Jurnal Ilmiah Universitas Batanghari Jambi **e-ISSN**: 2549-4236, **p-ISSN**: 1411-8939
Lembaga Penellitian dan Pengabdian kepada Masyarakat
Volume 23(2), Juli 2023, 1206-1209
DOI: 10.33087/jiubj.v23i2.3189

Kelebihan dan Kelemahan dalam Penggunaan *Object Oriented Programming*

Eva Darwisah Harahap, Zulaiha Siregar
Jurusan Ilmu Komputer, Fakultas Sains dan Teknologi, Universitas Islam Negeri Sumatra Utara.
Correspondence: evadarwisah2002@gmail.com, zulaihasiregarzulaiha@gmail.com

Abstrak. Bahasa pemrograman umumnya telah banyak disediakan untuk membuat aplikasi, baik yang bernilai kecil ataupun besar yang berguna untuk menjawab berbagai kebutuhan pengguna lainnya, Pemrograman berorientasi objek adalah paradigma pemrograman yang berorientasikan kepada objek semua data dan fungsi yang ada didalamnya dibungkus dalam kelas kelas ataupun objek objek, dibandingkan dengan logika pemrograman terstruktur yang setiap objeknya dapat menerima pesan, memproses data, dan juga mengirim pesan ke objek lainnya, program berorientasi objek ditemukan pada tahun 1960-an disitu berawal dari suatu pembuatan program yang terstruktur dan selanjutnya metode ini dikembangkan dari bahasa C dan pascal.

Kata Kunci: pengertian Oop, kelemahan, kelebihan Oop

Abstrak. *In general, many programming languages have been provided for making applications, both of small and large value, which are useful for answering various other user needs. Object-oriented programming is an object-oriented programming paradigm, all data and functions contained in it are wrapped in classes or object objects. , compared to structured programming logic where each object can receive messages, process data, and also send messages to other objects, object-oriented programs were discovered in the 1960s there starting from a structured program creation and then this method was developed from C and Pascal.*

Keywords: *definition of Oop, weaknesses, strengths of Oop*

PENDAHULUAN

PBO merupakan paradigma pemrograman yang popular saat ini yang telah menggantikan teknik pemrograman berbasis prosedur. Object oriented programing yang berarti pula pemrograman beorientasi objek sudah ditemukan sekitar tahun 1960 dan dikembangkan pada permulaan tahun 1970. Pemrograman berorientasi objek (*object oriented programming* – OOP) merupakan pemrograman yang berorientasikan kepada objek, dimana semua data dan fungsi dibungkus dalam class-class atau object-object. Setiap objek dapat menerima pesan, memproses data, mengirim, menyimpan dan memanipulasi data. Beberapa objek berinteraksi dengan saling memberikan informasi satu terhadap yang lainnya. Masing-masing objek harus berisikan informasi mengenai dirinya sendiri dan dapat dihubungkan dengan objek yang lain. Pemrograman berorientasi objek berbeda dengan pemrograman prosedural yang hanya menggunakan satu halaman ke bawah untuk mengerjakan banyak perintah atau statement. Penggunaan pemrograman berorientasi objek sangat benyak sekali, contoh: JAVA, PHP, PERL, C#, COBOL, dan lainnya. Tujuan penelitian ini adalah untuk memahami pemrograman objek berorientasi, karakteristik objek, konsep dasar pbo dan evolusi teknik dasar pemrograman, dan fundamental bahasa pemrograman berorientasi objek atau oop, serta dapat memahami aplikasi *library* bahasa pemrograman java, aplikasi library (Douglas dan Covington, 1992)

METODE

Metode dalam penelitian ini adalah dengan berbagai referensi di internet ataupun buku-buku dan juga jurnal

HASIL

Object oriented programing (OOP) merupakan paradigma pemrograman yang popular saat ini yang telah menggantikan teknik pemrograman berbasis prosedur. *Object oriented programing* yang berarti pula Pemrograman Beorientasi Objek (PBO) sudah ditemukan sekitar tahun 1960 dan dikembangkan pada permulaan tahun 1970. Pemrograman berorientasi objek (*object oriented programming* – OOP) merupakan pemrograman yang berorientasikan kepada objek, dimana semua data dan fungsi dibungkus dalam objek-objek.

Setiap objek dapat menerima pesan, memproses data, mengirim, menyimpan dan memanipulasi data. (Nugroho, 2005),

Beberapa objek berinteraksi dengan saling memberikan informasi satu terhadap yang lainnya. Prinsip OOP ada 3 (tiga), yaitu: (1) encapsulation, adalah mekanisme pemrograman yang membungkus kode dan data yang dimanipulasi dan menjaganya supaya terhindar dari interferensi dan penggunaan yang tidak perlu. Salah satu caranya dengan membentuk objek. Sedangkan manfaat *encapsulation* adalah menyembunyikan implementasi detil sebuah class, dan menyediakan public methode memaksa pengguna untuk menggunakan *method* untuk mengakses data membuat kode lebih terpelihara; (2) *inheritance*, memungkinkan programer meletakkan *member* yang sama dalam satu *class* dan *class-class* lain dapat mewarisi member tersebut. *Class* yang mengandung *member* yang sama dari beberapa class lain dinamakan superclass atau parent class. *Class* yang mewarisi dinamakan *subclass* atau *child class*. *Inheritance* menghasilkan *class hierarchy*; dan (3) *polymorphisme*, artinya mempunyai banyak bentuk. Dua objek atau lebih dikatakan sebagai *polymorphic*, bila objek-objek itu mempunyai antar muka yang identik namun mempunyai perilaku-perilaku yang berbeda.

Programmer telah menulis bahasa pemrograman modern sejak tahun 1940. Bahasa pemrograman terdahulu mengharuskan programmer untuk bekerja dengan alamat memory dan membuat kode program yang berhubungan dengan bahasa mesin. Bahasa pemrograman yang baru lebih mudah digunakan dengan beberapa alasan diantaranya: baris perintah menggunakan bahasa yang lebih mudah dimengerti. Sebagai contoh: perintah untuk menampilkan keluaran, menggunakan kata kerja yang umum diantaranya seperti *output*, *write* atau *print*. Memungkinkan programmer untuk menggunakan nama dari lokasi memory dari pada alamat memory aslinya. Sebagai contoh, sebuah data yang akan disimpan pada memory komputer, programmer dapat menunjukkan lokasi memory komputer menggunakan notasi heksadesimal dari pada menggunakan alamat sesungguhnya yang membutuhkan digit yang panjang.

Sifat-sifat dari teknik pemrograman terstruktur diantaranya adalah (1) memuat teknik pemecahan masalah yang logis dan sistematis; (2) membuat algoritma yang efisien, efektif dan sederhana; (3) program disusun dengan logika yang mudah dipahami; (4) tidak menggunakan perintah GOTO; (5) biaya pengujian program relatif rendah; (6) memiliki dokumentasi yang baik; dan (7) biaya perawatan dan dokumentasi yang dibutuhkan relatif rendah.

Object oriented programming (OOP) adalah sebuah bahasa pemrograman yang memandang segala sesuatu menjadi sebuah objek. Paradigma dari OOP adalah menyelesaikan masalah dengan merepresentasikan masalah ke model objek. Karakteristik dari *object oriented programming* adalah: (dictio.id, 2017) dan (Jagoan Hosting, 2023)

1. *Class* dan Objek. Class dapat dikatakan sebagai cetak biru (*blue print*) dari sebuah objek ataupun beberapa objek. Class diartikan juga menggambarkan ciri-ciri objek secara umum. Sebagai contoh nokia, samsung, blackberry dan Motorola merupakan objek dari class handphone. Nokia dan objek yang lainnya mempunyai atribut (apa yang dimiliki oleh objek/*what they have*) yang sama diantaranya memiliki tipe, tahun rilis, dimensi, berat, fitur dan lain sebagainya. Selain atribut, objek-objek tersebut memiliki methode (apa yang bisa dikerjakan objek/*what they do*) yang sama diantaranya kunci tombol, menampilkan teks dilayar, kirim sms dan lain sebagainya.
2. *Encapsulation*/Enkapsulasi. Disebut juga dengan pembungkusan, yaitu melindungi program dan data yang sedang diolah agar tidak diakses secara sembarangan oleh program lain. Dalam java, dasar enkapsulasi adalah *class*. Variabel atau methode pada sebuah *class* tidak dapat diakses oleh *class* lain dengan menjadikan class tersebut bersifat private, atau menjadikan class tersebut bersifat protected, yang hanya bisa diakses oleh turunannya (*inheritance*) atau menjadikan class tersebut bersifat *public*, sehingga bisa diakses oleh sembarang class.
3. *Inheritance*/Inheritansi, Disebut juga dengan turunan. Prinsipnya adalah sebuah *class* dapat diturunkan dari *class* yang lain. *Class* yang menurunkan ke class lain disebut dengan *superclass*, *parent class* atau *base class* atau kelas induk, sedangkan class yang merupakan turunan disebut sebagai *subclass*, *child class* atau *derived class* atau class turunan. Class turunan secara otomatis memiliki sifat (*variable*) dan kelakuan (*behavior, m*ethod) yang dimiliki oleh super class-nya. Class turunan dapat menambahkan

fitur atau *behavior* dengan mendefinisikan suatu method di dalam class turunan tersebut.

4. *Polimorphism*e/Polimorfisme. Polimorfisme secara bahasa dapat diartikan dengan memiliki banyak bentuk. kegunaan dari polimorfisme adalah agar dapat mendefinisikan beberapa konstruktor atau metode dengan karakteristik yang berbeda-beda agar nantinya dapat digunakan untuk kasus-kasus yang berbeda. Method atau perilaku yang sama tapi implementasinya/caranya yang berbeda-beda

Kelebihan oop adalah sebagai berikut: (Rizan dkk, 2017)

1. *Parallel Development*. Ketika bekerja sama dengan tim, masing-masing programmer dapat membangun class-class secara individual. Dengan begitu, komponen yang sudah dibentuk kemudian dapat digabung menjadi satu kesatuan, hal ini tentu saja menghemat banyak waktu dibanding harus membangun class satu per satu.
2. *Reusable*. Jika sudah berhasil membangun class, bukan berarti class-class tersebut hanya dapat digunakan sekali dalam proyek itu saja. Dalam OOP, *class-class* yang sudah ada dapat digunakan kembali di *project* dan development lainnya. Baik langsung pakai atau hanya butuh sedikit perubahan, tetapi jauh lebih mudah daripada harus membangun class lagi
3. *Coding* lebih mudah diatur, OOP adalah metode yang memiliki basis coding terpusat dan mudah dikelola. Data-data yang sudah kamu miliki lebih mudah diakses ketika dibutuhkan. Proses ini juga meningkatkan keamanan programming karena terdapat sistem otentikasi dinilai cukup kuat.

Sementara kelemahan oop adalah tidak efisien, membutuhkan tingkat manajemen data yang tinggi, dan kemungkinan diplikasi

Ada beberapa aplikasi (IDE – *Integrated Development Environment*) yang dapat digunakan untuk mengimplementasikan bahasa java, diantaranya adalah sebagai berikut: *Notepad*, *Texplus*, *Editplus*, *Jcreator*, *NetBeans*, *Crimson Editor*, *Elipase*, *Bluej*, *Dr java*, *JDevelopor*.

Object merupakan sesuatu yang memiliki identitas (nama), pada umumnya juga memiliki data tentang dirinya maupun *object* lain dan mempunyai kemampuan untuk melakukan sesuatu dan bisa bekerja sama dengan objek lainnya. Pada dasarnya semua benda yang ada di dunia dapat dianggap sebagai sebuah *object*, rumah, mobil, sepeda motor, meja, dan komputer merupakan contoh-contoh *object* yang ada di dunia nyata. *Object* adalah implementasi dari *class*. Secara sederhananya, dapat dikatakan terdiri dari properti (atribut) dan method. Pemahaman singkatnya adalah: (a) objek merupakan sebuah objek khusus yang digunakan untuk mengakses anggota objek (atribut atau method) dari dalam objek itu sendiri; dan (b) apabila terdapat variabel lain yang memiliki nama yang sama dengan nama atribut di dalam objek, "this" dapat digunakan untuk merujuk ke atribut objek

Setiap object memiliki dua karakteristik yang utama, yaitu atribut dan *behavior*. Atribut merupakan status object dan behavior merupakan tingkah laku dari object tersebut. Contoh sederhana adalah sepeda. Object sepeda memiliki atribut : pedal, roda, dan gigi, kecepatan serta memiliki tingkah laku seperti pacu dan rem. Sama seperti penggambaran di dunia nyata, pengembangan program dengan konsep OOP juga menggunakan model object yang akan menyimpan atributnya dalam variabel dan tingkah lakunya dalam method atau fungsi-fungsi/prosedur (Binus.ac.id., 2017)

SIMPULAN

PBO merupakan paradigma pemrograman yang popular saat ini yang telah menggantikan teknik pemrograman berbasis prosedur. *Object oriented programing* yang berarti pula Pemrograman Beorientasi Objek sudah ditemukan sekitar tahun 1960 dan dikembangkan pada permulaan tahun 1970 atau *object oriented programmin*g – OOP) merupakan pemrograman yang berorientasikan kepada objek, dimana semua data dan fungsi dibungkus dalam *class-class* atau *object-object*. Setiap *object* dapat menerima pesan, memproses data, mengirim, menyimpan dan memanipulasi data. beberapa object berinteraksi dengan saling memberikan informasi satu terhadap yang lainnya. Sifat-sifat dari teknik pemrograman terstruktur diantaranya adalah (1) memuat teknik pemecahan masalah yang logis dan sistematis; (2) membuat algoritma yang efisien, efektif dan sederhana; (3) program disusun dengan logika yang mudah dipahami; (4) tidak menggunakan perintah GOTO; (5) biaya pengujian program relatif rendah; (6) memiliki dokumentasi yang baik; dan (7) biaya perawatan dan dokumentasi yang dibutuhkan relatif rendah.

DAFTAR PUSTAKA

Binus.ac.id., 2017. *Mengidentifikasi Objek*, diakases pada website https://sis.binus.ac.id/2017/09/07/mengidentifikasi-object/,

Dictio.id., 2017. *Apakah Prinsip Pemrograman Berbasis Objek atau Oop?,* diakses pada website https://www.dictio.id/t/apakah-prinsip-pemrograman-berbasis-objek-atau-oop/13304/2

Douglas, Downing dan Michael Covington. 1992. *Kamus Istilah Komputer*. Jakarta: Erlangga.

Jagoanhosting.com, 2023, *Pengertian Oop (Object Oriented Programing) dan 4 Prinsipnya*, diakses pada website https://www.jagoanhosting.com/blog/oop-adalah/

Nugroho, Adi. 2005. *Analisis dan Perancangan Sistem Informasi Dengan Metodologi Berorientasi Objek*. Bandung: Informatika.

Rizan, Okkita., Anubhakti, Dian., Darmawan, Putu., 2017. *Pemrograman Berorientasi Objek*. Palembang : Tunas Gemilang Press

smkn1panjalu.sch.id, 2020, *Pengertian Pemrograman Berorientasi Objek (PBO)*, diakses pada website https://smkn1panjalu.sch.id/pengertian-pemrograman-berorientasi-objek-pbo/#:~:text=Pemrograman%20berorientasi%20objek%20ditemukan%20pada,begitu%20sulit%20dengan%20lebih%20mudah