

BAB I

PENDAHULUAN

A. Latar Belakang

Anak, dalam beragam usia dengan berbagai perilakunya biasanya menarik perhatian orang dewasa. Dunia anak adalah dunia yang penuh dengan canda tawa dan kegembiraan sehingga orang dewasa akan terhibur dengan melihat tingkah polah mereka. Anak memiliki karakteristik tertentu yang khas dan tidak sama dengan orang dewasa, mereka selalu aktif, dinamis, antusias dan ingin tahu terhadap apa yang dilihat, didengar, dirasakan, mereka seolah-olah tak pernah berhenti bereksplorasi dan belajar. Anak usia dini adalah sosok individu yang sedang menjalani suatu proses perkembangan dengan pesat dan fundamental bagi kehidupan selanjutnya. Yang dimaksud anak usia dini yaitu dalam rentang umur dari lahir sampai enam tahun merupakan usia yang sangat menentukan dalam pembentukan karakter dan kepribadian anak.

Maka dari itu pendidikan merupakan salah satu upaya untuk membuat masyarakat mengembangkan potensi manusia agar memiliki kekuatan spiritual keagamaan, pengendalian diri, berkepribadian, memiliki kecerdasan, berahlak mulia, serta memiliki ketrampilan yang diperlukan sebagai anggota masyarakat dan warga negara. Dalam hal ini pendidikan penting bagi anak usia dini. Dalam Jari jemari anak menggoreskan cairan warna-warni di atas selembar kertas. Goresan jari- jemari mungil itu akhirnya menghasilkan sebuah karya lukisan abstrak

yang penuh warna. Bahan yang digunakan ini adalah tepung kanji yang dicampur dengan pewarna. Kegiatan ini merupakan salah satu metode yang bermanfaat untuk merangsang atau menstimulan motorik anak.

Berdasarkan permasalahan diatas maka peneliti tertarik untuk melakukan penelitian dengan judul “Pembelajaran Aplikasi Finger Painting Untuk Meningkatkan Keterampilan Anak Usia Dini Di Kota Parepare”.

B. Rumusan Masalah

Dari latar belakang masalah yang ditemukan diatas maka pokok permasalahan yang terdapat dalam penelitian ini adalah Sebagai Berikut:

1. Aplikasi Finger Painting Untuk Meningkatkan Keterampilan Anak Usia Dini?
2. Bagaimana cara mengetahui cara menggunakan Aplikasi Finger Painting Untuk Meningkatkan Keterampilan Anak Usia Dini?

C. Batasan Masalah

Batasan masalah dalam penelitian tugas akhir ini adalah sebagai berikut :

1. Aplikasi yang dirancang pada aplikasi Website dan Android.
2. Aplikasi yang ditampilkan dalam bentuk gambar yang akan di warnai oleh anak-anak

D. Tujuan Penelitian

Adapun tujuan yang ingin dicapai pada proposal penelitian ini yaitu Pembelajaran Aplikasi Finger Painting Untuk Meningkatkan Keterampilan Anak Usia Dini.

E. Manfaat Penelitian

1. Manfaat bagi mahasiswa

Sebagai penerapan teori yang telah di dapatkan dari bangku kuliah dan diharapkan dapat menambah pengetahuan dan wawasan tentang bagaimana cara membuat aplikasi pada website .

2. Manfaat bagi penulis

Menambah pengetahuan, pengalaman, serta wawasan tentang metode perkembangan dan melatih diri dalam menyelesaikan masalah untk maju dan berkembang.

BAB II

TINJAUAN PUSTAKA

A. Penelitian Terdahulu

Dewi Dewi Fitriani, Aisyah, Novi Arvionta, Jurnal Bunayya Vol. 6 No. 1 Tahun 2020, *Aplikasi Kegiatan Main Finger Painting dalam Peningkatan Kreativitas Motorik Halus Anak Usia Dini*. Penelitian ini bertujuan melihat aktivitas apa pada guru yang dapat meningkatkan mutu proses pembelajaran terutama pada kegiatan main *finger painting*. Penelitian ini juga melihat apakah terjadi peningkatan pada nilai kreativitas anak di aspek perkembangan motorik halus. Jenis penelitian ini adalah penelitian tindakan kelas yang menggunakan dua siklus. Data dikumpulkan melalui *check list* observasi guru dan kreativitas anak dan dianalisis secara naratif deskriptif. Hasil penelitian menunjukkan bahwa dengan adanya panduan *check-list* sebanyak 26 kegiatan aktivitas guru, terjadi peningkatan yang lebih baik pada hasil kinerja guru didalam kelas. Hasil penelitian aktivitas guru menunjukkan peningkatan persentase pada kegiatan yang dilakukan oleh guru di siklus 1 dari 76, 91% menjadi 92,29% di siklus II. Kemudian, kemampuan kreativitas motoric halus anak usia 5-6 tahun dalam kegiatan finger painting dengan menggunakan jari-jari tangan juga dinilai berhasil. Nilai kreativitas ini meningkat dari siklus I sebesar 53,9% menjadi 84,7% pada siklus II. Hal ini terjadi dengan adanya

1. pemberian variasi jenis gambar dan warna sesuai dengan keinginan anak serta intensitas penggunaan jari jemari anak pada saat bermain *finger painting*.
2. Dewa Ayu Putri Ariska Pinatih, Dr. M.G. Rini Kristiantari, M.Pd., Drs. I Ketut Ardana, M.Pd, e-Journal PG-PAUD Undiksha, *Meningkatkan Kemampuan Motorik Halus dalam Menulis Dengan Metode Pemberian Tugas Berbantuan Media Gambar Pada Anak Kelompok B2 Semester II di PAUD Kusuma 2 Denpasar Tahun Ajaran 2014/2015*. Penelitian ini bertujuan untuk mengetahui peningkatan kemampuan motorik halus anak dalam menulis setelah diterapkannya metode pemberian tugas berbantuan media gambar pada anak kelompok B2 Semester II di PAUD Kusuma 2 Denpasar Tahun Ajaran 2014/2015. Jenis Penelitian ini adalah Penelitian Tindakan Kelas (PTK) yang dilaksanakan dalam dua siklus. Subjek penelitian adalah 20 orang anak. Pengumpulan data dalam penelitian ini dilakukan dengan metode observasi dengan instrumen lembar observasi. Data hasil penelitian yang diperoleh dianalisis dengan menggunakan teknikanalisis statistik deskriptif kualitatif dan analisis statistik deskriptif kuantitatif. Hasil penelitian bahwa terjadi peningkatan kemampuan motorik halus dalam menulis dengan menerapkan metode pemberian tugas berbantuan media gambar pada siklus I sebesar 61.1% menunjukkan pada kategori rendah kemudian mengalami peningkatan pada siklus II menjadi 85.7 % pada kategori tinggi. Berdasarkan hasil penelitian dan uraian tersebut dapat disimpulkan bahwa melalui penerapan metode pemberian tugas berbantuan media gambar dapat meningkatkan

kemampuan motorik halus dalam menulis pada anak kelompok B2 semester II di PAUD Kusuma 2 Denpasar sebesar 24.6%.

3. Resty Lisdayanti, M. Syukri, Desni Yuniarni, *Journal Pendidikan dan Pembelajaran Khatulistiwa*, *Pembelajaran Melukis Teknik Finger Painting Untuk Meningkatkan Perkembangan Motorik Halus di TK Islamiyah Pontianak*. Penelitian ini bertujuan untuk mengetahui peningkatan perkembangan motorik halus pada pembelajaran seni rupa melalui teknik lukis jari pada anak usia 5-6 tahun di Pontianak Tenggara TK Islamiyah. Tujuan umum dalam penelitian ini adalah untuk mendeskripsikan belajar melukis melalui teknik lukis jari untuk meningkatkan motorik halus perkembangan pada anak usia 5-6 tahun. Metode penelitian yang digunakan adalah deskriptif dengan pendekatan kualitatif dan jenis penelitian studi kasus. Sumber data ini penelitian melibatkan 1 orang guru dan anak kelompok B3 di Islamiyah yang berjumlah 16 orang orang sebagai subjek. Setelah data terkumpul, maka data disajikan untuk menarik kesimpulan.

B. Finger Painting

1. Konsep *Finger Painting*

Finger painting atau menggambar dengan jari adalah teknik melukis dengan jari tangan secara langsung tanpa menggunakan bantuan alat. Jenis kegiatan ini dilakukan dengan mengoleskan adonan warna (bubur warna) menggunakan jari tangan di atas bidang gambar. Batasan jari yang digunakan adalah semua jari tangan, telapak tangan, sampai pergelangan tangan.¹⁰ Jadi *finger painting*

adalah suatu kegiatan yang mengajak anak-anak untuk bermain warna-warni dengan menggunakan jari. Anak-anak bebas mengaplikasikan warna-warni tersebut hingga membentuk suatu gambar yang diinginkan. Menurut Siti Aisyah, *finger painting* adalah salah satu bentuk menggambar yang berharga dan merupakan ekspresi spontan.¹¹ Jadi *finger painting* adalah salah satu melukis yang dilakukan secara langsung. Dengan menuangkan adonan warna dan menggambar di atas bidang kertas tanpa dapat dihapus. Untuk itu satu kali hasil sudah cukup.

Sementara itu, *finger painting* adalah teknik melukis dengan tangan secara langsung menggunakan adonan warna.¹² Jadi *finger painting* adalah salah satu cara yang dipergunakan untuk melukis dengan menggunakan tangan dari ujung jari hingga pergelangan tangan. Jadi dapat disimpulkan bahwa *finger painting* adalah sebuah kegiatan di mana anak diajak untuk bermain warna tanpa takut kotor, dengan mengaplikasikan adonan warna ke atas buku gambar. Dalam mengaplikasikan warna anak menggunakan jari tangan, telapak tangan, sampai pergelangan tangan secara langsung. Anak bebas mengaplikasikan adonan warna sesuai dengan apa yang ada dipikirannya, seperti apa yang pernah anak lihat di masa lalu ataupun berdasarkan pengembangan dari imajinasi anak secara langsung.

Teknik *finger painting* dipergunakan untuk memahami tatacara melakukan kegiatan *finger painting* yang benar diantaranya;

- a. Melakukan *blocking*. *Blocking* dalam teknik *finger painting* adalah membuat sketsa atau gambar rancangan langsung menggunakan adonan

warna. Lalu seluruh area sketsa warna diwarnai dengan adonan warna.

Tahapan ini dilakukan langsung dengan menggunakan jari.

- b. Menerapkan berbagai teknik tarikan jari untuk membuat gambar yang diinginkan. Seperti 1 jari spiral dan titik, 1 jari lurus, 1 jari putar, 1 jari melingkar, 1 jari lengkung angkat, 2 garis serong, 2 jari putar, 3 jari bergelombang, 3 jari lengkung putar, dan 3 jari putar. Khusus teknik 1 jari spiral dan titik dapat digunakan tanpa melakukan blocking terlebih dahulu.¹³ Jadi *finger painting* dapat dilakukan dengan dua teknik yang pertama adalah teknik memenuhi semua bidang kertas dengan adonan warna. Selanjutnya teknik menarik garis dimana peserta didik dapat mengaplikasikan warna-warni dengan tarikan jari hingga membentuk gambar yang diinginkan

2. Manfaat *Finger Painting* bagi Perkembangan Anak

Setiap kegiatan pasti memiliki tujuan yang akan dicapai oleh anak yang melakukan kegiatan tersebut. Selain tujuan yang dapat dicapai suatu kegiatan juga dapat bermanfaat bagi anak yang melakukan kegiatan tersebut. *Finger painting* memiliki banyak tujuan dan manfaat yang dapat diperoleh atau dirasakan oleh anak usia dini. Tujuan akan tercapai apabila terjadi interaksi antara guru dengan peserta didik sehingga ada proses timbal baliknya. Berikut merupakan tujuan *finger painting* :

- a. Mengembangkan ekspresi melalui media lukis dengan gerakan tangan
Mengembangkan fantasi, imajinasi, dan kreativitas anak. Melatih otot-otot tangan/jari, koordinasi otot, dan mata. Melatih kecakapan

mengkombinasikan warna primer. Memupuk perasaan terhadap gerakan tangan. Memupuk perasaan keindahan.

Menurut Anies Listyowati dan Sugiyanto kegiatan *finger painting* sangat bermanfaat bagi perkembangan anak yaitu melatih kemampuan motorik halus anak karena jari-jari anak akan bergerak dan bergesekan dengan cat dan media lukisnya, mengembangkan dan mengenalkan berbagai warna dan bentuk, meningkatkan daya imajinasi dan kreativitas anak, meningkatkan koordinasi mata dan tangan, melatih konsentrasi, serta dapat dijadikan sebagai media mengekspresikan emosi anak. Dari pendapat di atas dapat penulis pahami bahwa manfaat *finger painting* itu dapat melatih motorik halus anak serta mengembangkan imajinasi ataupun kreativitas anak.

Dari beberapa pendapat di atas dapat penulis simpulkan bahwa manfaat *finger painting* itu dapat melatih motorik halus anak, serta mengembangkan imajinasi, kreativitas, dan mengenalkan bentuk serta warna-warna terhadap anak.

Finger painting dapat mempergunakan berbagai media dan warna, dengan menggunakan tepung kanji, adonan kue, pasir dan sebagainya. Aktifitas ini penting dilakukan sebab akan memberikan sensasi pada jari sehingga dapat merasakan kontrol gerakan jarinya dan membentuk konsep gerak membuat huruf. Anak *Down Syndrome* mengalami permasalahan dalam motorik halusnya. Salah satunya karena adanya kekakuan pada tangan dengan *finger painting* ini diharapkan anak bisa mengikuti dengan baik.

Perkembangan motorik halus melibatkan otot-otot halus yang mengendalikan tangan dan kaki. Kemampuan anak dalam mengontrol, mengkoordinasikan dan

ketangkasan dalam menggunakan tangan dan jemari, adalah menjadi fokus dari perkembangan motorik halus anak. Walaupun perkembangan ini berlangsung bersamaan dengan perkembangan motorik kasar, namun otot-otot dekat batang tubuh matang sebelum otot-otot kaki dan tangan yang mengendalikan pergelangan dan tangan. Oleh karena itu, sangat penting bagi anak-anak untuk berlatih menggunakan otot-otot besar saat terlibat dalam aktivitas motorik halus.

Dalam perkembangan motorik halus. Ada tiga hal yang harus diperhatikan, yaitu: gerakan refleks, waktu dan ketangkasan dan dominasi penggunaan tangan:

1) Gerakan refleks

Sebagai besar gerakan refleks dilakukan oleh bayi. Bayi menggerakkan tangan dan jemari mereka lewat gerakan refleks, bukan gerakan sadar. Elliot menggambarkan beberapa gerakan refleks yang dilakukan oleh bayi, diantaranya: mengayunkan lengan ke sembarang arah, menangis, gerakan refleks menghisap, membuka mulutnya ketika pipinya disentuh, dan sebagainya. Namun demikian, ketika gerakannya refleks awal ini memudar, maka anak-anak harus benar-benar belajar menggunakan dan mengendalikan tangan dan jemari mereka. Gerakan refleks yang paling terkait dengan kemampuan tangan motorik halus adalah gerakan refleks menggenggam, di mana bayi merapatkan jemarinya melingkupi sesuatu di telapaknya. Gerakan refleks menggenggam berlangsung hingga sekitar usia 9 bulan. Sebelum usia ini, bayi tidak bias mengontrol tindakan tangan dan jemarinya secara sadar.

2) Waktu (*timing*)

Seperti halnya kemampuan motorik kasar, kemampuan motorik halus anak-anak pun harus mendapatkan dorongan dari orang tua dalam menggunakan otot-otot kecil mereka. Yang harus dipahami, dalam mengajarkan kemampuan motorik halus kepada anak, apakah harus menunggu mereka siap? Menurut Beaty, tidak juga. Hal ini karena perkembangan anak itu berbeda, begitupun periode waktu antara anak yang satu dan yang lainnya pastilah berbeda. Oleh karena itu, tidak ada salahnya orang tua memberikan dorongan dan arahan kepada anak-anak untuk mengembangkan kemampuan motorik halus mereka, asalkan jangan memaksa anak. Menurut Beaty, pada dasarnya setiap orang tua di manapun mempunyai “jam biologi” dalam memprediksi perkembangan anak, begitu halnya dengan perkembangan motorik anak. Namun demikian, setiap anak mempunyai “jam biologis” yang berbeda satu sama lain. Oleh karena perkembangan setiap anak terjadi dalam urutan tertentu (walau periode waktu setiap anak berbeda), maka sebagai orang tua harus menilai perkembangan anak melalui pengamatan dan memberinya kegiatan, arahan, dan dorongan yang sesuai.

3) Ketangkasan dan dominasi penggunaan tangan

Ketangkasan mengharuskan gerakan cepat dan tepat tangan dan jari. Anak-anak yang berusia 4 dan 5 tahun sudah bisa mengatur kancing, dan retsleting kecil, juga menuliskan beberapa huruf dan angka. Anak usia 3 tahun belum matang untuk melakukan hal tersebut. Hal itu tergantung dari proses neurologi, dengan kemampuan tertentu ditempatkan di belahan otak kiri dan kanan. Laura E. Berk mengungkapkan bahwa gerak motorik halus

adalah meningkatnya pengkoordinasian gerak tubuh, yang melibatkan kelompok otot dan saraf kecil lainnya. Sedangkan Janet W. Lerner, menjelaskan gerak motorik halus merupakan keterampilan menggunakan media dengan koordinasi antara mata dan tangan. Kemampuan motorik halus, mengembangkan kemampuan anak dalam menggunakan jari-jarinya, khususnya ibu jari dan jari telunjuk. Beberapa kegiatan yang bias merangsang kemampuan motorik halus anak adalah sebagai berikut:

4) Menyusun *puzzle*

Permainan ini biasanya dilakukan oleh anak usia 3 tahun ke atas. Ketika anak masih berusia 2 tahun, biasanya juga mencoba permainan ini, namun ia akan menyerah di tengah jalan. Sedangkan anak yang lebih tua akan terus berusaha mencocokkan dan memutar potongan *puzzle* menjadi bentuk yang utuh. Dalam permainan ini dibutuhkan kesadaran *perceptual*, (yaitu seperti apa bentuk potongan *puzzle*), tetapi pertama-tama anak membutuhkan kemampuan memutar motorik halus untuk menggunakan kemampuan pengenalan bentuk mereka. *Puzzle* yang terbuat dari kayu atau sejenisnya, dengan berbagai kerumitan, sebaiknya menjadi salah satu bentuk permainan di sekolah atau dirumah. Permainan *puzzle* melatih ketangkasan jari dan koordinasi mata dan tangan, serta konsep kognitif mencocokkan bentuk. Selain itu permainan *puzzle* juga sangat bermanfaat bagi orang tua maupun guru dalam menilai dan mengamati anak, hal tersebut seperti yang disampaikan Maldonado, bahwa *puzzle* bias menjadi alat pengamatan dan penilaian yang baik bagi guru, memungkinkan guru mudah mengamati anak-anak konsentrasi,

gerakan tubuh, bahasa mereka, berfikir dari segi menyelesaikan masalah, dan melakukan pilihan, dapat diketahui dalam permainan ini.

C. Pengertian Aplikasi

Aplikasi menurut Jogiyanto (2018:12) adalah penggunaan dalam suatu komputer, instruksi (*instruction*) atau pernyataan (*statement*) yang disusun sedemikian rupa sehingga komputer dapat memproses input menjadi output. Menurut Kamus Besar Bahasa Indonesia (2018 : 52) , “Aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu”. Aplikasi adalah suatu program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari pengguna. Aplikasi merupakan rangkaian kegiatan atau perintah untuk dieksekusi oleh komputer. Program merupakan kumpulan *instruction set* yang akan dijalankan oleh pemroses, yaitu berupa *software*. Bagaimana sebuah sistem komputer berpikir diatur oleh program ini. Program inilah yang mengendalikan semua aktifitas yang ada pada pemroses. Program berisi konstruksi logika yang dibuat oleh manusia, dan sudah diterjemahkan ke dalam bahasa mesin sesuai dengan format yang ada pada *instruction set* . Program aplikasi merupakan program siap pakai. Program yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain. Contoh-contoh aplikasi ialah program pemroses kata dan *Web Browser*. Aplikasi akan menggunakan sistem operasi (OS) komputer dan aplikasi yang lainnya yang mendukung. Istilah ini mulai perlahan masuk ke dalam istilah Teknologi Informasi semenjak tahun 1993, yang biasanya juga disingkat dengan app. Secara *historis*, aplikasi adalah *software* yang dikembangkan oleh sebuah perusahaan. *App* adalah

software yang dibeli perusahaan dari tempat pembuatnya. Industri PC tampaknya menciptakan istilah ini untuk merefleksikan medan pertempuran persaingan yang baru, yang paralel dengan yang terjadi antar sistem operasi yang dimunculkan.

D. Java

Menurut Shalahuddin (2017:1), *java* adalah bahasa pemrograman yang berorientasi objek (OOP) dan dapat dijalankan pada berbagai *platform* sistem operasi. Perkembangan *Java* tidak hanya terfokus pada satu sistem operasi, tetapi dikembangkan untuk berbagai sistem operasi dan bersifat *open source*.

1. Sebagian Fitur dari Java

a. Java Virtual Machine (JVM)

Menurut Shalahuddin (2017:10), *JVM* adalah sebuah mesin imajiner (*maya*) yang bekerja dengan menyerupai aplikasi pada sebuah mesin nyata. *JVM* menyediakan spesifikasi *hardware* dan *platform* dimana kompilasi kode *Java* terjadi. Spesifikasi inilah yang membuat aplikasi berbasis *Java* menjadi bebas dari *platform* manapun karena proses kompilasi diselesaikan oleh *JVM*. Aplikasi program *Java* diciptakan dengan *file* teks berekstensi *Java*. Program ini dikompilasi menghasilkan satu berkas *bytecode* berekstensi *.class* atau lebih. *Bytecode* adalah serangkaian instruksi serupa instruksi kode mesin. Perbedaannya adalah kode mesin harus dijalankan pada sistem komputer dimana kompilasi ditujukan, sementara *bytecode* berjalan pada *Java interpreter* yang tersedia di semua *platform* sistem komputer dan sistem operasi.

b. Garbage Collection

Banyak bahasa pemrograman lain yang mengizinkan seorang programmer mengalokasikan memori pada saat dijalankan. Program *Java* melakukan *garbage collection* yang berarti program tidak perlu menghapus sendiri objek–objek yang tidak digunakan lagi.

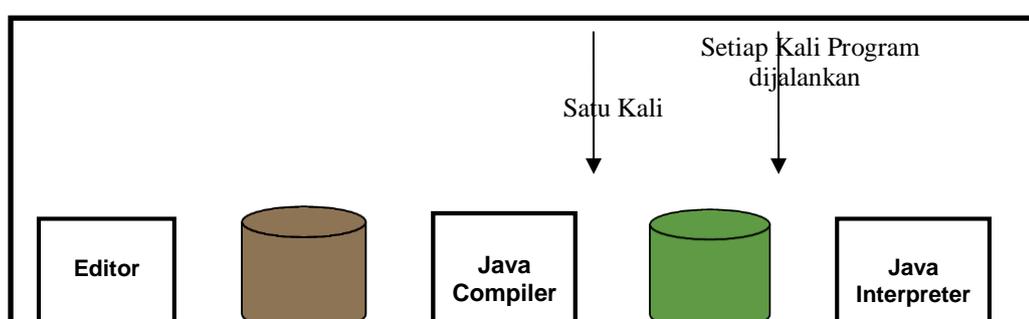
c. Code Security

Code Security terimplementasi pada *Java* melalui penggunaan *Java Runtime Environment (JRE)*. *Java* menggunakan model pengamanan 3 lapis untuk melindungi sistem dari *untrusted Java Code*.

1. Pertama, *class-loader* menangani pemuatan kelas *Java* ke runtime *interpreter*. Proses ini menyediakan pengamanan dengan memisahkan kelas–kelas yang berasal dari *localdisk* dengan kelas–kelas yang diambil dari jaringan. Hal ini membatasi aplikasi Trojan karena kelas–kelas yang berasal dari *localdisk* yang dimuat terlebih dahulu.
2. Kedua, *bytecode verifier* membaca *bytecode* sebelum dijalankan dan menjamin *bytecode* memenuhi aturan–aturan dasar bahasa *Java*.
3. Ketiga, manajemen keamanan menangani keamanan tingkat aplikasi dengan mengendalikan apakah program berhak mengakses sumber daya seperti sistem *file*, *port* jaringan, proses eksternal dan sistem *windowing*.

d. Fase-fase pemrogram *Java*

Gambar di bawah ini menjelaskan aliran proses kompilasi dan eksekusi sebuah program *Java*:





Keterangan :

1. *Editor* merupakan aplikasi untuk membuat atau mengedit aplikasi *java*.
2. *Hello.java* merupakan *file java* yang sudah mempunyai kode-kode *java* dan di simpan dengan extensi *java*.
3. *Java compiler* merupakan merubah *file java* yang extensi *java* menjadi *file* yang extensi *class*.
4. *Hello.class* merupakan hasil *file java compiler* yang asalnya berupa *file* extensi *java*.
5. *Java interpreter* merupakan menterjemahkan baris per baris kode ke dalam bahasa mesin dan menampilkan hasilnya.

E. Android

Android adalah sebuah sistem operasi untuk perangkat mobile berbasis linux yang mencakup sistem operasi, *middleware* dan aplikasi. Android menyediakan platform yang terbuka bagi para pengembang untuk menciptakan aplikasi baru. Android merupakan generasi baru *platform mobile*, *platform* yang memberikan pengembangan untuk melakukan pengembangan sesuai dengan yang diharapkan. Pada saat perilisan perdana android, 5 November 2007, android

bersama *open handset alliance* menyatakan mendukung pengembangan *open source* pada perangkat lunak *mobile*. Di lain pihak, google meriliskode-kode android dibawah lisensi *apache*, sebuah lisensi perangkat lunak dan *open platform* perangkat seluler.

1. Android SDK (*Software Development Kit*)

Android SDK adalah tools API (*Application Programming Interface*) yang diperlukan untuk memulai pengembangan aplikasi pada *platform* Android menggunakan bahasa pemrograman *Java*. Android merupakan subset perangkat lunak untuk ponsel yang meliputi, sistem operasi, *middleware* dan aplikasi yang di *release* oleh *google*. Saat ini disediakan android SDK (*Software Development Kit*) sebagai alat bantu dan API untuk mulai mengembangkan aplikasi platform android menggunakan bahasa pemrograman *java*. Sebagai platform aplikasi-netral, android memberi anda kesempatan untuk membuat aplikasi yang kita butuhkan yang buka merupakan aplikasi bawaan *handphone*.

F. Database SQLite

Menurut Jay A. Kreibich (2010,12) SQLite merupakan paket perangkat lunak yang bersifat *public domain* yang menyediakan sistem manajemen basis data relasional atau RDBMS. Sistem basis data relasional digunakan untuk menyimpan *record* yang didefinisikan oleh pengguna pada ukuran tabel yang besar dan memproses perintah *query* yang kompleks dan menggabungkan data dari berbagai

tabel untuk menghasilkan laporan dan rangkuman data. Kata '*Lite*' pada SQLite tidak menunjuk pada kemampuannya, melainkan menunjuk pada sifat dari SQLite, yaitu ringan ketika dihubungkan dengan kompleksitas pengaturan, administrative overhead, dan pemakaian sumber. SQLite memiliki fitur-fitur sebagai berikut :

1. Tidak memerlukan *server* Arsitektur SQLite tidak memiliki arsitektur client server. Kebanyakan sistem database skala besar memiliki paket server yang besar yang membentuk mesin *database*.
2. Single File *Database* SQLite mengemas seluruh *database* ke dalam suatu *single file*. Single file tersebut berisi *layout database* dan data aktual yang berada pada tabel dan indeks yang berbeda. Format file dapat digunakan pada banyak *platform* dan dapat diakses pada mesin manapun tanpa memperhatikan *native byte order* ataupun ukuran kata. Pengemasan database kedalam suatu *file* tunggal memudahkan pengguna untuk membuat, menyalin, ataupun mem-*backupimage database* yang berada di dalam media penyimpanan
3. *Zero Configuration* SQLite tidak membutuhkan apapun untuk melakukan instalasi dan konfigurasi. Dengan mengeliminasi server dan menggabungkan database secara langsung ke dalam aplikasi, maka pengguna tidak perlu mengetahui bahwa mereka sedang menggunakan *database*.
4. *Embedded Device Support* Ukuran code dari SQLite bersifat kecil dan penggunaan sumber daya yang konservatif membuatnya cocok digunakan untuk *embedded system* yang berjalan terbatas pada sistem operasi.

5. Fitur-fitur yang unik SQLite menggunakan sistem dengan tipe dinamis untuk tabel-tabel. SQLite memungkinkan pengguna untuk memasukkan nilai ke dalam kolom tanpa memperhatikan tipe data. Pada beberapa cara pemakaiannya, sistem yang bertipe dinamis pada SQLite mirip dengan sistem yang ditemukan pada bahasa *scripting* yang populer, yang sering memiliki sebuah tipe skalar yang dapat menerima semua tipe data dari *integer* sampai *string*. Fitur lainnya adalah kemampuan untuk memanipulasi lebih dari satu basis data pada satu waktu. SQLite mempunyai kemampuan dalam menghubungkan sebuah koneksi *database* tunggal dengan banyak file basis data secara bersamaan. Hal ini memungkinkan SQLite untuk memproses SQL statement yang menjembatani beberapa basis data sekaligus.
6. *Compatible License* SQLite dan SQLite *code* tidak memiliki lisensi pengguna dan tidak dilindungi oleh GNU's Not Unix (GNU) *General Public License* (GPL) atau lisensi *open source* sejenisnya. Hal ini berarti pengguna dapat melakukan apapun dengan *source code* SQLite, sehingga *library code* dapat digunakan dengan berbagai cara, dimodifikasi dengan berbagai cara dan didistribusikan dengan berbagai cara.
7. *Highly reliable* Sejumlah tes telah dilakukan sebelum *library* SQLite masing-masing dirilis. Hal ini dilakukan untuk mempertahankan tingkat kehandalan yang tinggi.

G. UML (*Unified Modeling Language*)

Menurut Adi nugroho dalam buku “Rekayasa Perangkat Lunak Berorientasi Objek” (2010:6) UML (*Unified Modeling Language*) adalah ‘bahasa’ pemodelan untuk *system* atau perangkat lunak yang berparadigma ‘berorientasi objek’. Pemodelang (*Modeling*) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami.

a. *Use Case Diagram*

Use case diagram bersifat statis, memperlihatkan himpunan *use-case* dan actor-aktor. Diagram ini sangat penting terutama untuk memodelkan ataupun mengorganisasikan perilaku dari system yang dibutuhkan pengguna.

b. *Activity Diagram*

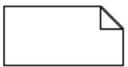
Activity Diagram bersifat dinamis, merupakan tipe khusus dari diagram state yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainya dalam suatu system.

c. *Class Diagram*

Class Diagram bersifat statis tetapi sering pula memuat kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi.

Tabel 2.1 Simbol Diagram *Use Case*

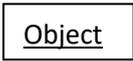
NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya sehingga elemen yang tidak mandiri (<i>independent</i>).

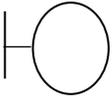
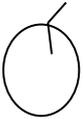
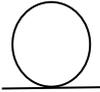
NO	GAMBAR	NAMA	KETERANGAN
3		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

Tabel 2.2 Simbol Diagram *Activity*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Activity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

Tabel 2.3 Simbol *Sequence Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Object (Partisipan)</i>	Object atau biasa juga disebut partisipan merupakan instance dari sebuah class dan dituliskan tersusun secara horizontal. Digambarkan sebagai sebuah class (kotak) dengan nama objek didalamnya yang diawali dengan sebuah titik koma.
2		<i>Actor</i>	Actor juga dapat berkomunikasi dengan object, maka actor juga dapat diurutkan sebagai kolom.
3		<i>Life line</i>	Life line mengidentifikasi sebuah keberadaan object dalam basis waktu. Notasi untuk lifeline adalah garis putus-putus vertical yang ditarik oleh sebuah object.
4		<i>Collaboration</i>	Activation dinotasikan sebagai kotak segi empat yang digambar pada sebuah lifeline action mengidentifikasi

			sebuah object yang akan melakukan sebuah aksi.
5		<i>Boundary</i>	Boundary terletak diantara system dengan dunia sekelilingnya. Semua form, laporan-laporan, antar muka ke perangkat keras seperti printer atau scanner dan antar muka ke system lainya adalah termasuk dalam kategor.
6		<i>Control</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		<i>Entity</i>	Entity digunakan menangani informasi yang mungkin akan disimpan secara permanen. Entity bias juga merupakan sebuah table.

BAB III

METODE PENELITIAN

A. Jenis Penelitian

Jenis penelitian yang digunakan adalah penelitian deskriptif dimana memberikan gambaran mengenai apa yang sesungguhnya terjadi. Dalam pembuatan Skripsi ini digunakan metode deskripsif yang menggambarkan fakta-fakta dan informasi secara sistematis, faktual dan akurat.

Penelitian ini dilakukan melalui internet yang dapat memberikan sumber data dan pengetahuan mengenai sistem yang diteliti, kemudian mencocokkan dengan kemungkinan yang terjadi dalam usaha penyelesaian masalah.

B. Lokasi dan Waktu Penelitian

Lokasi penelitian dilaksanakan di Sekolah TK dan PIUD Kota Parepare. Adapun waktu penelitian dilakukan selama 2 (dua) bulan tahun 2024.

C. Metode Pengumpulan Data

Untuk memperoleh data-data yang dibutuhkan dalam rangka melakukan penelitian, maka penulis mengumpulkan data melalui beberapa cara yaitu :

1. Analisis Data

Menganalisa data-data yang sebelumnya telah dikumpulkan.

2. Perancangan Program

Sebagai pedoman dalam penulisan program atau kode-kode agar berjalan sesuai rencana.

3. Uji Coba Program

Pengujian program dilakukan untuk memastikan bahwa program yang dibuat dapat berjalan dengan baik.

4. Evaluasi

Sistem yang telah selesai dibangun perlu adanya evaluasi untuk menemukan kelemahan yang terdapat pada program yang telah dibangun tadi, yang nantinya bisa digunakan sebagai acuan untuk memperbaiki program sehingga lebih sempurna.

D. Jenis Data

Data yang digunakan dalam penelitian ini berupa data-data yang telah dikumpulkan melalui Penelitian Pustaka (*Library Research*). Adapun jenis data primer dan data sekunder yang relevan dengan masalah yang akan dibahas.

1. Data primer

Data Primer adalah data yang berasal atau data yang diperoleh langsung dari sumber data dan pengetahuan.

2. Data Sekunder

Data sekunder adalah data yang diperoleh tidak secara langsung dari objek penelitian. Peneliti mendapatkan data yang sudah jadi dari internet, *website* dan jurnal.

E. Tahapan Penelitian

Tahapan penelitian yang dimaksud dalam penelitian ini ada beberapa tahapan yaitu persiapan penelitian, pengumpulan data, analisis perancangan, pengujian dan implementasi. Adapun Uraian dari tahapan tersebut adalah sebagai berikut :

1. Persiapan Penelitian

Pada tahapan ini peneliti melakukan persiapan penelitian. Persiapan penelitian yang dimaksud adalah menyiapkan buku-buku, artikel-artikel tentang topik penelitian serta *software* yang digunakan selama penelitian.

2. Pengumpulan Data

Pada tahap ini peneliti melakukan observasi dengan peninjauan, pencatatan dan pengamatan langsung di tempat penelitian.

3. Analisis

Pada tahap analisis, peneliti melakukan analisa terhadap sistem yang di terapkan sekarang berdasarkan kemudian merumuskan masalah yang menjadi pokok penelitian sehingga dapat dibuat alternatif pemecahan masalah.

4. Perancangan

Peneliti kemudian merancang aplikasi yang ingin dibuat berdasarkan alternatif pemecahan masalah.

5. Pengujian

Setelah melakukan perancangan, peneliti kemudian menguji hasil perancangan yang telah dibuat. Jika hasil perancangan terdapat kekurangan atau kelemahan maka kembali ke tahap analisis.

6. Implementasi

Setelah pada perancangan tidak terdapat kekurangan maka aplikasi siap untuk di gunakan oleh user.

F. Metode Pengujian Aplikasi

Beberapa *test-case* harus dilaksanakan dengan beberapa perbedaan strategi, query, atau jalur navigasi yang mewakili penggunaan sistem yang *typical*, kritis atau abnormal. Isu kunci pada pengembangan sistem adalah pemilihan sekelompok *test-case* yang cocok, sekecil dan secepat mungkin, untuk meyakinkan perilaku sistem secara detail. Pengujian harus mencakup *unit testing*, yang mengecek validasi dari prosedur dan fungsi-fungsi secara independen dari komponen sistem yang lain. Kemudian modul testing harus menyusul dilakukan untuk mengetahui apakah penggabungan beberapa unit dalam satu modul sudah berjalan dengan baik, termasuk eksekusi dari beberapa modul yang saling berelasi, apakah sudah berjalan sesuai karakteristik sistem yang diinginkan.

Jika struktur kendali antar modul sudah terbukti bagus, maka pengujian yang tak kalah pentingnya adalah pengujian unit. Pengujian unit digunakan untuk menguji setiap modul untuk menjamin setiap modul menjalankan fungsinya dengan baik. Ada 2 metode untuk melakukan unit testing, yaitu :

1. *White Box Testing*

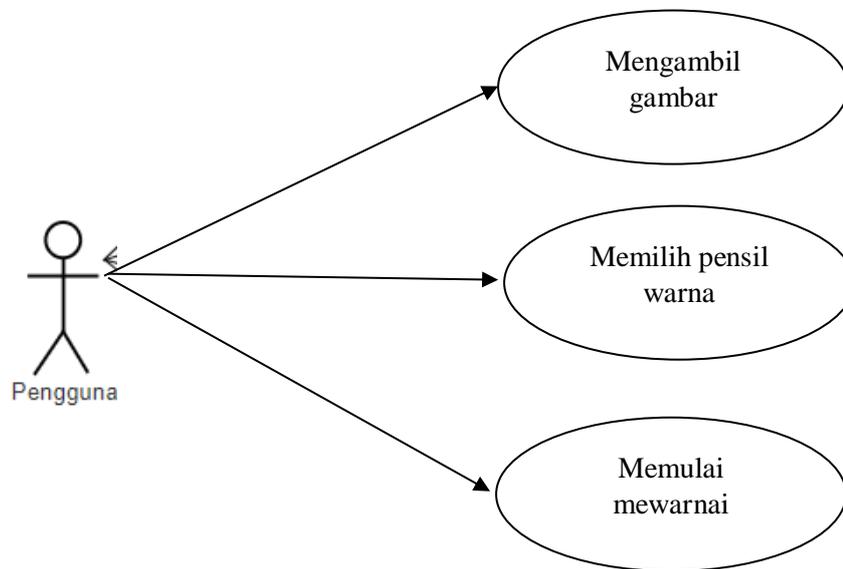
Uji coba *white box testing* merupakan metode perancangan *testcase* yang menggunakan struktural untuk mendapatkan *testcase*, test ini digunakan untuk meramal cara kerja perangkat lunak secara rinci kepada *logic path* (jalur logika), perangkat lunak di tes dengan kondisi dan perulangan secara fisik.

Contoh pengujian *white box testing* ini merupakan peringatan ketika *user* menginputkan password user yang salah, untuk kesalahan semacam ini akan memberikan suatu informasi kepada user mengenai kesalahan yang di lakukan.

2. *Black Box*

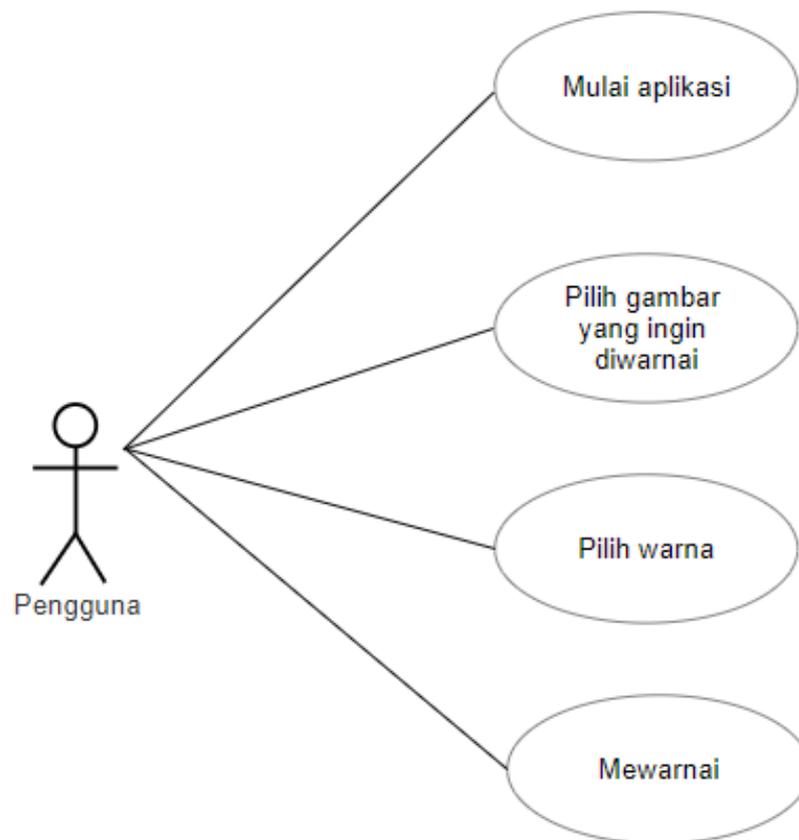
Berdasarkan hasil uji coba yang dilakukan, seluruh navigasi dan tombol fasilitas program lainnya serta proses yang di jalankan tidak terjadi kesalahan, tetapi aplikasi mempunyai aturan-aturan yang sudah di tetapkan dan harus di ikuti karena apabila di hiraukan maka sistem akan menolak perintah yang tidak sesuai seperti kesalahan ketika user belum menginput data yang harusnya di input sesuai ketentuan sistem yang di jalankan dan sistem memberikan informasi kepada user karena data yang ingin diproses belum lengkap atau tidak memenuhi ketentuan untuk proses selanjutnya.

G. Rancangan Sistem Yang Berjalan



pada system yang berjalan diatas menjelaskan yaitu pengguna mengambil buku gambar atau gambar yang ingin mewarnai kemudian memilih pensil warna sesuai yang diinginkan kemudian memulai mewarnai pada gambar yang sudah dipilih.

H. Rancangan Sistem Yang Diusulkan



Gambar 3.1. *Use Case Diagram*

Penjelasan gambar *usecase* diatas yaitu Dimana pengguna/user memulai aplikasi kemudian user memilih gambar yang ingin di warnai kemudian memilih warna yang sesuai dengan gambar aslinya setelah itu memulai mewarnai pada gambar tersebut.

BAB IV

ANALISIS DAN PERANCANGAN

A. Analisis Sistem

Analisis dan perancangan sistem ialah tahapan pengumpulan kebutuhan data dan informasi dan kemudian diolah sehingga dapat di definisikan kebutuhan yang harus dipenuhi oleh sistem yang akan dibangun. Jenis data yang dikumpulkan selama proses pengambilan data berupa data sekunder. Data sekunder merupakan berupa data yang di peroleh langsung oleh *user* yang mendaftar pada aplikasi kami dengan cara memperkenalkan aplikasi kami pada grup grup diskusi yang secara online, baik grup *whatshap*, *facebook* dan lain lain.

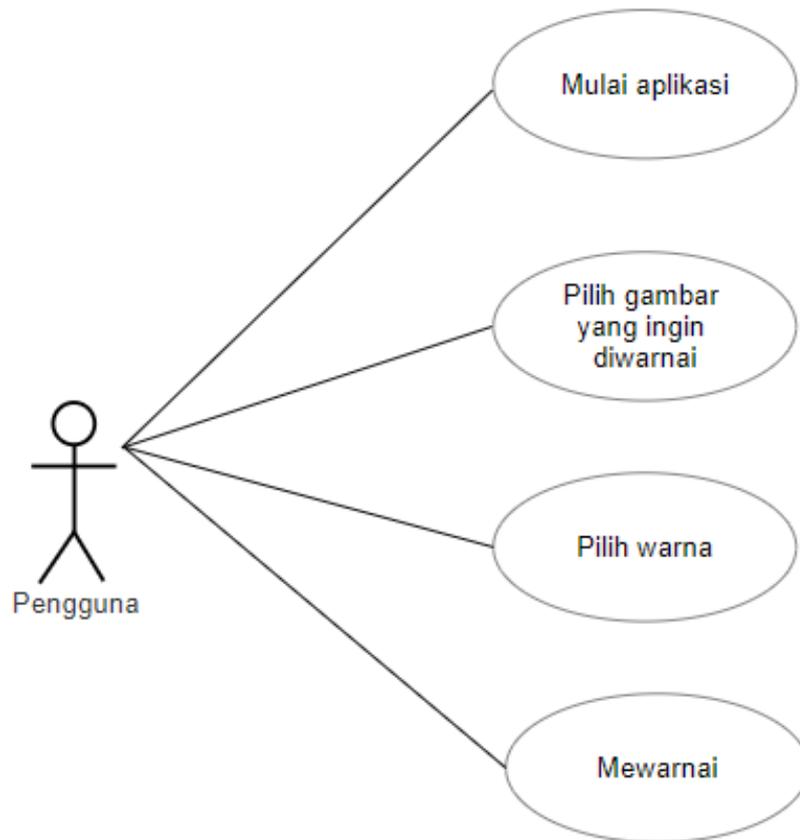
B. Use Case Diagram

Perancangan sistem adalah pendefinisian dari kebutuhan-kebutuhan fungsional dan persiapan untuk rancang bangun implementasi yang menggambarkan bagaimana suatu sistem dibentuk. Adapun kegiatan dalam tahapan ini antara lain :

1. Use Case diagram

Use Case Diagram adalah suatu bentuk diagram yang menggambarkan fungsionalitas yang diharapkan dari sebuah sistem dilihat dari perspektif pengguna di luar sistem. Sebuah *Use Case diagram* merepresentasikan interaksi yang terjadi antara *user* dengan proses atau sistem yang dibuat. *Use Case diagram* menggambarkan kegiatan yang dilakukan pada *user* pada sistem yang akan

dibangun sehingga *Use Case diagram* dari sistem yang dibangun dapat dilihat pada gambar berikut ini :



Gambar 4.1 *Use Case Diagram*

Pada gambar 4.1, dapat dijelaskan bahwa *User* dapat melakukan memilih gambar yang telah disiapkan, setelah itu *user* dapat melakukan proses mewarnai gambar data dapat melihat hasil mewarnai.

2. Tabel *Use Case diagram*

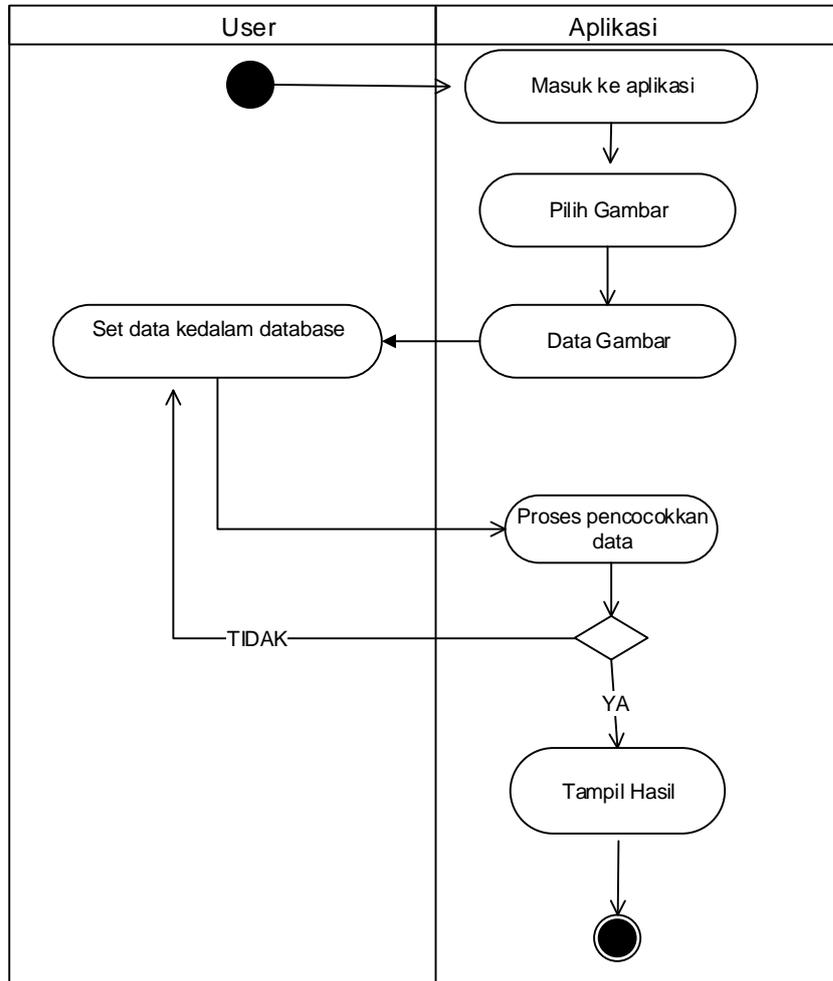
Penjelasan *Use Case Diagram* di atas :

a. Actor Pengguna

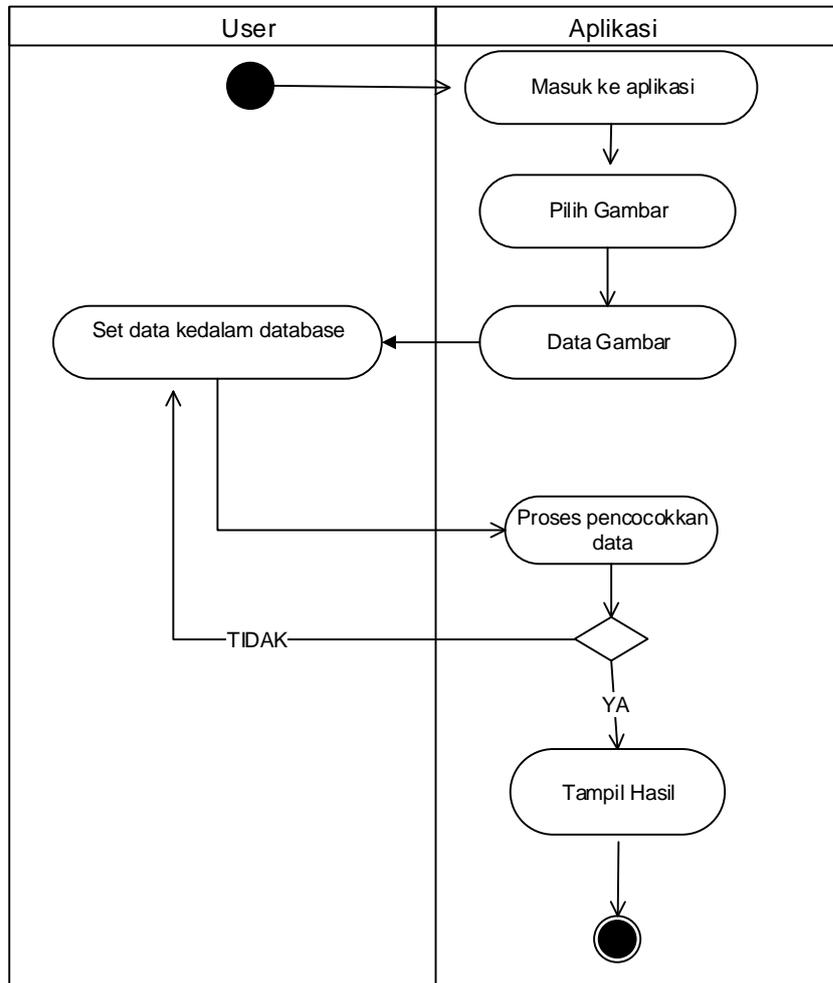
Tabel 4.1. Tabel *usecase*

Nama <i>Use Case</i>	Deskripsi <i>Use Case</i>
Mulai Aplikasi	<i>Use Case</i> ini menjelaskan bahwa setelah user mengaktifkan aplikasi, akan terbuka tampilan awal sistem.
Pilih gambar yang ingin di warnai	<i>Use Case</i> ini menjelaskan bahwa setelah user mengaktifkan aplikasi, akan terbuka form pilih gambar yang disiapkan aplikasi.
Pilih warna	<i>Use Case</i> ini menjelaskan bahwa setelah user pilih gambar maka user dapat memilih warna yang diinginkan didalam aplikasi.
Mewarnai	<i>Use Case</i> ini menjelaskan bahwa setelah user memilih warna maka user dapat warnai gambar tersebut didalam aplikasi.

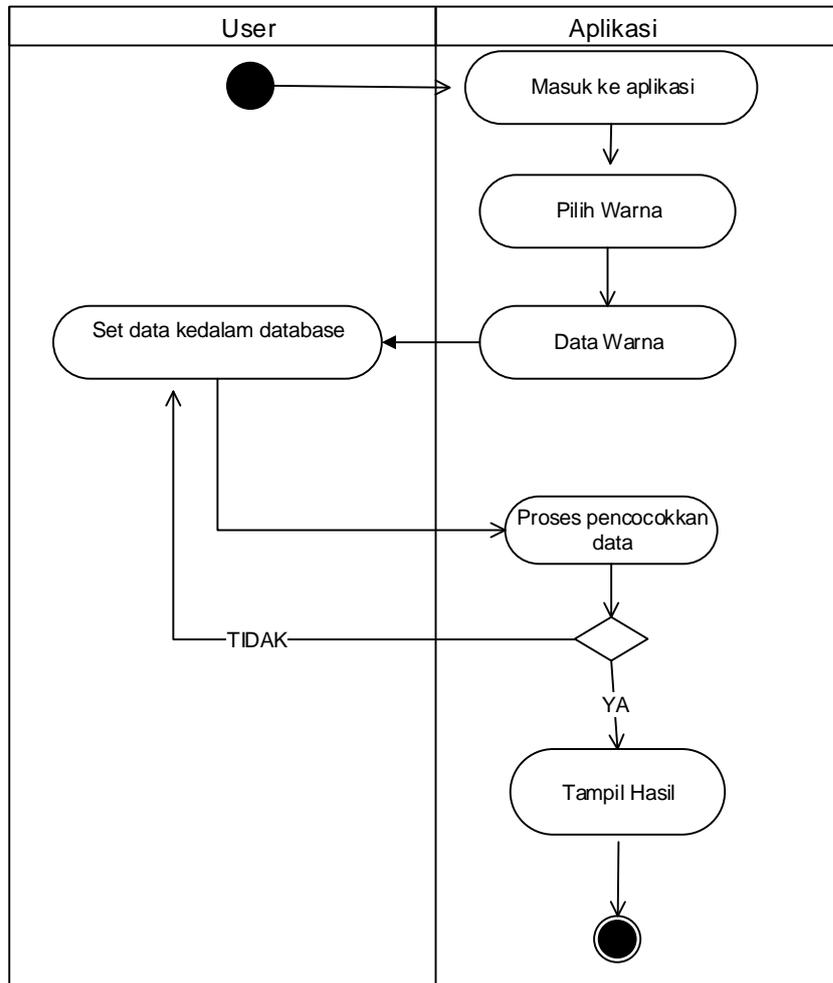
1. Activity Diagram
a. Activity Diagram User



Gambar 4.2 . Activity Diagram user

b. *Activity Diagram* Pilih GambarGambar 4.3 . *Activity Diagram* pilih gambar

c. Activity Diagram pilih warna

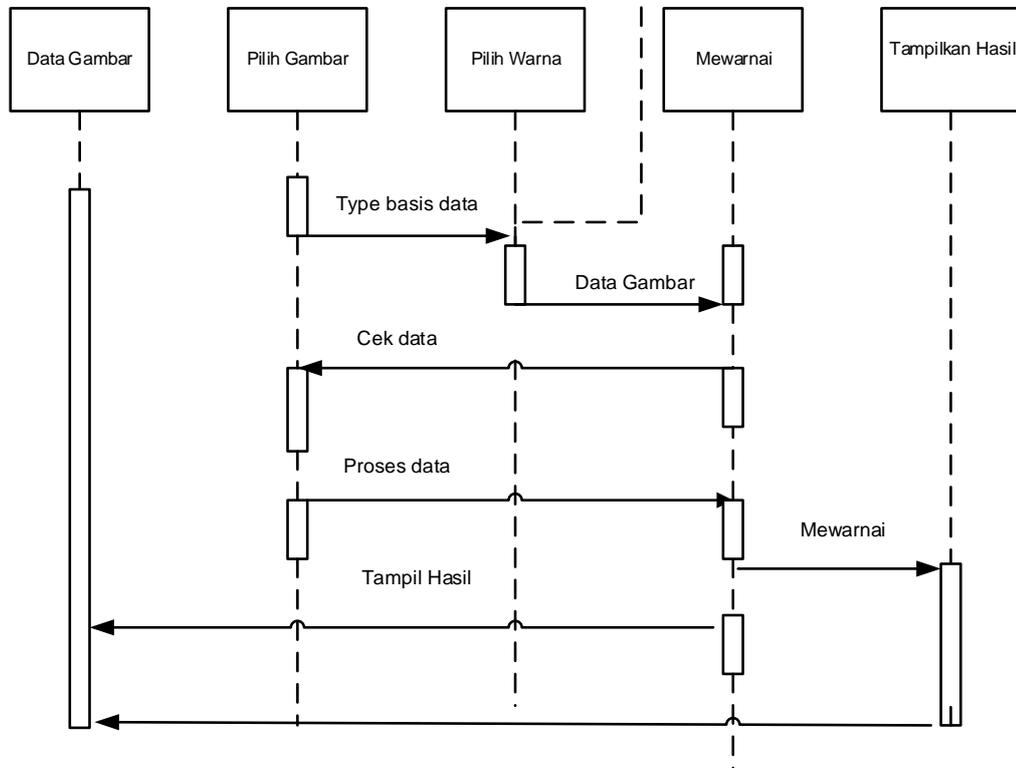


Gambar 4.4 . Activity Diagram pilih warna

Activity diagram di atas menjelaskan bahwa ketika *user* mulai masuk ke aplikasi kemudian memilih gambar dan memilih warna yang di inginkan dan melakukan proses mewarnai dan dapat melihat hasil hasil mewarnai.

2. Sequence Diagram

a. Sequence Diagram Aplikasi

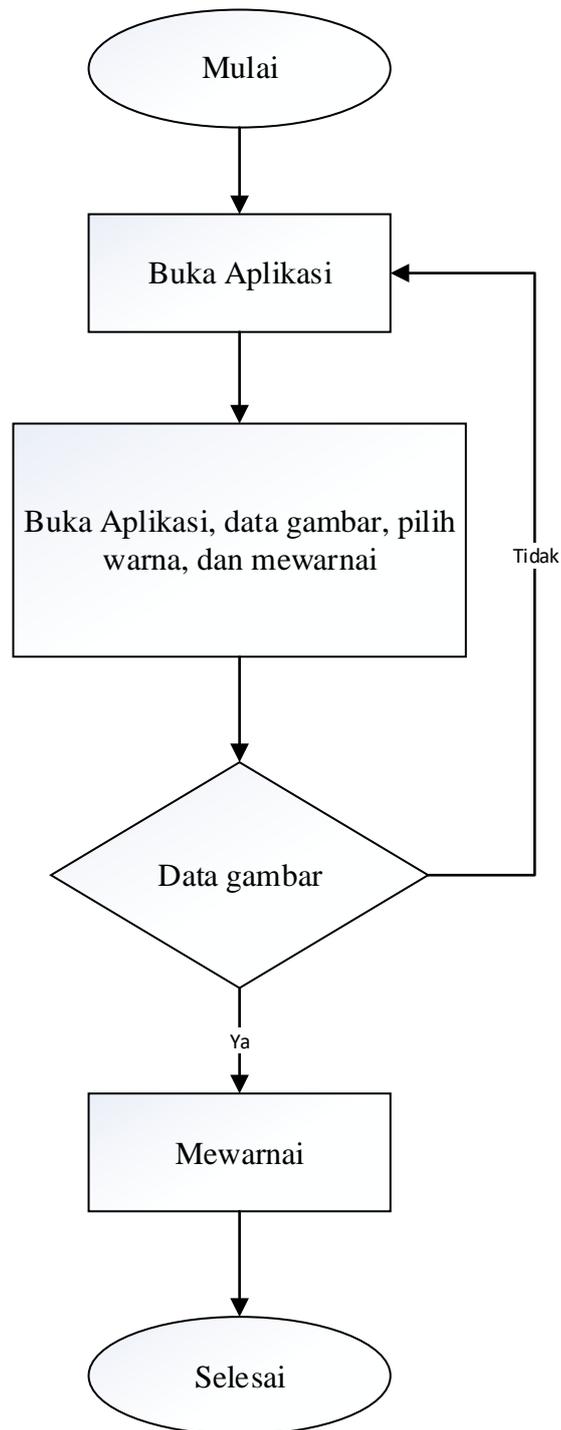


Gambar 4.5. Sequence Diagram

Sequence diagram di atas menjelaskan bahwa ketika user mulai masuk ke aplikasi kemudian melakukan pilih gambar dan melakukan mewarnai pada gambar tersebut pada system tersebut kemudian user dapat melihat data gambar yang sudah di warnai tersebut.

3. Rancangan Flowchart

Flowchart merupakan bagan yang menunjukkan alur kerja atau apa yang sedang dikerjakan di dalam sistem secara keseluruhan dan menjelaskan urutan dari prosedur-prosedur yang ada di dalam alur penelitian. Pada *flowchart* dibawah ini akan menjelaskan mengenai cara aplikasi.



Gambar 4.6 *Flowchart*

C. Implementasi

Program ini adalah aplikasi yang mempunyai fungsi untuk mempermudah seseorang untuk memperoleh data diri dari hasil aplikasi.

Implementasi aplikasi ini dilakukan setelah proses memilih gambar diaplikasi selesai dilakukan. Untuk menjalankan aplikasi ini maka dibutuhkan sebuah perangkat *smartphone* atau *komputer*.

- **Tampilan menu utama**

Ini adalah tampilan menu utama pada aplikasi sebagai berikut;

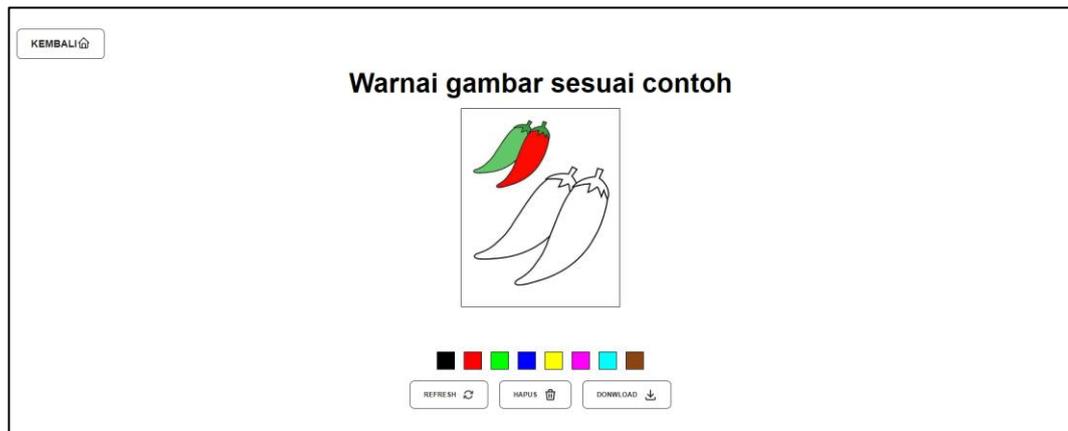


Gambar 4.7 Tampilan Menu Utama

- Tombol pilih gambar, untuk memilih sebuah gambar diatas.
- Tombol, masuk untuk melakukan proses mewarnai gambar

- **Tampilan pilihan gambar**

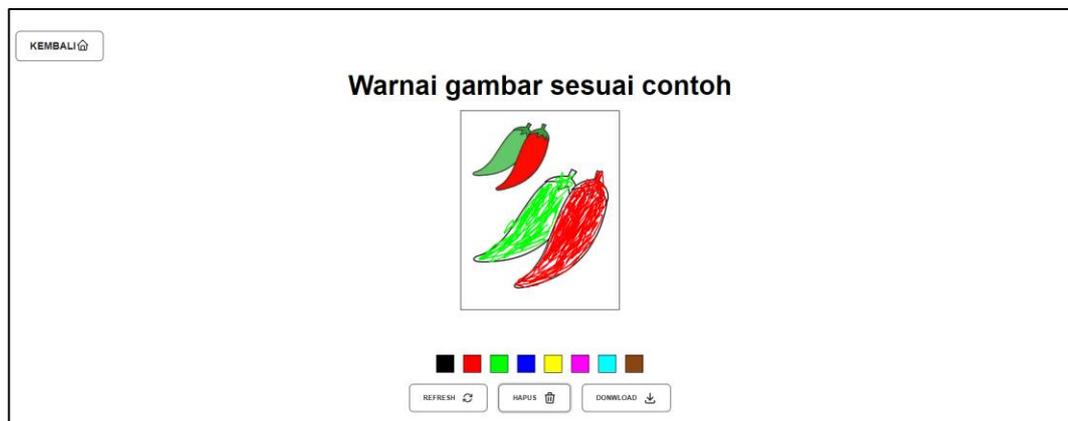
Setelah menekan Tombol pilih gambar dibawah maka *user* akan diarahkan ke form pilihan gambar.



Gambar 4.8 Tampilan pilih gambar

- **Tampilan halaman hasil mewarnai**

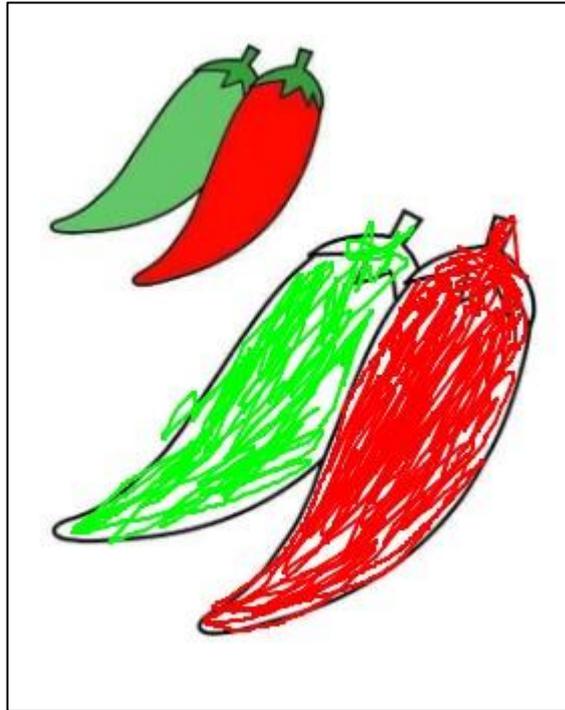
Setelah *user* melakukan melakukan mewarnai gambar maka *user* dapat melihat data tersebut didalam aplikasi.



Gambar 4.9 Tampilan hasil mewarnai

- **Tampilan download hasil mewarnai**

Tampilan pada Gambar dibawah hasil download gambar yang sudah diwarnai oleh *user*.



Gambar 4.10 Tampilan download gambar

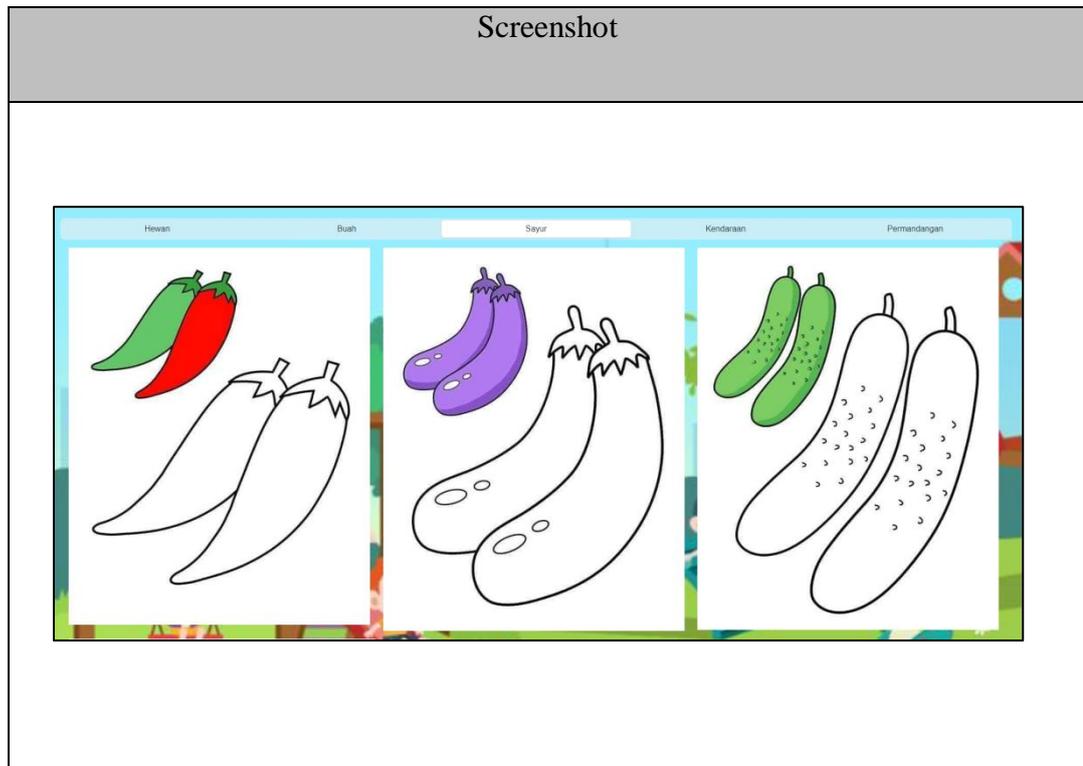
D. PENGUJIAN

Pengujian Sistem dengan Pengujian *Black box*. Pengujian ini bersifat menguji waktu perintah sesuai dengan fungsinya yang tanpa ada kesalahan atau error.

1. *Black box* melakukan pilih gambar .

Tabel 4.2. *Black Box* Melakukan proses pilih gambar

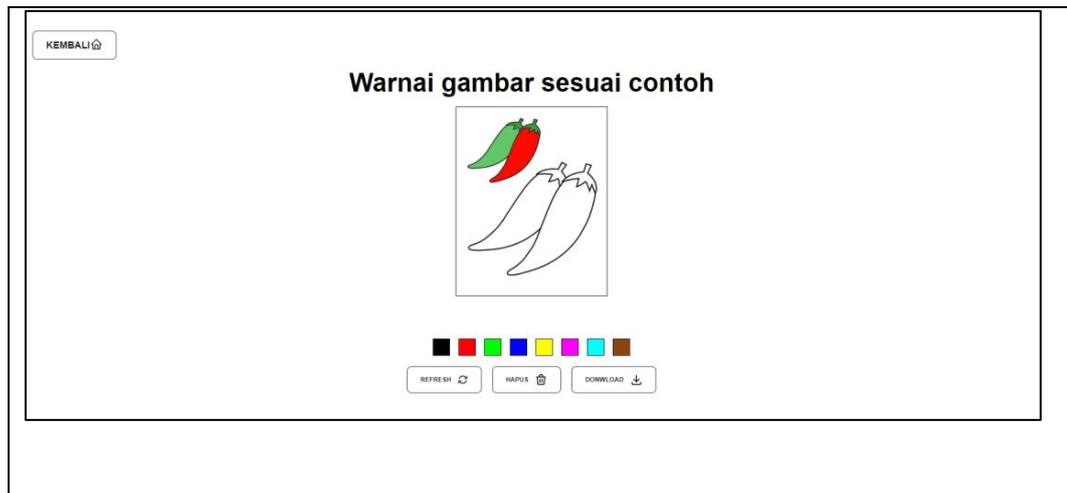
<i>Test Factor</i>	Hasil	Kesimpulan
Jika tombol data gambar di tekan akan menampilkan proses pemilihan data gambar	✓	Berhasil karena proses data gambar dapat di tampilkan data detail pada sistem.



2. *Black box* pilih gambar .

Tabel 4.3. *Black Box* Melakukan proses pilih gambar

Test Factor	Hasil	Kesimpulan
Jika tombol data gambar di tekan akan menampilkan proses gambar yang dipilih	✓	Berhasil karena proses data gambar dapat di tampilkan data detail pada sistem.
Screenshot		



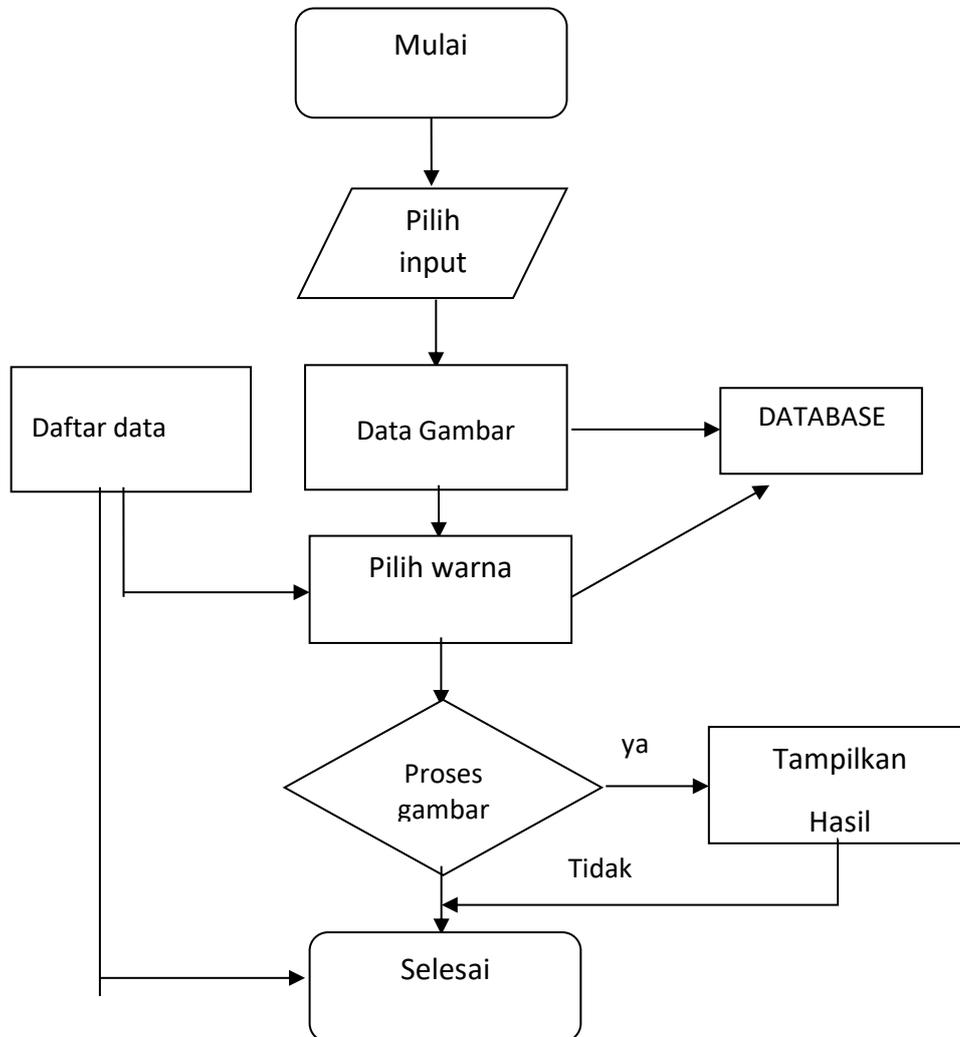
3. *Black box* mewarnai .

Tabel 4.3. *Black Box* Melakukan proses mewarnai

Test Factor	Hasil	Kesimpulan
Jika tombol data gambar di tekan akan menampilkan proses gambar yang diwarnai	✓	Berhasil karena proses data gambar dapat di warnai di tampilan data detail pada sistem.
Screenshot		

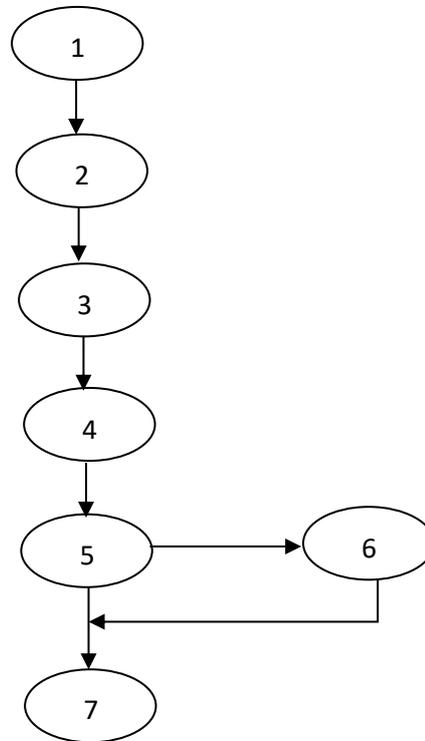
1. Pengujian *White box*

a. *Flowchart* Aplikasi



Gambar 4.26. *Flowchart* Aplikasi

b. *Flowgraph* Aplikasi



Gambar 4.27 . *Flowgraph* Aplikasi

1) Proses Perhitungan

Dari Gambar *FlowGraph* di atas dapat di lakukan proses perhitungan sebagai berikut :

a. Menghitung *Cyclomatic Complexity* $V(G) = E - N + 2$

$$N \text{ (node) } = 7$$

$$E \text{ (edge) } = 7$$

$$P \text{ (predikat node) } = 1$$

$$\begin{aligned}
 \text{Penyelesaian : } V(G) &= E - N + 2 \\
 &= 7 - 7 + 2 \\
 &= 2
 \end{aligned}$$

$$\begin{aligned}
 \text{Predikat Node (N)} &= P + 1 \\
 &= 1 + 1 \\
 &= 2
 \end{aligned}$$

b. Berdasarkan perhitungan *Cyclomatic Complexity* dari Flowgraph diatas memiliki *Region* = 2

c. *Independent Independent Path* pada *flowgraph* diatas adalah :

$$\text{Path 1} = 1 - 2 - 3 - 4 - 5 - 6 - 7$$

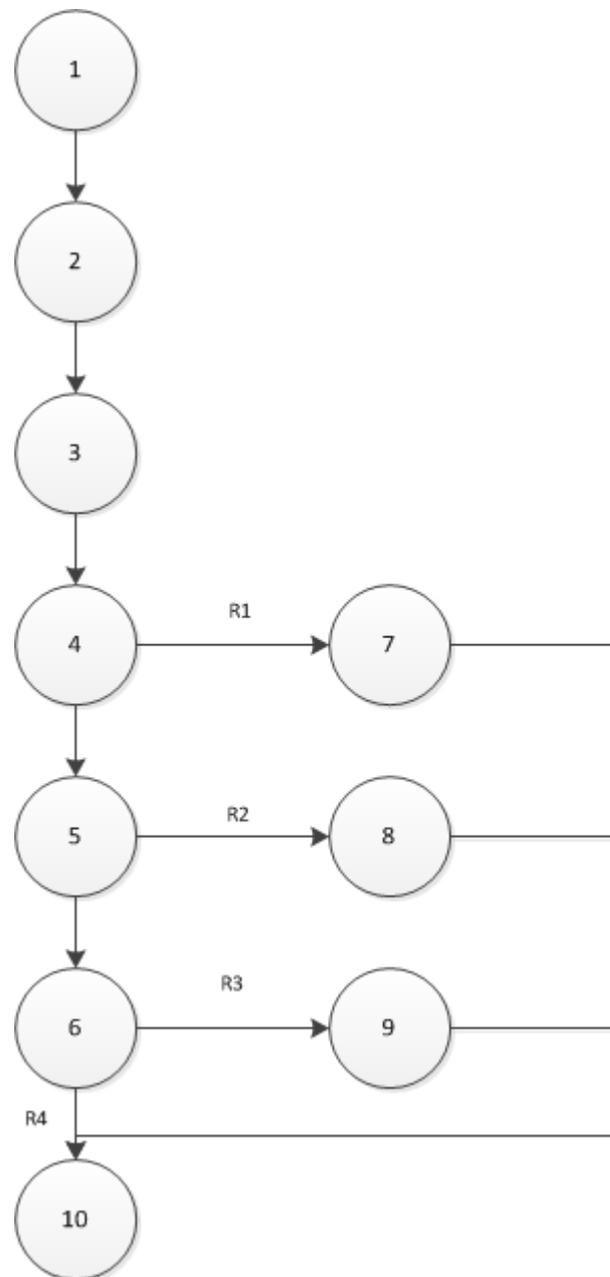
$$\text{Path 2} = 1 - 2 - 3 - 4 - 5 - 7$$

d. Grafik *Matriks*

Tabel 4.7. Grafik *Matriks*

	1	2	3	4	5	6	7	E - 1
1		1						1-1 = 0
2			1					1-1 = 0
3				1				1-1 = 0
4					1			1-1 = 0
5						1	1	2-1 = 1
6							1	1-1 = 1
7								
Zum (E+1)								1+1 = 2

1) *Flowgraph* menjalankan aplikasi



Gambar 4.28 *Flowgraph* menjalankan aplikasi

2) Proses perhitungan menjalankan aplikasi

Dari gambar *flowgraph* diatas dapat dilakukan proses perhitungan sebagai berikut :

a) Menghitung *Cyclomatic Complexity* $V(G)$ dari *Edge* dan *Node*:

Dengan Rumus : $V(G) = E - N + 2$

$$N(\text{node}) = 10$$

$$E(\text{edge}) = 12$$

$$P(\text{predikat node}) = 3$$

$$\begin{aligned} \text{Penyelesaian : } V(G) &= E - N + 2 \\ &= 12 - 10 + 2 \\ &= 4 \end{aligned}$$

$$\begin{aligned} \text{Predikat Node (N)} &= P + 1 \\ &= 3 + 1 \\ &= 4 \end{aligned}$$

b) Berdasarkan perhitungan *Cyclomatic Complexity* dari *Flowgraph* diatas memiliki *Region* = 4c) *Independent Path* pada *flowgraph* diatas adalah :

$$\text{Path 1} = 1 - 2 - 3 - 4 - 7 - 10$$

$$\text{Path 2} = 1 - 2 - 3 - 4 - 5 - 8 - 10$$

$$\text{Path 3} = 1 - 2 - 3 - 4 - 5 - 6 - 9 - 10$$

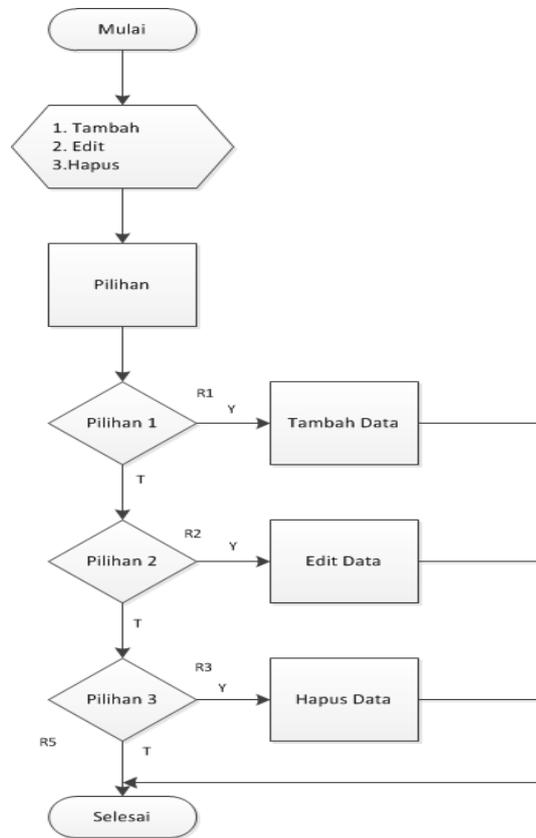
$$\text{Path 4} = 1 - 2 - 3 - 4 - 5 - 6 - 10$$

d) Grafik Matriks Menjalankan Aplikasi

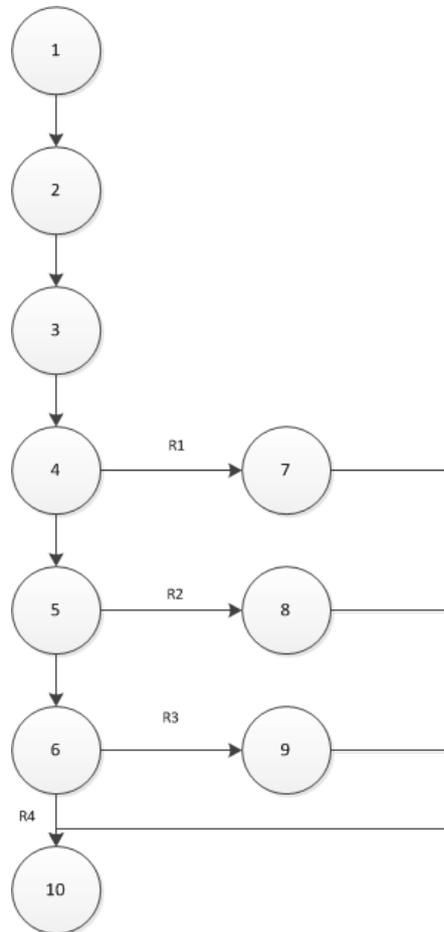
Tabel 4.8 Grafik matriks menjalankan aplikasi

	1	2	3	4	5	6	7	8	9	10	E-1
1		1									1-1=0
2			1								1-1=0
3				1							1-1=0
4					1		1				2-1=1
5						1		1			2-1=1
6									1	1	2-1=1
7										1	1-1=0
8										1	1-1=0
9										1	1-1=0
10											0
SUM(E+1)											3+1=4

1. Flowchart input data



Gambar 4.29 Flowchart input data

1) *Flowgraph* input dataGambar 4.30 *Flowgraph* input data

2) Proses perhitungan android data

Dari gambar *flowgraph* diatas dapat dilakukan proses perhitungan sebagai berikut :

3) Menghitung *Cyclomatic Complexity* $V(G)$ dari *Egde* dan *Node*:

Dengan Rumus : $V(G) = E - N + 2$

$N(\text{node}) = 10$

$E(\text{edge}) = 12$

$$P \text{ (predikat node)} = 3$$

$$\begin{aligned} \text{Penyelesaian : } V(G) &= E - N + 2 \\ &= 12 - 10 + 2 = 4 \end{aligned}$$

$$\begin{aligned} \text{Predikat Node (N)} &= P + 1 \\ &= 3 + 1 \\ &= 4 \end{aligned}$$

4) Berdasarkan perhitungan *Cyclomatic Complexity* dari *Flowgraph* diatas memiliki *Region* = 4

5) *Independent Path* pada *flowgraph* diatas adalah :

$$\text{Path 1} = 1 - 2 - 3 - 4 - 7 - 10$$

$$\text{Path 2} = 1 - 2 - 3 - 4 - 5 - 8 - 10$$

$$\text{Path 3} = 1 - 2 - 3 - 4 - 5 - 6 - 9 - 10$$

$$\text{Path 4} = 1 - 2 - 3 - 4 - 5 - 6 - 10$$

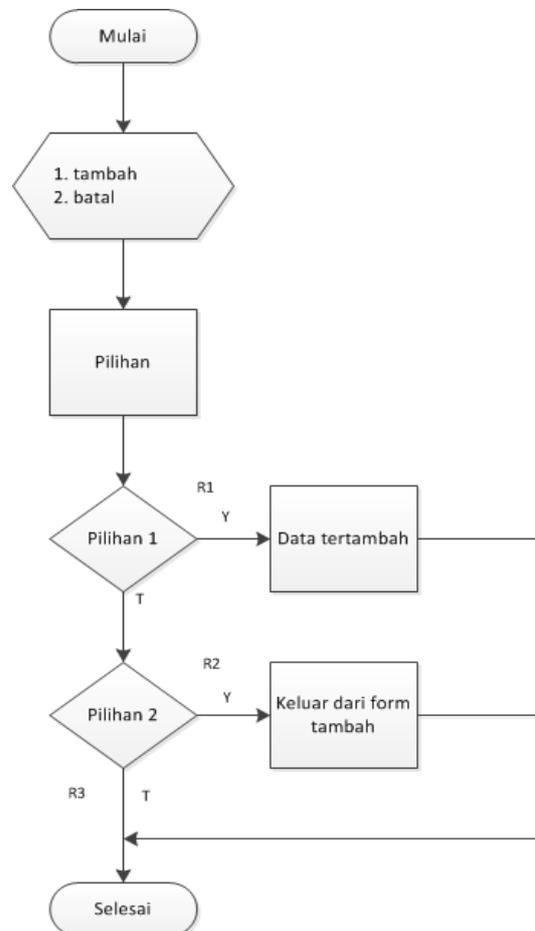
6) Grafik Matriks

Tabel 4.9 Grafik Matriks

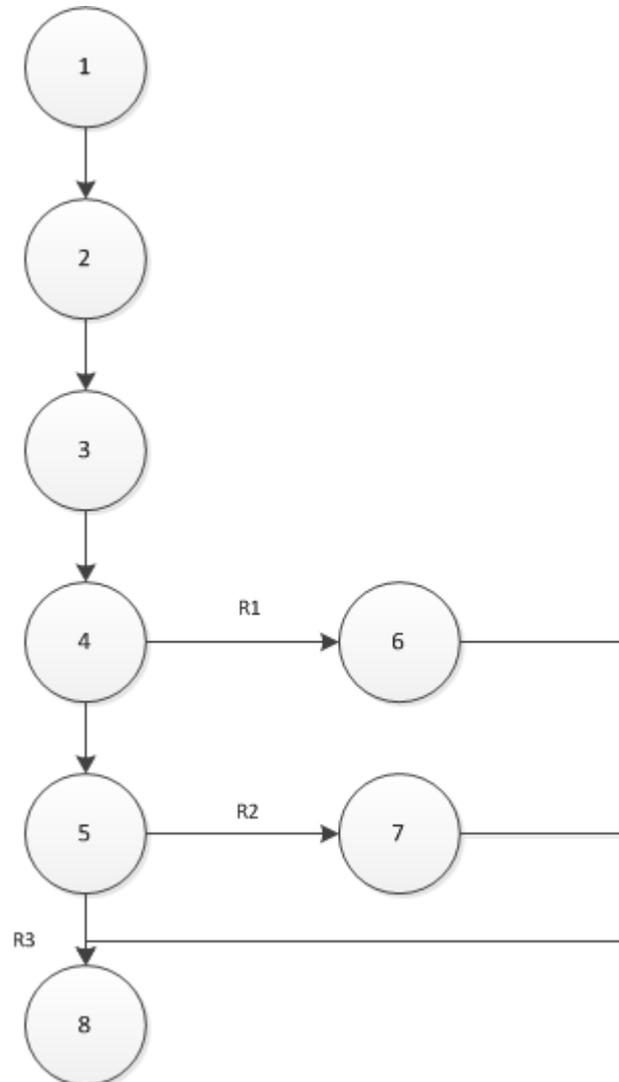
	1	2	3	4	5	6	7	8	9	10	E-1
1		1									1-1=0
2			1								1-1=0
3				1							1-1=0
4					1		1				2-1=1
5						1		1			2-1=1
6									1	1	2-1=1
7										1	1-1=0

8										1	1-1=0
9										1	1-1=0
10											0
SUM(E+1)											3+1=4

a. *White box* berdasarkan *form* tambah



Gambar 4.31 *Form* tambah data

1) *Flowgraph form* tambah dataGambar 4.32 *Flowgraph form* tambah2) Proses perhitungan *form* tambah

Dari gambar *flowgraph* diatas dapat dilakukan proses perhitungan sebagai berikut :

- a) Menghitung *Cyclomatic Complexity* $V(G)$ dari *Egde* dan *Node*:

Dengan Rumus : $V(G) = E - N + 2$

4					1	1					2-1=1
5							1	1			2-1=1
6								1			1-1=0
7								1			1-1=0
8											0
SUM(E+1)											2+1=3

E. Pengujian Aplikasi Pada Anak

Berikut ini adalah hasil pengujian Pembelajaran Aplikasi Finger Painting Untuk Meningkatkan Keterampilan Anak Usia Dini Di Kota Parepare. Pengujian ini bertujuan untuk mengukur efektivitas aplikasi dalam membantu anak-anak untuk meningkatkan motorik anak dengan cara yang lebih interaktif dan menyenangkan. Penilaian dilakukan dengan menggunakan *pre-test* dan *post-test*, untuk mengevaluasi kepuasan subjek terhadap aplikasi yang digunakan. Subjek yang terlibat dalam pengujian ini adalah anak-anak di Parepare. Berikut adalah tabel hasil pengujian dengan rincian subjek, skor *pre-test*, skor *post-test*, peningkatan skor, dan kepuasan subjek terhadap aplikasi yang digunakan.

Tabel 4.11. Pengujian Aplikasi Terhadap anak

No	Nama	Hasil	Skor Pre-Test	Skor Post-Test	Usia
1	Abrizan	Puas	50	80	3
2	Alina	Puas	60	90	5
3	Yoan	Puas	55	70	3
4	Azurah	Puas	40	70	4
5	Naila kamilah	Puas	65	90	3

6	Rafardan	Puas	70	80	3
7	Nurul Hikmah	Puas	65	90	4
8	Humairah Ahzahrah	Puas	55	85	5
9	Andi Ariza	Puas	70	80	5
10	Qosim	Puas	65	90	4

Keterangan:

- No.: Nomor urut subjek
- Nama Subjek: Nama lengkap subjek yang diuji
- Hasil: Hasil evaluasi subjek terhadap aplikasi (puas atau tidak puas)
- Skor Pre-Test: Pengetahuan nilai-nilai pre-test subjek sebelum menggunakan prediksi
- Skor Post-Test: Pengetahuan nilai-nilai post-test subjek setelah menggunakan aplikasi dengan system prediksi
- Peningkatan: Peningkatan skor nilai dari pre-test ke post-test

Dari tabel di atas, kita dapat melihat bahwa dari 10 anak terdapat 10 anak merasa puas dengan aplikasi. Selain itu, kita juga dapat melihat peningkatan skor nilai pre-test dan post-test serta perbedaan peningkatan skor antara subjek yang merasa puas dan subjek yang merasa tidak puas.

BAB V

SIMPULAN DAN SARAN

A. Kesimpulan

Hasil penelitian ini menunjukkan:

Dengan adanya aplikasi ini kegiatan aktivitas guru, terjadi peningkatan yang lebih baik pada hasil kinerja guru didalam kelas terutama dalam meningkatkan kreativitas motorik halus anak melalui kegiatan finger painting.

Kemampuan kreativitas motoric halus anak usia 5-6 tahun dalam kegiatan finger painting dengan menggunakan jari-jari tangan dinilai berhasil. Hal ini terjadi dengan adanya pemberian variasi jenis gambar dan warna sesuai dengan keinginan anak serta intensitas penggunaan jari jemari anak pada saat bermain finger painting.

B. Saran

Saran yang diberikan peneliti pada penelitian ini adalah :

1. Control pada bagian pilih gambar harus diutamakan agar pelayanan terhadap pada pengguna dapat berjalan dengan baik.
2. Aplikasi monotorik ini ini diharapkan dapat dikembangkan dengan menambahkan menu-menu sesuai kebutuhan.

DAFTAR PUSTAKA

Adi nugroho 2017 dalam buku “Rekayasa Perangkat Lunak Berorientasi Objek”
Penerbit ANDI

1. Dwi Sunar Prasetyo (2013), Membaca Wajah Orang, Jakarta, Penerbit : Diva Press
- 2.
3. Daryanto (2010: 168) e-learning merupakan sistem pembelajaran yang dapat membantu kegiatan pembelajaran dengan memanfaatkan media elektronik

HMM. *Internasional Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)* Volume 5, Issue 3, March 2016

Hossein Amerkashi, 2015, *Absolute App Inventor 2: Android Programming for all ages* Kindle Edition, Kindle Store

Irawan. 2018. *Membuat Alikasi Android untuk Orang Awam*. Palembang: Maxkom.

Jay A. Kreibich 2018, *Pengenalan SQLite*, Bandung, Penerbit ANDI

Jugianto, 2018. *Aplikasi Pada Sebuah Komputer*. Penerbit. Andi

Kamriani, F. & Roy, K., 2015, *App Inventor 2 Essentials A Step-By-Step Introductory Guide to mobile app development with app inventor 2*, Birmingham: Packt

Nana Syaodih Sukmadinata (2007: 206-207), e-learning tidak hanya singkatan dari electronic

Rusman (2012: 293) e-learning merupakan segala aktivitas belajar yang menggunakan bantuan teknologi elektronik

Rusman, Deni, & Cipi (2012: 265), WBL merupakan salah satu bentuk e-learning

Simamora (Made Wena, 2009: 215), WBL merupakan suatu sistem atau proses untuk melaksanakan kegiatan belajar mengajar jarak jauh melalui aplikasi web dan jaringan internet

Shalahuddin 2017, *Pengenalan Pemrograman java* Penerbit Andi

Shivam Sharma, 2015. *Speech Recognition with Hidden Markov Model: A Review*. *international Journal of Scientific & Engineering Research*, Volume 6, Issue 11, November-2015.

