**PERANCANGAN SISTEM BANK**

Logo

Description automatically generated

Dibuat oleh:

Abdullah Farhan (2019230100)

Adi Firmansyah (2019230083)

Dwi Putro Sulistyo (2019230004)

M. Ananda Rizky Audriansyah (2018230220)

# PROGRAM STUDI TEKNOLOGI INFORMASI FAKULTAS TEKNIK

**UNIVERSITAS DARMA PERSADA JAKARTA**

# 2021

# KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Tuhan Yang Maha Esa karena berkat limpahan rahmat dan karunia-Nya sehingga penulis dapat menyusun makalah tentang *Sistem Perbankan* ini dapat diselesaikan dengan tepat waktu. Adapun tujuan dari penulisan makalah ini adalah untuk memenuhi Tugas Akhir pada mata kuliah Pemrograman Berorientasi Objek yang diberikan oleh Bapak Aji Setiawan, S.Kom, M.Si.

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, penulis 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 Sistem Bank

# 

# DAFTAR ISI

[KATA PENGANTAR 2](#_TOC_250015)

[DAFTAR ISI 3](#_TOC_250014)

[BAB I PENDAHULUAN 4](#_TOC_250013)

1. [Latar Belakang 4](#_TOC_250012)
2. [Rumusan Masalah 7](#_TOC_250011)
3. [Tujuan Membuat Makalah 8](#_TOC_250010)
4. [Manfaat Makalah 8](#_TOC_250009)

[BAB II METODOLOGI 9](#_TOC_250008)

1. Observasi 9
2. [Studi Pustaka 10](#_TOC_250007)

[BAB III HASIL 23](#_TOC_250003)

1. [Analisis Kebutuhan 23](#_TOC_250002)
2. [Analisis Sistem 24](#_TOC_250001)

[BAB III KESIMPULAN DAN REFERENSI 23](#_TOC_250003)

1. [Kesimpulan 23](#_TOC_250002)
2. [Referensi 24](#_TOC_250001)

[DAFTAR PUSTAKA 50](#_TOC_250000)

# BAB I PENDAHULAN

# Latar Belakang

Perubahan teknologi berkembang dengan pesat, hal ini ditandai dengan banyak kemudahan yang didapatkan seiring kemajuan teknologi, tidak terkecuali Bank dalam memberikan pelayanan terhadap para nasabahnya. Perubahan yang dirasakan oleh para nasabah dibandingkan beberapa tahun terakhir dalam melakukan transaksi di Bank terdapat beberapa keuntungan dalam pengambilan uang tunai. Jika dulu, para nasabah harus sabar mengantri apabila akan melakukan transaksi dalam pengambilan uang tunai, maka dengan adanya Automatic Teller Machine (ATM) para nasabah tidak harus menunggu waktu yang cukup lama dalam pengambilan uang tunai, dikarenakan pihak Bank telah menyediakan banyak tempat ATM untuk memudahkan transaksi dalam pengambilan uang tunai.

Dalam rekayasa perangkat Lunak terdapat tahapan-tahapan sebelum memulai membuat sebuah perangkat lunak. Salah satu tahapan yang penting adalah membuat perancangan terhadap perangkat lunak yang akan dikerjakan. perancangan disini sangat penting, karena selain mempengaruhi bagaimana perangkat lunak akan dibuat juga mempermudah ketika ada revisi dari pihak klien.

Perancangan model sendiri terbagi menjadi dua, yaitu perancangan struktural dan perancangan OO (Object Oriented). Kedua jenis perancangan tersebut tidak terlepas dari paradigma pemrograman yang ada sekarang yaitu paradigma pemrograman struktural dan paradigma pemrograman OO.

# Konsep Pemrograman Berorientasi Objek2

Pemrograman Berorientasi Objek atau OOP didukung dengan bahasa pemrograman yang bersifat berbayar (lisensi) dan juga open source (free). Bahasa Java (Netbeans 8.2 hingga versi terbaru) merupakan bahasa pemrograman yang fully object oriented programming dan bersifat open source. Selain Java dan C++ ada juga Python, PHP5, Syphony, CI, Ruby, VB .NET, C#, dan ASP .NET untuk digunakan pada Pemrograman Berorientasi Objek.

Dalam pemrograman berorientasi objek, suatu sistem dibungkus (encapsulate) menjadi kelompok data dan fungsi yang dapat mewarisi atribut dan sifat dari komponen lainnya dan komponen-komponen tersebut saling berinteraksi satu sama lain.

OOP memandang perangkat lunak sebagai sebuah interaksi antar bagian dalam sebuah sistem. Beberapa karakteristik dalam OOP yaitu:

* + 1. Inheritance (pewarisan)

Pewarisan atau inheritance merupakan proses penciptaan kelas baru (subclass/kelas turunan) dengan mewarisi karakteristik dari kelas yang telah ada (superclass/kelas induk). Child class dapat memiliki karakteristik baru yang tidak dimiliki oleh superclass. Kelas turunan akan mewarisi anggota-anggota suatu kelas berupa data (atribut) dan fungsi (operasi), pada kelas turunan memungkinkan penambahan data serta fungsi yang baru.

* + 1. Encapsulation (pembungkusan)

Encapsulation (hiding) merupakan proses pemaketan data objek bersama method-methodnya. Encapsulation bermanfaat untuk menyembunyikan rincian-rincian implementasi dari pemakai/objek lain yang tidak berhak. Encapsulation memproteksi suatu proses dari kemungkinan interfensi atau penyalahgunaan dari luar sistem.

Fungsi Encapsulation adalah:

* + - 1. Menyembunyikan data

Objek disusun dari interface public method dan private data. Bermanfaat untuk mengubah bagian internal tanpa mempengaruhi bagian-bagian program yang lain.

* + - 1. Modularitas

Objek dapat dikelola secara independen. Karena kode sumber bagian internal objek dikelola secara terpisah dari interface, maka dapat dengan bebas melakukan modifikasi yang tidak menyebabkan masalah pada bagian- bagian lain. Manfaat modularitas adalah mempermudah mendistribusikan objek-objek di sistem.

* + 1. Polymorphisms

Objek yang memiliki perilaku (behaviour) yang sama dapat memiliki lebih dari satu interpretasi dalam konteks yang berbeda. Konsep ini memungkinkan digunakannya interface yang sama untuk memerintah suatu objek agar dapat melakukan aksi atau tindakan yang mungkin secara prinsip sama tapi secara proses berbeda. Mekanisme Polymorphism dapat dilakukan dengan beberapa cara yakni, Overloading Method, Overloading Constructor, dan Overriding Method.

* + 1. Abstraction (Abstraksi)

Abstraksi adalah menemukan hal-hal yang esensial pada suatu objek dan mengabaikan hal-hal yang sifatnya insidental. Fokus dengan menentukan apa ciri-ciri (atribut) yang dimiliki oleh objek tersebut serta apa saja yang bisa dilakukan (metode/perilaku) oleh objek tersebut.

Kelebihan paradigma OOP adalah sebagai berikut:

1. Code Reuse and Recycling

Objek yang dibuat dalam OOP dapat dengan mudah digunakan dalam program lain.

1. Abstraction

Ketika suatu objek dibuat, user hanya mendapatkan konsep yang familiar, sedangkan tingkat kerumitan dalam objek tidak ditampilkan pada user.

1. Data hiding

Objek-objek memiliki kemampuan untuk menyembunyikan beberapa bagian tertentu, hal ini dapat membantu para programmer untuk menjaga keutuhan kode program yang sudah dirancang dengan benar.

1. Design benefit

Program dengan OOP akan membuat designer merancang penuh saat perencanaan, yang memungkinkan akan diperoleh suatu perancangan yang lebih baik.

1. Software maintenance

Program yang ditulis dengan OOP jauh lebih mudah dilakukan modifikasi dan pemeliharaan.

Kelemahan paradigma OOP:

1. Size/ukuran

Program yang dibangun dengan konsep OOP akan menghasilkan ukuran yang jauh lebih besar daripada program-program lainnya.

1. Effort/usaha

Prorgam yang dibangun dengan konsep OOP memerlukan usaha yang lebih kompleks untuk pengerjaannya.

1. Speed/kecepatan

Program yang dibangun dengan konsep OOP lebih lambat daripada program- program lainnya. Hal ini karena ukuran program dengan OOP yang besar dan memerlukan sumber-sumber sistem.

# JAVA3

Java adalah sebuah bahasa pemrograman yang diciptakan oleh James Gosling, seorang developer dari Sun Microsytem pada tahun 1991. Selanjutnya Java dikembangkan Sun Microsystem dan banyak digunakan untuk menciptakan Executable Content yang dapat didistribusikan melalui network. Java adalah bahasa pemrograman Object-Oriented dengan unsur-unsur seperti bahsa C++ dan bahasa- bahasa lainnya yang memiliki libraries yang cocok untuk lingkunan internet.

# Netbeans4

Netbeans adalah Integrated Development Environtment (IDE) berbasis java dari Sun Microsystem yang berjalan di swing. Swing sebuah teknologi Java untuk pengembangan aplikasi Desktop yang dapat berjalan diberbagai macam platform seperti Windows, Linux, Mac OS X an Solaris. Suatu IDE adalah lingkup programan yang diintegrasikan kedalam suatu aplikasi perangkat lunak yang menyediakan pembangunan Graphical User Interface (GUI), suatu text atau kode editor, suatu compiler atau interpreter dan suatu debugger. Software Netbeans sebagai platform memungkinkan berbagai aplikasi java untuk dikembangkan dari suatu set modular komponen software yang disebut modul.

# Rumusan Masalah

1. Apa itu Pemrograman Berorientasi Objek?
2. Apa bahasa yang digunakan pada Pemrograman Berorientasi Objek?
3. Bagaimana konsep Pemrograman Berorientasi Objek?
4. Bagaimana cara membuat system bank mengunakan Java (OOP)?

# Tujuan Membuat Makalah

Tujuan pembuatan makalah ini yaitu:

1. Membuat system bank menggunakan Java (OOP)
2. Memahami konsep dasar dari OOP.
3. Menghasilkan sistem, aplikasi atau produk berkualitas dalam kerangka waktu yang dapat diterima oleh pengguna.

# Manfaat Makalah

1. Dapat dijadikan refrensi untuk membuat system berbasis objek.
2. Memudahkan para pengguna dalam mengakses system.
3. Dapat memberikan informasi tentang tata cara Pemrograman Berorientasi Objek.

# BAB II

# METODOLOGI PENELITIAN

1. **Metode Pengumpulan Data**

# Observasi

Metode yang dilakukan untuk mengindentifikasi data yang dilakukan secara sistematis baik dengan cara memperhatikan secara langsung atau tidak langsung objek yang diteliti, sehingga tidak ada data yang terlewatkan dalam Sistem Bank.

# Studi Pustaka

Menghimpun informasi yang relevan dengan topik dan masalah yang akan atau sedang di teliti untuk memperoleh data dengan literatur yang berisi teori-teori yang berkaitan dengan permasalahan yang di bahas atau sebagai sumber data termasuk data publik dan data yang berasal dari internet, dan dokumen-dokumen pendukung lainnya.

# BAB III

# HASIL DAN PEMBAHASAN

1. **Analisa Kebutuhan**

Adanya media ATM sebagai alat transaksi pengambilan atau pengiriman uang oleh para nasabah membuat lebih memudahkan dalam bertransaksi. Namun kemudahan dalam bertransaksi tersebut masih minim, oleh karena nasabah sebagai pengguna sehingga kurang optimal dalam pemakaiannya.

Untuk memenuhi kebutuhan user mengenai kemudahan dalam bertransaksi di Bank. Pembuatan system dengan mengunakan Bahasa java berorintasi objek diharapkan dapat membantu nasabah dalam setor dan Tarik tunai di bank

1. **Analisis Sistem**

Berdasarkan sistem berjalan bahwa proses transaksi membutuhkan sistem yang berbasis objek *(OO)* sehingga dapat memudahkan nasabah dalam pengecekan saldo, tarik tunai dan deposit, dan juga mempermudah dalam pembuatan akun rekening bank.

# Sistem Berjalan

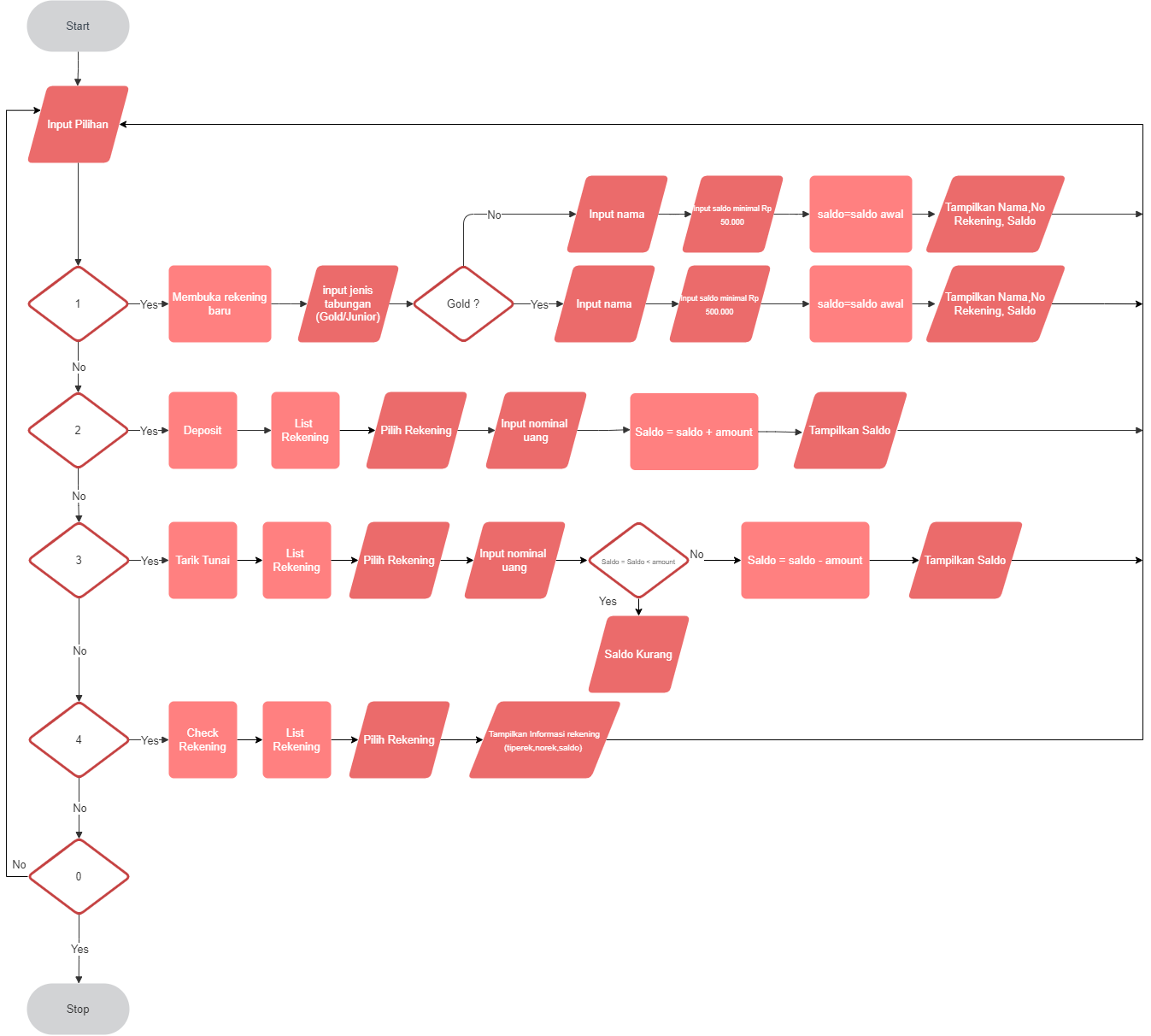
1. Nasabah datang ke Bank menemui bagian *Customer Service* (CS) untuk memberitahu pembukaan akun rekening bank. Sebelum itu nasabah harus mengambil nomor antrian. Setelah dipanggil CS akan memberikan ke nasabah satu lembar formulir untuk menciptakan nomor CIF (Customer Identification File). Nasabah harus mengisi formulir tersebut untuk pembukaan rekening.
2. Setelah itu nasabah dapat membuka akun rekening bank.
3. Setelah akun bank jadi, nasabah juga bisa memasukan setor tunai/deposit melalui mesin ATM.
4. Selain itu nasabah juga bisa menarik tunai dengan fitur yang tersedia dalam sistem.
5. System juga menyediakan fitur check daftar rekening untuk memudahkan pelanggan melakukan transaksi
   1. ***Use Case* Diagram**

Chart, diagram

Description automatically generated

1. Nasabah, dapat melakukan Buka rekening tabungan.
2. Nasabah dapat melakukan login.
3. Nasabah dapat melakukan deposit/setor tunai jika sudah selesai melakukan setor tunai dapat logout.
4. Nasabah dapat mengakses transaksi dan menampilkan beberapa menu, diantaranya data nasabah, daftar rekening nasabah, data user menampilkan data admin dan data nasabah
5. Nasabah dapat melihat rincian transaksi, jika sudah logout.

***b) Flowchart***



1. Menu Utama

Text

Description automatically generated

1. Buku Rekening

Text

Description automatically generated

1. Deposit

Text

Description automatically generated

1. Tarik Tunai

Text

Description automatically generated

1. Tampilan List Rekening

Text

Description automatically generated

**BAB III**

**KESIMPULAN**

Berdasarkan uraian dan hasil analisa yang telah dilakukan selama pengerjaan Sistem Bank Pemrograman Berorientasi Objek Dengan Bahasa Java ini, penulis mengambil kesimpulan sebagai berikut :

1. Dengan adanya Sistem Bank Pemrograman Berorientasi Objek Dengan Bahasa Java diharapkan nasabah menjadi lebih mudah dalam melakukan transaksi.

# DAFTAR PUSTAKA

Iskandar, Ade Rahmat. 2020. *Menguasai Pemrograman Berorientasi Objek*. Jakarta: Informatika Bandung.

Siahaan, Vivian dan RH. Sianipar. *Tutorial Pemrograman JAVA untuk Pemula*. Sumatera Utara: Balige Publishing.

Fajar. “Sejarah Object Oriented Programming”. *Belajar C++*. <https://www.belajarcpp.com/tutorial/cpp/sejarah-oop/> (diakses 7 Desember 2021).

Tidak diketahui. “Pengertian Variabel dan 3 Jenis Variabel pada Pemrograman Java”.

*Gotutorid.* <https://www.gotutorid.com/java/java-variable/>(diakses 10 Desember 2021).

Tidak diketahui. “Pengertian Tipe Data Non-Primitif pada Java | Reference data type”. *Gotutorid.* <https://www.gotutorid.com/java/tipe-referensi-java/> (diakses 12 Desember 2021).

Sri Astuti, Suendri. "Perancangan Sistem Parkir Mall dengan Konsep Object Oriented Programming (OOP) Menggunakan Bahasa Pemrograman Java dengan Editor Netbeans 8.2." *Jurnal Sistem Informasi*, 1-7.

Destiningrum, Mara dan Qadhli Jafar Adrian, 2017, ***Sistem Informasi Penjadwalan Dokter Berbasis Web Dengan Menggunakan Framework Codeigniter (StudiKasus: Rumah Sakit Yukum Medical Center),*** Jurnal TEKNOINFO, 11(2).