

# **LAPORAN**

## **SISTEM STOCK MANAGEMENT**

Diajukan Sebagai Tugas Akhir Mata Kuliah Praktikum Berorientasi Obyek



### **DOSEN PENGAMPU MATA KULIAH :**

M. Bahrul Subkhi, M. Kom

### **Disusun Oleh :**

Anggota Kelompok

1. Apliana Sriyanti Bili (2213020242)
2. Virginia Abuk Klau (2213020019)
3. Danika Ambadar Alfari (2213020256)

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNIK  
UNIVERSITAS NUSANTARA PGRI KEDIRI**

**2024**

## DAFTAR ISI

|                                       |    |
|---------------------------------------|----|
| HALAMAN JUDUL .....                   | i  |
| DAFTAR ISI .....                      | ii |
| BAB I .....                           | 1  |
| PENDAHULUAN .....                     | 1  |
| A. Latar Belakang .....               | 1  |
| B. Rumusan Masalah.....               | 1  |
| C. Tujuan.....                        | 1  |
| D. Claas Diagram .....                | 1  |
| BAB II .....                          | 2  |
| A. Hasil Program dan Penjelasan ..... | 2  |
| BAB III .....                         | 2  |
| A. Kesimpulan .....                   | 3  |
| B. Saran .....                        | 3  |
| DAFTAR PUSTAKA.....                   | 4  |

## **BAB I**

### **PENDAHULUAN**

#### **A. Latar Belakang**

Perkembangan teknologi informasi khususnya teknologi informasi berbasis system stok management ini, dirasa sangat pesat dan hal ini berpengaruh terhadap aspek pekerjaan. Hampir semua perusahaan dalam hal pengambilan keputusan, penyebaran informasi, peningkatan efektifitas pekerjaan dan pelayanan telah menggunakan sistem informasi komputer, dan tidak menutup kemungkinan lebih memudahkan lagi apabila semua itu dilakukan dengan menggunakan sistem aplikasi system management, Bagi suatu perusahaan yang sedang berkembang seperti pada perusahaan produk barang yang bergerak di bidang produksi, tentunya suatu sistem inventory yang berguna untuk mengelola persediaan stok barang. Namun pendokumentasian yang digunakan saat ini kebanyakan secara manual atau masih terkomputerisasi sehingga menimbulkan kendala waktu maupun teknis dan pengetahuan dalam kinerja perusahaan. Pada mulanya perdagangan melalui internet dilakukan di desktop website, dimana untuk mengakses sebuah desktop website dibutuhkan perangkat komputer dan laptop. Meskipun dapat diakses menggunakan perangkat system stok managemen ( handphone dan smartphone ) situs-situs yang berbasis desktop website ini akan menjadi beban untuk perangkat tersebut. Selain karena membutuhkan koneksi internet yang besar, perangkat yang digunakan untuk mengakses desktop website tidak memiliki mobilitas tinggi khususnya komputer (Stefanus, 2013). Menurut (Fatmanto, 2013) dilakukan perancangan aplikasi yang difokuskan untuk membuat forum jual beli suatu barang dan proses transaksi yang aman bagi pengguna. Dikarenakan semakin banyaknya penjual barang online perseorangan dan juga pembeli atau penjual yang tidak tahu dimana dia dapat membeli suatu barang atau mengiklankan barangnya,

maka aplikasi ini ditujukan untuk mengumpulkan penjual dan pembeli di dalam suatu forum agar informasi yang didapatkan lebih mudah dan akurat.

## B. Rumusan Masalah

### 1. Ketidakefisienan Persediaan:

Bagaimana meningkatkan efisiensi penyimpanan dan manajemen persediaan agar dapat mengurangi biaya penyimpanan yang tidak perlu?

### 2. Kehilangan Stok dan Kecurangan:

Bagaimana mencegah atau mengurangi kehilangan stok dan risiko kecurangan dalam sistem manajemen stok?

### 3. Optimalisasi Pemesanan:

Bagaimana meningkatkan proses pemesanan untuk menghindari kekurangan stok atau kelebihan persediaan yang tidak perlu?

### 4. Pengendalian Kualitas dan Keamanan Persediaan:

Bagaimana memastikan kualitas dan keamanan persediaan, terutama untuk produk yang memiliki batas waktu kadaluwarsa?

### 5. Pelacakan Stok Real-time:

Bagaimana meningkatkan kemampuan untuk melacak stok secara real-time guna mengidentifikasi perubahan cepat dalam permintaan atau persediaan?

### C. Tujuan

1. Optimalisasi Persediaan:

Meminimalkan biaya persediaan sambil memastikan ketersediaan produk yang cukup.

2. Efisiensi Operasional:

Meningkatkan efisiensi proses pengelolaan stok untuk mengurangi biaya operasional.

3. Peningkatan Akurasi Prediksi:

Meningkatkan akurasi dalam meramalkan permintaan produk untuk menghindari kelebihan atau kekurangan stok.

4. Transparansi Stok:

Memberikan visibilitas yang tinggi terhadap stok secara real-time untuk memudahkan pengambilan keputusan.

5. Keamanan dan Pelacakan:

Memastikan keamanan stok dan kemampuan untuk melacak pergerakan barang dari gudang hingga pelanggan.

### D. Class Diagram

Diagram memperlihatkan class, atribut, metode, dan hubungan antar kelas dalam sistem Class diagram adalah salah satu jenis diagram struktur dalam UML yang digunakan

untuk menggambarkan detail struktur dengan jelas Class Berikut adalah beberapa fungsi utama dari class diagram:

1. Menunjukkan struktur dari suatu sistem dengan jelas
2. Meningkatkan pemahaman tentang gambaran umum atau skema dari suatu sistem
3. Membantu kamu untuk menyampaikan kebutuhan dari suatu system

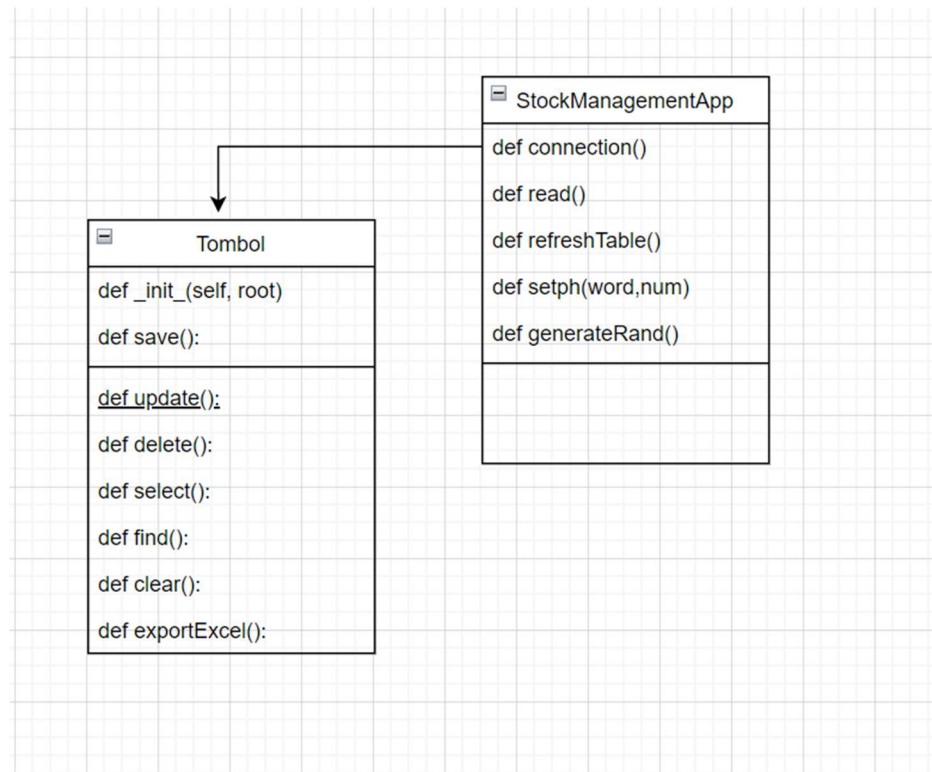
Class diagram memiliki komponen penyusun, yang mencakup nama class, atribut, dan metode

Ada tiga hubungan dalam diagram kelas: asosiasi, agregasi, dan pewarisan

- a. Asosiasi: Hubungan antara dua class yang bersifat statis, dapat diartikan sebagai hubungan antara class yang memiliki atribut tambahan seperti class lain
- b. Agregasi: Hubungan antara dua class di mana salah satu class merupakan bagian dari class lain, tetapi dua class ini dapat berdiri masing-masing
- c. Pewarisan: Kemampuan suatu class untuk mewarisi seluruh manfaat dari class yang diwarisi

Class diagram sering disebut dengan diagram struktur dan memiliki sifat statis, yakni hanya mampu menjelaskan hubungan apa yang terjadi, bukan menjelaskan apa yang terjadi jika kelasnya berhubungan

Class diagram dapat digunakan untuk memodelkan sistem dari sebuah perspektif bisnis dan merupakan satu-satunya diagram UML yang dapat dipetakan langsung menggunakan bahasa pemrograman



Gambar Diatas Merupakan Gambar Class Diagram.

## BAB II

### PEMBAHASAN

#### A. Hasil Program dan Penjelasan

Class untuk Membuat GUI dan Fungsi lainnya.

```
class StockManagementApp:
    def __init__(self, window):
        self.window = window
        self.window.title("Stock Management App")
        self.create_widgets()

window=tkinter.Tk()
window.title("Stock Management System")
window.geometry("720x640")
my_tree=ttk.Treeview(window,show='headings',height=20)
style=ttk.Style()

placeholderArray=['',' ',' ',' ',' ']
numeric='1234567890'
alpha='ABCDEFGHIJKLMNOPQRSTUVWXYZ'
```

Fungsi untuk mengkoneksikan Ke database Stockmanagementsystem

```
def connection():
    conn=pymysql.connect(
        host='localhost',
        user='root',
        password='',
        db='stockmanagementsystem'
    )
    return conn

conn=connection()
cursor=conn.cursor()

for i in range(0,5):
    placeholderArray[i]=tkinter.StringVar()
```



## Fungsi Untuk Menambahkan Atau Menyimpan Data Yang dimasukan ke Database

```
def read():
    cursor.connection.ping()
    sql=f"SELECT `item_id`, `name`, `price`, `quantity`, `category`, `date` FROM stocks ORDER BY `id` DESC"
    cursor.execute(sql)
    results=cursor.fetchall()
    conn.commit()
    conn.close()
    return results

def refreshTable():
    for data in my_tree.get_children():
        my_tree.delete(data)
    for array in read():
        my_tree.insert(parent='',index='end',iid=array,text="",values=(array),tag="orow")
    my_tree.tag_configure('orow',background="#EEEEEE")
    my_tree.pack()

def setph(word,num):
    for ph in range(0,5):
        if ph == num:
            placeholderArray[ph].set(word)

def generateRand():
    itemId=''
    for i in range(0,3):
        randno=random.randrange(0,(len(numeric)-1))
        itemId=itemId+str(numeric[randno])
    randno=random.randrange(0,(len(alpha)-1))
    itemId=itemId+'-'+str(alpha[randno])
    print("generated: "+itemId)
    setph(itemId,0)
```

## Kelas yang berisi kan Fungsi Tombol-Tombol Pada Program

```
class Tombol:
    def __init__(self, root):
        self.root = root
        self.create_widgets()
```

## Fungsi Untuk Tombol Simpan Data.

```
def save():
    itemId=str(itemIdEntry.get())
    name=str(nameEntry.get())
    price=str(priceEntry.get())
    qnt=str(qntEntry.get())
    cat=str(categoryCombo.get())
    valid=True
    if not(itemId and itemId.strip()) or not(name and name.strip()) or not(price and price.strip()) or not(qnt and qnt.strip()):
        messagebox.showwarning("", "Silakan isi semua entri")
        return
    if len(itemId) < 5:
        messagebox.showwarning("", "Id Barang Tidak Valid")
        return
    if(not(itemId[3]=='-')):
        valid=False
    for i in range(0,3):
        if(not(itemId[i] in numeric)):
            valid=False
            break
    if(not(itemId[4] in alpha)):
        valid=False
    if not(valid):
        messagebox.showwarning("", "Id Barang Tidak Valid")
        return
    try:
        cursor.connection.ping()
        sql=f"SELECT * FROM stocks WHERE `item_id` = '{itemId}' "
        cursor.execute(sql)
        checkItemNo=cursor.fetchall()
        if len(checkItemNo) > 0:
            messagebox.showwarning("", "Id Item sudah digunakan")
            return
```

## Fungsi Untuk Tombol Update

```
def update():
    selectedItemId = ''
    try:
        selectedItem = my_tree.selection()[0]
        selectedItemId = str(my_tree.item(selectedItem)['values'][0])
    except:
        messagebox.showwarning("", "Silakan pilih baris data")
    print(selectedItemId)
    itemId = str(itemIdEntry.get())
    name = str(nameEntry.get())
    price = str(priceEntry.get())
    qnt = str(qntEntry.get())
    cat = str(categoryCombo.get())
    if not(itemId and itemId.strip()) or not(name and name.strip()) or not(price and price.strip()) or not(qnt and qnt.strip()) or not(cat and cat.strip()):
        messagebox.showwarning("", "Silakan isi semua entri")
        return
    if(selectedItemId!=itemId):
        messagebox.showwarning("", "Anda tidak dapat mengubah ID Item")
        return
    try:
        cursor.connection.ping()
        sql=f"UPDATE stocks SET `name` = '{name}', `price` = '{price}', `quantity` = '{qnt}', `category` = '{cat}' WHERE `item_id` = '{itemId}'"
        cursor.execute(sql)
        conn.commit()
        conn.close()
        for num in range(0,5):
            setph('',(num))
    except Exception as err:
        messagebox.showwarning("", "Terjadi kesalahan ref: "+str(err))
        return
    refreshTable()
```

## Fungsi untuk Tombol Delete

```
def delete():
    try:
        if(my_tree.selection()[0]):
            decision = messagebox.askquestion("", "Hapus data yang dipilih?")
            if(decision != 'yes'):
                return
            else:
                selectedItem = my_tree.selection()[0]
                itemId = str(my_tree.item(selectedItem)['values'][0])
                try:
                    cursor.connection.ping()
                    sql=f"DELETE FROM stocks WHERE `item_id` = '{itemId}' "
                    cursor.execute(sql)
                    conn.commit()
                    conn.close()
                    messagebox.showinfo("", "Data telah berhasil dihapus")
                except:
                    messagebox.showinfo("", "Maaf, terjadi kesalahan")
                    refreshTable()
    except:
        messagebox.showwarning("", "Silakan pilih baris data")
```

Fungsi Untuk tombol Select atau pilih

```
def select():
    try:
        selectedItem = my_tree.selection()[0]
        itemId = str(my_tree.item(selectedItem)['values'][0])
        name = str(my_tree.item(selectedItem)['values'][1])
        price = str(my_tree.item(selectedItem)['values'][2])
        qnt = str(my_tree.item(selectedItem)['values'][3])
        cat = str(my_tree.item(selectedItem)['values'][4])
        setph(itemId,0)
        setph(name,1)
        setph(price,2)
        setph(qnt,3)
        setph(cat,4)
    except:
        messagebox.showwarning("", "Silakan pilih baris data")
```

Fungsi Untuk tombol Menemukan

```
def find():
    itemId = str(itemIdEntry.get())
    name = str(nameEntry.get())
    price = str(priceEntry.get())
    qnt = str(qntEntry.get())
    cat = str(categoryCombo.get())
    cursor.connection.ping()
    if(itemId and itemId.strip()):
        sql = f"SELECT `item_id`, `name`, `price`, `quantity`, `category`, `date` FROM stocks WHERE `item_id` LIKE '%{itemId}%' "
    elif(name and name.strip()):
        sql = f"SELECT `item_id`, `name`, `price`, `quantity`, `category`, `date` FROM stocks WHERE `name` LIKE '%{name}%' "
    elif(price and price.strip()):
        sql = f"SELECT `item_id`, `name`, `price`, `quantity`, `category`, `date` FROM stocks WHERE `price` LIKE '%{price}%' "
    elif(qnt and qnt.strip()):
        sql = f"SELECT `item_id`, `name`, `price`, `quantity`, `category`, `date` FROM stocks WHERE `quantity` LIKE '%{qnt}%' "
    elif(cat and cat.strip()):
        sql = f"SELECT `item_id`, `name`, `price`, `quantity`, `category`, `date` FROM stocks WHERE `category` LIKE '%{cat}%' "
    else:
        messagebox.showwarning("", "Silakan isi salah satu entri")
        return
    cursor.execute(sql)
    try:
        result = cursor.fetchall();
        for num in range(0,5):
            setph(result[0][num],(num))
        conn.commit()
        conn.close()
    except:
        messagebox.showwarning("", "Tidak ada data ditemukan")
```

Fungsi Untuk tombol Bersikan

```
def clear():
    for num in range(0,5):
        setph('',(num))
```

## Fungsi tombol Untuk mengunduh File Di excel Dari Database

```
def exportExcel():
    cursor.connection.ping()
    sql=f"SELECT `item_id`, `name`, `price`, `quantity`, `category`, `date` FROM stocks ORDER BY `id` DESC"
    cursor.execute(sql)
    dataraw=cursor.fetchall()
    date = str(datetime.now())
    date = date.replace(' ', '-')
    date = date.replace(':', '-')
    dateFinal = date[0:16]
    with open("stocks_"+dateFinal+".csv", 'a', newline='') as f:
        w = csv.writer(f, dialect='excel')
        for record in dataraw:
            w.writerow(record)
    print("saved: stocks_"+dateFinal+".csv")
    conn.commit()
    conn.close()
    messagebox.showinfo("", "File Excel diunduh")

frame=tkinter.Frame(window,bg="#02577A")
frame.pack()

btnColor="#196E78"
```

- Penjelasan dari setiap alur codingan tersebut :

### 1. Imports:

Skrip ini dimulai dengan mengimpor modul-modul yang diperlukan seperti tkinter untuk GUI, ttk untuk widget Tkinter berthemes, messagebox untuk menampilkan kotak dialog, random untuk menghasilkan angka acak, pymysql untuk interaksi database MySQL, csv untuk menangani file CSV, dan datetime untuk bekerja dengan tanggal dan waktu.

### 2. Pengaturan Jendela:

Membuat jendela Tkinter (window) dengan judul, ukuran, dan geometri tertentu.

### 3. Pengaturan Treeview:

Membuat widget ttk.Treeview bernama my\_tree untuk menampilkan data tabular (dalam hal ini, informasi stok). Mendefinisikan kolom dan judul untuk treeview.

### 4. Koneksi Database:

Mendefinisikan fungsi connection() untuk menjalin koneksi ke database MySQL. Parameter koneksi (host, user, password, db) ditentukan.

### 5. Inisialisasi Placeholder:

Membuat array placeholderArray untuk menyimpan instansi Tkinter StringVar. Ini akan digunakan untuk mengatur dan mendapatkan nilai dari berbagai widget masukan. Fungsi

6. Baca (Read):

Fungsi read() didefinisikan untuk mengambil data dari tabel database bernama 'stocks' dan mengembalikan hasilnya. Fungsi Perbarui Tabel (Refresh Table): Fungsi refreshTable() didefinisikan untuk menghapus data yang ada di treeview dan mengisinya dengan data terbaru dari database.

7. Fungsi Set Placeholder:

Fungsi setph(word, num) didefinisikan untuk mengatur nilai placeholderArray pada indeks tertentu.

8. Fungsi Generate Random ID:

Fungsi generateRand() menghasilkan ID item acak berdasarkan kombinasi angka dan alfabet.

9. Fungsi Simpan (Save):

Fungsi save() dipicu saat tombol "SIMPAN" diklik. Memvalidasi data masukan, memeriksa duplikasi ID item, dan menyisipkan rekaman baru ke dalam database.

10. Fungsi Perbarui (Update):

Fungsi update() dipicu saat tombol "PERBAHARUI" diklik. Memperbarui rekaman yang dipilih di database dengan nilai baru.

11. Fungsi Hapus (Delete):

Fungsi delete() dipicu saat tombol "HAPUS" diklik. Menghapus rekaman yang dipilih dari database setelah dikonfirmasi dengan pengguna.

12. Fungsi Pilih (Select):

Fungsi select() dipicu saat tombol "PILIH" diklik. Mengambil nilai rekaman yang dipilih dan mengisi widget masukan.

13. Fungsi Temukan (Find):

Fungsi find() dipicu saat tombol "MENEMUKAN" diklik. Mencari rekaman berdasarkan kriteria yang ditentukan pengguna dan mengisi widget masukan dengan rekaman pertama yang cocok.

14. Fungsi Bersihkan (Clear):

Fungsi `clear()` membersihkan nilai-nilai di widget masukan.

#### 15. Fungsi Ekspor Excel:

Fungsi `exportExcel()` mengekspor seluruh data stok ke file CSV dengan nama berdasarkan tanggal dan waktu saat ini.

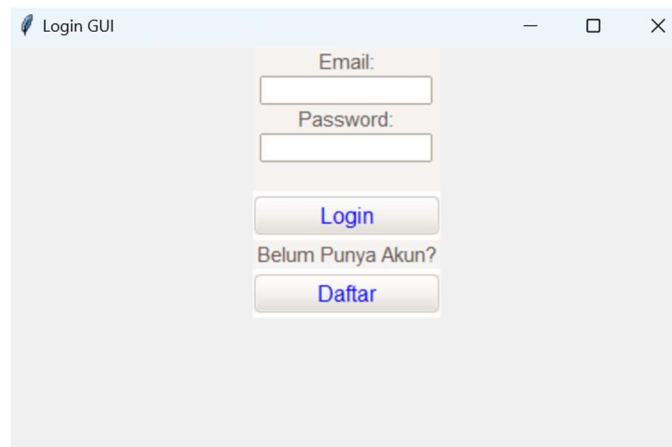
#### 16. Tata Letak GUI:

Skrip menentukan tata letak GUI menggunakan frame, label, tombol, widget masukan, dan treeview untuk menampilkan data.

#### 17. Perulangan Utama Jendela:

`mainloop()` Tkinter dipanggil untuk menjalankan aplikasi. Catatan: Skrip mengasumsikan adanya database MySQL bernama 'stockmanagementsystem' dengan tabel bernama 'stocks'. Struktur tabel 'stocks' tidak disediakan dalam kode, tetapi sepertinya memiliki kolom-kolom seperti 'item\_id', 'name', 'price', 'quantity', 'category', dan 'date'. Kode tidak mengatasi pengecualian secara efektif dalam operasi database; perbaikan dapat dilakukan dalam hal penanganan kesalahan dan umpan balik pengguna.

#### ▪ Tampilan Halaman Login :



#### ▪ Tampilan halman registrasi

Login GUI

Masukkan email baru:

Masukkan password baru:

Registrasi

- Tampilan Halaman Aplikasi GUI

Stock Management System

Manage

SIMPAN PERBAHARUI HAPUS PILIH MENEMUKAN BERSIHKAN EXPOR EXCEL

Form

IDENTITAS  MEMBUAT ID

NAMA

HARGA

KUANTITAS

KATEGORI

| Identitas | Nama        | harga   | Kuantitas | Kategori        | Tanggal & waktu     |
|-----------|-------------|---------|-----------|-----------------|---------------------|
| 429-X     | AMD RYZEN 5 | 2000000 | 125       | BAGIAN KOMPUTER | 2024-01-02 05:54:12 |

## **BAB II**

### **PENUTUP**

#### **A. KESIMPULAN**

Setelah menyelesaikan penelitian Aplikasi Stok Barang Berbasis Web maka diperoleh kesimpulan sebagai berikut:

1. Aplikasi dapat digunakan untuk menyimpan data supplier, data harga, data barang, jenis barang, satuan barang, dan data user
2. Aplikasi dapat menampilkan dan mengunduh laporan transaksi barang masuk dan keluar
3. Meminimalisir terjadinya kehilangan dan kerusakan data karena data disimpan di dalam database
4. Waktu untuk mencari data barang lebih cepat sehingga dapat menghemat waktu
5. Keamanan data terjamin karena ketika ingin mengakses aplikasi, harus terlebih dahulu login dan perlu memasukkan username dan password yang terdaftar di database.

#### **B. SARAN**

Berdasarkan kesimpulan yang sudah diuraikan di atas, hal yang diharapkan atau saran untuk ke depannya adalah agar sistem yang dibangun dapat bekerja dengan lebih baik, perbaikan dari segi tampilan agar lebih menarik untuk dilihat serta penambahan fitur berupa monitoring stok agar memudahkan dalam mengendalikan stok barang baik itu masuk dari pengadaan maupun keluar untuk penjualan atau produksi.



## **DAFTAR PUSTAKA**

A. S. Rosa., Shalahuddin, M. (2013). Rekayasa Perangkat Lunak Terstruktur Dan Berorientasi Objek. Bandung: Informatika.