

**ALGORITMA PEMROGRAMAN 2**  
**KELOMPOK 6: Aplikasi Dealer Mobil**



**Disusun Oleh:**

Arvel Mahsa Athallah Firdaus (2311102188)  
Tegar Bangkit Wijaya (2311102027)  
Avriel Fitria Rachma Suryantari (2311102036)  
Muhammad Djidzan Naufal Nugraha (2311102189)

**Dosen Pengampu :**

Wahyu Andi Saputra, S.Pd., M.Eng

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**  
**FAKULTAS INFORMATIKA**  
**UNIVERSITAS TELKOM PURWOKERTO**  
**2024**

**Arvel Mahsa Athallah Firdaus** - Tambah, Edit, Hapus Data Pabrik, Hapus mobil yang diproduksi suatu pabrik

```
package main

import "fmt"

type Pabrik struct {
    ID          int
    Nama         string
    Wilayah     string
    HitungMobil int
    TotalSales  int
}

type Mobil struct {
    ID_mobil int
    NamaMobil string
    IDpabrik int
    Tahun int
    HargaMobil float64
}

const maxCars = 100
var mobilList [maxCars]Mobil

const maxPabrik = 100 // Maksimum jumlah pabrik
var pabrikList [maxPabrik]Pabrik
var pabrikCount int = 0 // Jumlah pabrik saat ini

var mobilCount int = 0
var nextCarID int = 1

// Fungsi Untuk Menambahkan Pabrik
func tambahPabrik(nama, negara string) {
    if pabrikCount >= maxPabrik {
        fmt.Println("Tidak dapat menambahkan pabrik baru. Kapasitas penuh!")
        return
    }
    Manufaktur := Pabrik{
        ID: pabrikCount + 1,
        Nama: nama,
        Wilayah: negara,
        HitungMobil: 0,
    }
}
```

```

        pabrikList[pabrikCount] = Manufaktur
        pabrikCount++
        fmt.Println("Pabrik berhasil ditambahkan!")
    }

// Fungsi untuk mengubah data pabrik
func ubahPabrik(id int, nama, negara string) {
    for i := 0; i < pabrikCount; i++ {
        if pabrikList[i].ID == id {
            pabrikList[i].Nama = nama
            pabrikList[i].Wilayah = negara
            fmt.Println("Pabrik berhasil diperbarui!")
            return
        }
    }
    fmt.Println("Pabrik tidak ditemukan!")
}

// Fungsi menghapus pabrik dan mobilnya.
func hapusPabrik(id int) {
    // Hapus pabrik
    for i := 0; i < pabrikCount; i++ {
        if pabrikList[i].ID == id {
            // Geser elemen-elemen setelah yang dihapus
            for j := i; j < pabrikCount-1; j++ {
                pabrikList[j] = pabrikList[j+1]
            }
            pabrikCount-- // Kurangi jumlah elemen
            fmt.Println("Pabrik ditemukan dan dihapus.")
            break
        }
    }

    // Hapus mobil terkait
    for i := 0; i < mobilCount; {
        if mobilList[i].IDpabrik == id {
            // Geser elemen-elemen setelah yang dihapus
            for j := i; j < mobilCount-1; j++ {
                mobilList[j] = mobilList[j+1]
            }
            mobilCount-- // Kurangi jumlah elemen
            fmt.Println("Mobil terkait berhasil dihapus.")
        } else {
            i++ // Hanya lanjut jika elemen tidak dihapus
        }
    }
}

```

```

// Fungsi Untuk Menambahkan Mobil Ke Pabrik
func tambahMobil(namaMobil string, idPabrik, tahun int,
harga float64) {
    // Cari pabrik berdasarkan ID
    for i := 0; i < pabrikCount; i++ {
        if pabrikList[i].ID == idPabrik {
            if mobilCount >= maxCars {
                fmt.Println("Tidak dapat menambahkan
mobil baru. Kapasitas penuh!")
                return
            }

            // Tambahkan mobil ke array
            mobilList[mobilCount] = Mobil{
                ID_mobil: nextCarID,
                NamaMobil: namaMobil,
                IDpabrik: idPabrik,
                Tahun: tahun,
                HargaMobil: harga,
            }
            mobilCount++
            pabrikList[i].HitungMobil++
            nextCarID++
            fmt.Println("Mobil berhasil ditambahkan!")
            return
        }
    }
    fmt.Println("Pabrik tidak ditemukan!")
}

// ubahMobil mengedit data mobil yang ada.
func ubahMobil(id int, nama string, idPabrik, tahun int,
harga float64) {
    for i := 0; i < mobilCount; i++ {
        if mobilList[i].ID_mobil == id {
            // Perbarui data mobil
            mobilList[i].NamaMobil = nama
            mobilList[i].IDpabrik = idPabrik
            mobilList[i].Tahun = tahun
            mobilList[i].HargaMobil = harga
            fmt.Println("Mobil berhasil diperbarui!")
            return
        }
    }
    fmt.Println("Mobil tidak ditemukan!")
}

```

```

}

// Fungsi Untuk Menampilkan Data Mobil
func tampilkanMobil() {
    for i := 0; i < mobilCount; i++ {
        fmt.Printf("\nID Mobil : %d Nama: %s ID Pabrikan:
%d\nTahun: %d\nHarga: %.2f\n",
            mobilList[i].ID_mobil, mobilList[i].NamaMobil,
mobilList[i].IDpabrikan, mobilList[i].Tahun,
mobilList[i].HargaMobil)
        fmt.Println()
    }
}

// Fungsi Menghapus Data Mobil
func hapusMobil(id int) {
    for i := 0; i < mobilCount; i++ {
        if mobilList[i].IDpabrikan == id {
            // Geser elemen-elemen setelah elemen yang
dihapus
            for j := i; j < mobilCount-1; j++ {
                mobilList[j] = mobilList[j+1]
            }
            mobilCount-- // Kurangi jumlah mobil
            fmt.Println("Mobil berhasil dihapus!")
            return
        }
    }
    fmt.Println("Mobil tidak ditemukan!")
}

// Fungsi Untuk Menampilkan Pabrikan
func tampilkanPabrikan() {
    for i := 0; i < pabrikanCount; i++ {
        fmt.Printf("ID: %d\n Nama: %s\n Wilayah: %s\n
Hitung Mobil: %d\n",
            pabrikanList[i].ID, pabrikanList[i].Nama,
pabrikanList[i].Wilayah, pabrikanList[i].HitungMobil)
        fmt.Println()
    }
}

func main() {
    for {
        fmt.Println("1.Tambah Pabrikan")
        fmt.Println("2.Ubah Data Pabrikan")
    }
}

```

```

    fmt.Println("3.Tampilkan Pabrikan")
    fmt.Println("4.Hapus Pabrikan")
    fmt.Println("5.Tambah Mobil")
    fmt.Println("6.Ubah Data Mobil")
    fmt.Println("7.Tampilkan Mobil")
    fmt.Println("8.Hapus Mobil")
    fmt.Print("Pilih Nomor : ")
    var number int
    fmt.Scan(&number)
    switch number {
    case 1 :
        var nama, negara string
        fmt.Print("\nInput nama Pabrikan : ")
        fmt.Scan(&nama)
        fmt.Print("Input Wilayah Pabrikan : ")
        fmt.Scan(&negara)
        tambahPabrik(nama, negara)
        fmt.Println()

    case 2 :
        fmt.Println()
        var id int
        var nama, negara string
        fmt.Print("Masukan ID Pabrikan Yang Akan Di ubah : ")

        fmt.Scan(&id)
        fmt.Print("Nama Pabrikan      : ")
        fmt.Scan(&nama)
        fmt.Print("Wilayah Pabrikan : ")
        fmt.Scan(&negara)
        ubahPabrikan(id, nama, negara)
        fmt.Println()

    case 3 :
        fmt.Println()
        tampilkanPabrikan()

    case 4 :
        fmt.Println()
        var id int
        fmt.Print("Input ID Pabrikan Yang Akan Dihapus : ")
        fmt.Scan(&id)
        hapusPabrikan(id)
        fmt.Println()

    case 5 :
        var namaMobil string

```

```

        var idPabrikan, tahun int
        var harga_mobil float64
        fmt.Print("\nNama Mobil : ")
        fmt.Scan(&namaMobil)
        fmt.Print("Tambahkan dipabrikan ke- : ")
        fmt.Scan(&idPabrikan)
        fmt.Print("Tahun : ")
        fmt.Scan(&tahun)
        fmt.Print("Harga Mobil : Rp ")
        fmt.Scan(&harga_mobil)
        tambahMobil(namaMobil, idPabrikan, tahun,
harga_mobil)
        fmt.Println()

    case 6 :
        var id, idPabrikan, tahun int
        var nama string
        var harga float64
        fmt.Print("Input ID mobil yang akan di ubah : ")
        fmt.Scan(&id)
        fmt.Print("Ubah Nama Mobil : ")
        fmt.Scan(&nama)
        fmt.Print("Ubah ID Pabrikan : ")
        fmt.Scan(&idPabrikan)
        fmt.Print("Ubah Tahun : ")
        fmt.Scan(&tahun)
        fmt.Print("Ubah Harga : ")
        fmt.Scan(&harga)
        ubahMobil(id, nama, idPabrikan, tahun, harga)
        fmt.Println()

    case 7 :
        fmt.Println()
        tampilkanMobil()
        fmt.Println()

    case 8 :
        var id_mobil int
        fmt.Print("Inputkan ID Mobil Untuk Menghapus : ")
        fmt.Scan(&id_mobil)
        hapusMobil(id_mobil)

    default:
        fmt.Println("Pilihan Tidak Valid")
    }
}
}

```

```
func tambahMobil(namaMobil string, idPabrikan, tahun int, harga float64) {  
    pabrikanList[i].HitungMobil++  
    nextCarID++  
}
```

1.Tambah Pabrikan  
2.Ubah Data Pabrikan  
3.Tampilkan Pabrikan  
4.Hapus Pabrikan  
5.Tambah Mobil  
6.Ubah Data Mobil  
7.Tampilkan Mobil  
8.Hapus Mobil  
Pilih Nomor : 6  
Input ID mobil yang akan di ubah : 1  
Ubah Nama Mobil : Yaris  
Ubah ID Pabrikan : 2  
Ubah Tahun : 2019  
Ubah Harga : 30000000  
Mobil berhasil diperbarui!

1.Tambah Pabrikan  
2.Ubah Data Pabrikan  
3.Tampilkan Pabrikan  
4.Hapus Pabrikan  
5.Tambah Mobil  
6.Ubah Data Mobil  
7.Tampilkan Mobil  
8.Hapus Mobil  
Pilih Nomor : 7  
ID Mobil : 1 Nama: Yaris ID Pabrikan: 2

```
func tambahMobil(namaMobil string, idPabrikan, tahun int, harga float64) {  
    pabrikanList[i].HitungMobil++  
    nextCarID++  
}
```

4.Hapus Pabrikan  
5.Tambah Mobil  
6.Ubah Data Mobil  
7.Tampilkan Mobil  
8.Hapus Mobil  
Pilih Nomor : 3  
ID: 1  
Nama: Toyota  
Wilayah: Jepang  
Hitung Mobil: 2  
ID: 2  
Nama: Mercedes  
Wilayah: Sampang  
Hitung Mobil: 0  
1.Tambah Pabrikan  
2.Ubah Data Pabrikan  
3.Tampilkan Pabrikan  
4.Hapus Pabrikan  
5.Tambah Mobil  
6.Ubah Data Mobil  
7.Tampilkan Mobil  
8.Hapus Mobil  
Pilih Nomor : 7  
ID Mobil : 1 Nama: Avanza ID Pabrikan: 1

**Keterangan:** untuk keseluruhan fungsi sudah aman, yang menjadi masalah yakni pada fungsi ubahmobil, ketika user menginputkan id pabrikan untuk merubah data mobil yang semula dari pabrikan A ke B, inputannya tidak terhitung kedalam data pabrikan yang dituju. Semisal pada data tersebut pabrikan A sudah memiliki 2 mobil, kemudian user ingin mengubah data dari mobil tersebut yang semula dari pabrikan A menjadi B. Output yang seharusnya adalah “hitung mobil” pada pabrikan A berkurang satu dan bertambah satu pada pabrikan B, namun output diatas tidak menunjukan perubahan apapun alias “hitung mobil” pada pabrikan A masih tetap bernilai 2.

([https://github.com/arvelmahsa/ALGORITHMMA\\_PEMROGRAMAN\\_2/tree/9346fe4187dc944c7530b98e03e1a34e1ce0f720/TugasBesar\\_\(Diler%20Mobil\)](https://github.com/arvelmahsa/ALGORITHMMA_PEMROGRAMAN_2/tree/9346fe4187dc944c7530b98e03e1a34e1ce0f720/TugasBesar_(Diler%20Mobil)) )



**Tegar Bangkit Wijaya** - Tampilkan daftar mobil berdasarkan nama pabrikan yang di input, pencarian mobil berdasarkan input disertakan data datanya

```
package main

import (
    "fmt"
    "strings"
)

// Tipe bentukan untuk data mobil
type Car struct {
    Manufacturer string
    Model        string
    Year         int
    Price        int
}

// Deklarasi array statis untuk menyimpan data mobil
var cars = [5]Car{
    {"Toyota", "Avanza", 2020, 200000000},
    {"Honda", "Civic", 2019, 350000000},
    {"Toyota", "Corolla", 2021, 300000000},
    {"Honda", "Jazz", 2018, 250000000},
    {"Toyota", "Fortuner", 2022, 500000000},
}

// Fungsi sequential search untuk mencari mobil berdasarkan
nama pabrikan
func searchCarsByManufacturer(manufacturer string, cars
[5]Car) []Car {
    var results []Car
    manufacturer = strings.ToLower(manufacturer) //
Normalisasi input untuk pencarian

    for _, car := range cars {
        if strings.ToLower(car.Manufacturer) ==
manufacturer {
            results = append(results, car)
        }
    }
    return results
}

// Fungsi selection sort untuk mengurutkan mobil berdasarkan
harga (ascending atau descending)
```

```

func sortCarsByPrice(cars []Car, ascending bool) {
    for i := 0; i < len(cars)-1; i++ {
        minOrMaxIdx := i
        for j := i + 1; j < len(cars); j++ {
            if ascending {
                if cars[j].Price <
cars[minOrMaxIdx].Price {
                    minOrMaxIdx = j
                }
            } else {
                if cars[j].Price >
cars[minOrMaxIdx].Price {
                    minOrMaxIdx = j
                }
            }
        }
        cars[i], cars[minOrMaxIdx] = cars[minOrMaxIdx],
cars[i]
    }
}

// Fungsi untuk menampilkan daftar mobil berdasarkan nama
pabrikan
func displayCarsByManufacturer(manufacturer string) {
    results := searchCarsByManufacturer(manufacturer, cars)
    if len(results) == 0 {
        fmt.Printf("Tidak ditemukan mobil dari pabrikan
%s'\n", manufacturer)
        return
    }

    // Mengurutkan hasil berdasarkan harga secara ascending
    sortCarsByPrice(results, true)

    fmt.Printf("Daftar mobil dari pabrikan '%s':\n",
manufacturer)
    for _, car := range results {
        fmt.Printf("%s %s (%d) - Rp%d\n", car.Manufacturer,
car.Model, car.Year, car.Price)
    }
}

func main() {
    var manufacturer string

    fmt.Print("Masukkan nama pabrikan: ")
    fmt.Scanln(&manufacturer)
}

```

```
    displayCarsByManufacturer (manufacturer)
}
```

```
PS D:\alpro_2\Tubes_Alpro> go run "d:\alpro_2\Tubes_Alpro\Fungsi_Tampil&pencarian.go"
Masukkan nama pabrikan: Toyota
Daftar mobil dari pabrikan 'Toyota':
Toyota Avanza (2020) - Rp200000000
Toyota Corolla (2021) - Rp300000000
Toyota Fortuner (2022) - Rp500000000
PS D:\alpro_2\Tubes_Alpro> go run "d:\alpro_2\Tubes_Alpro\Fungsi_Tampil&pencarian.go"
Masukkan nama pabrikan: Honda
Daftar mobil dari pabrikan 'Honda':
Honda Jazz (2018) - Rp250000000
Honda Civic (2019) - Rp350000000
PS D:\alpro_2\Tubes_Alpro> 
```

**Keterangan:** Fungsi Tampilkan daftar mobil berdasarkan nama pabrikan yang di input, pencarian mobil berdasarkan input disertakan data datanya.

<https://github.com/tegarbangkitwijaya>

**Avriel Fitria Rachma Suryantari** - Mengurutkan nama pabrikan berdasarkan jumlah mobil yang ada, menampilkan mobil terurut berdasarkan tahun keluar, nama, pabrikan, dan lainnya

```
package main

import (
    "fmt"
    "sort"
)

// Tipe bentukan untuk data mobil
type Mobil struct {
    ID_mobil    int
    NamaMobil   string
    IDpabrikan  int
    Tahun       int
    HargaMobil  float64
}

// Tipe bentukan untuk data pabrikan
type Pabrikan struct {
    ID          int
    Nama        string
    Wilayah     string
}
```

```

        HitungMobil int
    }

    // Deklarasi array statis untuk menyimpan data mobil dan
    pabrik
    const maxCars = 100

    var mobilList [maxCars]Mobil

    const maxPabrik = 100

    var pabrikList [maxPabrik]Pabrik

    var mobilCount int = 0
    var pabrikCount int = 0

    // Fungsi untuk mengurutkan nama pabrik berdasarkan jumlah
    mobil yang ada
    func sortPabrikByCarCount(ascending bool) {
        sort.SliceStable(pabrikList[:pabrikCount], func(i, j
    int) bool {
            if ascending {
                return pabrikList[i].HitungMobil <
    pabrikList[j].HitungMobil
            }
            return pabrikList[i].HitungMobil >
    pabrikList[j].HitungMobil
        })
    }

    // Fungsi untuk menampilkan mobil terurut berdasarkan tahun
    keluar, nama, pabrik, dan lainnya
    func displaySortedCars() {
        sort.SliceStable(mobilList[:mobilCount], func(i, j int)
    bool {
            return mobilList[i].Tahun < mobilList[j].Tahun
        })

        fmt.Println("Daftar mobil terurut:")
        for _, mobil := range mobilList[:mobilCount] {
            fmt.Printf("Nama: %s, Pabrik ID: %d, Tahun: %d,
    Harga: %.2f\n",
                mobil>NamaMobil, mobil.IDpabrik,
    mobil.Tahun, mobil.HargaMobil)
        }
    }

```

```

func main() {
    // Urutkan pabrikan berdasarkan jumlah mobil (ascending)
    sortPabrikanByCarCount(true)

    // Tampilkan pabrikan yang terurut berdasarkan jumlah
    mobil
    fmt.Println("Pabrikan terurut berdasarkan jumlah
    mobil:")
    for _, pabrikan := range pabrikanList[:pabrikanCount] {
        fmt.Printf("ID: %d, Nama: %s, Jumlah Mobil: %d\n",
        pabrikan.ID, pabrikan>Nama, pabrikan.HitungMobil)
    }

    // Tampilkan mobil terurut berdasarkan tahun
    displaySortedCars()
}

```

```

PS C:\Users\user> go run "d:\DATA (data d)\TUGAS KULIAH\Semester 3\Alpro2\tugas BESAAAAAAAAAAAAAAAAAAAAARR
RRRRRRRR\mengurutkan pabrikan - avriel.go"
Pabrikan terurut berdasarkan jumlah mobil:
Daftar mobil terurut:
PS C:\Users\user>

```

Keterangan: Urutan nama mobil, pabrikan, tahun, dan harga mobil akan terlihat setelah user menginputkan nama mobil, pabrikan, tahun, dan harga mobil.

[https://github.com/AvrielFRS/Avriel\\_TubesAlpro2](https://github.com/AvrielFRS/Avriel_TubesAlpro2)

**Muhammad Djidzan Naufal Nugraha** - Menampilkan 3 daftar mobil dan pabrikan dengan jumlah penjualan tertinggi.

```
package main

import (
    "fmt"
)

type Pabrikan struct {
    ID          int
    Nama         string
    Negara       string
    JumlahMobil int
}

type Mobil struct {
    ID          int
    Nama         string
    IDPabrikan  int
    Tahun       int
    Harga        float64
    Penjualan   int
}

var pabrikanList = []Pabrikan{
    {ID: 1, Nama: "Toyota", Negara: "Jepang", JumlahMobil: 2},
    {ID: 2, Nama: "Ford", Negara: "Amerika Serikat", JumlahMobil: 1},
    {ID: 3, Nama: "Honda", Negara: "Jepang", JumlahMobil: 1},
}

var mobilList = []Mobil{
    {ID: 1, Nama: "Corolla", IDPabrikan: 1, Tahun: 2020, Harga: 300,000,000, Penjualan: 5,000},
    {ID: 2, Nama: "Mustang", IDPabrikan: 2, Tahun: 2021, Harga: 800,000,000, Penjualan: 6,000},
    {ID: 3, Nama: "Civic", IDPabrikan: 3, Tahun: 2022, Harga: 400,000,000, Penjualan: 4,000},
}

func daftarMobilTerlaris() {
    var pilihanUrutan int
    fmt.Println("Pilih :")
}
```

```

    fmt.Println("1. Tertinggi ke Terendah")
    fmt.Println("2. Terendah ke Tertinggi")
    fmt.Print("Masukkan pilihan: ")
    fmt.Scan(&pilihanUrutan)

    for i := 0; i < len(mobilList)-1; i++ {
        indexTerpilih := i
        for j := i + 1; j < len(mobilList); j++ {
            if (pilihanUrutan == 1 &&
mobilList[j].Penjualan > mobilList[indexTerpilih].Penjualan)
||
                (pilihanUrutan == 2 &&
mobilList[j].Penjualan < mobilList[indexTerpilih].Penjualan)
{
                    indexTerpilih = j
                }
            }
        mobilList[i], mobilList[indexTerpilih] =
mobilList[indexTerpilih], mobilList[i]
    }

    fmt.Println("\n3 Mobil Terlaris:")
    for i := 0; i < 3 && i < len(mobilList); i++ {
        mobil := mobilList[i]
        var namaPabrikan string
        for _, pabrikan := range pabrikanList {
            if pabrikan.ID == mobil.IDPabrikan {
                namaPabrikan = pabrikan.Nama
                break
            }
        }
        fmt.Printf("Nama: %s, Pabrikan: %s, Penjualan:
%d\n",
                mobil.Nama, namaPabrikan, mobil.Penjualan)
    }
}

func main() {
    for {
        fmt.Println("\nMenu:")
        fmt.Println("1. Lihat Mobil Terlaris")
        fmt.Println("2. Keluar")
        fmt.Print("Masukkan pilihan: ")

        var pilihan int
        fmt.Scan(&pilihan)
    }
}

```

```

        switch pilihan {
        case 1:
            daftarMobilTerlaris()
        case 2:
            fmt.Println("Keluar dari program.")
            return
        default:
            fmt.Println("Pilihan tidak valid.")
        }
    }
}

```

```

PS C:\Users\rikay\Documents\Backup Data C:\KULIAH\TUGAS SEMESTER 3\P. ALPRO> cd .\Tubes\
PS C:\Users\rikay\Documents\Backup Data C:\KULIAH\TUGAS SEMESTER 3\P. ALPRO\Tubes> go run .\coba.go

Menu:
1. Lihat Mobil Terlaris
2. Keluar
Masukkan pilihan: 1
Pilih :
1. Tertinggi ke Terendah
2. Terendah ke Tertinggi
Masukkan pilihan: 1

3 Mobil Terlaris:
Nama: Mustang, Pabrikan: Ford, Penjualan: 6000
Nama: Corolla, Pabrikan: Toyota, Penjualan: 5000
Nama: Civic, Pabrikan: Honda, Penjualan: 4000

Menu:
1. Lihat Mobil Terlaris
2. Keluar
Masukkan pilihan: 1
Pilih :
1. Tertinggi ke Terendah
2. Terendah ke Tertinggi
Masukkan pilihan: 2

3 Mobil Terlaris:
Nama: Civic, Pabrikan: Honda, Penjualan: 4000
Nama: Corolla, Pabrikan: Toyota, Penjualan: 5000
Nama: Mustang, Pabrikan: Ford, Penjualan: 6000

Menu:
1. Lihat Mobil Terlaris
2. Keluar
Masukkan pilihan: 2
Keluar dari program.
PS C:\Users\rikay\Documents\Backup Data C:\KULIAH\TUGAS SEMESTER 3\P. ALPRO\Tubes>

```

**Keterangan:** Fungsi ini untuk mengurutkan dan menampilkan mobil terlaris berdasarkan penjualan. Fungsi ini juga memungkinkan User untuk memilih urutan pengurutan yang berbeda (Ascending atau Descending).

<https://github.com/MendoanGeprek/TUBES>