktur Data Universitas Terbuka

<https://bpmbkm.uma.ac.id/2025/06/04/penggunaan-array-dan-linked-list/>

[Tautan permanen](#) [Tampilkan induk](#)

**Re: Diskusi.1**oleh [F. LIA DWI CAHYANTI 04003643](#) - Minggu, 19 Oktober 2025, 20:46

Wa'alaikumussalam wr. wb., Selamat malam.

Terima kasih atas tanggapan diskusi yang jelas dan ringkas ini. Tidak ada kesalahan, pendapat Anda sangat tepat.

[Tautan permanen](#) [Tampilkan induk](#)

Hide sidebars

**Re: Diskusi.1**oleh [CHRISTOFER CHRESENTI WAGOLEBO 053109243](#) - Minggu, 19 Oktober 2025, 12:36

## Keunggulan Utama Linked List dibandingkan Array

**Linked List** adalah struktur data yang terdiri dari **kumpulan node** di mana setiap node berisi *data* dan *pointer* (penunjuk) ke node berikutnya.

Berbeda dengan **Array**, yang menyimpan data secara berurutan di blok memori yang berdekatan, Linked List dapat menyimpan elemen di lokasi memori yang tersebar.

Berikut beberapa **keunggulan utama Linked List dibandingkan Array**:

### 1. Ukuran Dinamis

- Linked List dapat **bertambah atau berkurang panjangnya secara fleksibel** tanpa perlu mengalokasikan ulang memori.
- Sementara Array memiliki ukuran tetap sejak awal dideklarasikan.

- *Contoh:* Jika kita tidak tahu jumlah data yang akan dimasukkan (misalnya input pengguna yang terus berubah), Linked List lebih efisien.

### 2. Kemudahan dalam Menyisipkan dan Menghapus Elemen

- Operasi **insert** atau **delete** pada Linked List cukup mengubah pointer antar node tanpa perlu memindahkan elemen lain.
- Pada Array, setiap kali menambah atau menghapus data di tengah, elemen-elemen setelahnya harus digeser.

- *Contoh:* Pada aplikasi antrian dinamis atau daftar tugas yang sering berubah urutan, Linked List lebih praktis.

### 3. Penggunaan Memori yang Efisien untuk Data Tidak Tetap

- Linked List hanya menggunakan memori sesuai jumlah node yang ada, sedangkan Array sering kali memesan ruang lebih besar dari yang digunakan.
- Ini membuat Linked List lebih hemat memori pada data berukuran tidak menentu.

## Kapan Linked List Lebih Efisien Digunakan

Linked List lebih efisien pada kondisi berikut:

1. Saat jumlah data **tidak diketahui sebelumnya** atau sering berubah.
2. Ketika operasi **penyisipan dan penghapusan** data dilakukan **sering di posisi acak** (bukan hanya di akhir).
3. Saat tidak memerlukan akses langsung (random access) ke elemen tertentu, karena Linked List harus dilalui secara berurutan.
4. Dalam implementasi **struktur data dinamis** seperti *stack*, *queue*, dan *graph adjacency list*.

## Namun, kekurangan Linked List:

- Tidak mendukung akses langsung ke elemen (harus traversal).
- Menggunakan memori lebih besar karena setiap node butuh pointer tambahan.
- Tidak efisien untuk pencarian cepat seperti pada Array.

## Kesimpulan

- **Keunggulan utama Linked List** terletak pada **fleksibilitas ukuran dan efisiensi dalam operasi sisip/hapus data**. Linked List lebih efisien digunakan ketika data bersifat dinamis dan sering mengalami perubahan, seperti pada daftar antrian, playlist musik, atau manajemen memori dinamis di sistem operasi.

[Tautan permanen](#) [Tampilkan induk](#)

### Re: Diskusi.1

oleh [F. LIA DWI CAHYANTI 04003643](#) - Minggu, 19 Oktober 2025, 20:46

Secara keseluruhan, pemahaman Anda mengenai kedua struktur data fundamental ini dan trade-off masing-masing sudah sangat mendalam. 🌟

[Tautan permanen](#) [Tampilkan induk](#)

### Re: Diskusi.1

oleh [MOCHAMAD HUSIN DEKI SANTOSO 044987848](#) - Minggu, 19 Oktober 2025, 20:11

selamat malam, berikut tanggapan saya mengenai diskusi 1 ini.

menurut saya linked list lebih fleksibel digunakan dibandingkan array karena data yang diperlukan tidak harus tetap dan node dapat di tambah atau dihapus kapan pun sedangkan array data harus tetap dan jika ada penambahan data harus disesuaikan agar bisa posisinya. walapun cara pemakaiannya berbeda masing masing ada keunggulan dalam pemakaian untuk beberapa data.

contoh

linked list lebih cocok untuk data yang sering berubah ubah jumlahnya, tidak perlu akses secara langsung seperti menghitung pelanggan di mall yang masuk/keluar, sistem undo/redo

array lebih cocok untuk data yang tetap yang sudah diketahui sejak awal dan sering di akses seperti daftar sales bulanan di suatu toko

Terima kasih

sumber:

Ginting, Subhan Hafiz Nanda, et al. "Pengantar struktur data." Penerbit Mifandi Digital 1.01 (2024).

[Tautan permanen](#) [Tampilkan induk](#)

### Re: Diskusi.1

oleh [F. LIA DWI CAHYANTI 04003643](#) - Minggu, 19 Oktober 2025, 20:47

Terima kasih atas tanggapan Anda yang padat dan informatif mengenai perbandingan Linked List dan Array.

[Tautan permanen](#) [Tampilkan induk](#)

### Re: Diskusi.1

oleh [M. DIMAS PRAYOGA 055097753](#) - Minggu, 19 Oktober 2025, 20:39

Menurut Pendapat saya, keunggulan utama dari Linked List dibandingkan dengan Array terletak pada sifatnya yang dinamis. Linked List tidak memerlukan ukuran tetap seperti Array, sehingga penambahan atau juga penghapusan elemen dapat dilakukan dengan mudah tanpa harus mengalokasikan ulang seluruh memori.

Selain itu, operasi seperti penyisipan (insert) dan penghapusan (delete) pada Linked List lebih efisien karena hanya perlu mengubah arah pointer, sedangkan pada Array, elemen lain harus digeser agar tetap berurutan.

Linked List lebih efisien digunakan pada kondisi di mana ukuran data sering berubah atau ketika operasi penambahan dan penghapusan elemen dilakukan secara intensif. Contohnya dapat ditemukan pada struktur antrian printer, playlist lagu, atau fitur undo/redo pada aplikasi yang memerlukan perubahan data secara dinamis.

[Tautan permanen](#) [Tampilkan induk](#)

### Re: Diskusi.1

oleh [F. LIA DWI CAHYANTI 04003643](#) - Minggu, 19 Oktober 2025, 20:47

Tanggapan Anda mengenai keunggulan Linked List dibandingkan Array sangat akurat, ringkas, dan tepat sasaran!



[Tautan permanen](#) [Tampilkan induk](#)

### Re: Diskusi.1

oleh [052646246 RANGGA HARMAWAN](#) - Minggu, 19 Oktober 2025, 22:53

#### Keuntungan Utama Linked List

Keunggulan utama Linked List dibandingkan Array adalah efisiensi dalam operasi penyisipan (insertion) dan penghapusan (deletion) data, serta fleksibilitas alokasi memori.

##### 1. Efisiensi Penyisipan/Penghapusan (Insertion/Deletion) .

a. Pada Array, jika kita ingin menyisipkan atau menghapus elemen di awal atau tengah, kita harus menggeser (shifting) semua elemen lain setelahnya.

b.Pada Linked List, kita hanya perlu mengubah beberapa pointer untuk menyambungkan atau melepas sebuah node. Operasi ini sangat cepat (kompleksitas O(1)), dengan asumsi kita sudah berada di lokasi node yang diinginkan.

##### 2. Alokasi Memori Dinamis (Dynamic Memory Allocation)

a. Array (statis) memiliki ukuran tetap yang ditentukan saat kompilasi. Jika penuh, kita tidak bisa menambah data.

b. Dynamic Array (seperti Vector di Java/C++ atau list di Python) bisa tumbuh, tetapi proses "tumbuh" (resizing) ini mahal: sistem harus mengalokasikan blok memori baru yang lebih besar dan menyalin semua elemen lama ke blok baru tersebut.

c. Linked List tumbuh dan menyusut secara alami. Setiap node baru dialokasikan di memori di mana saja dan disambungkan. Tidak ada operasi copy-paste massal, sehingga tidak ada memori yang terbuang percuma (selain overhead untuk menyimpan pointer itu sendiri).

Berdasarkan keunggulan di atas, Linked List lebih efisien digunakan pada kondisi berikut:

1.Ketika ukuran data tidak diketahui atau sangat fluktuatif. Jika kita tidak tahu berapa banyak data yang akan disimpan, atau jika jumlah data akan sering bertambah dan berkurang secara drastis, Linked List adalah pilihan yang

lebih baik. Kita tidak perlu menebak ukuran awal (seperti pada Array) dan tidak akan terkena penalti performa akibat resizing.

2. Ketika sering melakukan operasi penyisipan dan penghapusan. Jika aplikasi lebih banyak melakukan insert dan delete data (terutama di awal atau tengah koleksi) daripada membaca data, Linked List akan jauh lebih cepat.

- Contoh bagus: Implementasi Queue (antrean) atau Stack (tumpukan), di mana penambahan/penghapusan sering terjadi di ujung (kepala/ekor).

3. Ketika akses acak (Random Access) tidak diperlukan. Ini adalah kelemahan utama Linked List. Kita tidak bisa langsung melompat ke elemen ke-500 (indeks [500]) seperti pada Array ( $O(1)$ ). kita harus menelusuri (traversing) dari head node satu per satu.

- Jadi, jika kita tidak butuh akses acak dan lebih sering memproses data secara berurutan (sequential), maka Linked List efisien digunakan.

Singkatnya:

- Gunakan Array jika kita butuh akses data (lookup) yang cepat berdasarkan indeks dan ukurannya relatif stabil.
- Gunakan Linked List jika kita butuh modifikasi data (insert/delete) yang cepat dan fleksibilitas ukuran memori.

[Tautan permanen](#) [Tampilkan induk](#)



### Re: Diskusi.1

oleh [F. LIA DWI CAHYANTI 04003643](#) - Senin, 20 Oktober 2025, 15:04

Terima kasih atas penjelasan yang sangat lengkap dan terstruktur mengenai keunggulan utama Linked List dibandingkan Array.

[Tautan permanen](#) [Tampilkan induk](#)

◀ Materi Inisiasi 2

Lompat ke...

Materi Inisiasi 3 ►

## Navigasi

### ▼ Dasbor

[Beranda situs](#)

> [Laman situs](#)

### ▼ Kelasku

> [STSI4203.108](#)

> [STSI4202.42](#)

> [STSI4103.119](#)

> [MKKI4201.278](#)

> [STSI4201.161](#)

> [STSI4205.331](#)

> [STSI4104.284](#)

> [Peserta](#)

■■■ [Nilai](#)

Hide sidebars

> [AKTIVITAS BELAJAR 1](#)✓ [AKTIVITAS BELAJAR 2](#) [Materi Inisiasi 2](#) [Diskusi.1](#)> [AKTIVITAS BELAJAR 3](#)> [AKTIVITAS BELAJAR 4](#)> [AKTIVITAS BELAJAR 5](#)> [MKDI4202.1514](#)> [Kelas](#) **Administrasi**

✓ Forum administrasi

Berlangganan dinonaktifkan

Follow Us:      

UNIVERSITAS TERBUKA ©2025

Anda masuk sebagai [INDRAWAN LISANTO 053724113](#) ([Keluar](#))[Dapatkan aplikasi seluler](#)