**MAKALAH PEMOGRAMAN BEORIENTASI OBJEK**

**“SISTEM SEKOLAH*”***

****

Dibuat Oleh:

**Kelompok 3**

1. **DIMAS GILANG PRAYOGO (2019230046)**
2. **WAHYU AJU NUGROHO (2019230052)**
3. **AQIL MIFTACH PUTRA (2019230051)**

**PEMROGRAMAN BERBASIS OBJEK**

**SEMESTER 5**

**TEKNOLOGI INFORMASI**

**FAKULTAS TEKNIK**

**UNIVERSITAS DARMA PERSADA  
JAKARTA  
2022**

**KATA PENGANTAR**

Puji syukur penulis panjatkan kehadirat Tuhan Yang Maha Esa karena berkat limpahan rahmat dan karunia-Nya sehingga penulis dapat menyusun makalah tentang “Aplikasi penjualan motor”dengan menggunakan java (Netbeans) ini dapat diselesaikan dengan tepat waktu. Makalah ini disusun untuk memenuhi penyelesaian tugas kelompok pada mata kuliah Pemrograman Beorientasi Objek.

Penulis menyadari bahwa makalah yang penulis susun ini masih jauh dari sempurna, oleh karena itu kritik dan saran sangat penulis harapkan dari berbagai pihak. Sebagai manusia biasa, kami berusaha dengan sebaik-baiknya dan semaksimal mungkin, dan sebagai manusia biasa, penulis tidak luput dari segala kesalahan serta kekhilafan dalam menyusun makalah ini.

Untuk menyempurnakan karya ini, penulis dengan senang hati akan menerima kritik dan saran yang sifatnya membangun dari berbagai pihak. Sehingga di kemudian hari penulis dapat menyempurnakan makalah ini dan penulis dapat belajar dari kesalahan-kesalahan yang telah penulis lakukan. Demikian penulis berharap semoga makalah ini dapat bermanfaat khususnya bagi penulis dan umumnya bagi semua pihak yang berkepentingan.

Jakarta, 11 Januari 2022

Penulis

Kelompok 3

# **DAFTAR ISI**

[**DAFTAR ISI** 3](#_Toc94727467)

[**BAB I** 4](#_Toc94727468)

[**1.1** **Latar Belakang** 4](#_Toc94727469)

[**1.2** **Rumusan Masalah** 5](#_Toc94727470)

[**1.3** **Tujuan dan Manfaat** 5](#_Toc94727471)

[**BAB II** 6](#_Toc94727472)

[2.1 OPP 6](#_Toc94727473)

[2.2 Struktur yang terdapat di OPP 6](#_Toc94727474)

[o **Kelas** 6](#_Toc94727475)

[o **Objek** 6](#_Toc94727476)

[o **Abstraction** 7](#_Toc94727477)

[o **Inheritance** 7](#_Toc94727478)

[o **Encapsulation** 7](#_Toc94727479)

[o **Polymorphism** 8](#_Toc94727480)

[2.3 Use Case Diagram 9](#_Toc94727481)

[2.4 Activity Diagram 9](#_Toc94727482)

[2.5 Sequence Diagram 10](#_Toc94727483)

[2.6 Class Diagram 10](#_Toc94727484)

[**BAB III** 11](#_Toc94727485)

[**Daftar Pustaka** 13](#_Toc94727486)

# 

# **BAB I**

**PENDAHULUAN**

## **Latar Belakang**

**Perkembangan teknologi** menjadi hal yang sudah tidak asing lagi perkembangan teknologi informasi saat ini memang jauh lebih pesat dari tahun tahun sebelumnya transformasi dari teknologi masalalalu menjadi teknologi yang lebih cangggih mudah dan cepat**.** Kemajuan teknologi digital yang dipadu dengan telekomunikasi telah membawa komputer memasuki masa-masa “revolusi”-nya. Di awal tahun 1970-an, teknologi PC atau Personal Computer mulai diperkenalkan sebagai alternatif pengganti mini computer. Dengan seperangkat komputer yang dapat ditaruh di meja kerja (desktop), seorang manajer atau teknisi dapat memperoleh data atau informasi yang telah diolah oleh komputer (dengan kecepatan yang hampir sama dengan kecepatan mini computer, bahkan mainframe). Kegunaan komputer di perusahaan tidak hanya untuk meningkatkan efisiensi, namun lebih jauh untuk mendukung terjadinya proses kerja yang lebih efektif. Tidak seperti halnya pada era komputerisasi dimana komputer hanya menjadi “milik pribadi” Divisi EDP (Electronic Data Processing) pada suatu perusahaan, di era kedua ini setiap individu di organisasi dapat memanfaatkan kecanggihan komputer, seperti untuk mengolah database, spreadsheet, maupun data processing (end-user computing). Pemakaian komputer di kalangan perusahaan semakin marak, terutama didukung dengan alam kompetisi yang telah berubah dari monompoli menjadi pasar bebas. Secara tidak langsung, perusahaan yang telah memanfaatkan teknologi komputer sangat efisien dan efektif dibandingkan perusahaan yang sebagian prosesnya masih dikelola secara manual.

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di Sun Microsystems, yang saat ini merupakan bagian dari Oracle dan dirilis tahun 1995. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada C dan C++ namun dengan sintaksis model objek yang lebih sederhana serta dukungan rutin-rutin aras bawah yang minimal. Aplikasi-aplikasi berbasis java umumnya dikompilasi ke dalam p-code (bytecode) dan dapat dijalankan pada berbagai Mesin Virtual Java (JVM). Java merupakan bahasa pemrograman yang bersifat umum/non-spesifik (general purpose), dan secara khusus didesain untuk memanfaatkan dependensi implementasi seminimal mungkin.

Sekolah mengemban tugas memberikan pelayanan pendidikan kepada masyarakat khususnya kepada anak – anak dibawah 18 tahun dengan lebih baik dan lebih propesional. Sekolah juga sebagai salah satu lembaga pendidikan, sekolah memiliki kewajiban untuk terus meningkatkan kinerja dan kualitas kerjanya. Publitas, khususnya siswa/i dan karyawan

Untuk memenuhi persyaratan tersebut, kami menyediakan aplikasi SISTEM INFORMASI AKADEMIK SEKOLAH ( SIAS ) berbasis aplikasi untuk mempermudah mengakses informasi dan menyebar luaskan ke tujuan dan memudahkan akses.

## **Rumusan Masalah**

1. Apa pengertian OPP?
2. Struktur apa saja yang terdapat di OPP?
3. Pengertian dan hasil pembuatan Use Case?
4. Pengertian dan hasil pembuatan Activity Diagram?
5. Pengertian dan hasil pembuatan Sequence Diagram?
6. Pengertian dan hasil pembuatan Class Diagram?

## **Tujuan dan Manfaat**

Tujuan dibuatnya aplikasi system sekolah berbasis Java Netbeans adalah untuk meringankan kinerja para staff dari sekolah tersebut agar lebih efisien dan mudah dalam memasukan nilai – nilai para murid. Dengan adanya aplikasi ini diharapkan dapat membantu:

* Pengelolaan data agar lebih rapih dan mengurai tingkat kehilangan data maupun kerusakan data
* Pengelolaan data berbasis NetBeans
* Menyediakan informasi lengkap terkait nilai – nilai siswa

# **BAB II**

**ISI**

1. OPP

OOP (Object Oriented Programming) adalah suatu metode pemrograman yang berorientasi kepada objek. Tujuan dari OOP diciptakan adalah untuk mempermudah pengembangan program dengan cara mengikuti model yang telah ada di kehidupan sehari-hari. Jadi setiap bagian dari suatu permasalahan adalah objek, nah objek itu sendiri merupakan gabungan dari beberapa objek yang lebih kecil lagi. Saya ambil contoh Pesawat, Pesawat adalah sebuah objek. Pesawat itu sendiri terbentuk dari beberapa objek yang lebih kecil lagi seperti mesin, roda, baling-baling, kursi, dll. Pesawat sebagai objek yang terbentuk dari objek-objek yang lebih kecil saling berhubungan, berinteraksi, berkomunikasi dan saling mengirim pesan kepada objek-objek yang lainnya. Begitu juga dengan program, sebuah objek yang besar dibentuk dari beberapa objek yang lebih kecil, objek-objek itu saling berkomunikasi, dan saling berkirim pesan kepada objek yang lain.

1. Struktur yang terdapat di OPP

### **Kelas**

Kelas merupakan kumpulan definisi dan fungsi data dalam suatu unit untuk tujuan tertentu. Misalnya ‘kelas anjing’ adalah unit yang terdiri dari definisi data dan fungsi yang merujuk pada berbagai jenis perilaku / turunan dari anjing. Kelas adalah dasar modularitas dan struktur dalam pemrograman berorientasi objek.

Kelas biasanya harus dikenali oleh non-programmer bahkan jika itu terkait dengan domain masalah yang ada dan kode yang terkandung dalam kelas harus (relatif) independen dan independen (karena kode digunakan jika tidak menggunakan OOP).

### **Objek**

Objek untuk embungkus data dan fungsi menjadi satu unit dalam program komputer; objek adalah dasar dari modularitas dan struktur dalam program komputer berorientasi objek.

### **Abstraction**

Prinsip selanjutnya yaitu abstraction*.*Prinsip ini sendiri berarti memungkinkam seorang developer memerintahkan suatu fungsi, tanpa harus mengetahui bagaimana fungsi tersebut bekerja.

Lebih lanjut, abstraction berarti menyembunyikan detail latar belakang dan hanya mewakili informasi yang diperlukan untuk dunia luar. Ini adalah proses penyederhanaan konsep dunia nyata menjadi komponen yang mutlak diperlukan. Seperti kala menggunakan *handphone*, kamu cukup memberikan suatu perintah, tanpa tahu bagaimana proses terlaksananya perintah tersebut.

### **Inheritance**

Inheritance dalam konsep OOP adalah kemampuan untuk membentuk *class* baru yang memiliki fungsi turunan atau mirip dengan fungsi yang ada sebelumnya.

Konsep ini menggunakan sistem hierarki atau bertingkat. Maksudnya, semakin jauh turunan atau *subclass*-nya, maka semakin sedikit kemiripan fungsinya.

### **Encapsulation**

Encapsulation atau pengkapsulan adalah konsep tentang pengikatan data atau metode berbeda yang disatukan atau “dikapsulkan” menjadi satu unit data.

Maksudnya, berbagai objek yang berada dalam *class* tersebut dapat berdiri sendiri tanpa terpengaruh oleh yang lainnya.

Encapsulation dapat mempermudah pembacaan kode. Hal tersebut terjadi karena informasi yang disajikan tidak perlu dibaca secara rinci dan sudah merupakan satu kesatuan.

Proses enkapsulasi mempermudah untuk menggunakan sebuah objek dari suatu kelas karena kita tidak perlu mengetahui segala hal secara rinci.

### **Polymorphism**

Prinsip terakhir dalam OOP adalah polymorphism. Pada dasarnya polymorphism adalah kemampuan suatu pesan atau data untuk diproses lebih dari satu bentuk.

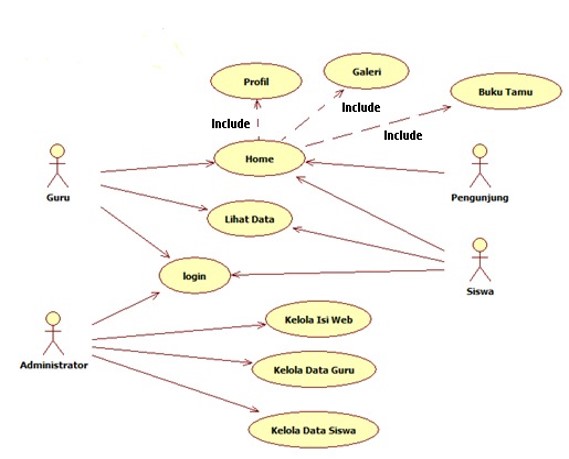
Salah satu ciri utama dari OOP adalah adanya polymorphism. Tanpa hal ini, suatu pemrograman tidak bisa dikatakan sebagai OOP.

Polymorphism sendiri adalah konsep di mana suatu objek yang berbeda-beda dapat diakses melalui interface yang sama.

Sebagai contoh, kamu memiliki fungsi untuk menghitung luas suatu benda, sementara benda tersebut berbentuk segitiga, lingkaran, dan persegi. Tentu, ketiga benda tersebut memiliki rumus perhitungan tersendiri.

Dengan polymorphism, kamu dapat memasukkan fungsi perhitungan luas ke tiga benda tersebut, dengan tiap benda memiliki metode perhitungannya sendiri. Ini tentu akan mempermudah perintah yang sama untuk beberapa *class* atau *subclass* tertentu.

1. Use Case Diagram



Use Case Diagram adalah satu jenis dari diagram UML (Unified Modelling Language) yang menggambarkan hubungan interaksi antara sistem dan aktor. Use Case dapat mendeskripsikan tipe interaksi antara si pengguna sistem dengan sistemnya. Use Case merupakan sesuatu yang mudah dipelajari. Langkah awal untuk melakukan pemodelan perlu adanya suatu diagram yang mampu menjabarkan aksi aktor dengan aksi dalam sistem itu sendiri, seperti yang terdapat pada Use Case.

1. Activity Diagram

Activity diagram atau dalam bahasa Indonesia berarti diagram aktivitas, merupakan sebuah diagram yang dapat memodelkan berbagai proses yang tejadi pada sistem. Seperti layaknya runtutan proses berjalannya suatu sistem dan digambarkan secara vertikal. Activity diagram adalah salah satu contoh diagram dari UML dalam pengembangan dari Use Case.

1. Sequence Diagram

Sequence diagram merupakan diagram yang menjelaskan interaksi objek berdasarkan urutan waktu. *Sequence* dapat menggambarkan urutan atau tahapan yang harus dilakukan untuk dapat menghasilkan sesuatu, seperti yang tertera pada Use Case diagram.

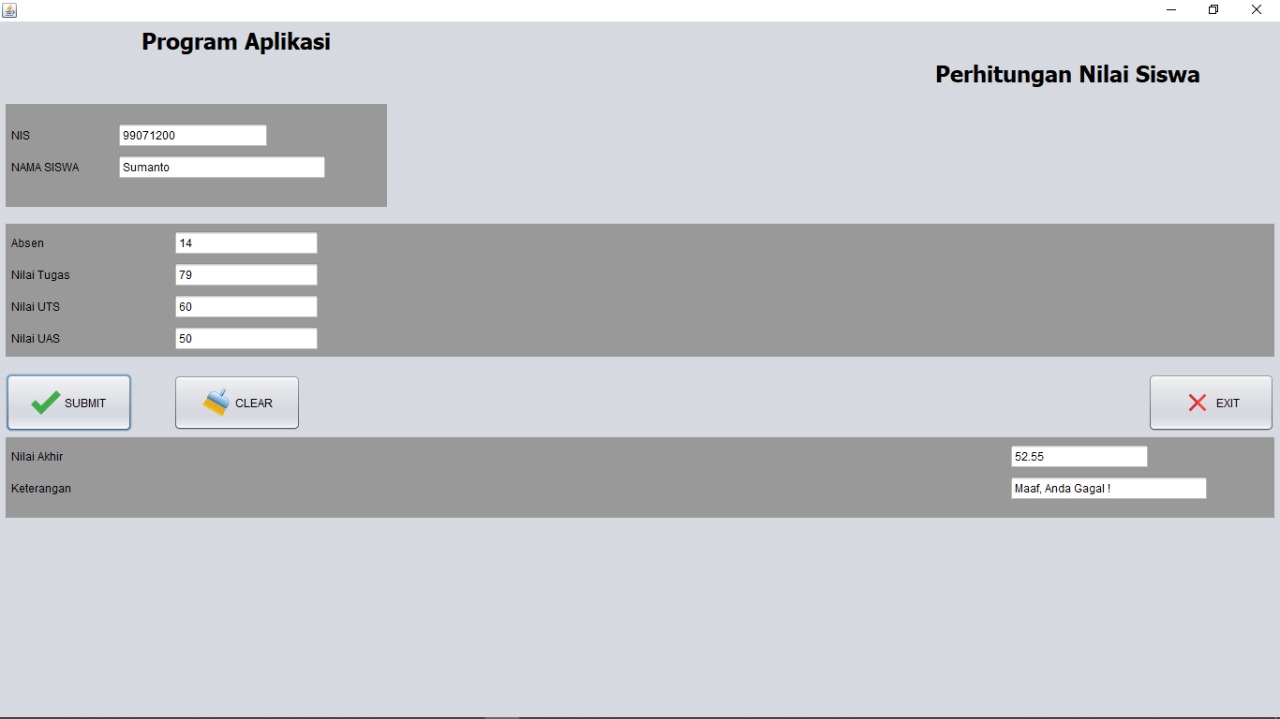
1. Class Diagram

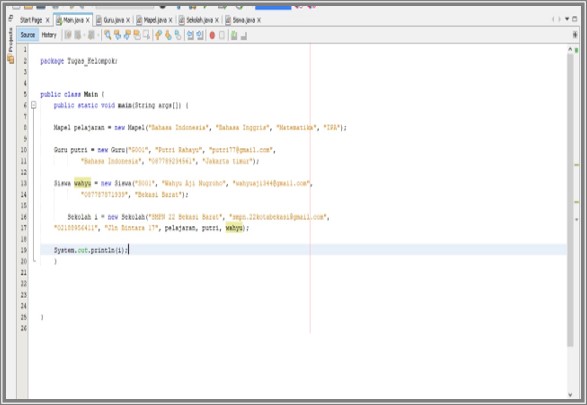
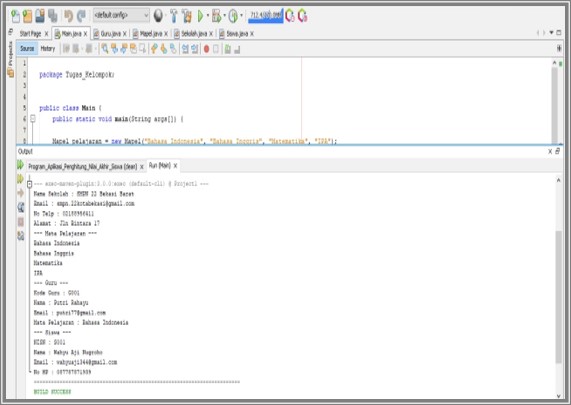
Class diagram atau diagram kelas merupakan suatu diagram yang digunakan untuk menampilkan kelas-kelas berupa pake-paket untuk memenuhi salah satu kebutuhan paket yang akan digunakan nantinya.

Namun, pada Class diagram desain modelnya dibagi menjadi 2 bagian. Class diagram yang pertama merupakan penjabaran dari domain model yang merupakan abstraksi dari basis data. Class diagram yang kedua merupakan bagian dari modul program MVC pattern (Model View Controller), di mana terdapat class boundary sebagai class interface, class control sebagai tempat ditemukannya algoritma, dan class entity sebagai tabel dalam basis data dan query program.

# **BAB III**

IMPLEMENTASI





# **Daftar Pustaka**

1. <https://www.dicoding.com/blog/apa-itu-uml/>
2. <https://idcloudhost.com/panduan/mengenal-pengertian-dan-konsep-oop/>
3. <https://glints.com/id/lowongan/oop-adalah/#.YeYW2_5BzIU>
4. <https://webcache.googleusercontent.com/search?q=cache:f8j2q90UIEEJ:https://osf.io/x5mfc/download+&cd=7&hl=id&ct=clnk&gl=sg>
5. <https://kodingin.com/apa-itu-pemrograman-visual/>