

Pemrograman Menggunakan Java NetBeans

Oktaviyanus¹, Mugi yanto¹, Panji Eky Bima¹, Muhammad Bahron¹, Ines Heidiani Ikasari¹

¹Ilmu Komputer, Teknik Informatika, Universitas Pamulang, Kota Tangerang, Indonesia.

Email : oktaviyanus02@gmail.com, mugi.goki199@gmail.com, bimabsaku@gmail.com,
muhamadbaron21@gmail.com, dosen01374@unpam.ac.id

Abstrak - Pemrograman menggunakan Java NetBeans merupakan pendekatan yang efektif dalam pengembangan perangkat lunak berbasis Java. Java NetBeans adalah integrated development environment (IDE) yang menyediakan berbagai fitur dan fasilitas untuk mempermudah proses pengkodean, pengujian, dan penyebaran aplikasi Java. Abstrak ini membahas penerapan pemrograman Java NetBeans dalam konteks pengembangan perangkat lunak yang modular, dapat diandalkan, dan efisien. Penelitian ini fokus pada analisis dan implementasi berbagai aspek pemrograman Java NetBeans, termasuk manfaat penggunaan framework GUI, kelebihan modularitas, dan integrasi dengan basis data. Kami juga mengeksplorasi pendekatan-pendekatan terbaru dalam pengembangan perangkat lunak Java menggunakan NetBeans, termasuk konsep-konsep seperti multithreading, desain antarmuka yang responsif, dan keamanan aplikasi. Hasil penelitian ini memberikan wawasan mendalam tentang potensi dan keterbatasan penggunaan Java NetBeans dalam konteks pemrograman. Dengan memanfaatkan fitur-fitur kunci yang ditawarkan oleh Java NetBeans, pengembang dapat meningkatkan produktivitas dan kualitas perangkat lunak yang dihasilkan.

Kata Kunci : *Java NetBeans, Integrated Development Environment (IDE), Software Development, GUI Framework, Modular Programming, Database Integration, Java Persistence API (JPA), JDBC (Java Database Connectivity).*

Abstract - Programming using Java NetBeans is an effective approach in developing Java-based software. Java NetBeans is an integrated development environment (IDE) that provides various features and facilities to simplify the process of coding, testing, and deploying Java applications. This abstract discusses the application of Java NetBeans programming in the context of modular, reliable and efficient software development. This research focuses on the analysis and implementation of various aspects of Java NetBeans programming, including the benefits of using a GUI framework, the advantages of modularity, and integration with databases. We also explore the latest approaches to Java software development using NetBeans, including concepts such as multithreading, responsive interface design, and application security. The results of this research provide deep insight into the potential and limitations of using Java NetBeans in a programming context. By taking advantage of the key features offered by Java NetBeans, developers can increase productivity and the quality of the resulting software.

Keywords : *Java NetBeans, Integrated Development Environment (IDE), Software Development, GUI Framework, Modular Programming, Database Integration, Java Persistence API (JPA), JDBC (Java Database Connectivity)*

1 PENDAHULUAN

Di era sekarang ini, perkembangan teknologi informasi yang semakin pesat, pengembangan perangkat lunak menjadi salah satu aspek yang krusial dalam mendukung berbagai aktivitas dan kebutuhan masyarakat. Java, sebagai salah satu bahasa pemrograman yang populer, menjadi landasan utama dalam proses pembangunan aplikasi yang handal dan efisien. Untuk mendukung pengembangan perangkat lunak berbasis Java, banyak pengembang memanfaatkan integrated development environment (IDE) seperti Java NetBeans.

Java NetBeans, sebagai IDE yang terkenal dan matang, menyediakan sejumlah fitur dan fasilitas yang mendukung para pengembang dalam menangani kompleksitas pengkodean, menguji fungsionalitas, dan mendistribusikan aplikasi dengan lebih mudah. Penelitian ini bertujuan untuk

menjelajahi aplikasi praktis pemrograman Java NetBeans dalam konteks pengembangan perangkat lunak yang modular, dapat diandalkan, dan efisien

Fokus utama penelitian ini adalah menganalisis dan mengimplementasikan berbagai aspek dari pemrograman Java NetBeans, termasuk manfaat penggunaan framework antarmuka grafis (GUI), keunggulan dalam modularitas, serta integrasi yang lancar dengan basis data. Selain itu, penelitian ini juga akan mengeksplorasi pendekatan-pendekatan terkini dalam pengembangan perangkat lunak berbasis Java melalui perspektif NetBeans, mencakup konsep-konsep seperti multithreading, desain antarmuka yang responsif, dan keamanan aplikasi.

Hasil dari penelitian ini diharapkan dapat memberikan wawasan mendalam terkait potensi dan keterbatasan penggunaan Java NetBeans dalam konteks pemrograman. Dengan memahami fitur-fitur kunci yang ditawarkan oleh Java NetBeans, diharapkan para pengembang dapat meningkatkan produktivitas serta kualitas perangkat lunak yang dihasilkan. Penelitian ini diharapkan dapat memberikan sumbangan berharga dalam memahami landasan pemrograman Java NetBeans dan memberikan panduan praktis bagi para pengembang yang ingin mengoptimalkan upaya pengembangan perangkat lunak mereka.

2 METODE PENELITIAN

2.1 Studi Literatur

Studi literatur dilakukan untuk mengumpulkan informasi terkini tentang pengembangan perangkat lunak berbasis Java menggunakan Java NetBeans. Sumber-sumber literatur mencakup buku, jurnal ilmiah, artikel, dan dokumentasi resmi Java NetBeans. Tujuan dari studi literatur adalah untuk memahami konsep-konsep dasar, perkembangan terkini, dan praktik terbaik dalam pengembangan perangkat lunak Java.

2.2 Analisis Fitur Java NetBeans

Pada tahap ini, dilakukan analisis mendalam terhadap fitur-fitur kunci yang ditawarkan oleh Java NetBeans. Fokus analisis mencakup fitur-fitur yang mendukung modularitas, integrasi dengan basis data, dan keamanan aplikasi. Hasil analisis ini akan menjadi dasar untuk implementasi dalam tahap selanjutnya.

2.3 Implementasi Praktis

Implementasi praktis dilakukan untuk mengaplikasikan konsep-konsep yang telah dianalisis sebelumnya. Hal ini mencakup pengembangan perangkat lunak menggunakan Java NetBeans dengan memanfaatkan fitur-fitur kunci yang relevan. Penerapan modularitas, desain antarmuka yang responsif, dan aspek keamanan menjadi fokus utama dalam pengembangan perangkat lunak.

2.4 Eksperimen dan Pengujian

Pada tahap eksperimen, dilakukan pengujian terhadap perangkat lunak yang dikembangkan. Pengujian mencakup uji fungsionalitas, uji modularitas, dan uji keamanan aplikasi. Hasil pengujian digunakan untuk mengevaluasi performa perangkat lunak dan mengidentifikasi potensi perbaikan atau pengembangan lebih lanjut.

2.5 Analisis Hasil

Hasil dari implementasi dan pengujian dievaluasi dan dianalisis secara mendalam. Hasil analisis ini mencakup kelebihan dan keterbatasan dari penggunaan Java NetBeans dalam konteks pengembangan perangkat lunak. Temuan ini digunakan untuk menyusun kesimpulan dan rekomendasi bagi pengembang perangkat lunak.

2.6. Penyusunan Kesimpulan dan Rekomendasi

Berdasarkan hasil analisis, disusun kesimpulan yang mencakup temuan utama dari penelitian ini. Selain itu, diberikan rekomendasi untuk pengembang perangkat lunak yang ingin memanfaatkan Java NetBeans secara optimal dalam pengembangan perangkat lunak mereka.

3 ANALISA DAN PEMBAHASAN

3.1 Java

Java adalah sebuah bahasa pemrograman tingkat tinggi yang bersifat object-oriented, platform-independent, dan general-purpose. Dikembangkan pertama kali oleh James Gosling pada tahun 1991, Java dirancang agar dapat berjalan di berbagai platform tanpa perlu kompilasi ulang. Salah satu konsep utama dalam Java adalah portabilitas, yang memungkinkan program Java dapat dijalankan di berbagai sistem operasi tanpa modifikasi. Java juga dikenal karena penggunaan mesin virtual (Java Virtual Machine atau JVM), yang memungkinkan eksekusi kode Java di lingkungan yang terisolasi dari sistem operasi host.

Java digunakan secara luas dalam pengembangan aplikasi berbagai jenis, termasuk aplikasi desktop, perangkat lunak server, aplikasi seluler (Android), dan pengembangan web (JavaServer Pages).

3.2 NetBeans

NetBeans merupakan sebuah Integrated Development Environment (IDE) yang digunakan untuk pengembangan perangkat lunak berbasis Java. NetBeans menyediakan lingkungan yang terintegrasi untuk kebutuhan pengembangan, mulai dari penulisan kode hingga pengujian dan penyebaran aplikasi. NetBeans mencakup berbagai fitur seperti editor kode yang cerdas, debugger, manajemen proyek, dan integrasi dengan berbagai alat pengembangan.

Fitur utama NetBeans antara lain adalah kemampuannya dalam pengembangan aplikasi desktop, web, dan perangkat lunak berbasis Java Enterprise Edition (EE). NetBeans juga mendukung pengembangan aplikasi untuk platform mobile, termasuk Android.

Dengan antarmuka pengguna yang intuitif dan dukungan untuk berbagai teknologi Java, NetBeans menjadi pilihan populer di kalangan pengembang Java untuk meningkatkan produktivitas dan efisiensi dalam proses pengembangan perangkat lunak.

3.3 Penggunaan Java NetBeans

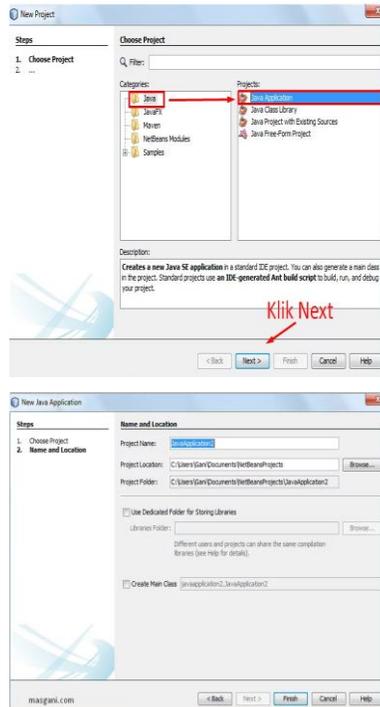
Penggunaan Java NetBeans dalam pengembangan perangkat lunak memberikan berbagai keuntungan kepada para pengembang. Sebagai Integrated Development Environment (IDE) yang khusus dirancang untuk bahasa pemrograman Java, NetBeans menyederhanakan langkah-langkah pengkodean, pengujian, dan penyebaran aplikasi.

Keunggulan utamanya termasuk dukungan untuk berbagai jenis pengembangan, mulai dari aplikasi desktop hingga web, dan perangkat lunak berbasis Java Enterprise Edition (EE). Kemampuan NetBeans dalam mengintegrasikan framework GUI mempermudah pengembangan antarmuka pengguna yang responsif melalui fitur drag-and-drop.

Selain itu, pendekatan modular dalam pengembangan perangkat lunak dengan NetBeans meningkatkan keterbacaan dan struktur kode. NetBeans juga memberikan dukungan untuk integrasi dengan basis data, mendukung keamanan aplikasi, serta memfasilitasi pengembangan aplikasi mobile, khususnya untuk platform Android.

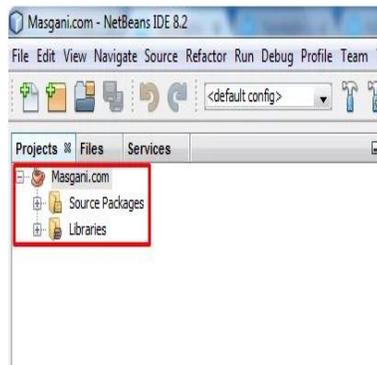
Berikut adalah Langkah-langkahnya cara menggunakan NetBeans IDE

a. Pembuatan Project

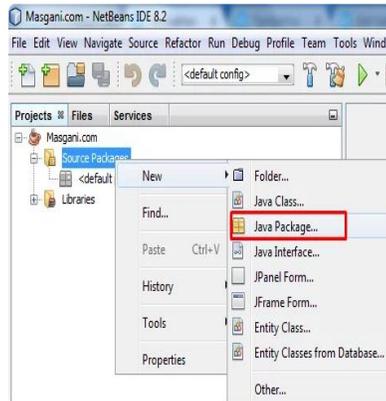


- Project Name adalah nama untuk proyek yang akan dibuat, silahkan beri nama sesuai keinginan Anda.
- Project Location adalah lokasi di mana proyek akan di simpan.
- Use Dedicated Folder for Storing Libraries biarkan seperti bawaannya.
- Create Main Class: Jika opsi ini dipilih, maka secara otomatis akan membuat kelas utama bersamaan dengan proyek baru.

Jika sudah selesai memberi nama projectnya, selanjutnya klik tombol Finish. Berikut adalah tampilan Projectnya :



b. Membuat Package



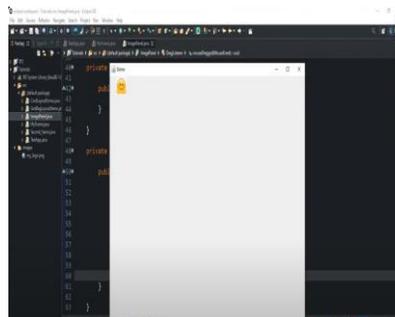
Jika sudah di klik, maka akan muncul tampilan window untuk memberi detail nama package, setelah di isi lalu klik Finish.

3.4 Framework NetBeans GUI

Framework NetBeans GUI adalah kerangka kerja (framework) yang disediakan oleh Integrated Development Environment (IDE) NetBeans untuk memudahkan pengembangan antarmuka pengguna (GUI) dalam aplikasi Java. Framework ini dirancang untuk membantu pengembang membangun antarmuka pengguna secara visual tanpa harus menulis kode secara manual. Berikut adalah beberapa fitur dan konsep kunci terkait Framework NetBeans GUI:

a. Drag-and-Drop Interface

"Drag and drop" (tarik dan lepas) pada NetBeans adalah fitur yang memungkinkan pengembang untuk dengan mudah menambahkan komponen atau elemen ke dalam proyek atau formulir aplikasi dengan cara menariknya dari palet atau tempat lain, lalu melepaskannya pada tempat yang diinginkan pada tampilan desain aplikasi. Ini memberikan pengalaman pengembangan yang lebih visual dan intuitif.

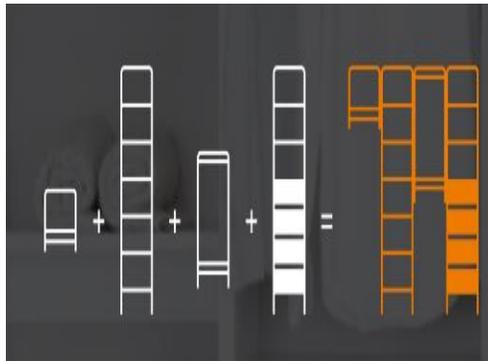


b. Visual Design

Visual Design adalah pendekatan dalam pengembangan perangkat lunak yang memfasilitasi proses desain antarmuka pengguna (GUI) secara visual. Dengan menggunakan alat desain grafis atau editor visual, pengembangan dalam suatu aplikasi dapat di rancang dan mengatur elemen-elemen GUI tanpa harus menulis kode secara manual.

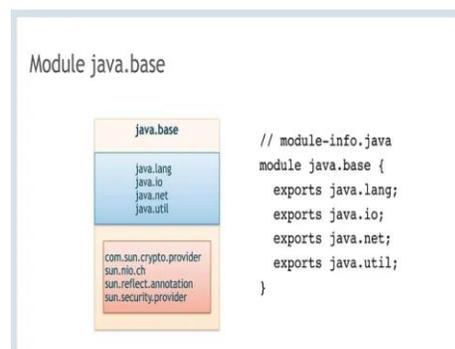
Fitur ini melibatkan "drag-and-drop" yang memungkinkan pengembangan suatu aplikasi dapat dengan mudah menambahkan komponen GUI ke dalam formulir atau tampilan aplikasi dengan menariknya dari palet dan melepaskannya pada tempat yang diinginkan pada tampilan desain.

3.5 Modularitas Java

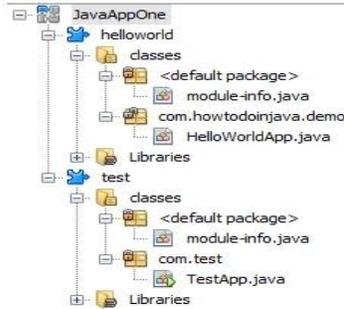


Modularitas adalah konsep umum yang diterapkan pada pengembangan perangkat lunak dengan cara yang memungkinkan *modul* individual dikembangkan, seringkali dengan antarmuka standar untuk memungkinkan modul berkomunikasi. Faktanya, jenis pemisahan perhatian antar objek dalam bahasa OO memiliki konsep yang hampir sama dengan modul, kecuali dalam skala yang lebih besar. Biasanya, mempartisi sistem menjadi beberapa modul membantu meminimalkan penggandengan yang akan mempermudah pemeliharaan kode.

Pentingnya modularitas dalam Java terlihat pada JDK sebelum Java SE 9. JDK sebelumnya sangat besar dan monolitik, menyebabkan peningkatan waktu pengunduhan, waktu pengaktifan, dan jejak memori. Dengan Java SE 9, JDK dipecah menjadi 90 modul. Setiap modul mewakili bagian fungsionalitas yang terdefinisi dengan baik, seperti logging, Swing, dan instrumentasi.



Manfaat penggunaan modularitas Java termasuk aplikasi yang lebih skalabel untuk perangkat kecil, peningkatan kinerja aplikasi, serta peningkatan keamanan dan pemeliharaan untuk Java dan aplikasi yang dikembangkan dengan pendekatan modular.



Modul-Proyek

Pada Java 9, modul dideklarasikan dengan menggunakan file `module-info.java`, di mana deklarasi dimulai dengan kata kunci `module`, diikuti dengan nama modul unik dan badan modul diapit tanda kurung kurawal. Modul memiliki kemampuan untuk mengekspor paket-paket yang dapat diakses oleh modul lain dan menentukan ketergantungan pada modul lainnya.

Pentingnya ekspor dan membutuhkan modul adalah untuk mengelola aksesibilitas antar modul. Sebuah modul dapat mengekspor paket-paket tertentu, dan modul lain dapat membutuhkan modul tersebut untuk mengakses paket-paket tersebut. Pengaturan ini membantu mengontrol akses terhadap kode dan mencegah penggunaan yang tidak diinginkan.

```
paket pkgA;

kelas publik KelasA {
    public int tahunPublic = 2001;
    dilindungi int tahunDilindungi = 2002;
    int paket tahun = 2003;
    private int tahunPrivate = 2004;
}

modul A {
    ekspor pkgA;
}

modul B {
    membutuhkan A;
}

paket pkgB;
import pkgA.ClassA;

kelas publik NewMain {
    public static void main(String[] args) {
        Tes KelasA = KelasA baru();
        System.out.println(test.yearPublic); //Ya
        System.out.println(test.yearProtected); //Tidak Ada
        System.out.println(test.yearPackage); //Tidak Ada
        System.out.println(test.yearPrivate); //TIDAK
    }
}
```

Transitivitas memungkinkan keterbacaan pada rantai kebutuhan, di mana apa pun yang memerlukan modul saat ini memiliki akses ke paket-paket yang diekspor oleh modul saat ini dan modul yang diperlukan secara transitif. Hal ini membantu dalam mengorganisir ketergantungan modul secara lebih terstruktur.

Modul dasar `java.base` adalah modul yang mencakup semua paket inti platform. Setiap modul di Java bergantung pada `java.base`, dan referensi modul ini secara implisit disertakan dalam semua modul lainnya.

```
// modul-info.java

modul java.base{
    ekspor java.lang;
    ekspor java.io;
    ekspor java.net;
    ekspor java.util;
}

modul A{
    memerlukan java.base ; // tersirat
    membutuhkan java.logging;
    membutuhkan C transitif;
    mengekspor pkgA ke B;
}
```

Dengan menggunakan modularitas dalam pengembangan Java, kita dapat mengelola kompleksitas proyek dengan lebih baik, meningkatkan keterbacaan dan pemeliharaan kode, serta meningkatkan keamanan dan kinerja aplikasi.

3.6 integrasi dengan basis data

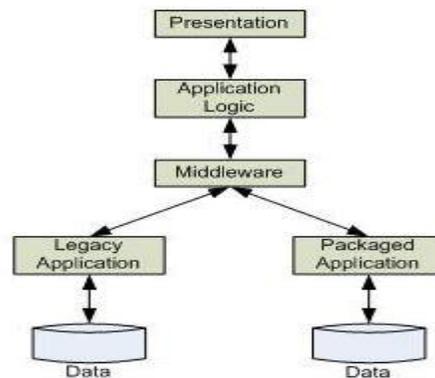
Integrasi dengan basis data merupakan aspek penting dalam pengembangan perangkat lunak, termasuk dalam konteks penggunaan Java NetBeans. Java NetBeans menyediakan berbagai fasilitas dan dukungan untuk mengintegrasikan aplikasi dengan basis data, memungkinkan pengembang untuk menyimpan, mengambil, dan memanipulasi data secara efisien.

Salah satu cara untuk mengintegrasikan aplikasi Java NetBeans dengan basis data adalah melalui penggunaan JDBC (Java Database Connectivity). JDBC adalah API Java yang menyediakan kelas dan antarmuka untuk mengakses dan mengelola data dalam database. Java NetBeans memfasilitasi penggunaan JDBC dengan menyediakan wizard dan alat bantu visual untuk membuat koneksi database, mengatur query SQL, dan menampilkan data dalam aplikasi dengan mudah.

Java NetBeans juga mendukung integrasi dengan berbagai sistem manajemen basis data (DBMS), termasuk MySQL, PostgreSQL, Oracle, dan lainnya. Melalui NetBeans, pengembang dapat membuat koneksi database, mengeksekusi query, dan mengelola skema database secara visual.

Selain JDBC, Java NetBeans juga mendukung pemrograman basis data berbasis objek menggunakan Java Persistence API (JPA). JPA adalah spesifikasi Java untuk mengelola objek-objek yang persisten ke dalam basis data relasional. NetBeans menyediakan alat bantu untuk membuat entitas JPA, mengonfigurasi koneksi ke penyimpanan data, dan melakukan operasi CRUD (Create, Read, Update, Delete) pada entitas.

Penggunaan fitur integrasi basis data pada Java NetBeans dapat mempercepat pengembangan aplikasi yang membutuhkan penyimpanan dan pengambilan data dari database. Dengan menggunakan alat visual dan dukungan yang disediakan oleh Java NetBeans, pengembang dapat mengelola aspek basis data dengan lebih efisien dan fokus pada pengembangan fitur-fitur utama dari aplikasi mereka.



3.7 Wawasan Terhadap Potensi dan Keterbatasan

Java NetBeans menawarkan potensi besar dalam pengembangan perangkat lunak berbasis Java. Lingkungan pengembangan terpadu ini memungkinkan pengembang untuk menulis, menguji, dan mendistribusikan aplikasi secara efisien. Keuntungan utama terletak pada kemampuan pengembangan antarmuka pengguna yang intuitif melalui framework GUI, memungkinkan pengembang untuk merancang GUI dengan cepat menggunakan fitur "drag-and-drop". Selain itu, pendekatan modularitas yang didukung oleh Java NetBeans mempromosikan pengembangan yang terstruktur, meningkatkan keterbacaan kode, dan memudahkan pemeliharaan.

Namun demikian, penggunaan Java NetBeans juga memiliki keterbatasan yang perlu dipertimbangkan. Adanya ketergantungan pada IDE tertentu dapat menjadi hambatan jika pengembang memutuskan untuk beralih ke IDE lain. Konsumsi sumber daya yang tinggi, terutama memori, dapat mempengaruhi kinerja terutama untuk proyek besar. Antarmuka dan fitur kaya NetBeans mungkin terasa kompleks bagi pengembang pemula, memerlukan waktu pembelajaran yang cukup.

3.8 Memanfaatkan Fitur Kunci

Kita selaku pengembang dapat memanfaatkan fitur-fitur kunci yang ditawarkan oleh Java NetBeans untuk meningkatkan produktivitas dan kualitas perangkat lunak yang dikembangkan. Salah satu fitur utama yang dapat dimanfaatkan adalah framework GUI yang memungkinkan pengembangan antarmuka pengguna (GUI) secara visual melalui pendekatan "drag-and-drop". Dengan menggunakan fitur ini, pengembang dapat dengan mudah menambahkan dan mengatur elemen-elemen GUI tanpa harus menulis kode secara manual, sehingga mempercepat proses desain antarmuka.

Selain itu, modularitas adalah fitur kunci yang dapat dimanfaatkan. Java NetBeans mendukung pendekatan modular dalam pengembangan perangkat lunak, yang memungkinkan pemisahan fungsionalitas ke dalam modul-modul terpisah. Ini tidak hanya meningkatkan keterbacaan kode, tetapi juga memudahkan pemeliharaan dan pengembangan berkelanjutan. Pengembang dapat membuat modul-modul yang bersifat independen dan dapat digunakan kembali dalam proyek-proyek berbeda.

Integrasi dengan basis data adalah fitur lain yang penting. Java NetBeans menyediakan dukungan penuh untuk mengintegrasikan aplikasi dengan berbagai sistem manajemen basis data (DBMS). Melalui alat visual yang disediakan, pengembang dapat membuat koneksi ke basis data, mengeksekusi query SQL, dan mengelola skema database dengan mudah, mempercepat pengembangan fitur-fitur yang melibatkan penyimpanan dan pengambilan data.

Selanjutnya, fitur pemrograman berbasis objek menggunakan Java Persistence API (JPA) dapat dimanfaatkan untuk pengembangan aplikasi dengan pemodelan data yang bersifat objek dan persisten. Java NetBeans menyediakan alat bantu untuk membuat entitas JPA, mengonfigurasi koneksi ke penyimpanan data, dan melakukan operasi CRUD pada entitas dengan mudah.

Dengan memahami dan memanfaatkan fitur-fitur kunci ini, Kita selaku pengembang untuk aplikasi dengan menggunakan Java NetBeans dapat meningkatkan efisiensi, keterbacaan kode, dan kemampuan pengembangan aplikasi secara keseluruhan menggunakan Java NetBeans

4 KESIMPULAN

Jurnal ini menguraikan pentingnya pemrograman menggunakan Java NetBeans dalam pengembangan perangkat lunak berbasis Java. Dalam konteks ini juga, Java NetBeans diakui sebagai Integrated Development Environment (IDE) yang efektif, menyediakan berbagai fitur untuk mempermudah proses pengkodean, pengujian, dan penyebaran aplikasi Java. Sejumlah aspek utama dibahas dalam jurnal ini, dan berikut adalah kesimpulan relevan:

1. Keuntungan Penggunaan Java NetBeans:

- Java NetBeans memberikan keuntungan signifikan dalam pengembangan antarmuka pengguna (GUI) melalui framework GUI yang mendukung pendekatan "drag-and-drop". Ini memungkinkan pengembang untuk merancang GUI dengan cepat dan intuitif.
- Modularitas Java NetBeans mendukung pengembangan yang terstruktur, meningkatkan keterbacaan kode, dan memfasilitasi pemeliharaan. Konsep ini memberikan keleluasaan dalam pemisahan fungsionalitas ke dalam modul-modul terpisah.
- Integrasi dengan basis data adalah fitur kunci yang mempercepat pengembangan fitur-fitur yang melibatkan penyimpanan dan pengambilan data. Dukungan penuh untuk JDBC dan Java Persistence API (JPA) memudahkan pengelolaan data dalam aplikasi.

2. Potensi dan Keterbatasan:

- Wawasan mendalam diberikan terkait potensi dan keterbatasan penggunaan Java NetBeans. Pengembang diuntungkan dari lingkungan pengembangan yang intuitif, tetapi ketergantungan pada IDE tertentu dapat menjadi hambatan jika beralih ke IDE lain.
- Konsumsi sumber daya yang tinggi, terutama memori, dan kompleksitas antarmuka NetBeans mungkin memerlukan waktu pembelajaran yang cukup, terutama bagi pengembang pemula.

3. Memanfaatkan Fitur Kunci:

- Fitur-fitur kunci seperti framework GUI, modularitas, dan integrasi basis data dapat dimanfaatkan untuk meningkatkan produktivitas dan kualitas perangkat lunak. Pendekatan modular memungkinkan pengembang untuk membuat modul-modul yang dapat digunakan kembali.
- Pemahaman dan pemanfaatan fitur-fitur ini dapat meningkatkan efisiensi, keterbacaan kode, dan kemampuan pengembangan aplikasi secara keseluruhan.

4. Studi Literatur dan Metode Penelitian:

- Kombinasi studi literatur dan metode penelitian memberikan dasar kuat untuk penelitian ini. Studi literatur memberikan pemahaman tentang konsep dasar dan praktik terbaik, sementara metode penelitian mencakup analisis fitur, implementasi praktis, eksperimen, dan analisis hasil.

Jadi secara garis besarnya, jurnal ini memberikan pandangan menyeluruh tentang potensi dan keterbatasan Java NetBeans dalam pengembangan perangkat lunak. Pengembang dapat memanfaatkan fitur-fitur kunci ini dengan baik untuk mencapai hasil pengembangan yang efisien dan berkualitas.

REFERENCES

- Robby Yuli Endra. 2020. MATERI #1 PENGENALAN JAVA, NETBEANS DAN CONTOH PROGRAM SEDERHANA. <https://youtu.be/omUrgahRCxg?si=Wgqm98C2gOxmIriW>
- Magsani. 2023. Cara Menggunakan NetBeans IDE Pada Pemrograman Java. <https://www.masgani.com/cara-menggunakan-netbeans-ide-pada-pemrograman-java/>
- Metha, Sanjaya. Agustus 2021. Modularitas Java . <https://medium.com/@sanjayofficial94/java-modularity-a848942d0820>
- Career & Tech HQ. 2021. Java GUI Tutorial #32 – Drag and Drop In Java GUI. <https://youtu.be/UE549Ae1C1U?si=bzJweQCyFfjRri07>
- DqLab. Januari 2021. Kenali Proses Integrasi Dara dalam Dunia Basis Data. <https://dqlab.id/kenali-proses-integrasi-data-dalam-dunia-basis-data>