## Penjelasan Skrip Todo App Sederhana (todoapp.sh)

Skrip ini adalah program kecil yang bisa dijalankan dari terminal (command line) untuk membantu Anda mencatat dan mengelola daftar pekerjaan (todo list). Cara kerjanya cukup sederhana dan semua data disimpan dalam sebuah file teks biasa bernama todo.txt.

### Cara Kerja Umum

1. **Penyimpanan Data:** Setiap kali Anda menambahkan pekerjaan baru, skrip akan menuliskannya ke dalam file todo.txt. Jika file ini belum ada, skrip akan membuatnya secara otomatis saat pertama kali dijalankan.
2. **Perintah:** Anda berinteraksi dengan skrip ini dengan memberikan perintah saat menjalankannya dari terminal. Misalnya, untuk menambahkan pekerjaan, Anda akan mengetik sesuatu seperti ./todoapp.sh add Belajar Bash.

### Bagian-bagian Skrip

Mari kita bedah fungsi utama dari skrip ini:

1. **Inisialisasi (TODO\_DB="todo.txt" dan touch "$TODO\_DB")**
   * Bagian ini menentukan nama file tempat menyimpan data todo, yaitu todo.txt.
   * Perintah touch memastikan file todo.txt ada. Kalau belum ada, file kosong akan dibuat.
2. **Fungsi show\_help()**
   * Fungsi ini bertugas menampilkan pesan bantuan. Pesan ini menjelaskan cara menggunakan skrip dan daftar perintah yang tersedia, seperti print, add, remove, finish (atau done), dan help.
3. **Fungsi print()**
   * Fungsi ini digunakan untuk menampilkan semua daftar pekerjaan yang tersimpan di todo.txt.
   * Jika file todo.txt kosong, ia akan memberi tahu bahwa “Todo list is empty”.
   * Jika ada isinya, ia akan menampilkan daftar pekerjaan tersebut, lengkap dengan nomor urut di depannya (misal: “1. [ ] Beli susu”).
4. **Fungsi add\_todo()**
   * Fungsi ini untuk menambahkan pekerjaan baru ke dalam daftar.
   * Anda perlu memberikan teks pekerjaan setelah perintah add (contoh: ./todoapp.sh add Cuci piring).
   * Jika Anda lupa memberikan teks pekerjaan, skrip akan mengingatkan “Please insert the todo text”.
   * Pekerjaan baru akan ditambahkan ke baris paling bawah di file todo.txt dengan format [ ] Teks Pekerjaan Anda. Tanda [ ] di depan menandakan pekerjaan ini belum selesai.
   * Setelah berhasil menambahkan, skrip akan memberi konfirmasi.
5. **Fungsi mark\_done()**
   * Fungsi ini untuk menandai pekerjaan yang sudah selesai.
   * Anda perlu memberikan nomor urut (ID) pekerjaan yang ingin ditandai selesai setelah perintah done (contoh: ./todoapp.sh done 1). *Catatan: Di pesan bantuan tertulis finish, tapi di kode menggunakan done.*
   * Skrip akan mencari baris sesuai nomor ID yang diberikan, lalu mengubah tanda [ ] di awal baris menjadi [x].
   * Jika nomor ID tidak valid (bukan angka), skrip akan meminta Anda memasukkan ID yang benar.
   * Setelah berhasil menandai, skrip akan memberi konfirmasi.
6. **Fungsi remove\_todo()**
   * Fungsi ini untuk menghapus pekerjaan dari daftar.
   * Anda perlu memberikan nomor urut (ID) pekerjaan yang ingin dihapus setelah perintah remove (contoh: ./todoapp.sh remove 2).
   * Skrip akan mencari baris sesuai nomor ID dan menghapusnya dari file todo.txt.
   * Jika nomor ID tidak valid, skrip akan meminta ID yang benar.
   * Setelah berhasil menghapus, skrip akan memberi konfirmasi.
7. **Logika Utama (case "$1" in ... esac)**
   * Ini adalah bagian utama yang memproses perintah yang Anda berikan saat menjalankan skrip.
   * Skrip akan memeriksa perintah pertama yang Anda ketik (misalnya print, add, done, remove, atau help).
   * Berdasarkan perintah tersebut, skrip akan menjalankan fungsi yang sesuai (misalnya, jika perintahnya add, maka fungsi add\_todo() akan dijalankan).
   * Jika perintah tidak dikenali atau Anda mengetik help, maka fungsi show\_help() akan dijalankan untuk menampilkan bantuan.

### Kesimpulan

Skrip todoapp.sh ini adalah contoh sederhana bagaimana skrip Bash bisa digunakan untuk membuat alat bantu kecil yang berguna. Ia memanfaatkan perintah-perintah dasar Linux/Unix seperti touch, echo, nl, dan sed untuk mengelola daftar pekerjaan dalam file teks biasa. Meskipun sederhana, ini menunjukkan dasar-dasar pemrograman skrip shell, termasuk penggunaan variabel, fungsi, kondisional (if), dan pemrosesan argumen baris perintah (case).