



BOOTCAMP BACK-END PROGRAMMING

Go Languages - Phase 0

Full Time Back End Programming

Kisi-Kisi Livecode 1 Phase 1



Program



***Hint* :**

Buat sebuah Program dengan CLI yang nantinya dalam Program ini akan punya beberapa release :

Release 1 :

1. Bisa menampilkan data statik kedalam CLI
2. User Experience : User bisa memilih data statik yang tampil kemudian menentukan action selanjutnya.
Sebagai gambaran. Bisa pahami konsep Add to Cart dari sisi User Experiences.

Release 2 :

3. Program CLI Melakukan Perhitungan dengan beberapa penerapan logika pemogramman (Kondisi / Perulangan)

References

***Hint* :**

1. Pelajari Terkait Data Structures & Data Storage : [Struct](#), [Slices](#)
2. Pelajari Terkait I/O : [Variable](#), [Const](#)
3. Pelajari Logic Programming : [Conditional](#) , [Looping](#), [Conditional2](#)
4. Pelajari Function : [Function](#), [Package](#), [Interface](#)



Case : Order Management System

***Hint* :**

Contoh Program Membuat Order Book Management System

```
Main Menu:
1. Show available books
2. Rent a book
3. Exit program

Please enter your choice (1/2/3): 1

Available Books:
1. Title: The Great Gatsby, Author: F. Scott Fitzgerald, Available: Yes
2. Title: To Kill a Mockingbird, Author: Harper Lee, Available: Yes
3. Title: 1984, Author: George Orwell, Available: No

Please enter your choice (1/2/3): 2

Enter the title of the book you want to rent: To Kill a Mockingbird

Book "To Kill a Mockingbird" has been successfully rented!

Do you want to rent another book? (yes/no): yes

Enter the title of the book you want to rent: 1984

Book "1984" is not available for rent.

Do you want to rent another book? (yes/no): no

Thank you for using the Library Book Rental System. Goodbye!
```

Case : Order Management System

***Hint* :**

Contoh Program Membuat Order Book Management System

```
// Book represents the details of a book
type Book struct {
    Title    string
    Author   string
    Available bool
}
```

```
// Initialize the list of books (you can add more books here)
var books = []Book{
    {Title: "The Great Gatsby", Author: "F. Scott Fitzgerald", Available: true},
    {Title: "To Kill a Mockingbird", Author: "Harper Lee", Available: true},
    {Title: "1984", Author: "George Orwell", Available: false},
}
```

```
func main() {
    fmt.Println("Welcome to the Library Book Rental System!\n")

    for {
        fmt.Println("Main Menu:")
        fmt.Println("1. Show available books")
        fmt.Println("2. Rent a book")
        fmt.Println("3. Exit program")

        var choice int
        fmt.Print("\nEnter your choice (1/2/3): ")
        fmt.Scanln(&choice)

        switch choice {
            case 1:
                showAvailableBooks()
            case 2:
                rentBook()
            case 3:
                fmt.Println("Thank you for using the Library Book Rental System")
                return
            default:
                fmt.Println("Invalid choice. Please try again.\n")
        }
    }
}
```

```
func showAvailableBooks(books []Book) {
    fmt.Println("\nAvailable Books:")

    for i, book := range books {
        if book.Available {
            fmt.Printf("%d. Title: %s, Author: %s, Available: Yes\n", i+1, book.Title, book.Author)
        } else {
            fmt.Printf("%d. Title: %s, Author: %s, Available: No\n", i+1, book.Title, book.Author)
        }
    }

    fmt.Println()
}

func rentBook(books []Book) {
    fmt.Print("\nEnter the title of the book you want to rent: ")
    var title string
    fmt.Scanln(&title)

    for i, book := range books {
        if book.Title == title {
            if book.Available {
                books[i].Available = false
                fmt.Printf("\nBook \"%s\" has been successfully rented!\n", title)
            } else {
                fmt.Printf("\nBook \"%s\" is not available for rent.\n", title)
            }
            return
        }
    }

    fmt.Printf("\nBook \"%s\" not found.\n", title)
}
```

REFERENCES

<https://golangbot.com/learn-golang-series/>

<https://www.w3schools.com/go/>

<https://www.programiz.com/golang/struct>