# 2. Koreksi untuk code pada soal nomer 2

package main

//menambahkan beberapa modul

import (

"fmt"

"log"

"gorm.io/driver/mysql"

"gorm.io/gorm"

)

model pada soal

type (

Area struct {

ID int64 `gorm:"column:id;primaryKey;"`

AreaValue int64 `gorm:"column:area\_value"`

AreaType string `gorm:"column:type"`

}

)

Implementasi model yang sehrusnya

type Model struct (

ID int64 `gorm:"column:id;primaryKey;"`

AreaValue int64 `gorm:"column:area\_value"`

AreaType string `gorm:"column:type"`

)

func (\_r \*AreaRepository) InsertArea(param1 int32, param2 int64, type []string, ar \*Model.Area) (err error) {

inst := \_r.DB.Model(ar)

Var area int

area = 0

switch type {

case 'persegi panjang':

var area := param1 \* param2 //harusnya tidak pakai var, karena =: sudah mewakili deklarasi

ar.AreaValue = area

ar.AreaType = 'persegi panjang'

err = \_r.DB.create(&ar).Error

if err != nil {

return err

}

case 'persegi':

var area = param1 \* param2

ar.AreaValue = area

ar.AreaType = 'persegi'

err = \_r.DB.create(&ar).Error

if err != nil {

return err

}

case segitiga: //kata segitiga harusnya berupa string yang mana harus disertai tanda petik

area = 0.5 \* (param1 \* param2)

ar.AreaValue = area

ar.AreaType = 'segitiga'

err = \_r.DB.create(&ar).Error

if err != nil {

return err

}

default:

ar.AreaValue = 0

ar.AreaType = 'undefined data'

err = \_r.DB.create(&ar).Error

if err != nil {

return err

}

}

}

func main() {

err = \_u.repository.InsertArea(10, 10, 'persegi')

if err != nil {

log.Error().Msg(err.Error())

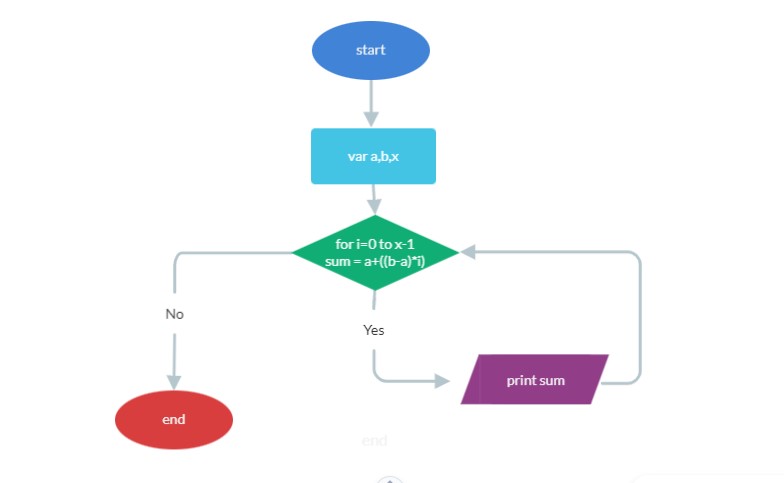
err = errors.New(en.ERROR\_DATABASE)

return err

}

}

# 3. Flow Chart



# Pseudocode

1. Begin
2. Declare number1, number2, x
3. for (i=0 to x-1)

sum = a + ((b-a)\*i)

1. Display sum
2. End

Berikut code yang sudah dicompile meggunakan golang

package main

import "fmt"

func main() {

a,b,x := 2,4,5

for i:=0; i < x; i++ {

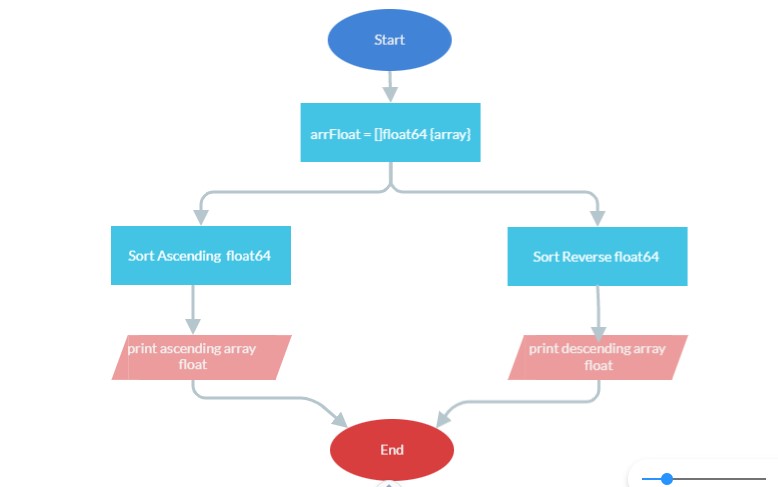
sum := a+((b-a)\*i)

fmt.Println(sum)

}

}

# 4. Flow Chart



# Pseudocode

1. Begin
2. Declare array as float64
3. Sorting by Ascending number
4. Display Ascending array
5. Sorting by Descending Array
6. Display Descending Array
7. End

Beriiku code yang sudah dicompile meggunakan golang

