**MAKALAH**

**MEMBUAT APLIKASI PENGADAAN BARANG DENGAN OPERASI CRUD MENGGUNAKAN BAHASA PEMROGRAMAN PYTHON**

**DAN DATABASE MYSQL**

Dibuat Sebagai Tugas Akhir Mata Kuliah Dasar Pemrograman Semester 1

Dosen Pengampu:

**Walim, M.Kom.**



Disusun Oleh:

**Kelompok 6**

Abdul Aziz

Ahmad Maulana Zuhdi 19240878

Defa Raihan Agis

Muhammad Nadhif Fadhilah

Nero Aziz Saputra

**KELAS 19.1A.24**

**FAKULTAS TEKNIK DAN INFORMATIKA**

**UNIVERSITAS BINA SARANA INFORMATIKA**

**TAHUN 2024**

**KATA PENGANTAR**

Puji syukur kami panjatkan kepada Tuhan Yang Maha Esa, karena berkat limpahan rahmatnya penyusun dapat menyelesaikan makalah ini tepat waktu tanpa ada halangan yang berarti dan sesuai dengan harapan.

Ucapan terima kasih kami sampaikan kepada Pak Walim, M.Kom. sebagai dosen pengampu Mata Kuliah Dasar Pemrograman yang telah memberi arahan dan pemahaman dalam penyusunan makalah ini.

Kami menyadari bahwa dalam penyusunan makalah ini masih banyak kekurangan karena keterbatasan kami. Maka dari itu kami mengharap kritik dan sarannya untuk kami agar dapat menyusun makalah dengan lebih baik lagi kedepannya. Semoga apa yang ditulis dapat bermanfaat bagi semua pihak yang membutuhkan.

Serang Baru, 05 November 2024

Kelompok 6

**DAFTAR ISI**

**DAFTAR TABEL**

**DAFTAR GAMBAR**

**DAFTAR LAMPIRAN**

# BAB I PENDAHULUAN

## Latar Belakang

Aplikasi pengadaan barang merupakan salah satu aspek penting dalam manajemen bisnis. Proses ini mencakup serangkaian aktivitas yang bertujuan untuk memastikan bahwa barang dan jasa yang dibutuhkan tersedia dengan cara efisien dan efektif. Namun, banyak organisasi masih menghadapi tantangan dalam mengelola pengadaan barang secara manual, yang dapat mengakibatkan pemborosan sumber daya.

Seiring dengan perkembangan teknologi informasi, penggunaan aplikasi dalam pengadaan barang menjadi semakin relevan. Aplikasi berbasis komputer dapat membantu dalam otomatisasi proses pengadaan barang sehingga mempermudah pengelolaan data dan pengawasan stok. Salah satu pendekatan yang popular untuk membangun aplikasi semacam itu adalah menggunakan bahasa pemrograman Python, yang dikenal akan kesederhanaan sintaksinya dan beragam pustaka yang mendukung pengembangan aplikasi. Selain itu, penggunaan MySQL sebagai system manajemen basis data memungkinkan pengelolaan data yang terstruktur dan efisien, serta kemudahan dalam menjalankan operasi CRUD (Create, Read, Update, Delete).

Melalui makalah ini, penulis akan membahas mengenai pengembangan aplikasi pengadaan barang dengan operasi CRUD menggunakan bahasa pemrograman Python dan database MySQL. Pembahasan ini diharapkan dapa memberikan gambaran yang jelas tentang langkah-langkah yang diperlukan dalam proses pengembangan. Dengan demikian, makalah ini diharapkan dapat menjadi referensi bagi pengembang dan praktisi dalam menerapkan teknologi untuk meningkatkan efisiensi dan efektivitas pengadaan barang.

## Rumusan Masalah

## Tujuan Penulisan

### Menganalisis kebutuhan user

Aplikasi pengadaan barang dapat meningkatkan efisiensi dan efektivitas bagi pengguna dalam pengelolaan data.

### Mengembangkan aplikasi dengan Operasi CRUD

Penulisan ini bertujuan untuk mengembangkan aplikasi pengadaan barang dengan menerapkan operasi CRUD (Create, Read, Update, Delete) menggunakan bahasa pemrograman Python dan database MySQL, serta menjelaskan Langkah-langkah teknis yang diperlukan dalam proses pengembangan tersebut.

### Sebagai projek Mata Kuliah Dasar Pemrograman

Tujuan lain dalam pembuatan aplikasi ini adalah sebagai tugas akhir semester 1 mata kuliah Dasar Pemrograman.

# BAB II PEMBAHASAN

## Penjelasan Sintaks

### Line 2-5

import mysql.connector

import time

import getpass

import os

Import mysql.connector adalah modul Python yang digunakan untuk membuat koneksi adalah program Python dengan database MySQL.

Import time adalah modul yang menyediakan fungsi untuk mengelola waktu dalam program Python.

Import getpass adalah modul yang berfungsi untuk user menginput value tanpa menampilkan karakter yang diinput oleh user. Jika kita pernah menggunakan Linux dengan basis CLI(Command Line Interface) maka pada halaman login saat user menginputkan password, karakter yang diinput user tidak di-print oleh program, maka fungsi getpass kurang lebih sama seperti itu.

Import os adalah modul untuk berinteraksi dengan sistem operasi yang sedang berjalan.

### Function clear

1. def clear():
2. if os.name == 'nt':
3. os.system('cls')
4. else:
5. os.system('clear')

Function clear berfungsi untuk pengoperasian membersihkan terminal. Nt adalah identitas untuk sistem operasi Windows, sedangkan untuk sistem operasi berbasis Unix seperti pada Linux atau macOS mempunyai identitas yaitu posix. Karna sistem operasi yang berjalan di komputer program adalah Windows akan mengeksekusi perintah os.system(‘cls’), jika yang berjalan bukan sistem operasi windows maka akan mengeksekusi os.system(‘clear’).

### Function db\_connection

def db\_connection():

    try:

        return mysql.connector.connect(

            host = 'localhost',

            user = 'root',

            password = '',

            database = 'db\_minimarket'

        )

    except mysql.connector.Error as e: *#*

        print(f"Error: {e}")

Function ini berfungsi untuk membuat koneksi kedalam MySQL sintaks akan mencoba mengeksekusi kolom try, jika dalam kolom try terdapat sintaks yang error maka akan mengeksekusi kolom except. Sintaks myql.connector.Error adalah untuk menampilkan letak kesalahan saat membuat koneksi.

### Function login

def login(username, password):

    try:

        db = db\_connection()

        cursor = db.cursor()

        cursor.execute("SELECT \* FROM users WHERE username = %s AND password = %s", (username, password))

        get\_user = cursor.fetchone()

        if get\_user:

            clear()

            print("\n\033[32m\033[1m!--Login Berhasil--!\033[0m\033[30m\n")

            time.sleep(1)

            clear()

            main()

        else:

            clear()

            print("\n\033[31m\033[1m!--Login Gagal--!\nusername atau password salah\033[0m\033[30m\n")

            time.sleep(1)

            clear()

    except ValueError as e:

        print(f"Error: {e}")

    finally:

        cursor.close()

        db.close()

Function login menampung segala kebutuhan untuk user login. Penjelasan query untuk cursor mengeksekusi SQL adalah:

SELECT \* FROM : memilih semua kolom pada table.

users : nama table di database.

username = %s : mencari kolom username sesuai dengan yang

diinput oleh user.

password = %s : mencari kolom password sesuai dengan yang

diinput oleh user.

### Function add\_new

def add\_new(name, category, weight, quantity, supplier):

    try:

        db = db\_connection()

        cursor = db.cursor()

        cursor.execute("SELECT \* FROM tb\_product WHERE name = %s AND weight = %s AND supplier = %s", (name, weight, supplier))

        if cursor.rowcount == 0:

            cursor.execute("INSERT INTO tb\_product (name, category, weight, quantity, supplier) VALUES(%s, %s, %s, %s, %s)", (name, category, weight, quantity, supplier))

            db.commit()

            print("\n\033[32m\033[1m!--Product Berhasil Ditambahkan--!\033[0m\033[30m\n")

            time.sleep(2)

            clear()

        else:

            print(f"\033[31m\033[1m!--Gagal Menambah Product Baru--!\nProduct Sudah Ada\033[0m\033[30m")

            time.sleep(2)

            clear()

            main()

    except ValueError as e:

        print(f"Error: {e}")

        time.sleep(2)

        clear()

    finally:

        cursor.close()

        db.close()

Function add\_new memberikan perintah untuk menambahkan item baru didalam database. Query pertama berfungsi untuk memeriksa kedalam database apakah kolom name, weight, dan supplier sudah ada atau belum, jika belum maka akan mengeksekusi query untuk menginput data kedalam database.

### Function update\_qty

def update\_qty(id, name, quantity, weight, supplier):

    try:

        db = db\_connection()

        cursor = db.cursor()

        cursor.execute("SELECT id FROM tb\_product WHERE name = %s AND weight = %s AND supplier = %s", (name, weight, supplier))

        select = cursor.fetchone()

        if select:

            cursor.execute("UPDATE tb\_product SET quantity = quantity + %s WHERE id = %s AND name = %s AND weight = %s AND supplier = %s", (quantity, id, name, weight, supplier))

            db.commit()

            print("\n\033[32m\033[1m!--Berhasil Menambahkan Jumlah Product--!\033[0m\033[30m\n")

            time.sleep(2)

            clear()

        else:

            print(f"\n\033[31m\033[1m!--Gagal Menambah Jumlah Product--!\033[0m\033[30m\n")

            time.sleep(2)

            clear()

    except ValueError as e:

        print(f"Error: {e}")

    finally:

        cursor.close()

        db.close()

Function update\_qty memungkinkan user untuk dapat menambah jumlah stok barang yang ada pada database.

### Function reduce\_qty

def reduce\_qty(id, name, quantity, weight, supplier):

    try:

        db = db\_connection()

        cursor = db.cursor()

        cursor.execute("SELECT id FROM tb\_product WHERE name = %s AND weight = %s AND supplier = %s", (name, weight, supplier))

        select = cursor.fetchone()

        if select:

            cursor.execute("UPDATE tb\_product SET quantity = quantity - %s WHERE id = %s AND name = %s AND weight = %s AND supplier = %s", (quantity, id, name, weight, supplier))

            db.commit()

            print("\n\033[32m\033[1m!--Berhasil Mengurangi Jumlah Product--!\033[0m\033[30m\n")

            time.sleep(2)

            clear()

        else:

            print(f"\n\033[31m\033[1m!--Gagal Mengurangi Jumlah Product--!\033[0m\033[30m\n")

            time.sleep(2)

            clear()

    except ValueError as e:

        print(f"Error: {e}")

    finally:

        cursor.close()

        db.close()

Sama seperti function update\_qty, function reduce\_qty memiliki query yang sama, hanya saja SET quantity = quantity - %s bukan SET quantity = quantity + %s.

### Function remove\_product

def remove\_product(id, name, weight, supplier):

    try:

        db = db\_connection()

        cursor = db.cursor()

        cursor.execute("DELETE FROM tb\_product WHERE id = %s AND name = %s AND weight = %s AND supplier = %s", (id, name, weight, supplier))

        db.commit()

        print("\n\033[32m\033[1m!--Product berhasil dihapus--!\033[0m\033[30m\n")

        time.sleep(2)

        clear()

    except ValueError as e:

        print(f"\033[31m\033[1m!--Gagal Menghapus Product--!\nError: {e}\033[0m\033[30m")

        time.sleep(2)

        clear()

    finally:

        cursor.close()

        db.close()

Function ini digunakan jika ingin menghapus product dari database dengan mencari ID, name, weight, supplier yang sesuai pada kolom database.

### Function show\_all

def show\_all():

    try:

        db = db\_connection()

        cursor = db.cursor()

        cursor.execute("SELECT \* FROM tb\_product")

        show\_result = cursor.fetchall()

        print("Daftar Produk:")

        print("="\*100)

        print(f"{'ID':<20} {'Name':<20} {'Categpry':<20} {'Weight':<10} {'Quantity':<10} {'Supplier':<20}")

        print("="\*100)

        for cell in show\_result:

            print(f"{cell[0]:<20} {cell[1]:<20} {cell[2]:<20} {cell[3]:<10} {cell[4]:<10} {cell[5]:<20}")

        print("="\*100)

    except ValueError as e:

        print(f"Error: {e}")

    finally:

        cursor.close()

        db.close()

Function show\_all berfungsi untuk menampilkan item yang ada pada database, ini akan membuat lebih mudah untuk melihat stok tanpa harus membuka table database.

### Function main

def main():

    while True:

        print("\n\033[35m\033[1mSelamat Datang Di Minimarket\033[0m\033[30m\n")

        print("1. Tambah Product Baru")

        print("2. Update Jumlah Product")

        print("3. Hapus Product")

        print("4. Lihat Semua Product")

        print("5. Keluar")

        user\_answer = int(input("Masukkan pilihan kamu(1/2/3/4/5): "))

        if user\_answer == 1:

            clear()

            print("Menambahkan  Product Baru\n")

            nama\_product = input("Nama Product: ")

            catg\_product = input("Kategoti Product: ")

            weig\_product = int(input(("Berat Product(gram): ")))

            qtyy\_product = int(input(("Jumlah Product: ")))

            splr\_product = input("Supplier Product: ")

            add\_new(nama\_product, catg\_product, weig\_product, qtyy\_product, splr\_product)

        elif user\_answer == 2:

            clear()

            print("1. Tambahi Jumlah Product: ")

            print("2. Kurangi Jumlah Product: ")

            upd\_user\_answer = int(input("Masukkan pilihan kamu(1/2): "))

            if upd\_user\_answer == 1:

                clear()

                print("Menambah Jumlah Product\n")

                frId\_product = int(input("ID Product: "))

                nama\_product = input("Nama Product: ")

                weig\_product = int(input("Berat Product(gram): "))

                splr\_product = input("Supplier Product: ")

                qtyy\_product = int(input("Tambah Jumlah Product: "))

                update\_qty(frId\_product, nama\_product, qtyy\_product, weig\_product, splr\_product)

            elif upd\_user\_answer == 2:

                clear()

                print("Mengurangi Jumlah Product\n")

                frId\_product = int(input("ID Product: "))

                nama\_product = input("Nama Product: ")

                weig\_product = int(input("Berat Product(gram): "))

                splr\_product = input("Supplier Product: ")

                qtyy\_product = int(input("Tambah Jumlah Product: "))

                reduce\_qty(frId\_product, nama\_product, qtyy\_product, weig\_product, splr\_product)

            else:

                clear()

                print("\033[31mJawaban tidak valid !\033[30m")

                clear()

        elif user\_answer == 3:

            clear()

            print("Menghapus Product\n")

            frId\_product = int(input("ID Product: "))

            nama\_product = input("Nama Product: ")

            weig\_product = int(input("Berat Product(gram): "))

            splr\_product = input("Supplier Product: ")

            remove\_product(frId\_product, nama\_product, weig\_product, splr\_product)

        elif user\_answer == 4:

            show\_all()

        elif user\_answer == 5:

            break

        else:

            clear()

            print("\033[31mJawaban tidak valid !\033[30m")

            clear()

Ini merupakan fungsi utama dalam aplikasi ini, fungsi ini yang menghandle fungsi fungsi lainnya agar berjalan sesuai dengan kebutuhan. Jika user menginput nilai 1 maka akan menuju ke function add\_new, tentunya user diminta untuk menginput beberapa persyaratan yang sesuai agar dapat menjalankan query sesuai pada functionnya.

### Jalankan program

if \_\_name\_\_ == "\_\_main\_\_":

    clear()

    while True:

        username = input("Username: ")

        password = getpass.getpass("Password: ")

        login(username, password)

Logika ini digunakan untuk mengeksekusi program pertama yang dijalankan. Pertama membuat loop untuk user menginput username dan password sampai value yang diisi bernilai True. Function login dijalankan pertama sebelum user dapat memasuki aplikasi