

LAPORAN
APLIKASI PENJUALAN RUMAH



OLEH:
NAMA : DIAH NUR ASTANTI
NIM: 13020210001
KELAS: B1

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS MUSLIM INDONESIA
MAKASSAR**

2023

DAFTAR ISI

DAFTAR ISI.....	2
ABSTRAK.....	4
BAB 1	5
PENDAHULUAN	5
1.1 Latar Belakang.....	5
1.2 Rumusan Masalah.....	6
1.3 Manfaat	6
BAB 2	8
TINJAUAN PUSTAKA.....	8
2.1 OOP	8
2.2 Metode MVC	8
2.3 Controller	9
2.4 CRUD	9
2.5 Enkapsulasi	10
2.6 Inheritance	10
BAB 3	11
ANALISIS DAN DESAIN	11
3.1 ANALISA KEBUTUHAN SISTEM	11
3.2 METODE PENGEMBANGAN	12
3.3 ALUR PROGRAM.....	14
3.4 USE CASE DIAGRAM.....	14
3.5 CLASS DIAGRAM.....	15
3.6 DATABASE.....	16
BAB 4	17
IMPLEMENTASI DAN PENGUJIAN	17
4.1 Implementasi dan Pengujian.....	17
4.2 Database Aplikasi Data Rumah	17
4.3 Pengujian Program.....	18
4.3.1 Create	18
4.3.2 Pengujian Update	18

4.3.3 Pengujian Delete	19
4.3.4 Mencari Data.....	19
BAB 5	19
PENUTUP	19
KESIMPULAN.....	19
DAFTAR PUSTAKA.....	21

ABSTRAK

Pembuatan aplikasi penjualan rumah dimulai dengan menganalisa kebutuhan utama dari aplikasi dan fungsionalitas yang diharapkan. Analisa ini menjadi landasan dari perancangan sistem penjualan rumah yang akurat dan sesuai dengan kebutuhan pengguna. Setelah melalui tahap analisa, dilakukan perancangan sistem. Pada tahap ini, struktur kelas-kelas utama dan hubungan antara kelas-kelas yang berbeda ditentukan. Selain itu, desain antarmuka pengguna juga dibuat agar dapat memberikan pengalaman pengguna yang intuitif dan mudah digunakan. Prinsip-prinsip OOP seperti enkapsulasi dan pewarisan digunakan untuk menciptakan desain yang fleksibel dan mudah dikembangkan di masa mendatang.

Selama proses implementasi, prinsip-prinsip OOP seperti modularitas dan reusabilitas diterapkan untuk memastikan kode yang bersih dan mudah dipelihara. Setelah tahap implementasi selesai, maka dilakukan tahap pengujian untuk memastikan aplikasi berjalan sesuai dengan yang diharapkan. Pengujian dilakukan untuk mengidentifikasi dan memperbaiki bug atau kesalahan dalam kode. Berbagai skenario pengujian dibuat untuk menguji fungsionalitas dan performa aplikasi, termasuk pengujian input validasi, pengujian transaksi penjualan, dan pengujian integrasi dengan sistem lain jika diperlukan. Pengujian ini penting untuk memastikan kualitas dan kehandalan aplikasi.

Terakhir, dokumen pengembangan aplikasi dibuat untuk memberikan panduan penggunaan dan pemeliharaan aplikasi. Dokumen ini mencakup penjelasan mengenai fungsionalitas aplikasi, cara penggunaan, dan troubleshooting. Selain itu, dokumentasi juga mencakup panduan pengembangan lanjutan, sehingga tim pengembang di masa depan dapat dengan mudah memahami dan memperbarui aplikasi jika diperlukan. Dokumentasi yang lengkap membantu dalam pemeliharaan dan pengembangan berkelanjutan dari aplikasi penjualan rumah.

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Laporan ini berisikan mengenai pembuatan aplikasi berbasis OOP (Object-Oriented Programming) yang bertujuan untuk mengelola sistem penjualan rumah. Aplikasi ini dikembangkan dengan menggunakan pendekatan OOP yang melibatkan konsep-konsep seperti enkapsulasi, pewarisan, dan polimorfisme. Melalui aplikasi ini, pengguna dapat melakukan berbagai tugas terkait penjualan rumah secara efisien dan terstruktur.

Tujuan utama pembuatan aplikasi ini adalah untuk memberikan alat yang handal dan efektif dalam mengelola sistem penjualan rumah. Dalam industri properti, penjualan rumah melibatkan berbagai aspek seperti pencatatan informasi rumah, manajemen pelanggan, dan pencatatan transaksi penjualan. Aplikasi ini ditujukan untuk mempermudah proses tersebut, meningkatkan efisiensi, dan mengurangi kesalahan manusia yang mungkin terjadi. Aplikasi ini dapat digunakan oleh berbagai pihak yang terlibat dalam penjualan rumah, seperti agen properti, pengembang, dan staf administrasi.

Dalam analisis awal, telah diidentifikasi beberapa kebutuhan dasar yang harus dipenuhi oleh aplikasi. Kebutuhan tersebut mencakup pencatatan data rumah, termasuk informasi mengenai spesifikasi, harga, dan lokasi. Selain itu, pengelolaan data pelanggan dan transaksi penjualan juga menjadi fokus utama. Aplikasi ini harus dapat menyimpan dan mengelola informasi pelanggan, mencatat transaksi penjualan, dan memberikan laporan yang berguna bagi pengguna. Dengan dibuatnya aplikasi ini diharapkan dapat memberikan solusi yang handal, efisien, dan mudah digunakan dalam mengelola sistem penjualan rumah.

1.2 Rumusan Masalah

Dari hasil analisa yang dilakukan oleh tim pengembang terdapat 2 masalah yang paling mencolok. Pertama, masalah utama adalah ketidakefisienan dalam pencatatan dan manajemen informasi terkait rumah yang dijual. Dalam sistem konvensional, hal ini menyebabkan data mengenai rumah sulit diakses oleh penjual/agen ataupun pembeli. Selain itu terdapat kesulitan dalam mengelola data karena data seringkali berupa dokumentasi fisik. Hal ini menyebabkan kesulitan dalam memperoleh informasi yang akurat dan terkini tentang rumah yang sedang dijual, seperti spesifikasi dan harga. Selain itu, sulitnya melacak riwayat perubahan data juga menjadi hambatan dalam pengambilan keputusan yang tepat.

Masalah kedua yang perlu diatasi adalah kurangnya pengelolaan yang efektif terhadap proses penjualan rumah secara keseluruhan. Dalam sistem konvensional, terdapat banyak kegiatan yang harus dilakukan secara manual, mulai dari penyiapan kontrak penjualan, pembayaran, hingga pengiriman dokumen-dokumen penting. Hal ini rentan terhadap kesalahan dan memakan waktu yang cukup lama. Selain itu, manajemen pelanggan juga seringkali kurang terorganisir, menyebabkan kesulitan dalam memantau dan mengelola hubungan dengan pelanggan potensial serta pelanggan yang sudah ada. Dengan adanya aplikasi berbasis OOP yang terintegrasi, diharapkan dapat meningkatkan efisiensi dalam melakukan proses-proses tersebut serta memastikan pengelolaan yang lebih efektif dan teratur.

1.3 Manfaat

Tujuan utama dari pembuatan aplikasi penjualan rumah berbasis OOP ini adalah untuk meningkatkan efisiensi dan efektivitas dalam proses penjualan rumah serta pengelolaan informasi terkait. Beberapa tujuan yang ingin dicapai melalui pengembangan aplikasi ini antara lain:

- 1.3.1 Meningkatkan efisiensi operasional: Dengan adanya aplikasi penjualan rumah berbasis OOP, diharapkan proses penjualan rumah dapat dilakukan dengan lebih efisien dan terstruktur. Pengguna dapat dengan mudah mencatat dan mengelola informasi rumah yang dijual, memantau ketersediaan, memperbarui spesifikasi dan harga, serta melacak riwayat perubahan data dengan cepat. Hal ini akan mengurangi waktu dan upaya yang diperlukan dalam mengelola penjualan rumah, sehingga menghemat sumber daya perusahaan.
- 1.3.2 Meningkatkan kualitas layanan pelanggan: Aplikasi ini juga bertujuan untuk meningkatkan kualitas layanan kepada pelanggan. Dengan adanya sistem yang terintegrasi, pengguna dapat dengan mudah memberikan informasi yang akurat dan terkini kepada calon pembeli, seperti spesifikasi rumah, harga, dan fasilitas yang tersedia. Selain itu, pengguna juga dapat memantau dan mengelola hubungan dengan pelanggan potensial serta pelanggan yang sudah ada, sehingga memberikan pelayanan yang lebih personal dan memuaskan.
- 1.3.3 Meningkatkan pengambilan keputusan yang tepat: Aplikasi penjualan rumah berbasis OOP ini juga memiliki tujuan untuk memberikan informasi yang relevan dan akurat kepada manajemen dalam pengambilan keputusan strategis. Dengan adanya sistem yang terintegrasi, manajemen dapat dengan mudah mengakses data penjualan, analisis performa rumah, dan tren pasar. Hal ini akan membantu dalam membuat keputusan yang lebih tepat dan berdasarkan data, seperti menentukan harga jual yang kompetitif, mengidentifikasi peluang pasar, dan mengoptimalkan strategi pemasaran.

Manfaat yang diharapkan dari pengembangan aplikasi ini adalah peningkatan efisiensi operasional, peningkatan kualitas layanan pelanggan, penghematan waktu dan biaya, serta peningkatan pengambilan keputusan yang lebih akurat

dan tepat. Dengan adanya aplikasi penjualan rumah berbasis OOP, diharapkan perusahaan dapat mengoptimalkan proses penjualan rumah dan meningkatkan kepuasan pelanggan, sehingga berkontribusi pada pertumbuhan dan kesuksesan bisnis secara keseluruhan.

BAB 2

TINJAUAN PUSTAKA

2.1 OOP

Object-Oriented Programming (OOP) adalah paradigma pemrograman yang berfokus pada konsep objek. OOP menggunakan objek sebagai unit utama untuk merepresentasikan data dan perilaku. Objek dapat berinteraksi satu sama lain melalui pesan dan metode, sehingga memungkinkan pemrograman yang modular, fleksibel, dan mudah dipahami. Beberapa konsep utama dalam OOP meliputi enkapsulasi, pewarisan, dan polimorfisme. Enkapsulasi memungkinkan penyembunyian informasi dan membatasi akses langsung ke data, sedangkan pewarisan memungkinkan pembuatan hierarki kelas yang mewarisi sifat-sifat dan perilaku dari kelas lain. Polimorfisme memungkinkan objek dari kelas yang berbeda untuk merespons metode dengan cara yang berbeda pula.

2.2 Metode MVC

Metode Model-View-Controller (MVC) adalah sebuah paradigma desain perangkat lunak yang memisahkan tiga komponen utama, yaitu Model, View, dan Controller. Pemisahan ini memungkinkan pengembang untuk memisahkan logika bisnis (Model), tampilan (View), dan pengendalian (Controller) dalam aplikasi GUI berbasis OOP Java. Dengan memisahkan tugas-tugas ini, MVC memungkinkan pengembangan yang terstruktur, mudah dipelihara, dan dapat mempermudah pengembangan kolaboratif.

Model berfungsi sebagai komponen yang menangani logika bisnis dan data dalam aplikasi. Ini mencakup manipulasi data, validasi input, dan logika proses yang berkaitan dengan pengolahan data. Model tidak langsung berinteraksi dengan tampilan (View) atau pengendali (Controller). Model dalam MVC merupakan representasi dari entitas atau objek bisnis dalam aplikasi.

View adalah komponen yang menangani tampilan atau presentasi data kepada pengguna. Ini dapat berupa antarmuka pengguna grafis (GUI) yang dihasilkan menggunakan komponen OOP Java seperti JFrame, JPanel, JButton, dan lainnya. View bertanggung jawab untuk menampilkan data dari Model dan menerima input pengguna. Namun, View tidak melakukan manipulasi langsung terhadap data.

Controller berfungsi sebagai penghubung antara Model dan View. Ini menangani interaksi pengguna dengan View, mengambil input dari pengguna, dan memperbarui Model sesuai dengan input tersebut. Controller juga dapat menerima pembaruan dari Model dan memperbarui tampilan View sesuai dengan perubahan data yang terjadi. Dengan demikian, Controller bertindak sebagai mediator antara Model dan View.

2.3 Controller

Controller dalam konteks aplikasi OOP Java GUI bertanggung jawab untuk mengontrol alur program, menerima input dari pengguna, dan memperbarui model serta tampilan sesuai dengan interaksi yang terjadi. Controller juga berfungsi sebagai mediator antara Model dan View, mengelola logika bisnis, serta mengatur perubahan data dan interaksi antarmuka.

2.4 CRUD

Merupakan kepanjangan dari *create*, *read*, *update*, dan *delete*. CRUD merupakan operasi dasar dalam pengolahan data yang

melibatkan pembuatan (Create), pembacaan (Read), pembaruan (Update), dan penghapusan (Delete) data. Dalam aplikasi OOP Java GUI, CRUD sering digunakan untuk melakukan manipulasi data pada model dan menghasilkan perubahan pada tampilan yang sesuai.

2.5 Enkapsulasi

Enkapsulasi adalah konsep dalam pemrograman berorientasi objek yang menggabungkan data dan metode yang beroperasi pada data tersebut ke dalam sebuah entitas tunggal yang disebut objek. Dengan enkapsulasi, data dalam objek hanya dapat diakses melalui metode-metode yang ditentukan, sehingga memastikan keamanan dan integritas data.

2.6 Inheritance

Inheritance (warisan) adalah konsep dalam pemrograman berorientasi objek yang memungkinkan kelas (subclass) untuk mewarisi atribut dan metode dari kelas lain (superclass). Dalam konteks pembuatan aplikasi OOP Java GUI, inheritance dapat digunakan untuk membangun hierarki kelas yang memungkinkan pemodelan objek yang lebih terstruktur, penggunaan kembali kode, dan meningkatkan efisiensi dalam pengembangan aplikasi GUI.

Inheritance memungkinkan subclass untuk mewarisi semua karakteristik (atribut dan metode) dari superclass. Subclass dapat menambahkan fungsi tambahan atau mengubah perilaku superclass melalui overriding (penimpaan) metode yang ada. Dalam pembuatan aplikasi OOP Java GUI, inheritance dapat digunakan untuk membangun hierarki kelas komponen GUI, seperti JFrame, JPanel, dan JButton, yang merupakan turunan dari kelas-kelas yang lebih umum.

BAB 3

ANALISIS DAN DESAIN

3.1 ANALISA KEBUTUHAN SISTEM

Program penjualan rumah yang menggunakan OOP CRUD dengan Java GUI akan memiliki beberapa kebutuhan sistem yang harus dipenuhi agar dapat berfungsi dengan baik. Berikut adalah beberapa kebutuhan utama yang harus diperhatikan:

1. Kebutuhan Fungsional (Manajemen Data)

Program ini harus mampu mengelola data penjualan rumah, termasuk informasi tentang properti, detail rumah, harga, lokasi, dan lain sebagainya. Sistem harus dapat menyimpan, mengubah, menghapus, dan mencari data dengan menggunakan operasi CRUD (Create, Read, Update, Delete) pada objek-objek yang terkait. Dalam hal ini, program harus memiliki kelas-kelas yang mewakili entitas seperti rumah, pelanggan, dan transaksi.

2. Kebutuhan Non Fungsional (Antarmuka Pengguna)

Program ini harus memiliki antarmuka pengguna yang interaktif dan intuitif menggunakan Java GUI. Antarmuka ini akan memungkinkan pengguna untuk memasukkan data penjualan rumah, melihat daftar rumah yang tersedia, mengubah informasi rumah, dan menghapus rumah yang sudah terjual. Selain itu, program juga dapat menampilkan laporan penjualan rumah dan informasi lainnya yang relevan.

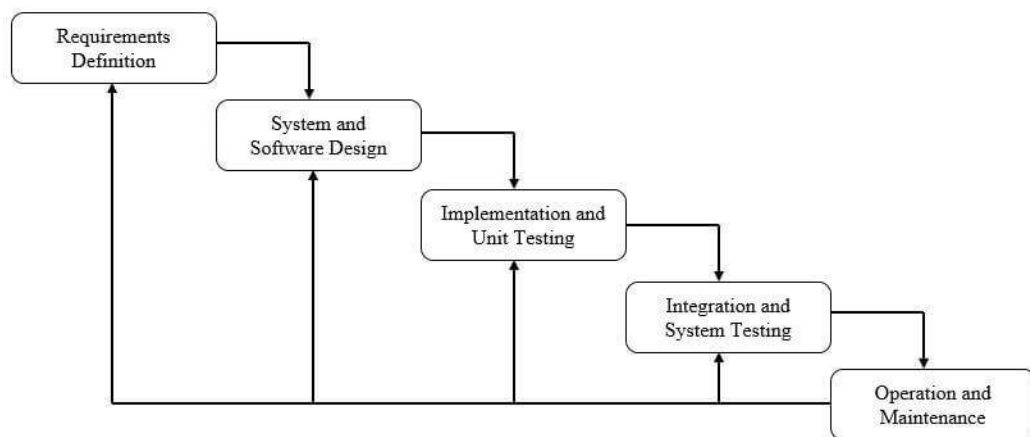
3. Validasi Data

Program harus melakukan validasi data untuk memastikan bahwa input yang dimasukkan pengguna sesuai dengan format yang diharapkan. Misalnya, memeriksa apakah harga rumah adalah angka positif, memastikan semua kolom wajib diisi, dan sebagainya. Validasi data ini penting untuk menjaga integritas dan keakuratan informasi dalam program.

4. Koneksi ke Database

Program harus dapat terhubung ke database untuk menyimpan dan mengambil data penjualan rumah. Dalam hal ini, dapat digunakan teknologi seperti Java Database Connectivity (JDBC) untuk berinteraksi dengan database. Dengan adanya koneksi ke database, data penjualan rumah dapat disimpan secara persisten dan dapat diakses kembali setelah program dimatikan dan dijalankan kembali.

3.2 METODE PENGEMBANGAN



Metode pengembangan yang akan digunakan dalam pengembangan program penjualan rumah menggunakan OOP CRUD dengan Java GUI adalah metode waterfall. Metode waterfall adalah pendekatan pengembangan perangkat lunak yang mengikuti tahapan-tahapan yang berurutan, dimulai dari analisis kebutuhan hingga implementasi dan pengujian. Berikut adalah tahapan-tahapan dalam metode waterfall untuk pengembangan program penjualan rumah:

1. Analisis Kebutuhan

Tahap ini melibatkan pengumpulan dan analisis kebutuhan sistem secara rinci. Tim pengembang harus berinteraksi dengan pemangku kepentingan untuk memahami persyaratan program penjualan rumah secara mendalam. Pada tahap ini, dapat dibuat dokumen analisis kebutuhan yang menjelaskan kebutuhan fungsional dan non-fungsional program.

2. Perancangan Sistem

Setelah kebutuhan dikumpulkan, langkah selanjutnya adalah merancang arsitektur sistem dan desain antarmuka pengguna. Pada tahap ini, akan dibuat desain kelas-kelas Java yang merepresentasikan entitas-entitas dalam program penjualan rumah dan desain antarmuka pengguna menggunakan Java GUI.

3. Implementasi

Tahap ini melibatkan implementasi kode program berdasarkan desain yang telah dibuat sebelumnya. Tim pengembang akan mulai menulis kode Java untuk mengimplementasikan fungsionalitas program penjualan rumah, termasuk operasi CRUD, validasi data, koneksi database, dan antarmuka pengguna.

4. Pengujian

Setelah implementasi selesai, program akan diuji untuk memastikan bahwa berfungsi sesuai dengan kebutuhan yang telah ditetapkan. Pengujian akan melibatkan pengujian fungsionalitas, pengujian integrasi, dan pengujian keamanan. Jika ditemukan kesalahan atau kekurangan, tim pengembang akan melakukan perbaikan yang diperlukan.

5. Penerapan

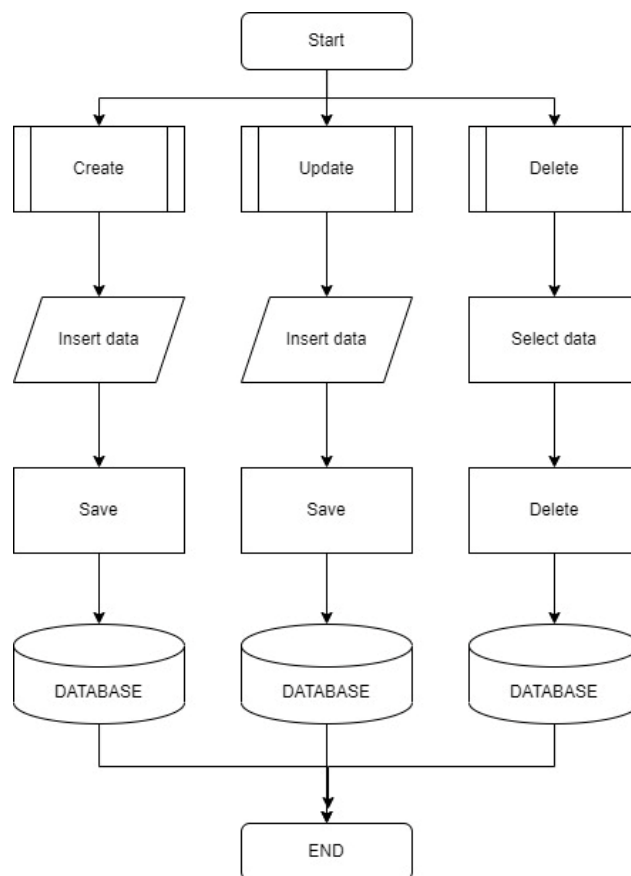
Setelah program telah diuji dengan baik dan ditemukan bekerja dengan baik, program dapat diimplementasikan dan digunakan secara langsung oleh pengguna. Pada tahap ini, program penjualan rumah akan diinstal dan dikonfigurasi pada lingkungan produksi yang sesuai.

6. Pemeliharaan

Setelah penerapan, program akan memasuki tahap pemeliharaan. Pada tahap ini, tim pengembang akan merespons umpan balik pengguna, melakukan pemeliharaan rutin, dan melakukan perbaikan atau peningkatan jika diperlukan.

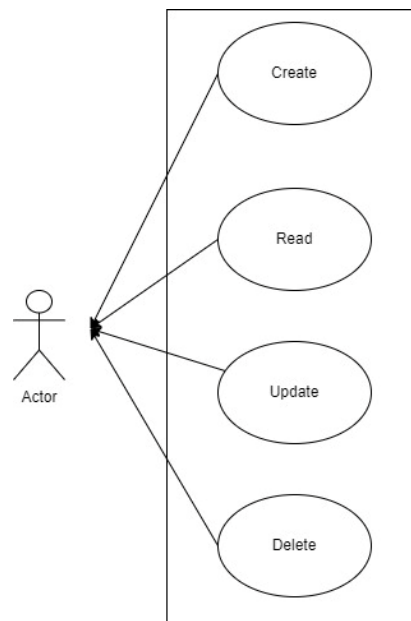
Metode waterfall memberikan pendekatan yang terstruktur dan linear dalam pengembangan program penjualan rumah. Setiap tahapan harus diselesaikan sebelum melanjutkan ke tahap berikutnya.

3.3 ALUR PROGRAM



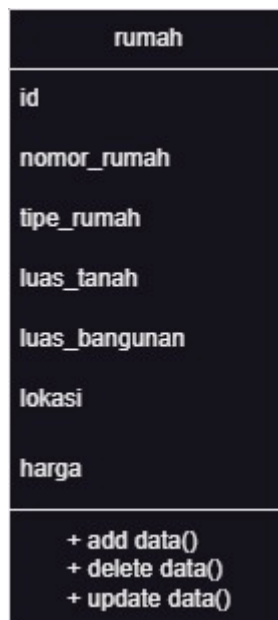
Bagan diatas merupakan diagram *flowcart* yang menggambarkan bagaimana alur program berjalan. Ketika program dimulai, maka akan ditampilkan table berisikan data serta form data. Pengguna bisa mengisikan data pada form dan menambahkan data atau mengubah data. Selain itu pengguna juga bisa memilih data yang tersedia pada table untuk dihapus.

3.4 USE CASE DIAGRAM



Bagan diatas merupakan *use case* diagram dari program ini. Pada program ini actor dapat melakukan *create*, *read*, *update*, dan *delete*.

3.5 CLASS DIAGRAM



Bagan diatas merupakan *class* diagram yang digunakan pada program ini.

3.6 DATABASE

Showing rows 0 - 4 (5 total, Query took 0.0005 seconds.)

SELECT * FROM `rumah`

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Explain SQL](#)] [[Create PHP code](#)] [[Refresh](#)]

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

				id	nomor_rumah	tipe_rumah	luas_tanah	luas_bangunan	lokasi	harga
<input type="checkbox"/>	Edit	Copy	Delete	2	AB01	AC05	2000	1222	BANDUNG	900000
<input type="checkbox"/>	Edit	Copy	Delete	3	AB02	AC10	23231	41211	JAKARTA	1000
<input type="checkbox"/>	Edit	Copy	Delete	4	AB20	FK90	23523	1000	JAWA TIMUR	10100
<input type="checkbox"/>	Edit	Copy	Delete	7	AB29	FK70	7777	1000	SULAWESI SELATAN	10100
<input type="checkbox"/>	Edit	Copy	Delete	9	AB01	AC05	2000	1222	BANDUNG	900000

☐ Check all | With selected: [Edit](#) [Copy](#) [Delete](#) [Export](#)

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

jualrumah rumah

id : int(11)

nomor_rumah : varchar(50)

tipe_rumah : varchar(50)

luas_tanah : int(225)

luas_bangunan : int(225)

lokasi : varchar(100)

harga : int(225)

Gambar diatas merupakan database yang digunakan pada program ini.

BAB 4

IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi dan Pengujian

ID	Nomor Rumah	Tipe Rumah	Luas Tanah	Luas Bangunan	Lokasi	Harga
2	AB01	AC05	2000	1222	BANDUNG	900000
3	AB02	AC10	23231	41211	JAKARTA	1000
4	AB20	FK90	23523	1000	JAWA TIMUR	10100
7	AB29	FK70	7777	1000	SULAWESI SE...	10100
9	AB01	AC05	2000	1222	BANDUNG	900000

Gambar diatas merupakan tampilan antarmuka dari program. Program telah diuji dan dianalisa. Dari hasil pengujian tersebut, program dapat berjalan dengan lancar.

4.2 Database Aplikasi Data Rumah

Showing rows 0 - 4 (5 total, Query took 0.0005 seconds.)

SELECT * FROM `rumah`

☐ Profiling [\[Edit inline \]](#) [\[Edit \]](#) [\[Explain SQL \]](#) [\[Create PHP code \]](#) [\[Refresh \]](#)

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

	id	nomor_rumah	tipe_rumah	luas_tanah	luas_bangunan	lokasi	harga
<input type="checkbox"/>	2	AB01	AC05	2000	1222	BANDUNG	900000
<input type="checkbox"/>	3	AB02	AC10	23231	41211	JAKARTA	1000
<input type="checkbox"/>	4	AB20	FK90	23523	1000	JAWA TIMUR	10100
<input type="checkbox"/>	7	AB29	FK70	7777	1000	SULAWESI SELATAN	10100
<input type="checkbox"/>	9	AB01	AC05	2000	1222	BANDUNG	900000

☐ Check all | With selected: [Edit](#) [Copy](#) [Delete](#) [Export](#)

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

4.3 Pengujian Program

4.3.1 Create

DATA JUAL RUMAH

REFRESH

ID	Nomor Rumah	Tipe Rumah	Luas Tanah	Luas Bangunan	Lokasi	Harga
2	AB01	AC05	2000	1222	BANDUNG	900000
3	AB02	AC10	23231	41211	JAKARTA	1000
4	AB20	FK90	23523	1000	JAWA TIMUR	10100
7	AB29	FK70	7777	1000	SULAWESI SE...	10100
9	AB01	AC05	2000	1222	BANDUNG	900000
10	AB02	AC10	23231	41211	JAKARTA	1000

Message

Data berhasil disimpan

OK

No.Rumah: AB28
Luas Tanah: 7777
Tipe Rumah: FK70
Luas Bangunan: 1000
Lokasi: SULAWESI SELATAN
Harga: 10100

TAMBAH HAPUS

UBAH

CARI

ID:

4.3.2 Pengujian Update

DATA JUAL RUMAH

REFRESH

ID	Nomor Rumah	Tipe Rumah	Luas Tanah	Luas Bangunan	Lokasi	Harga
2	AB01	AC05	2000	1222	BANDUNG	900000
3	AB02	AC10	23231	41211	JAKARTA	1000
4	AB20	FK90	23523	1000	JAWA TIMUR	10100
7	AB29	FK70	7777	1000	SULAWESI SE...	10100
9	AB01	AC05	2000	1222	BANDUNG	900000
10	AB02	AC10	23231	41211	JAKARTA	1000
11	AB28	FK70	7777	1000	SULAWESI SE	10100

Message

Data berhasil diupdate

OK

No.Rumah: AB28
Luas Tanah: 7777
Tipe Rumah: FK70
Luas Bangunan: 1000
Lokasi: SULAWESI SELATAN
Harga: 10100

TAMBAH HAPUS

UBAH

CARI: bandung

ID: 11

4.3.3 Pengujian Delete

DATA JUAL RUMAH

REFRESH

ID	Nomor Rumah	Tipe Rumah	Luas Tanah	Luas Bangunan	Lokasi	Harga
2	AB01	AC05	2000	1222	BANDUNG	900000
3	AB02	AC10	23231	41211	JAKARTA	1000
4	AB20	FK90	23523	1000	JAWA TIMUR	10100
7	AB29	FK70	7777	1000	SULAWESI SE...	10100
9	AB01	AC05	2000	1222	BANDUNG	900000
10	AB02	AC10	23231	41211	JAKARTA	1000
11	AB28	FK70	7777	1000	SULAWESI SE	10100

Message

Data berhasil dihapus

OK

No.RumahAB02
Luas Tanah23231
Tipe RumahAC10
Luas Bangunan41211
LokasiJAKARTA
Harga1000

TAMBAH
HAPUS
UBAH

CARI

ID: 10

4.3.4 Mencari Data

ID	Nomor Rumah	Tipe Rumah	Luas Tanah	Luas Bangunan	Lokasi	Harga
2	AB01	AC05	2000	1222	BANDUNG	900000
9	AB01	AC05	2000	1222	BANDUNG	900000

BAB 5

PENUTUP

KESIMPULAN

Dalam laporan tersebut, telah dianalisis kebutuhan sistem untuk pembuatan program penjualan rumah menggunakan OOP CRUD dengan Java GUI.

Berikut adalah kesimpulan dari laporan tersebut:

1. Program penjualan rumah menggunakan OOP CRUD dengan Java GUI adalah solusi yang efektif untuk manajemen penjualan rumah dengan mengoptimalkan penggunaan objek dan antarmuka pengguna yang interaktif.

2. Program harus mampu mengelola data penjualan rumah dengan operasi CRUD (Create, Read, Update, Delete) pada entitas seperti rumah, pelanggan, dan transaksi.
3. Antarmuka pengguna yang intuitif dan interaktif menggunakan Java GUI harus disediakan untuk memasukkan data penjualan rumah, melihat daftar rumah, mengubah informasi rumah, dan menghapus rumah yang sudah terjual.
4. Validasi data harus dilakukan untuk memastikan integritas dan keakuratan informasi dalam program, seperti memeriksa format harga rumah dan memastikan semua kolom wajib terisi.
5. Koneksi ke database harus tersedia untuk menyimpan dan mengambil data penjualan rumah secara persisten menggunakan JDBC atau teknologi serupa.
6. Keamanan program harus diperhatikan dengan menerapkan autentikasi dan otorisasi pengguna, sehingga hanya pengguna yang memiliki hak akses tertentu yang dapat melakukan operasi CRUD pada data penjualan rumah.
7. Metode pengembangan waterfall digunakan untuk pengembangan program ini, yang melibatkan tahapan analisis kebutuhan, perancangan sistem, implementasi, pengujian, penerapan, dan pemeliharaan.

Dengan memperhatikan analisis kebutuhan ini dan mengikuti metode pengembangan yang tepat, program penjualan rumah menggunakan OOP CRUD dengan Java GUI telah diuji dan dianalisa. Hasil dari pengujian tersebut menyatakan bahwa program telah memenuhi tujuan dari pembuatan program ini dan dapat berjalan dengan baik.

DAFTAR PUSTAKA

1. Deitel, P., & Deitel, H. (2017). Java: How to Program, Early Objects (11th Edition). Pearson.
2. Horstmann, C. S., & Cornell, G. (2016). Core Java Volume I: Fundamentals (10th Edition). Prentice Hall.
3. Freeman, E., Robson, E., & Bates, B. (2003). Head First Design Patterns. O'Reilly Media.
4. Ambler, S. W. (2002). Agile Modeling: Effective Practices for eXtreme Programming and the Unified Process. Wiley.
5. Gaddis, T. (2016). Starting Out with Java: Early Objects (6th ed.). Pearson Education.