

# PROGRAM KASIR

Kelompok C

# Pointer dan Array

- **Pointer:** Pointer (variabel penunjuk) adalah suatu variabel yang berisi alamat memori dari suatu variabel lain. Alamat ini merupakan lokasi dari obyek lain (biasanya variabel lain) di dalam memori. Contoh, jika sebuah variabel berisi alamat dari variabel lain, variabel pertama dikatakan menunjuk ke variabel kedua.
- **Array:** sekumpulan variable yang bertipe data sama yang dibedakan oleh indeks. Suatu Array mempunyai jumlah komponen yang banyaknya tetap. Banyaknya komponen dalam suatu larik ditunjukkan oleh suatu indek untuk membedakan variabel yang satu dengan variabel yang lainnya. Dalam bahasa C, index dari array dimulai dengan 0 (zero).

# W/R File

Operasi file terbagi menjadi 3 jenis. Read(R), Write(W), dan Append(A).

Struktur Dasarnya adalah:

`FILE *in=fopen(namafile,mode);` Ini adalah Langkah pertama yang wajib untuk membuka file dengan perintah `fopen` diikuti dengan 2 variabel diantaranya nama file dan mode.

Setelah dibuka, file tersebut wajib di close dengan perintah `fclose(namavariabel)`.

# LINKED LIST

---

- Linked List adalah suatu struktur data linier. Berbeda dengan array yang juga merupakan struktur data linier dan tipe data komposit, linked list dibentuk secara dinamik. Pada saat awal program dijalankan elemen linked list belum data. Elemen linked list (disebut node) dibentuk sambil jalan sesuai instruksi. Apabila setiap elemen array dapat diakses secara langsung dengan menggunakan indeks, sebuah node linked list diakses dengan menggunakan pointer yang mengacu (menunjuk) ke node tersebut. Awal atau kepala linked list harus diacu sebuah pointer yang biasa diberi nama head. Pointer current (disingkat curr) digunakan untuk memindahkan pengacuan kepada node tertentu.



# KEUNGGULAN LINKED LIST

---


- Fleksibilitas, dapat memasukkan atau menghapus dari posisi mana saja dalam waktu yang konstan.
- Alkasi memori dinamis, tidak perlu mengalokasikan memori.

Tetapi kekurangannya, waktu yang dibutuhkan untuk mengakses elemen tidak konstan, hal ini karena tidak memasukkan rumus perhitungan yang diterapkan untuk array untuk menghitung alamat memori, jadi linked list relative tidak efisien dibandingkan dengan array.



# 5 FUNCTIONS

- Account Log in
- Main Men
- Item Sell
- Item Add
- Item Add New
- Item Add Existed
- Item View All
- Item History
- Account View Al
- Account Ad
- Account Delete
- Swap Account
- Account Sort

- 
- Read File
  - Write File
  - Screen Clear
  - Screen Pause
  - Item Insert
  - Item Delete

# STACK

---

- Stack merupakan jenis linked list yang menerapkan konsep LIFO (Last In First Out) artinya elemen dari struktur (node) yang dimasukkan pertama kali ke dalam rangkaian akan dikeluarkan (diproses) terakhir kali sedangkan elemen yang masuk terakhir akan diproses / dikeluarkan pertama. Dengan kata lain setiap transaksi atau aktifitas yang ada dalam stack dilakukan pada kepalanya (head) saja. Pada stack hanya ada 2 fungsi utama yaitu Push atau menambah/memasukkan node (struktur) ke dalam stack, dan Pop atau mengeluarkan/menghapus node dari stack.



# Keunggulan stack

- ▶ Implementasi algoritma parsing, evaluasi dan backtracking.
- ▶ Digunakan OS untuk memungkinkan pemanggilan prosedur secara nested.
- ▶ Untuk mendukung compiler mengkonversi infix menjadi postfix dan kemudian mengevaluasi postfix menjadi assembly command.

# AVL Tree

- ▶ AVL Tree adalah Binary Search Tree yang memiliki perbedaan tinggi maksimal 1 antara sub tree kiri dan sub tree kanan. AVL Tree muncul untuk menyeimbangkan Binary Search Tree maka waktu pencarian dan bentuk tree dapat dipersingkat dan disederhanakan.
- ▶ Untuk menjaga tree tetap imbang, setelah penyisihan sebuah node, dilakukan pemeriksaan dari node baru  $\rightarrow$  root. Node pertama yang memiliki (balance factor)  $> 1$  diseimbangkan. Proses penyeimbangan dilakukan dengan Single Rotation dan Double Rotation.

# Keunggulan AVL Tree

- Untuk mengoptimasi pencarian data terutama untuk kasus tree yang condong ke kiri atau ke kanan sehingga pencarian akan jauh lebih mudah apabila pohon tersebut seimbang. Kasus tree yang condong ke kiri atau ke kanan itu mungkin saja terjadi terutama apabila penambahan elemen dan penghapusan elemen dilakukan terus menerus dan tidak dapat diketahui urutannya.

# Linear Search

- Linear search atau sequential search merupakan sebuah algoritma untuk pencarian sebuah data dari himpunan data. Cara kerja dari algoritma ini adalah data yang ada dibandingkan satu per satu secara berurutan dengan yang dicari sampai data tersebut ditemukan atau tidak ditemukan.

# KEUNGULAN LINEAR SEARCH

- Dapat di implementasikan di Unsorted List (Data yang tidak teratur) Jika Target ada di awal pencarian, maka prosesnya sangat singkat.



# QUICK SORT

- Quick Sort merupakan suatu algoritma pengurutan data yang menggunakan Teknik pemecahan data menjadi partisi-partisi, sehingga metode ini disebut juga dengan nama partition exchange sort. Untuk memulai interaksi pengurutan, pertama-tama sebuah elemen dipilih dari data, kemudian elemen-elemen data akan diurutkan diatur sedemikian rupa.
- Algoritma ini mengambil salah satu elemen secara acak yang disebut dengan pivot lalu menyimpan semua elemen yang lebih kecil di sebelah kiri pivot dan semua elemen yang lebih besar di sebelah kanan pivot. Hal ini dilakukan secara rekursif terhadap elemen di sebelah kiri dan kanannya sampai semua elemen sudah terurut.

# ALGORITMA QUICKSORT

- Pilih 1 elemen secara acak secara pivot
- Pindahkan semua elemen yang lebih kecil ke sebelah kiri pivot dan semua elemen yang lebih besar ke sebelah kanan pivot. Elemen yang nilainya sama bisa disisipkan di salah satunya.
- Lakukan sort secara rekursif terhadap sub array sebelah kiri dan kanan pivot.

# Keunggulan Quick sort

- ♦ Algoritmanya sederhana dan mudah diterapkan pada berbagai bahasa pemrograman secara efisien.
- ♦ Lebih cepat dari berbagai algoritma pengurutan dengan perbandingan seperti merge sort dan heap sort