BAB I

PENDAHULUAN

A. Latar Belakang

Kemajuan teknologi informasi dan komunikasi mempengaruhi aktivitas manusia yang telah mengalami banyak perubahan dan perkembangan. Perkembangan ilmu pengetahuan di bidang teknologi saat ini telah menciptakan aplikasi-aplikasi baru dengan produktivitas dan biaya teknologi yang lebih *efisien* (Atmaja & Azis, 2019). Menurut (Huda, 2020), pesatnya perkembangan teknologi saat ini telah mempengaruhi pekerjaan di berbagai organisasi.

Aplikasi adalah program komputer yang dibuat oleh teknologi untuk membantu orang melakukan tugas tertentu yang mencakup analisis dan manipulasi data (Maqvira, 2024). Sedangkan menurut (Dewi et al., 2021) aplikasi adalah suatu program di dalam komputer atau handphone yang digunakan untuk menjalankan suatu program yang telah dibuat.

Android merupakan sebuah sistem operasi modifikasi dari *linux* yang digunakan untuk ponsel hingga perangkat jam tangan sampai televisi pintar, dalam perkembanganya dalam bidang handphone, sistem operasi *android* sudah menjamur di kalangan masyarakat Indonesia dari anak-anak hingga dewasa. Ponsel berbasis *android* juga sangat berguna dalam berkomuniksai dan mencari sebuah informasi (Nasution et al., 2019). *Android* merupakan sebuah sitem operasi yang sangat di gemari masyarakat belakangan ini (Sifauttijani, 2017).

Menurut (Ilmi Abdillah & Basuki, 2022) cat adalah suatu bahan cair atau kental yang tersusun dari suatu medium yang merupakan bahan fluida cair. Menambahkan sejumlah bahan tambahan tertentu ke pewarna dan eksipien (partikel kecil tidak larut dengan pelepasan sedang), tergantung pada campuran dan dosis. Untuk mendapatkan hasil pengecatan yang baik dan berkualitas menurut (Rizky Ferdiansyah et al., 2023) maka pihak yang terkait dalam pengecatan perlu mengetahui dasar-dasar pengecatan baik teknis aplikasi maupun pengawasan sehingga perlakuan dan penanganan dapat dilakukan sedemikian rupa untuk memenuhi spesifikasi baik oleh aplikator pemilik inspektor atau konsultan sehingga selama proses pengecatan diharapkan dapat.

Proses pengecatan tembok merupakan salah satu kegiatan penting dalam renovasi atau pembangunan bangunan. Namun, perhitungan kebutuhan cat yang diperlukan seringkali menjadi tantangan tersendiri. Perhitungan manual yang dilakukan oleh pekerja sering kali kurang akurat dan memakan waktu lama. Dalam penyusunan estimasi harus sangat berhati-hati dalam memperkirakan bahan dan jumlah yang dibutuhkan, seiring berjalannya waktu harga suatu material dapat berubah, namun jumlah material yang dibutuhkan pada proyek tersebut adalah tetap, sehingga proses perhitungan kuantitas secara akurat merupakan hal yang krusal dalam menentukan kesuksesan suatu proyek (Agustin et al., 2023).

Menurut (Rivki et al., 2020) komparatif adalah membandingkan variabel yang sama yang sifatnya membandingkan persamaan dan perbedaan dua atau

lebih sifat-sifat dan fakta-fakta objek yang diteliti berdasarkan suatu kerangka pemikiran tertentu.

Beberapa penelitian terdahulu telah mencoba mengatasi masalah ini dengan berbagai pendekatan. (Hari & Zaqi Al Faritsy, 2022) mengembangkan analisis pengendalian kualitas produk ember cat tembok 5kg, yang membahas jenis jenis cat tembok 5kg. (Octaviana & Siang, 2015) mengembangkan aplikasi pemilihan dan penghitungan volume cat tembok berbasis web dengan menggunaan motede *analytical hierarchy process* (AHP). Sementara itu, (Utomo, 2020) membahas analisa perawatan lambung *single mooring buoy* (SBM) sebagai sarana penunjang alur pelayaran transportasi laut. Proses pengecatan ulang pada kajian penelitian ini telah dilakukan uji kelayakan untuk melakukan aplikasi sesuai dengan pengawas pengecatan, termasuk di dalamnya kondisi cuaca, kelembapan udara, dan temperatur ruangan.

Berdasarkan referensi diatas maka penelitian ini berfokus pada **"Aplikasi Penghitung Kebutuhan Cat Tembok Pada Suatu Dimensi Ruang"** dengan tujuan membantu pengguna mengetahui jumlah kebutuhan cat yang tepat sebelum melakukan pengecatan, sehingga dapat menghemat biaya dengan menghindari pembelian yang berlebihan atau kurang dari yang diperlukan.

B. Rumusan Masalah

Berdasarkan latar belakang diatas, maka diperoleh rumusan masalah sebagai berikut yaitu :

1. Bagaimana membuatkan aplikasi penghitung kebutuhan cat tembok yang sederhana dan mudah digunakan oleh pengguna?

2. Bagaimana tingkat akurasi aplikasi dalam memberikan informasi mengenai besaran kebutuhan cat tembok yang akan digunakan?
3. Apa saja faktor-faktor yang mempengaruhi selisih antara hasil perhitungan manual dan hasil perhitungan aplikasi komputer?

C. Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk memungkinkan pengguna mengetahui jumlah kebutuhan cat yang diperlukan dengan *efektif* dan akurat, sehingga tidak memakan waktu lama dan dapat menghemat biaya.

D. Batasan Masalah

Batasan masalah dalam peracangan dan pembangunan aplikasi ini sebagai berikut :

1. Aplikasi dirancangan menggunakan bahasa pemrograman java *android* dan berjalan pada *sistem android.*
2. Produk cat pada aplikasi ini antara lain kansai paint, aries, avitex, dan no drop.
3. Pada halaman perhitungan, bidang yang tidak dicat hanya dapat dihitung jika bentuknya persegi atau persegi panjang.
4. Basis data yang digunakan adalah *SQLite* pada *Android Studio.*

E. Manfaat Penelitian

Adapun manfaat dari penelitian ini yaitu :

1. Bagi penulis

Dapat menambah wawasan dan penerapan bidang ilmu selama masa perkuliahan terutama bidang informatika.

2. Bagi pembaca

Memberikan pengetahuan tentang cara membuat aplikasi penghitung kebutuhan cat tembok menggunakan bahasa pemrograman java *android studio*.

3. Bagi pengguna

membantu melakukan perhitungan kebutuhan cat yang akan di gunakan untuk melakukan pengecatan pada suatu bangunan.

F. Sistematika Penulisan

Dalam menyusun sistematika penulisan, penulis menguraikan ke dalam lima bab yaitu :

BAB I : PENDAHULUAN

Menguraikan tentang latar belakang penulisan, rumusan masalah, tujuan penelitian, manfaat penelitian, dan batasan masalah, dan juga sistematika penulisan.

BAB II : TINJAUAN PUSTAKA

Pada Bab ini berisi uraian - uraian tentang tinjauan penelitian sebelumnya dan teori - teori yang digunakan sebagai *referensi* dalam membuat Aplikasi Penghitung kebutuhan cat pada suatu dimensi ruang.

BAB III : METODE PENELITIAN

Membahas tentang waktu dan tempat penelitian, metode penelitian, jenis penelitian, alat dan bahan yang digunakan dalam penelitian serta desain sistem.

BAB IV : HASIL DAN PERANCANGAN APLIKASI

Bab ini membahas tentang analisis sistem dengan UML, Perancangan aplikasi dan Implementasi.

BAB V : PENGUJIAN SISTEM

Bab ini berisi hasil pengujian yang telah dilakukan terhadap aplikasi penghitung kebutuhan cat tembok pada suatu dimensi ruang.

BAB VI : PENUTUP

Bab ini berisi kesimpulan dari hasil laporan penelitian tugas akhir dan saran agar penerapan aplikasi ini dapat dikembangkan lagi untuk penelitian selanjutnya.

BAB II

TINJAUAN PUSTAKA

A. Kajian Teori

1. Aplikasi

Aplikasi perangkat lunak (*Software Application*) adalah kategori perangkat lunak komputer yang secara langsung memanfaatkan kemampuan komputer untuk melaksanakan tugas tertentu yang diinginkan oleh pengguna. Ini biasanya berbeda dengan perangkat lunak sistem yang mengintegrasikan berbagai kemampuan komputer tanpa secara langsung mengaplikasikannya untuk tugas-tugas yang bermanfaat bagi pengguna. Contoh umum dari perangkat lunak aplikasi termasuk pengolah kata, lembar kerja, dan pemutar media.

Seringkali, beberapa aplikasi digabungkan menjadi satu paket yang disebut *suite* aplikasi (*application suite*). Contohnya adalah *Microsoft Office* dan *OpenOffice.org*, yang mengkombinasikan aplikasi pengolah kata, lembar kerja, dan beberapa aplikasi lainnya. Aplikasi dalam suatu paket biasanya memiliki antarmuka pengguna yang mirip, sehingga memudahkan pengguna untuk mempelajari dan menggunakan setiap aplikasi. Selain itu, aplikasi-aplikasi ini sering kali dapat berinteraksi satu sama lain, memberikan manfaat tambahan bagi pengguna. Misalnya, lembar kerja dapat dimasukkan ke dalam dokumen pengolah kata meskipun dibuat di aplikasi lembar kerja yang berbeda.

Menurut Huda dan Priyatna (2019), aplikasi adalah perangkat lunak yang berfungsi sebagai antarmuka depan (*front end*) pada sebuah sistem yang

digunakan untuk mengelola berbagai jenis data, sehingga menghasilkan informasi yang berguna bagi penggunanya dan sistem terkait.

2. Cat

Cat merupakan salah satu produk industri yang sangat penting saat ini, digunakan untuk melapisi permukaan bahan agar tampak lebih indah atau memiliki nilai lebih tinggi. Cat didefinisikan sebagai cairan yang diaplikasikan pada permukaan suatu bahan dengan tujuan memperindah, memperkuat, atau melindungi bahan tersebut. Setelah diaplikasikan dan mengering, cat membentuk lapisan tipis yang melekat kuat pada permukaan tersebut.

Penerapan cat pada permukaan dapat dilakukan dengan berbagai metode, seperti diusapkan, dilumurkan, dikuaskan, atau disemprotkan (Bently dan Turner, 1997). Secara umum, pembuatan cat melibatkan teknologi yang berkaitan dengan kimia organik dan kimia polimer. Proses ini memanfaatkan kimia antar permukaan, kimia koloid, elektrokimia, dan petrokimia (Tadros, 2013).

Emulsi adalah jenis koloid di mana fase terdispersi berupa zat cair dalam medium pendispersi yang bisa berupa padat, cair, atau gas. Cat tembok berbasis air disebut juga cat emulsi, yang melibatkan emulsi antara air dan minyak dalam formulanya. Dalam emulsi, setiap komponen pembentuknya sudah mengandung emulsifier berupa surfaktan. Bahan penyusun cat meliputi bahan pengikat (binder), bahan pengisi (filler), pigmen, pelarut, dan bahan aditif (Hradil dkk., 2003; Kent, 2012).

3. Algoritma

Algoritma merupakan langkah-langkah sistematis dan logis yang digunakan untuk menyelesaikan masalah atau mencapai tujuan tertentu dalam pembuatan perangkat lunak. Dalam sistem kerja komputer, terdapat tiga komponen utama yaitu *brainware, hardware, dan software*. Tanpa salah satu dari ketiga komponen tersebut, komputer tidak akan berfungsi dengan baik. Fokus utama kita adalah pada *software,* yang terdiri dari program dan *syntax* (cara penulisan/pembuatan program).

Untuk menyusun program atau *syntax*, diperlukan langkah-langkah yang sistematis dan logis, yang dikenal sebagai algoritma. Algoritma sangat penting dalam penyusunan program atau *syntax* karena memberikan struktur yang logis dan sistematis untuk memecahkan masalah atau mencapai tujuan tertentu. Dalam dunia komputer, algoritma berperan penting dalam pembangunan perangkat lunak. Bahkan dalam kehidupan sehari-hari, kita sering menggunakan algoritma tanpa menyadarinya (Abdul Kadir dan Heriyanto, 2013).

Mendefinisikan algoritma dalam pemrograman adalah langkah dasar yang harus diketahui dan merupakan awal dari proses pemrograman. Algoritma dapat disajikan dengan dua cara: tertulis atau ilustratif, tergantung pada metode atau tahapan yang digunakan oleh programmer. Presentasi tertulis sering menggunakan *pseudocode,* sementara ilustrasi biasanya berupa diagram alur. Tantangan utama adalah menerapkan algoritma yang ditentukan sehingga kode atau skrip dapat diubah dan dijalankan sesuai kebutuhan.

Pada dasarnya, fungsi utama algoritma adalah untuk memecahkan masalah. Algoritma pemrograman menawarkan berbagai keuntungan dan fungsi penting dalam pemrograman, antara lain:

a. Membantu pemrogram dalam memecahkan masalah kompleks yang mungkin melibatkan perhitungan berkualitas tinggi, sehingga mengurangi risiko kesalahan dalam program.

b. Menyederhanakan program besar menjadi lebih sederhana dan *efisien* dengan menggunakan pendekatan *top-down* atau *divide* and *conquer*.

c. Meminimalkan penulisan ulang kode karena algoritma dapat digunakan berulang kali.

d. Memudahkan pemrogram dalam menemukan dan memperbaiki kesalahan dengan cepat.

e. Membantu dalam mendeteksi kesalahan karena program terorganisir dan urut.

Algoritma adalah langkah-langkah untuk menyelesaikan masalah, sedangkan program adalah realisasi algoritma dalam bahasa pemrograman. Program ditulis dalam bahasa pemrograman tertentu, dan proses pembuatan program disebut pemrograman. Orang yang membuat program disebut programmer. Setiap langkah dalam program disebut instruksi, dan program ini terdiri dari serangkaian instruksi. Ketika instruksi dijalankan, operasi dilakukan sesuai dengan instruksi tersebut.

Secara umum, komputer terdiri dari empat komponen utama: perangkat *input*, perangkat *output*, prosesor utama, dan memori. Unit pemrosesan pusat

(CPU) adalah "otak" komputer yang melakukan operasi dasar seperti perbandingan, perhitungan, operasi baca dan tulis. Memori digunakan untuk menyimpan program dan data. Perangkat *input* dan *output* adalah alat yang digunakan komputer untuk mengomunikasikan hasil operasinya, seperti *keyboard*, *scanner*, *monitor*, dan *printer*.

Sebagai contoh, algoritma sederhana untuk memasak mie rebus melibatkan langkah-langkah tertentu yang harus dilakukan. Dalam pemrograman, algoritma serupa digunakan untuk menyelesaikan masalah. Misalnya, algoritma untuk menghitung nilai y dari persamaan y = 3x + 8 adalah:

a. Mulai

b. Tentukan nilai x

c. Hitung nilai y = 3x + 8

d. Cetak nilai x dan y

e. Selesai

Algoritma ini menunjukkan bahwa langkah-langkah penyelesaian masalah tidak selalu identik dengan penggunaan komputer, tetapi juga dapat diterapkan dalam berbagai situasi kehidupan sehari-hari.

4. *Research and development* (R&D)

Sugiyono (2009:407) menyatakan bahwa metode penelitian dan pengembangan digunakan untuk menghasilkan dan menguji efektivitas produk tertentu. Untuk menghasilkan produk, diperlukan penelitian analisis kebutuhan, yang dapat dilakukan melalui metode survei atau kualitatif. Untuk menguji

efektivitas produk agar dapat berfungsi di masyarakat luas, diperlukan penelitian *eksperimental*.

Lebih lanjut, Borg dan Gall (dalam Sugiyono:2009:11) menyatakan bahwa untuk analisis kebutuhan yang menghasilkan produk hipotetik, sering digunakan metode penelitian dasar. Selanjutnya, untuk menguji produk hipotetik tersebut, digunakan metode eksperimen atau penelitian tindakan (action research). Setelah produk teruji, produk tersebut dapat diaplikasikan.

Proses pengujian produk melalui eksperimen disebut penelitian terapan. Penelitian dan pengembangan bertujuan menemukan, mengembangkan, dan memvalidasi produk, sehingga bersifat longitudinal atau bertahap (bisa multi-tahun). Penelitian Hibah Bersaing yang didanai oleh Direktorat Jenderal Pendidikan Tinggi menggunakan metode penelitian dan pengembangan karena menghasilkan produk.

Produk yang ditemukan bisa berupa model, pola, prosedur, atau sistem. Dalam bidang pendidikan, produk yang dihasilkan melalui penelitian R&D diharapkan meningkatkan produktivitas pendidikan, menghasilkan lulusan yang banyak, berkualitas, dan relevan dengan kebutuhan. Produk pendidikan yang dihasilkan dapat berupa kurikulum spesifik, metode mengajar, media pendidikan, buku ajar, modul, kompetensi tenaga kependidikan, sistem evaluasi, model uji kompetensi, penataan ruang kelas untuk model pembelajaran tertentu, model unit produksi, model manajemen, sistem pembinaan pegawai, dan sistem penggajian (Sugiyono:2009:412).

Sukmadinata (2008:190) mengemukakan bahwa penelitian dan pengembangan merupakan pendekatan penelitian untuk menghasilkan produk baru atau menyempurnakan produk yang sudah ada. Produk yang dihasilkan bisa berupa perangkat lunak (software) atau perangkat keras (hardware) seperti buku, modul, paket, program pembelajaran, atau alat bantu belajar. Penelitian dan pengembangan berbeda dengan penelitian biasa yang hanya menghasilkan saran-saran perbaikan; penelitian dan pengembangan menghasilkan produk yang langsung bisa digunakan.

5. Java



Gambar 2. 1 Java

Java merupakan salah satu bahasa pemrograman yang banyak diminati Java adalah salah satu bahasa pemrograman yang sangat diminati oleh programmer karena kemampuannya untuk dijalankan di berbagai platform, termasuk telepon genggam. Sebelum popularitas sistem operasi *Android*, Java sudah menjadi bahasa pemrograman yang dominan untuk telepon genggam. Java berasal dari proyek penelitian perusahaan Sun Microsystems dengan nama sandi Green pada tahun 1991.

Ada prediksi bahwa *mikroprosesor* akan digunakan secara luas pada berbagai perangkat elektronik. Karena terdapat berbagai jenis *mikroprosesor*, diperlukan bahasa pemrograman yang dapat berjalan di semua *mikroprosesor*.

Oleh karena itu, James Gosling menciptakan bahasa pemrograman baru yang awalnya dinamai Oak. Nama Oak diambil dari pohon Oak yang bisa dilihat melalui jendela kantor James Gosling di Sun Microsystems.

Namun, setelah beberapa waktu, nama Oak ternyata sudah digunakan untuk bahasa pemrograman lain. Akhirnya, nama Oak diganti menjadi Java. Nama ini diambil setelah beberapa pegawai Sun mengunjungi kedai kopi dan melihat bahwa salah satu jenis biji kopi di kedai tersebut adalah biji kopi Jawa (Java).

Pada tahun 1995, Sun Microsystems secara resmi mengumumkan kehadiran Java. Bahasa pemrograman ini segera disambut hangat oleh masyarakat luas, terutama seiring dengan perkembangan dan ledakan era internet.

Tabel 2. 1 Versi Java

Versi Java	Tahun	Jumlah Class
1.0	1996	211
1.1	1997	477
1.2	1998	1524
1.3	2000	1840
1.4	2004	2723
5	2007	3279
6	2005	>3777

Saat ini, ada tiga API utama dari Java yang dikenal, yaitu:

a. Java 2 *Standard Edition* (J2SE)

Menyediakan lingkungan pengembangan yang kaya fitur, stabil, aman, *cross-platform*, dan dapat digunakan untuk aplikasi *standalone client-side* atau *applet*.

b. Java 2 *Enterprise Edition* (J2EE)

Menyediakan alat dan kerangka kerja untuk membangun dan menjalankan aplikasi *enterprise multitier* yang bersifat *server-side.*

c. Java 2 *Micro Edition* (J2ME)

Menyediakan platform untuk berbagai produk elektronik konsumen, seperti pager, *smart card*, telepon seluler, PDA, dan *set-top box.*

6. *Android studio*



Gambar 2. 2 *Android Studio*

Android Studio adalah *integrated development environment* (IDE) resmi untuk pengembangan aplikasi *Android* yang didasarkan pada IntelliJ IDEA. Untuk mendukung pengembangan aplikasi dalam sistem operasi *Android*, *Android* Studio menggunakan sistem *build* berbasis *Gradle*, *emulator*, kode template, dan integrasi dengan GitHub.

Sistem operasi adalah perangkat lunak utama yang mengelola dan mengontrol perangkat keras secara langsung serta mengatur dan mengontrol perangkat lunak lainnya agar dapat berfungsi dengan baik. Salah satu sistem operasi untuk perangkat mobile adalah *Android*. Sistem operasi (OS) *Android* dibangun di atas platform *Linux* yang bersifat *open source*. Sejalan dengan *Linux*, *Android* juga bersifat *open source*.

Dengan dukungan besar dari Google dan konsep open source, *android* dengan cepat mampu bersaing dan mengalahkan sistem operasi mobile lainnya seperti *Symbian*, *Windows Mobile*, *Blackberry*, dan iOS.

Android menggunakan bahasa XML untuk *scripting layout*-nya. Beberapa istilah yang populer dalam pemrograman *android* termasuk *activity* dan *fragment*, yang merupakan container untuk *user interface* (UI) dalam pemrograman *android*.

7. *SQLite*



Gambar 2. 3 *SQLite*

Menurut Putra, Budi, dan Kadafi (2020), *SQLite* adalah basis data open source yang tertanam dalam *Android*. *SQLite* mendukung fitur-fitur database relasional standar seperti *sintaks* SQL, fungsi transaksi, dan prepared *statement*. Selain itu, *SQLite* hanya membutuhkan sedikit memori saat dijalankan.

SQLite adalah perpustakaan perangkat lunak yang menerapkan sifat mandiri, tanpa server, tanpa konfigurasi, dan merupakan mesin transaksional basis data SQL. *SQLite* adalah mesin database SQL yang paling banyak digunakan di dunia. Berikut adalah beberapa fitur manajemen basis data *SQLite*:

1. Mandiri : *SQLite* tidak memerlukan server terpisah untuk operasinya. Basis data diintegrasikan langsung dalam aplikasi.

2. Tanpa Server : Tidak memerlukan instalasi atau konfigurasi server basis data, yang berarti lebih mudah untuk digunakan dalam aplikasi yang membutuhkan basis data ringan dan cepat.

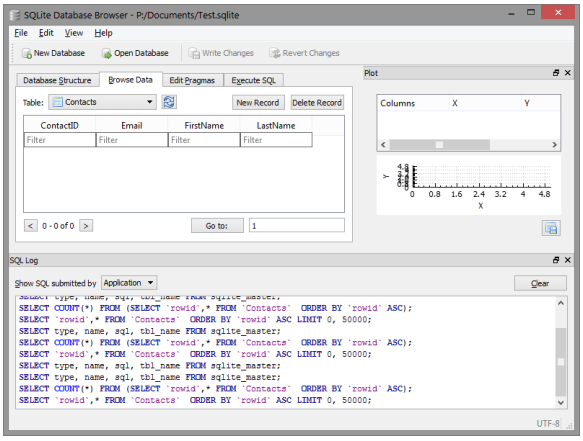
3. Transaksional : Mendukung transaksi yang memenuhi properti ACID (*Atomicity, Consistency, Isolation, Durability*).

4. Kecil dan Ringan : Dibandingkan dengan sistem basis data lainnya, *SQLite* memerlukan sedikit ruang memori dan penyimpanan.

5. Standar SQL : Mendukung sebagian besar fitur SQL standar, sehingga mudah digunakan oleh pengembang yang sudah terbiasa dengan SQL.

6. Cross-Platform : Dapat digunakan di berbagai platform, tidak hanya *Android* tetapi juga di sistem operasi lainnya.

Dengan fitur-fitur ini, *SQLite* menjadi pilihan populer untuk penyimpanan data dalam aplikasi *mobile* dan perangkat *embedded*.

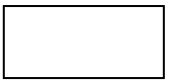
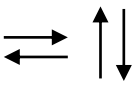
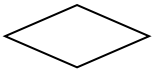
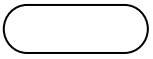
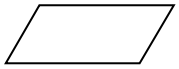


Gambar 2. 4 *SQLite Database Management*

8. *Flowchart*

Flowchart adalah prosedur untuk mendesain prosesor komputer (Miura F, 2013). Teknik ini bekerja untuk yang besar prosesor dan prosesor kecil. Prosesor memiliki dua bagian: "pengontrol" dan "eksekusi satuan. "Pengendali memberi tahu unit eksekusi apa yang harus dilakukan lakukan. Kontroler menentukan, lebih dari apapun yang lain, "kepribadian" prosesor.

Tabel 2. 2 Simbol *Flowchart*

No	Simbol-simbol	Keterangan Simbol
1		Simbol proses merupakan simbol yang digunakan untuk menunjukkan kegiatan proses dari operasi program komputer.
2		Simbol garis alir merupakan simbol yang digunakan untuk menunjukkan arus dari proses.
3		Simbol proses terdefinisi, merupakan simbol yang digunakan untuk menunjukkan suatu operasi yang rinciannya ditunjukkan di tempat lain.
4		Simbol keputusan merupakan simbol yang digunakan untuk suatu penyelesaian kondisi didalam program.
5		Simbol titik terminal merupakan simbol yang menunjukkan awal dan akhir dari suatu proses.
6		Simbol *input/output* merupakan simbol yang digunakan untuk mewakili data *input/output*.
7		Simbol proses berupa pencetakan laporan dan lain-lain

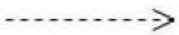
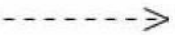
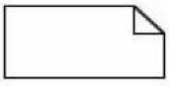
9. *Unified modelling langguage* (UML)

UML adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek.

a. *Use case diagram*

Use case adalah peringkat tertinggi dari fungsionalitas yang dimiliki sistem. Dengan kata lain, *use case* menggambarkan bagaimana seseorang akan menggunakan/memanfaatkan sistem (Sahrul, 2023).

Tabel 2. 3 Simbol *Use Case*

GAMBAR	NAMA	KETERANGAN
	*Actor*	Menspesifikasikan himpuan peran yang pengguna mainkan ketika berinteraksi dengan *use case*.
	*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
	Generalization	Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
	*Include*	Menspesifikasikan bahwa *use case* sumber secara *eksplisit*.
	Extend	Menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan.
	Association	Apa yang menghubungkan antara objek satu dengan objek lainnya.
	*System*	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
	*Use Case*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
	*Collaboration*	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
	*Note*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

b. *Activity diagram*

Activity diagram adalah salah satu cara untuk memodelkan *event-event* yang terjadi dalam suatu *use case*. *Activity* diagram digunakan untuk memodelkan aspek dinamis dari sistem. *Activity* diagram secara esensial mirip diagram alir (*flowchart*) (Sahrul, 2023).

Tabel 2. 4 Simbol *Activity Diagram*

Simbol	Nama simbol	Keterangan
	*Activity*	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
	*Action*	*State* dari *system* yang mencerminkan eksekusi dari suatu aksi
	*Initial Node*	Bagaimana objek dibentuk atau diawali
	*Activity Final Node*	Bagaimana objek dibentuk dan dihancurkan
	*Ford Node*	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

10. Metode pengujian

a. Pengujian *black box*

Pengujian *Black Box* adalah metode yang digunakan untuk memastikan bahwa program yang telah dikembangkan berjalan dengan baik. Pengujian ini sangat penting untuk menghindari kesalahan dalam alur program. Metode ini mengevaluasi perangkat lunak dari perspektif spesifikasi fungsional tanpa memeriksa desain atau kode internal program. Fokus utama pengujian *Black Box* adalah pada input dan output sistem, mengabaikan bagaimana *output* tersebut dihasilkan.

Pengujian ini bertujuan untuk mengidentifikasi berbagai jenis kesalahan, termasuk:

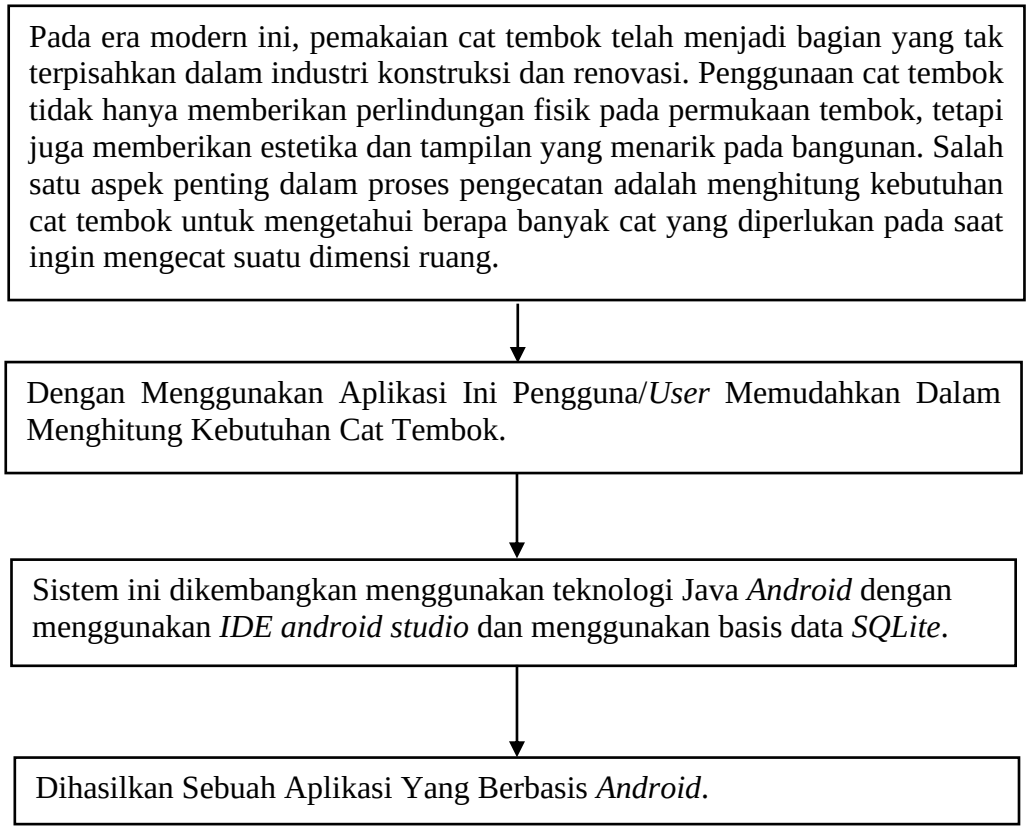
1. Fungsi yang tidak beroperasi dengan benar atau hilang. Pengujian ini memastikan bahwa semua fitur yang dijanjikan oleh perangkat lunak berfungsi sesuai harapan.
2. Kesalahan pada antarmuka. Pengujian ini memverifikasi bahwa komunikasi antara sistem dan pengguna berjalan dengan lancar tanpa adanya masalah.
3. Kesalahan pada struktur data atau akses database internal. Pengujian ini mengevaluasi bagaimana data disimpan, diakses, dan diolah dalam sistem.
4. Kesalahan kinerja. Pengujian ini memastikan bahwa perangkat lunak beroperasi secara efisien dan memenuhi standar kinerja yang telah ditentukan.

b. Pengujian *white box*

Pegujian *White Box testing* adalah metode pengujian yang didasarkan pada analisis detail desain dan struktur internal program. Dalam metode ini, penguji menggunakan struktur kontrol dari desain program secara prosedural untuk membagi pengujian menjadi beberapa kasus.

Fokus utama pengujian *white box* adalah pada logika internal dan kode program untuk memastikan bahwa semua jalur kode, kondisi, cabang, dan pernyataan diuji setidaknya sekali. Dalam pengujian *white box*, penguji memiliki akses penuh ke kode sumber dan berusaha menemukan kesalahan dengan menganalisis aliran data dan struktur kontrol dalam program.

11. Kerangka berfikir



12. Kajian hasil penelitian terdahulu

Dalam melakukan sebuah penelitian perlu adanya bahan pertimbangan sebagai acuan baik itu melalui berbagai jurnal maupun skripsi untuk memudahkan peneliti dalam menentukan kerangka awal. Dalam penelitian kali ini akan dicantumkan hasil dari penelitian sebelumnya :

Tabel 2. 5 Matriks Perbedaan Penelitian Terdahulu dan Sekarang

No	Judul penelitian	Persamaan	Perbedaan
1	Monica Octaviana (2020), Implementasi *Analytical Hierarchy Process* (AHP) Untuk Pemilihan Dan Penghitungan Volume Cat Tembok Berbasis *Web*	Tampilah pada menu user untuk rekomendasi pemilihan jenis cat	Aplikasi akan dibuat lebih simple (*frendly user*) sehingga semua kalangan *user* dapat dengan mudah menggunakan aplikasi tersebut

Adapun keterbaruan pada penelitian ini adalah aplikasi ini dapat Instal/diakses diperangkat *Android* sehingga semua pengguna yang memiliki *smarphone* dapat mengakses aplikasi ini.

BAB III

METODE PENELITIAN

A. Jenis Penelitian

Adapun jenis penelitian yang digunakan adalah jenis penelitian Kuantitatif. Menurut I Made Laut Martha Jaya (2020:12), Penelitian kuantitatif adalah jenis penelitian yang menghasilkan temuan-temuan baru yang dapat dicapai (diperoleh) dengan menggunakan prosedur-prosedur secara statistik atau cara lainnya dari suatu kuantifikasi (pengukuran). Penelitian dengan menggunakan pendekatan kuantitatif lebih memusatkan perhatian pada beberapa gejala yang mempunyai karakteristik tertentu di dalam kehidupan manusia, yaitu variabel. Dalam pendekatan kuantitatif, hakikat hubungan di antara variabel-variabel selanjutnya akan dianalisis dengan alat uji statistik serta menggunakan teori yang objektif.

Alasan pentingnya melakukan penelitian kuantitatif, antara lain sebagai berikut:

1. Peneliti dapat mengungkap fenomena-fenomena yang objektif dan dikaji secara kuantitatif.

2. Peneliti dalam menjawab pertanyaan penelitian dapat menggunakan pengukuran yang cermat terhadap variabel- variabel penelitian.

3. Peneliti ingin melakukan generalisasi dari hasil penelitian ke wilayah yang lebih luas.

B. Lokasi dan Waktu

Adapun lokasi tempat penelitian dilaksanakan di Toko Cat Warna-Warni tepatnya yang beralamat dijalan Andi Mappatola Ujung Sabbang Kec. Ujung Kota Parepare. Waktu penelitian dilakukan selama 1 (satu) bulan dimulai minggu ke 2 (kedua) bulan Februari sampai minggu ke 1 (pertama) pada bulan Maret 2024.

Tabel 3. 1 Jadwal Pelaksanaan Penelitian

No	Uraian kegiatan	February 2024			Maret 2024
		2	3	4	1
1	Studi literatur				
2	Perancangan sistem				
3	Pengujian hasil kerja				
4	Analisis hasil pengujian				

C. Alat dan Bahan

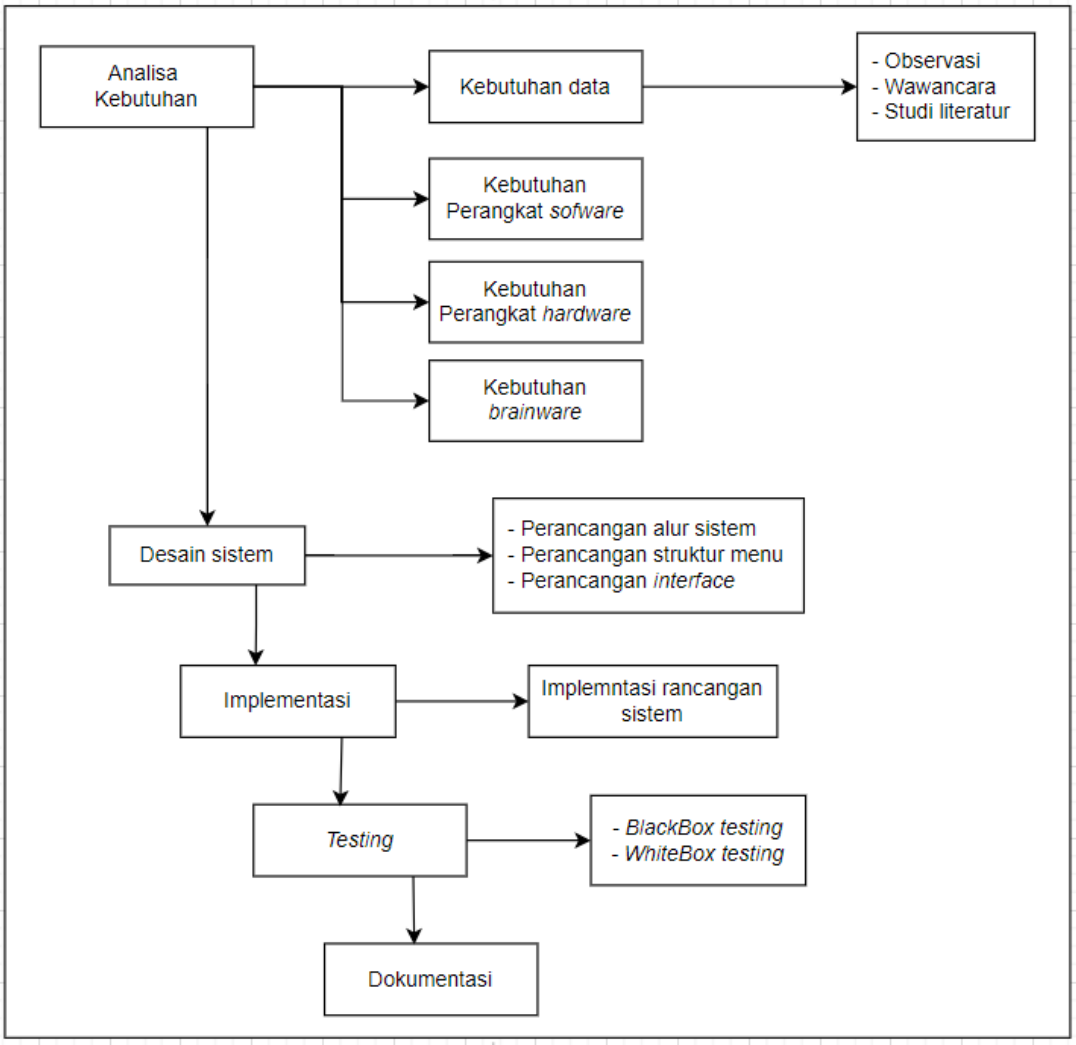
Dalam penelitian tugas akhir ini dibutuhkan alat dan bahan untuk menunjang produktifitas penelitian sehingga lebih *efisien* dalam merealisasikan rancangan sistem yang diinginkan.

Tabel 3. 2 Alat dan Bahan Penelitian

No	Uraian	Qty
1	Laptop Acer Swift SF314-57 Ram 4 GB	1 buah
2	Mouse Kabel	1 buah
3	Keyboard External	1 buah
4	Printer	1 buah

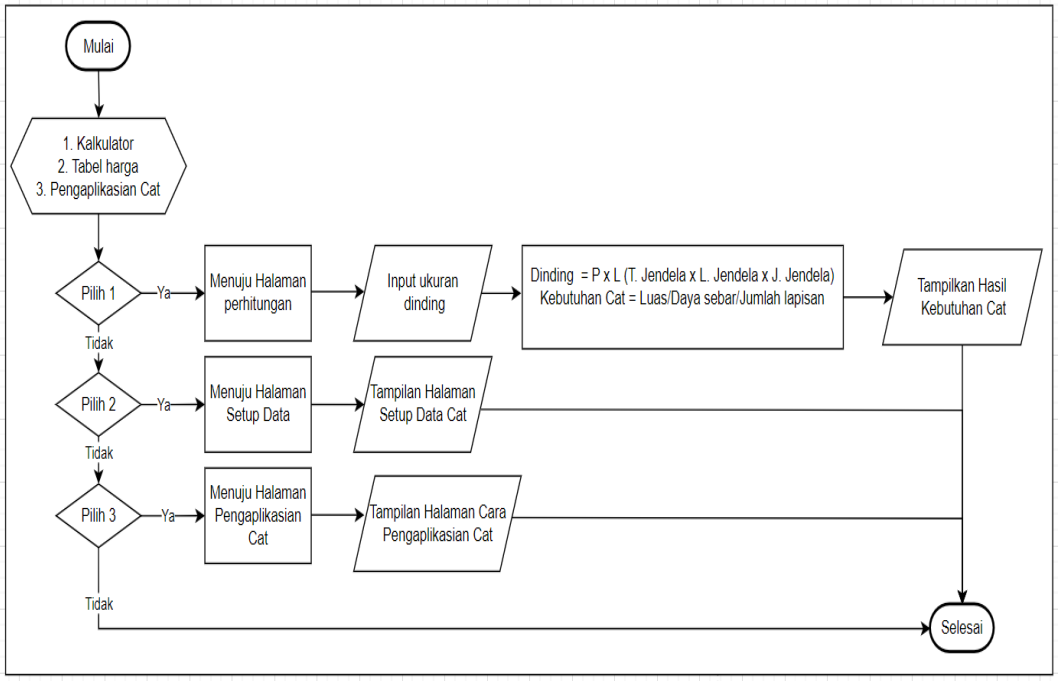
D. Alur Penelitian

Untuk menyelesaikan tugas akhir ini peneliti membuat gambar alur penelitian, berikut adalah alur penelitian yang dapat dilihat pada gambar dibawah ini.



Gambar 3. 1 Alur Penelitian

E. *Flowchart System*



Gambar 3. 2 *Flowchart System*

Pada gambar 3.2 menggambarkan alur proses dalam sebuah aplikasi yang memiliki tiga fitur utama: kalkulator kebutuhan cat, tabel harga cat, dan panduan pengaplikasian cat. Proses dimulai dengan pengguna memilih salah satu dari tiga opsi tersebut. Jika pengguna memilih kalkulator kebutuhan cat, mereka akan diarahkan ke halaman perhitungan di mana mereka perlu memasukkan ukuran dinding dan jendela. Aplikasi kemudian menghitung kebutuhan cat berdasarkan luas dinding, daya sebar cat, dan jumlah lapisan, serta menampilkan hasilnya. Jika pengguna memilih tabel harga, mereka akan diarahkan ke halaman yang memungkinkan mereka untuk melihat atau mengatur data harga cat. Pilihan ketiga adalah pengaplikasian cat, di mana pengguna akan diarahkan ke halaman yang memberikan panduan tentang cara menerapkan cat dengan benar. Setelah menyelesaikan salah satu dari tiga opsi tersebut, pengguna akan menuju ke tahap

akhir yang menandakan bahwa proses telah selesai. *Flowchart* ini menggambarkan bagaimana aplikasi memandu pengguna melalui berbagai langkah untuk memenuhi kebutuhan terkait cat dengan memberikan perhitungan, informasi harga dan panduan praktis.

F. Teknik Pengumpulan Data

Untuk menyelesaikan tugas akhir ini peneliti melakukan beberapa pendekatan yang meliputi :

1. Studi literatur

Pencarian referensi-referensi yang berhubungan dengan penyusunan tugas akhir, baik melalui internet ataupun buku- buku referensi yang meliputi Aplikasi menghitung kebutuhan cat, rumus besaran kebutuhan dan matematika sedeharna berdasarkan referensi yang telah terkumpul, dapat diambil kesimpulannya mengenai perancangan sistem, teknik pengerjaan, maupun metode-metode apa saja yang akan digunakan dalam penyelesaian tugas akhir ini. Pada tahap ini dilakukan pendalaman buku-buku *literature* yang berhubungan dengan menghitung kebutuhan cat serta berbagai macam materi tentang konsep pemrograman.

2. Perancangan sistem

Pada tahap ini, langkah awal yang dilakukan adalah membuat rancangan sistem secara keseluruhan. Setelah itu, setiap item akan diperjelas dengan detail-detail proses, beserta target yang akan dicapai pada setiap prosesnya.

G. Teknik Analisis Data

Teknik analisis data bertujuan untuk menguraikan dan memecahkan masalah yang berdasarkan data yang diperoleh. Analisis yang digunakan adalah analisis statistik Deskriptif. Teknik ini digunakan untuk menyajikan dan meringkas data yang terkumpul. Peneliti dapat menggunakan statistik deskriptif untuk menggambarkan jumlah total cat yang digunakan, rata-rata volume cat per area, atau distribusi kebutuhan cat untuk berbagai jenis permukaan.

BAB IV

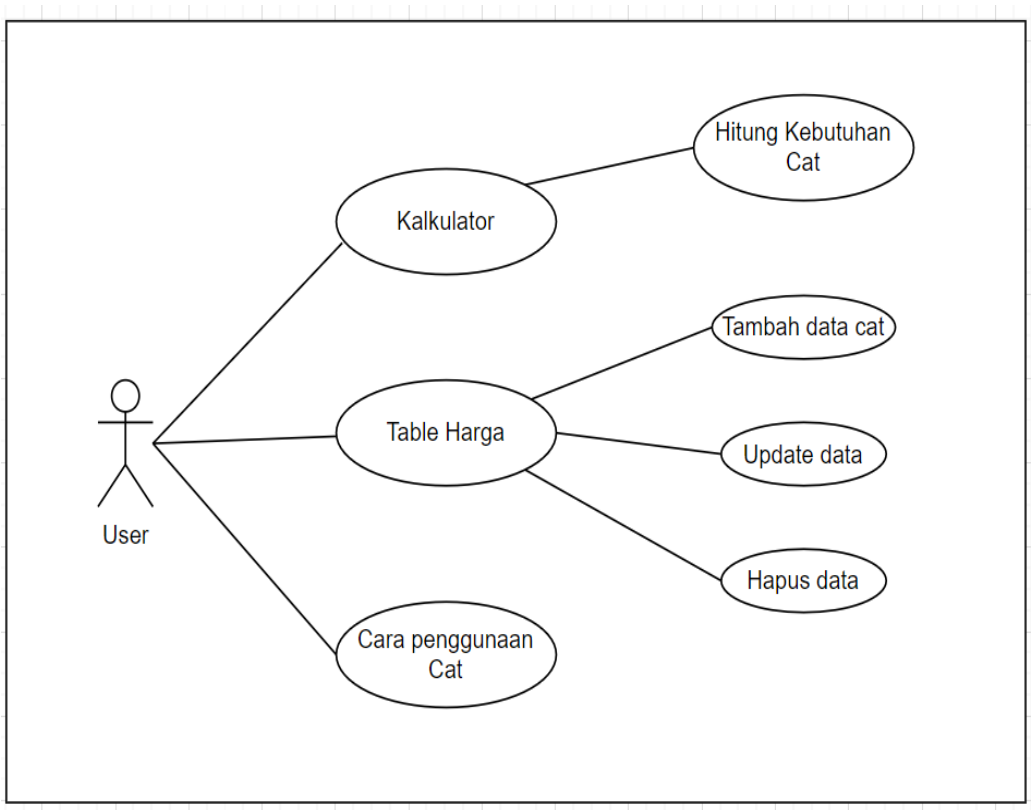
HASIL DAN PEMBAHASAN

A. Hasil Penelitian

1. *Use case diagram*

Use case diagram ialah sebuah kegiatan atau interaksi yang saling berkaitan antara aktor dan sistem. Atau secara umum, dapat diartikan sebagai sebuah teknik yang dimanfaatkan untuk pengembangan sebuah sistem, guna mengetahui kebutuhan fungsional dari sistem tersebut.

a. Aktor *user*



Gambar 4. 1 *Use Case Diagram User*

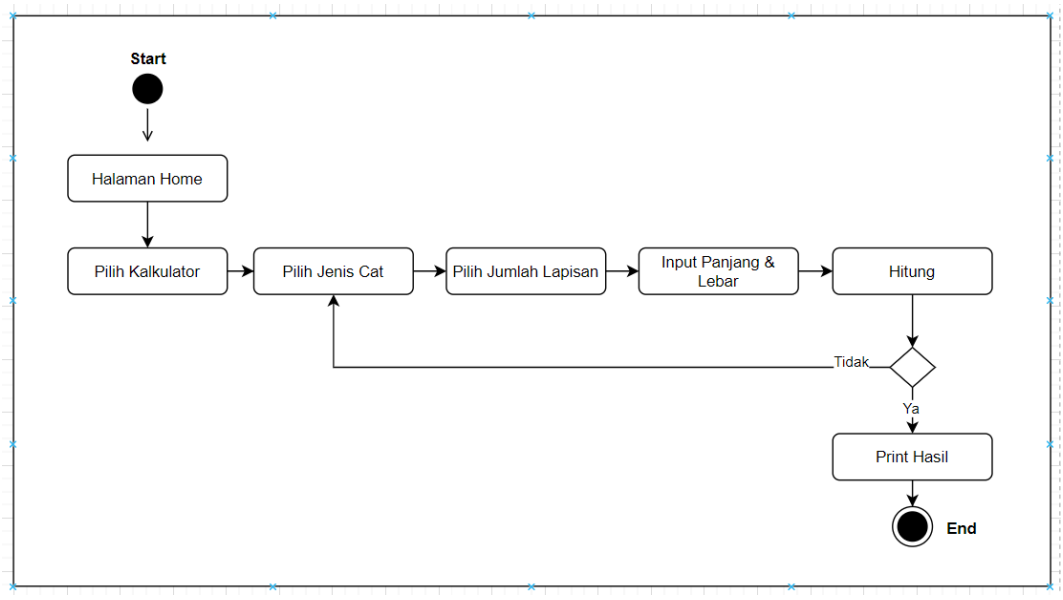
Tabel 4. 1 *Use Case Diagram*

No	Nama	Deskripsi
1	Kalkulator	*Use case* yang merupakan pokok dari aplikasi ini yaitu menghitung besaran kebutuhan cat yang akan digunakan
2	Tabel harga	*Use case* ini merupakan tempat data set merk cat sebelum memulai perhitungan.
3	Cara penggunaan	*Use case* ini merupakan *activity* yang didalam terdapat akses youtube online yang memutar video singkat mengenai cara penggunaan cat

2. *Activity diagram*

Activity Diagram menggambarkan berbagai alur aktivitas dalam system yang sedang di rancang, bagaimana masing-masing alur berawal, *decision* yang mungkin terjadi dan bagaiman mereka berakhir.

a. *Activity diagram* kalkutalor

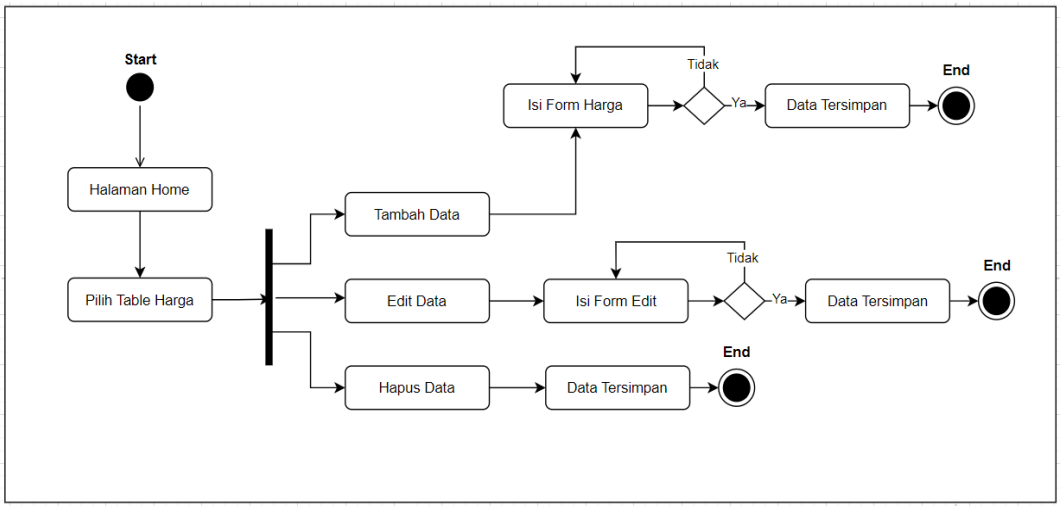


Gambar 4. 2 *Activity Diagram* Menu Kalkulator

Gambar 4.2 merupakan diagram aktivitas perhitungan cat yang akan diimplementasikan. Aktivitas ini diawali dengan memilih kalkulator kemudian akan tampil halaman perhitngan dan user memilih jenis cat yang akan digunakan,

memilih jumlah lapisan sesuai kebutuhan, dan memasukkan ukuran bidang yang akan dicat serta memasukan ukuran jendela jika bidang yang ingin dicat memiliki jendela.

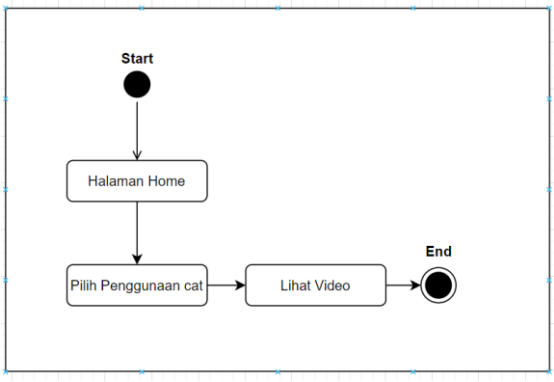
b. *Activity diagram* tabel harga



Gambar 4. 3 *Activity Diagram* Menu Tabel Harga

Gambar 4. 3 merupakan aktivitas diagram untuk mengelola *setup* data cat pada aplikasi yang akan diimplementasikan. Aktivitas ini diawali memilih table harga, kemudian akan tampil halaman setup data harga yang terdapat tambah data, edit data dan hapus data.

c. *Activity diagram* penggunaan cat



Gambar 4. 4 *Activity Diagram* Pengguaan Cat

Gambar 4. 4 Merupakan aktivitas diagram untuk mengetahui bagaimana cara penggunaan cat agar sesuai dengan prosedur yang akan diimplementasikan. Aktivitas ini diawali dengan memilih penggunaan cat kemudian akan tampil halaman penggunaan cat dan user diharapkan untuk menonton video yang telah disediakan

3. Perancangan sistem

Sistem yang akan dibangun adalah aplikasi berbasis *android* dimana semua *user android* dapat mengunduh aplikasi tersebut dan menggunakan aplikasi ini. *User* dapat menggunakan aplikasi ini menhitung berapa kebutuhan cat tembok yang perlukan untuk mengecat suatu dimensi ruang dan *user* dapat menambahkan merk cat kedalam aplikasi sehingga aplikasi *update* dengan pilihan cat yang tersedia dipasaran/toko cat.

Tabel 4. 2 Data Merk Cat

Merk	Daya Sebar	Estimasi Kering	Kemasan
KANSAI ECOCLEN	10-12 m^2/kg	15 menit	5 kg
KANSAI PROPERTI	10 m^2/kg	60-90 menit	5 kg
KANSAI TROPIC	10 m^2/kg	60 menit	5 kg
KANSAI OK-100	9-11 m^2/kg	60 menit	5 kg
KANSAI PEARL	10-12 m^2/kg	60 menit	5 kg
AVITEX GOLD	8-10 m^2/kg	60 menit	5 kg
NO DROP	10-12 m^2/kg	60 menit	5 kg
ARIES BLING	8-10 m^2/kg	60 menit	5 kg
ARIES GOLD	8-10 m^2/kg	60 menit	5 kg
ARIES TEMBOK	8-10 m^2/kg	60 menit	5 kg
NO DROP BASIC	10-12 m^2/kg	60 menit	5 kg

Data perhitungan kebutuhan cat pada dinding juga diperoleh dari pemilik toko. Banyaknya kebutuhan cat yang dibutuhkan dalam pengecatan dinding dapat diperkirakan sesuai dengan merk cat yang digunakan (daya sebar) dan kondisi pada dinding yang akan dicat.

Contoh kasus, untuk mengecat ruangan baru yang diplamir menggunakan cat merk KANSAI kemasan 5 kg dengan data seperti tampak pada tabel 4.2 perhitungan yang dilakukan adalah sebagai berikut:

Tabel 4. 3 Data untuk Penghitungan Kebutuhan Cat

Ruangan	**Jendela**
Panjang = 6 m	Tinggi = 2 m
Lebar = 5 m	Lebar = 1 m
	Jumlah = 1

Dengan rumus:

Luas dinding ruangan = panjang x lebar

= 6 meter x 5 meter = $30m^2$

Luas pintu dan jendela = (tinggi jendela x lebar jendela x jumlah jendela)

= 2 meter x 1 meter x 1 = $2m^2$

Luas bidang dinding = luas dinding ruangan – luas jendela

= $30m^2 – 2m^2 = 28m^2$

Untuk hasil yang optimal maka dibutuhkan 2 lapisan sehingga,

Kebutuhan cat = luas bidang dinding / daya sebar x jumlah lapisan

= $28m^2 / 9m^2/kg$ x 2 = 6.2kg

Berdasarkan perhitungan di atas, untuk mengecat permukaan dinding ruangan dengan luas $30m^2$, setelah mengurangi luas pintu dan jendela yang

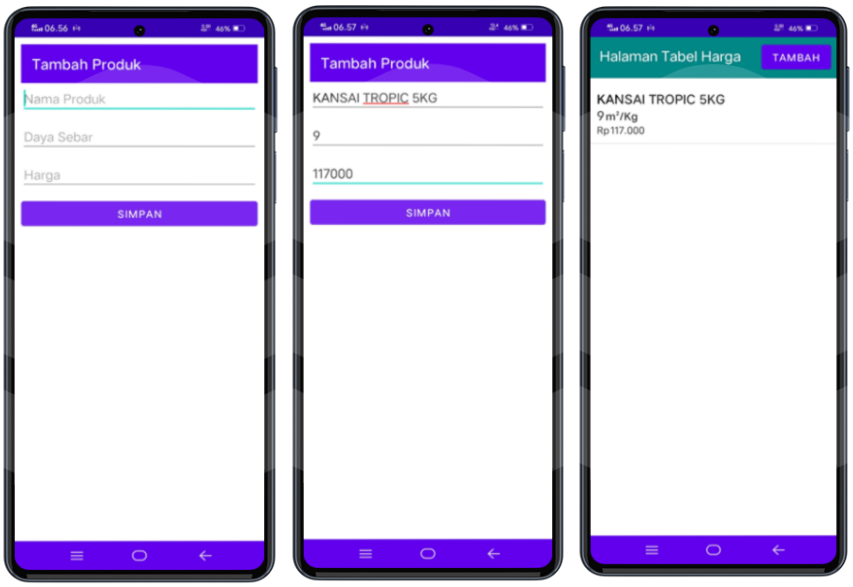
sebesar 2m², maka luas dinding yang akan dicat adalah 28m². Dengan menggunakan cat yang memiliki daya sebar 9m² per kilogram dan menerapkan dua lapisan cat, diperlukan kebutuhan cat sebanyak 6.2 kilogram. Dengan demikian, untuk mendapatkan hasil pengecatan yang optimal pada permukaan dinding ruangan ini, Anda perlu menyediakan 6.2 kg cat merek KANSAI.

a. Rancangan *input*

Bentuk rancangan *input* secara rinci Aplikasi penghitung kebutuhan cat tembok pada suatu dimensi ruang adalah sebagai berikut:

1). Tampilan tambah data cat

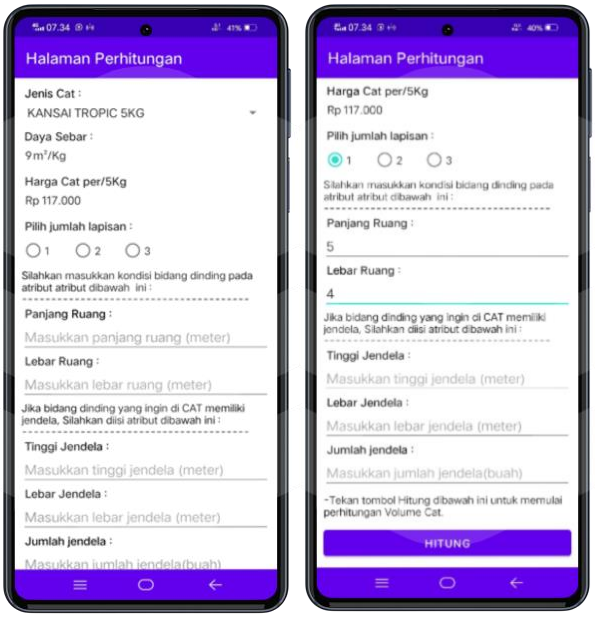
Gambar dibawah ini merupakan tampilan halaman untuk menambah data cat dimana ini merupakan salah satu parameter hasil *output* nantinya.



Gambar 4. 4 Tampilan Tambah Data

2). Tampilan halaman perhitungan

Gambar dibawah ini merupakan tampilan untuk menghitungan kebutuhan cat dimana *user* akan diminta untuk memasukkan panjang dan lebar pada bidang dinding.



Gambar 4. 5 Tampilan Halaman Perhitungan

a. Rancangan proses

Proses *count* pada aplikasi ini ialah dengan cara menggunakan rumus persegi kemudian dibagi dengan daya sebar cat dan mengalikan dengan jumlah lapisan yang dinginkan.

Pertama, *user* memasukkan nilai panjang dan lebar dinding pada aplikasi kemudian setelah itu, nilai panjang dan lebar akan dikalikan dan hasil luas itu akan proses dengan membagi antara luas dinding dengan daya sebar cat yang telah ditentukan serta mengalikan dengan jumlah lapisan yang pilih oleh user dan nilainya akan diproses menjadi hasil dari aplikasi ini.

b. Rancangan *output*

Bentuk rancangan *output* secara rinci dari Aplikasi penghitung kebutuhan cat tembok pada suatu dimensi ruang adalah sebagai berikut:

1). Tampilan Utama pada aplikasi setelah hasil proses pada tombol hitung



Gambar 4. 6 Halaman *Output* pada Halaman Hasil

4. Perancangan database

Rancangan database aplikasi ini menggunakan basis data *SQLite*, Berikut adalah tabel basis datanya.

Tabel 4. 4 Basis Data Harga

No	Nama	Tipe data	Keterangan
1	id	Int(12)	*Primary key*
2	nama_produk	Text(15)	
3	daya_sebar	Int(12)	
4	harga	Int(12)	

Tabel 4.5 merupakan struktur untuk menyimpan data produk dalam sebuah aplikasi. Tabel ini terdiri dari empat kolom, yaitu “id”, “nama_produk”, “daya_sebar”, dan “harga”. Kolom “id” memiliki tipe data “int(12)” dan berfungsi sebagai kunci utama (*Primary Key*) yang mengidentifikasi setiap entri secara unik. Kolom “nama_produk” menggunakan tipe data “Text(15)” untuk

menyimpan nama produk dengan panjang maksimum 15 karakter. Selanjutnya, kolom “daya_sebar” memiliki tipe data “Int(12)”, yang menyimpan informasi mengenai daya sebar produk, sedangkan kolom “harga” juga bertipe data “Int(12)” dan digunakan untuk menyimpan harga produk. Secara keseluruhan, tabel ini dirancang untuk mengelola informasi dasar tentang produk, termasuk identifikasi unik, nama, daya sebar, dan harga.

B. Implementasi

Penelitian ini berhasil mengembangkan aplikasi *android* yang bernama "Aplikasi penghitung kebutuhan cat tembok pada suatu dimensi ruang" yang dirancang untuk memudahkan pengguna dalam menghitung jumlah cat yang dibutuhkan untuk melapisi dinding suatu ruang dengan tepat. Implementasi dilakukan di Toko Cat Jalan Andi Mappatola Kel Ujung Kota Parepare, di mana pengguna dapat memanfaatkan aplikasi ini secara *efektif* untuk berbagai proyek pengecatan.

Proses implementasi melibatkan pelatihan kepada karyawan toko agar mereka dapat mengoperasikan aplikasi dengan mudah. Para pengguna diberikan pemahaman mengenai cara memasukkan data dimensi ruang, memilih tipe cat, serta menghitung jumlah cat yang diperlukan secara otomatis. Hasilnya, karyawan toko dapat memberikan estimasi kebutuhan cat kepada pelanggan dengan lebih cepat dan akurat. Hal ini tidak hanya meningkatkan efisiensi kerja karyawan, tetapi juga kepuasan pelanggan yang mendapatkan pelayanan yang lebih cepat dan tepat.

Evaluasi terhadap implementasi aplikasi menunjukkan hasil yang positif. Aplikasi "Penghitung kebutuhan cat" mampu mengurangi kesalahan perhitungan manual yang sering terjadi dan menghemat waktu dalam proses penentuan jumlah cat. Selain itu, *feedback* dari pengguna menunjukkan bahwa antarmuka aplikasi mudah digunakan dan sangat membantu dalam menentukan kebutuhan cat sesuai dengan spesifikasi yang diberikan. Secara keseluruhan, aplikasi ini berhasil memenuhi tujuan penelitian untuk mempermudah proses perhitungan kebutuhan

cat di Toko Cat Warna Warni, memberikan dampak positif bagi operasional toko, serta meningkatkan pengalaman pelanggan dalam berbelanja cat.

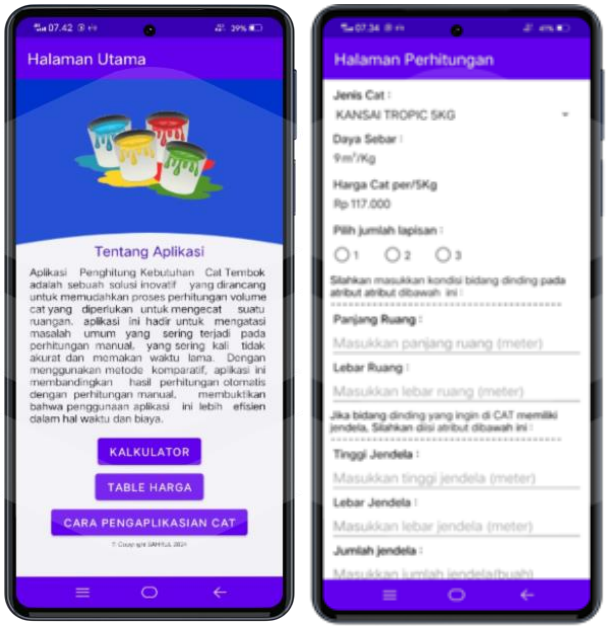
C. Metode Pengujian

1. *Blackbox*

Pengujian *black-box* berkaitan dengan pengujian yang dilakukan pada interface perangkat lunak. Pengujian *black-box* menguji beberapa aspek dasar suatu sistem dengan sedikit memperhatikan struktur logika internal perangkat lunak tersebut.

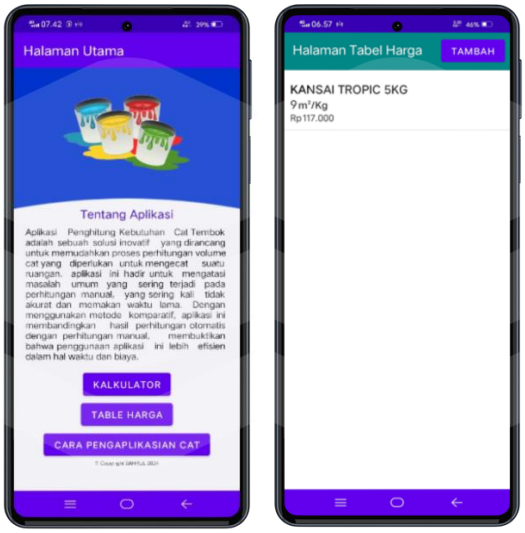
Gambar 4. 7 Halaman Menu Kalkulator

Test Faktor	Hasil	Kesimpulan
Proses memilih Kalkulator pada halaman utama	✓	Berhasil menampillkan halaman untuk perhitungan kebutuhan cat tembok
Sreenshot		



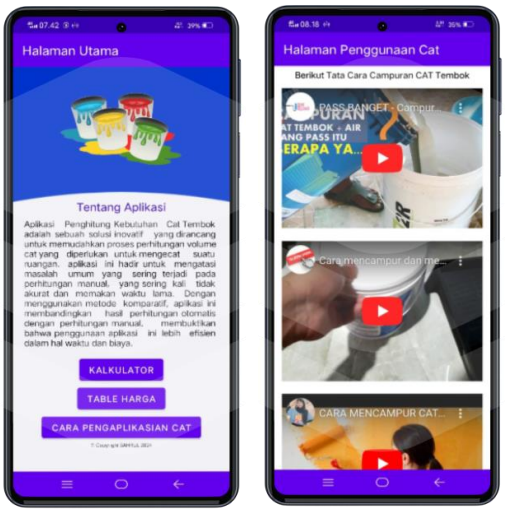
Gambar 4. 8 Halaman Menu Harga

Test Faktor	Hasil	Kesimpulan
Proses memilih Tabel harga pada halaman utama	✓	Berhasil menampillkan halaman untuk setup harga cat
Sreenshot		



Gambar 4. 9 Halaman Penggunaan Cat

Test Faktor	Hasil	Kesimpulan
Proses memilih Pengaplikasian cat pada halaman utama	✓	Berhasil menampillkan halaman untuk melihat video cara cat
Sreenshot		

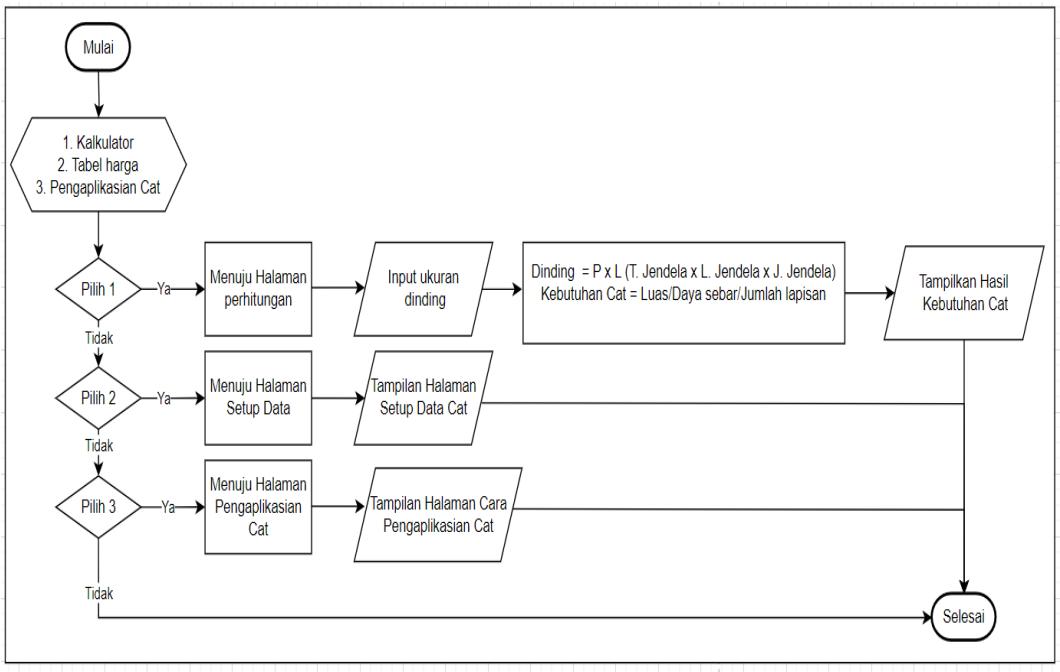


Dari beberapa pengujian *blackbox* diatas, dapat ditarik kesimpulan bahwa setiap fitur pada aplikasi ini dapat berfungsi sebagaimana rancangan yang usulkan pada penelitian ini.

2. *WhiteBox*

Pengujian ini merupakan pengujian yang didasarkan pada pengecekan terhadap detail perancangan, menggunakan struktur kontrol dari desain program secara *procedural* untuk membagi pengujian kedalam beberapa kasus pengujian.

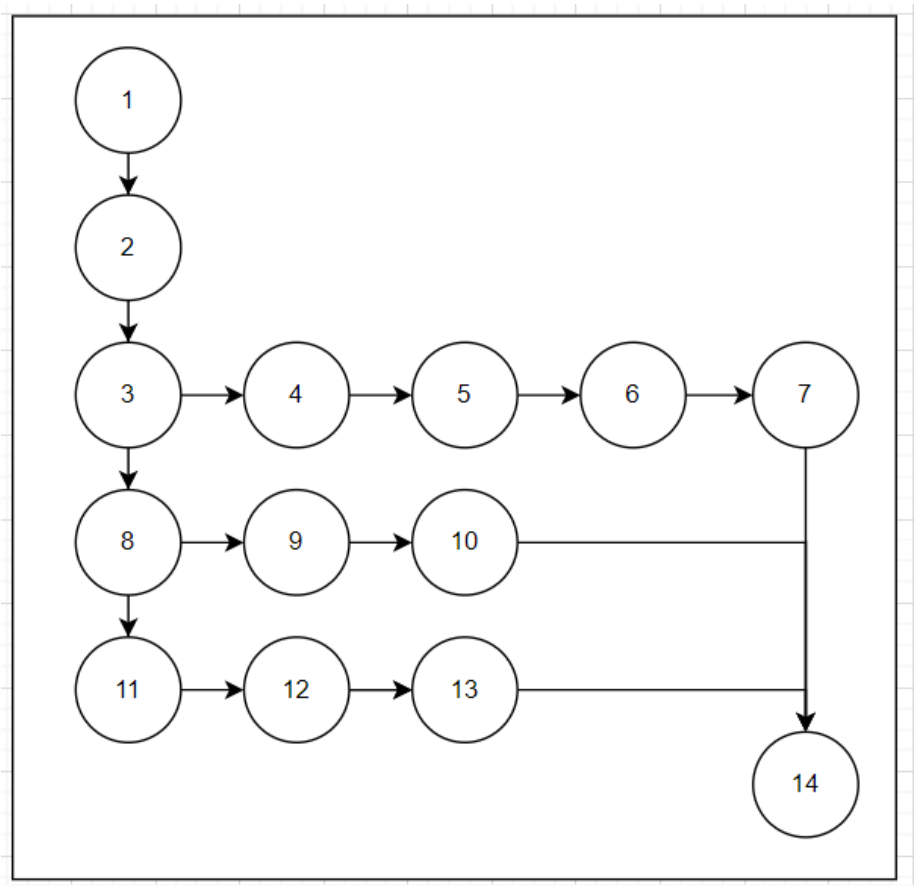
a. *Flowchart* halaman utama



Gambar 4. 10 *Flowchart* Halaman Utama

b. *Flowgraph*

Dari *Flowchart* yang digunakan untuk pengujian perangkat lunak, maka ditentukan *Flowgraph* sebagai berikut :



Gambar 4. 11 *Flowgraph* Halaman Utama

Dari diagram alir kegiatan pengelolaan di atas maka perhitungannya dapat dilakukan sebagai berikut:

1. *Counting cyclomatic complexity*

   Cyclomatic complexity (CC) adalah metrik yang digunakan untuk mengukur kompleksitas sebuah program. Rumus untuk menghitung CC adalah:

   CC = E – N + 2

   Di mana:

   E = jumlah *edge* (panah)

   N = jumlah *node* (simpul)

Dari *flowgraph* yang diberikan:

E = 15 (ada 15 panah)

N = 14 (ada 14 simpul)

Menggunakan rumus:

CC = 15 – 14 + 2 = 3

Jadi, *cyclomatic complexity* adalah

2. Jumlah *region*

   Jumlah *region* dalam *flowgraph* adalah jumlah area tertutup oleh *edge* dan *node* di dalam *graph*, termasuk area luar.

   Karena CC juga dapat dihitung dengan menghitung jumlah region:

   Jumlah *region = cyclomatic complexity*

   Jadi, jumlah *region* adalah 3.

   Region = 3.

3. *Independent path*

   Dari hasil perhitungan *cylomatic complexity* terdapat empat *independent path* yaitu:

   Path 1 = 1 -> 2 -> 3 -> 4 -> 5 -> 6 -> 7 -> 14

   Path 2 = 1 -> 2 -> 3 -> 8 -> 9 -> 10 -> 14

   Path 3 = 1 -> 2 -> 3 -> 8 ->11 -> 12 -> 13 -> 14

4. Matriks aktivitas admin

Berikut adalah tabel matriks *adjacency* dan matriks aktivitas admin berdasarkan *flowgraph* diatas.

Tabel 4. 5 Matriks *Adjacency*

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**E - 1**
1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1-1=0
2	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1-1=0
3	0	0	0	1	0	0	0	1	0	0	0	0	0	0	2-1=1
4	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1-1=0
5	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1-1=0
6	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1-1=0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1-1=0
8	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1-1=0
9	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1-1=0
10	0	0	0	0	0	0	1	0	0	0	0	0	0	1	2-1=1
11	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1-1=0
12	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1-1=0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1-1=0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E															2
SUM (E + 1)															2+1=3

Berdasarkan tabel 4.9 kita mendapatkan gambaran lengkap mengenai struktur dan kompleksitas *flowgraph*, serta jalur independen yang diperlukan untuk memastikan semua bagian dari *flowgraph* diuji.

3. Pengujian

Berikut adalah tabel pengujian menggunakan metode manual dengan hasil perhitungan aplikasi. Dengan melihat tabel yang disajikan kita dapat mengetahui berapa persentasi error pada aplikasi.

Tabel 4. 6 Perbandingan Aplikasi dengan Manual

No	**Nama produk**	**Ukuran bidang**		**Kebutuhan cat**			**Persentase**	
		Panjang (m)	**Lebar (m)**	**Manual (Kg)**	**Aplikasi (Kg)**	**Real (Kg)**	**Error (Kg)**	**Error (%)**
1	KANSAI	7	5	4	3,89	3,5	0,39	11%
2	KANSAI	15	4	7	6,6	6,5	0,1	1,5 %

Lanjutan Tabel 4. 6

No	Nama produk	Ukuran bidang		Kebutuhan cat			Persentase	
		Panjang (m)	Lebar (m)	Manual (Kg)	Aplikasi (Kg)	Real (Kg)	Error (Kg)	Error (%)
3	KANSAI	10	4	5	4,4	4	0,4	10 %
4	KANSAI	5	4	3	2,22	2	0,22	20 %
Total persentase error pada aplikasi								10,6 %

Total persentase error pada aplikasi adalah 10.6%. Ini mencerminkan bahwa secara keseluruhan, aplikasi memberikan hasil yang cukup akurat, namun masih ada beberapa penyimpangan yang dapat diperbaiki. Penyimpangan ini terutama disebabkan oleh faktor-faktor seperti permukaan dinding yang kasar dan campuran air dengan cat yang tidak sesuai. Dengan mempertimbangkan faktor-faktor ini dalam algoritma aplikasi, akurasi perhitungan dapat ditingkatkan.

BAB V

PENUTUP

A. Kesimpulan

Untuk membuat aplikasi penghitung kebutuhan cat tembok yang sederhana dan mudah digunakan, penting untuk fokus pada antarmuka yang *user-friendly* dan *intuitif*. Aplikasi harus memiliki fitur-fitur dasar seperti *input* dimensi ruang (panjang, lebar, tinggi) dan jenis cat yang akan digunakan. Selain itu, perlu ada instruksi yang jelas dan bantuan visual untuk membantu pengguna memahami cara menggunakan aplikasi.

Hasil pembuatan aplikasi menunjukkan bahwa aplikasi penghitung kebutuhan cat tembok memiliki tingkat akurasi yang cukup baik, dengan total persentase keberhasilan sebesar 89.4%. Meskipun terdapat beberapa perbedaan antara hasil perhitungan manual dan aplikasi, aplikasi ini masih memberikan estimasi yang cukup tepat untuk sebagian besar kasus.

Untuk meningkatkan akurasi, aplikasi harus memperhitungkan berbagai variabel seperti kondisi permukaan dinding (kasar atau halus) dan konsistensi campuran cat. Dengan memperbarui algoritma perhitungan untuk memasukkan faktor-faktor ini, aplikasi dapat memberikan estimasi yang lebih presisi dan mengurangi persentase error.

Dua faktor utama yang mempengaruhi selisih antara hasil perhitungan manual dan aplikasi adalah kondisi permukaan dinding dan campuran air dengan cat. Permukaan dinding yang kasar membutuhkan lebih banyak cat untuk

menutupi area secara merata, yang dapat menyebabkan aplikasi memberikan hasil yang berbeda dari perhitungan manual yang mungkin mengasumsikan permukaan yang lebih halus.

Selain itu, campuran air dengan cat yang tidak tepat dapat mengubah viskositas cat dan mempengaruhi jumlah cat yang diperlukan. Campuran yang terlalu encer atau terlalu kental dapat menyebabkan perbedaan signifikan dalam kebutuhan cat. Oleh karena itu, aplikasi harus memungkinkan pengguna untuk menginput kondisi permukaan dan konsistensi campuran cat untuk memberikan estimasi yang lebih akurat.

B. Saran

Untuk meningkatkan akurasi aplikasi penghitung kebutuhan cat tembok, sangat disarankan agar algoritma perhitungan diperbarui dengan mempertimbangkan kondisi permukaan dinding seperti tekstur kasar atau halus serta konsistensi campuran cat. Dengan demikian, aplikasi akan mampu memberikan estimasi kebutuhan cat yang lebih presisi dan andal bagi pengguna.

DAFTAR PUSTAKA

Abdul Kadir, 2013. *“ Pengenalan Algoritma “.,C.V. Andi Offset.* Yogyakarta: Andi.

Atmaja, A. P., & Azis, A. (2019). *Sistem Informasi Terintegrasi Evaluasi Kegiatan Mengajar Dosen Sebagai Implementasi Sistem Penjaminan Mutu Internal. Matrix:* Jurnal Manajemen Teknologi Dan Informatika, 9(1), 1–6.

Bently, J., Turner, G.P.A. (1997). *Introduction to Paint Chemistry and Principles of Paint Technology*, CRC Press, Bristol UK.

Dewi, N. K., Irawan, B. H., Fitry, E., & Putra, A. S. (2021*). Konsep Aplikasi E-Dakwah Untuk Generasi Milenial Jakarta.* Jurnal IKRA-ITH Informatika, 5(2), 26–33.

F. Miura et al., *“Flowchart for the Management of Acute Cholangitis and Cholecystitis*,” J. Hepatobiliary. Pancreat. Sci., 2013.

Faizal Okie, Prabowo. (2015*). Aplikasi Pemetaan Pelayanan Umum Di Kabupaten Brebes Dengan Gis.* Semarang: Faizal.

Huda, B., & Priyatna, B. (2019). *Penggunaan Aplikasi Content Management System (CMS) Untuk Pengembangan Bisnis Berbasis E-commerce.* Systematics, 1(2), 81. https://doi.org/10.35706/sys.v1i2.2076

Huda, I. A. (2020). *Perkembangan Teknologi Informasi Dan Komunikasi (Tik) Terhadap Kualitas Pembelajaran Di Sekolah Dasar*. Jurnal Pendidikan Dan Konseling (JPDK), 2(1), 121–125. https://doi.org/10.31004/jpdk.v1i2.622

H. Dhika, *“Implementasi Teknologi J2me pada Sistem Administrasi Akademik Mahasiswa di Universitas Indraprasta PGRI Jakarta,”* Factor Ecaxta, vol. 5 (1), pp. 15– 27, 2015.

Haryati, Sri. (2012). *Researh and Development (R&D) sebagai salah satu model penelitian dalam bidang pendidikan.* FKIP-UTM: Universitas Trunojoyo Madura.

Hari, H., & Zaqi Al Faritsy, A. (2022*). Analisis Pengendalian Kualitas Produk Ember Cat Tembok 5kg Menggunakan Metode New Seven Tools.* Jurnal Teknologi Dan Manajemen Industri Terapan, 1(3), 231–242.

Hradil, D., Grygar, T., Hradilova, J., Bezdicka, P. (2003). *Clay and iron oxide pigments in the history of painting.* Applied Clay Science , 22, 223 – 236.

Ilmi Abdillah, R. G., & Basuki, M. (2022). *Analisis Teknis Dan Ekonomis Pemakaian Ketebalan Cat Dan Metode Pengecatan Pada Lambung Bmpp (Barge Mounted Power Plant) Di Pt. Pal Indonesia (Persero).* Jurnal

Sumberdaya Bumi Berkelanjutan (SEMITAN), 1(1), 423–429. https://doi.org/10.31284/j.semitan.2022.3258

Maqvira. (2024). *Jurnal Fakultas Teknik Aplikasi Edukasi Anak Usia Dini Berbasis Kamera*. 4(2), 1–12.

Made Laut Mertha Jaya, I. (2020*). Metode Penelitian Kuantitatif dan Kualitatif: Teori, Penerapan, dan Riset Nyata.* Yogyakarta: Anak Hebat Indonesia.p

Maya. (2015). *Membangun Sistem Informasi dengan Java NetBeans dan MySQL*. Semarang: C.V ANDI OFFSET.

Nasution, A., Efendi, B., & Kamil Siregar, I. (2019). *Pelatihan Membuat Aplikasi Android Dengan Android Studio Pada Smp Negeri 1 Tinggi Raja.* Jurdimas (Jurnal Pengabdian Kepada Masyarakat) Royal, 2(1), 53–58. https://doi.org/10.33330/jurdimas.v2i1.321

NetBeans, *"NetBeans IDE,"* Features, 2015.Putra, Raden Bagus Dimas, Eko Setia Budi, and Abdul Rahman Kadafi. 2020. *"Perbandingan Antara SQLite, Room, Dan RBDLiTe Dalam Pembuatan Basis Data Pada Aplikasi Android."* JURIKOM (Jurnal Riset Komputer)

Rivki, M., Bachtiar, A. M., Informatika, T., Teknik, F., & Indonesia, U. K. (2020). *No Title*. *112*.

Rizky Ferdiansyah, Iyan Bachtiar, & Selamat. (2023). *Pengendalian Kualitas dengan Metode Taguchi pada Produk Cat Tembok di Pt XYZ.* Jurnal Riset Teknik Industri, 129–138. https://doi.org/10.29313/jrti.v3i2.2890

Susandi, D., & Sukisno, S. (2017). *Sistem Penjualan Berbasis E-Commerce Menggunakan Metode Objek Oriented pada Distro Dlapak Street Wear*. JSiI (Jurnal Sistem Informasi), 4, 5–8.

Sifauttijani, F. (2017). *Pencarian rumah makan berbasis android.* Jurnal SIMETRIS, 8(1), 309–316.

Sugiyono. (2009). *Metode Penelitian Pendidikan Pendekatan Kuantitatif, Kualitatif, dan R & D.* Bandung: Alfabeta.

Sukmadinata, Nana Syaodih. (2008). *Metode Penelitian Pendidikan.* Bandung: Remaja Rosdakarya.

Sitohang, H.T. (2018*). Sistem Informasi Pengagendaan Surat Berbasis Web Pada Pengadilan Tinggi Medan.* Jurnal Of Informatic Pelita Nusantara. 3(1). 6-9.

Tadros, T. (2013). *Paints and Coatings, Encyclopedia of Colloid and Interface Science.* Springer-Verlag Berlin Heidelberg.

Universitas Muhammadiyah Parepare. (2021). *Panduan Penulisan Proposal & Skripsi.* Parepare : FT-UMPAR

Widodo, Joko. (2005). *Perencanaan Pendidikan di Sekolah Kejuruan; Disertasi*. Bandung: PPS IKIP Bandung.

Y. Bai, *"Introduction to NetBeans IDE," in Practical Database Programming with Java,* 2011, pp. 155–315.

Y. Setiowati, *"Pendahuluan tentang Netbeans,"* 2006.