APLIKASI PERHITUNGAN KAPASITOR *SURFACE MOUNT DEVICE*

Agus Irawan[1], Desmulyati[2]
Program Studi Teknik Informatika, Sekolah Tinggi Manajemen Informatika dan Komputer Nusa Mandiri Jakarta
agusirawan965@gmail.com[1], desmulyati.dmy@nusamandiri.ac.id[2]

Abstrak - Menghitung kapasitor dengan memanfaatkan teknologi akan menjadi lebih menyenangkan dan mudah dibandingkan menghitung kapasitor menggunakan avometer analog. Menghitung kapasitor akan mudah menyenangkan dan lebih menarik jika menghitung menggunakan aplikasi di dalam *gadget* yang sedang berkembang pesat saat ini terutama android. Aplikasi perhitungan kapasitor *surface mount device* berbasis android dengan menggunakan bahasa java dan *software* eclipse sebagai kompilernya. Terdapat tiga jenis kalkulator kapasitor, diantaranya kalkulator v*entax standard two place code*, kalkulator samsung *alternate two code numbers* dan kalkulator samsung *standard single place code. Design system* yang digunakan ada empat yaitu: *Activity Diagram*, *Usecase Diagram*, *Sequence Diagram*, dan *Deployment Diagram*. Keempat komponen tersebut adalah bagian dari UML (*Unified Modelling Language*).

Kata Kunci: Android, Kapasitor, *Mobile*

I. PENDAHULUAN

Seiring dengan berkembangnya perangkat elektronika di pasaran, baik itu berupa televisi, laptop, *handphone* dan perangkat elektronika lainnya. Tentunya komponen-komponen elektronika semakin berkembang dari ukuran yang besar menjadi semakin kecil, maka setiap perusahaan berlomba-lomba mengeluarkan produknya dengan ukuran yang lebih simpel, ramping dan praktis, dengan bentuk fisik yang lebih kecil.

Salah satu komponen elektro yang berkembang adalah kapasitor atau kondensator, fungsi dari kapasitor adalah untuk menyimpan muatan listrik atau energi listrik. Kapasitor yang banyak digunakan sekarang merupakan perkembangan dari kapasitor primitif yang kemudian menjadi cikal bakal kapasitor yang ada sekarang. Berdasarkan bahan isolator dan nilainya, kapasitor dapat dibagi menjadi dua jenis yaitu kapasitor nilai tetap dan kapasitor variabel. Kapasitor nilai tetap adalah kapasitor yang nilainya konstan atau tidak berubah-ubah. Berikut adalah jenis-jenis kapasitor nilainya tetep, diantaranya kapasitor keramik, kapasitor polyester, kapasitor kertas, kapasitor mika, kapasitor elektrolit, dan kapasitor tantalum. Sedangkan kapasitor variabel adalah kapasitor yang nilai kapasitasnya dapat diatur dan berubah-ubah. Berikut adalah jenis kapasitor yang nilainya berubah yaitu kapasitor varco dan kapasitor trimmer.

Sedangkan kapasitor yang dibahas dalam penulisan ini adalah kapasitor tantalum atau disebut juga dengan kapasitor SMD (*surface mount device*). Kapasitor tantalum adalah jenis dari kapasitor elektrolit yang elektrodenya dibuat dari bahan tantalum. Komponen ini mempunyai polaritas, pembedanya dengan yang lain ialah adanya tanda positif di badan kapasitor, tanda tersebut memiliki arti bahwa pin yang berada di bawahnya mempunyai polaritas yang positif, kelebihannya ialah frekuensi dan temperatur yang lebih baik dari pada kapasitor elektrolit yang dibuat dengan matrial alumunium.

Menurut Syamsudin (2014:55), pada umumnya beban pada jaringan listrik adalah beban induktif seperti motor listrik, *heater*, neon, lampu mercuri dan lain-lain. Jadi beban listrik kebanyakan adalah beban induktif yang membutuhkan daya reaktif. Jika beban listrik ini dipikul oleh pembangkit tenaga listrik, maka arus yang mengalir di jaringan juga semakin besar yang berakibat faktor dayanya menurun dan jatuh tegangan pada ujung saluran meningkat. Salah satu langkah efisiensi penggunaan energi listrik di konsumen adalah dengan memasang peralatan penghemat energi listrik yang digunakan baik disektor industri, bisnis komersial maupun rumah tangga. Peralatan penghemat energi listrik tersebut adalah kapasitor bank yang gunanya untuk menginjeksi daya reaktif pada titik-titik dimana terjadi tegangan jauh, sehingga diperoleh profil tegangan yang baik dan rugi daya yang lebih kecil.Kapasitor bank dapat memperbaiki power factor untuk meningkatkan kualitas daya sekaligus meningkatkan efisiensi pemakaian peralatan listrik konsumen dan akhirnya efisiensi energi listrik yang disediakan oleh penyedia tenaga listrik.

Aplikasi ini dibuat untuk memudahkan teknisi *handphone* untuk mengenali komponen kapasitor SMD, dikarenakan setiap perusahaan yang memproduksi kapasitor SMD mempunyai standar yang berbeda dalam setiap penilaian kodenya. Berdasarkan latar belakang masalah di atas, penulis akan membuat sebuah aplikasi *mobile* berbasis android yang difungsikan untuk perhitungan kapasitor SMD, mengingat perangkat ini sudah tidak asing lagi di masyarakat atau kalangan teknisi. Oleh sebab itu, penulis akan melakukan penelitian dengan judul **"**Aplikasi Perhitungan Kapasitor *Surface Mount Device*".

II. KAJIAN PUSTAKA

Menurut Wicaksono dan Hakim (2012:129) kapasitor adalah materi pada pelajaran fisika yang memerlukan media pembelajaran dalam penyampaiannya. Hal ini dikarenakan beberapa sebab, salah satunya kapasitor mempunyai bentuk, aturan, sifat, dan warna yang berbeda sehingga perlu media pembelajaran yang dapat memperlihatkan bentuk, ukuran, sifat, dan warna dari kapasitor serta menyajikan pembahasan tentang kapasitor secara lebih jelas dan menarik agar pemahaman terhadap materi kapasitor dapat diserap oleh siswa dengan baik.

Menurut Khairurrijal dkk (2009:93) menjelaskan pengertian kapasitor sebagai berikut: "Kapasitor adalah komponen listrik pasif yang memiliki kemampuan menyimpan muatan listrik, oleh karena itu dapat menyimpan energi listrik. Dalam sejarah penemuan, Ewald von Kleist membuat kapasitor pertama kali pada tahun 1745. Dari segi struktur, kapasitor terdiri dari dua konduktor yang dipisahkan oleh dielektrik (insulator atau isolator). Dalam memperkenalkan konsep komponen listrik yang dinamakan kapasitor, ada dua besaran bergantung waktu yang terlibat, yaitu muatan Q dan tegangan V. Muatan Q yang tersimpan di dalam kapasitor sebanding dengan tegangan V antara ujung-ujung kapasitor. Hubungan antara Q dan V secara matematis dinyatakan dengan persamaan Q = CV, dengan C adalah tetapan kesebandingan antara muatan dan tegangan. Besaran C dikenal sebagai kapasitansi. Formulasi kapasitansi untuk kapasitor keping sejajar adalah C = εA/l. Pada persamaan tersebut, ε adalah permitivitas listrik dari bahan dielektrik di antara dua pelat sejajar tersebut, A adalah luas penampang konduktor, dan l adalah jarak antara dua pelat konduktor. Jika tegangan berubah, maka muatan juga berubah. Sesuai dengan persamaan di atas, maka perubahan tegangan, Perubahan muatan q = q2 – q1 sebanding dengan perubahan tegangan v = v2 – v1.

Android

Safaat (2011:1) mengungkapkan bahwa "*Android* adalah sebuah sistem informasi untuk perangkat *mobile* berbasis linux yang mencakup sistem informasi, *middleware* dan aplikasi". Secara umum Android adalah platform yang terbuka (*Open Source*) bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh berbagai piranti bergerak.

Hingga tahun 2012, Android telah berkembang dengan pesat. Dalam kurun tiga tahun Android telah diproduksi dalam versi, dan versi terakhir yang diproduksi disebut sebagai Android versi 4.1 atau Android Jelly Bean.

Berikut ini versi android sampai saat ini:

1. Android versi 1.1
2. Android versi 1.5 (Cupcake)
3. Android versi 1.6 (Donut)
4. Android versi 2.0/2.1 (Eclair)
5. Android versi 2.2 (Froyo: Frozen Yoghurt)
6. Android versi 2.3 (Gingerbread)
7. Android versi 3.0/3.1 (Honeycomb)
8. Android versi 4.0 (ICS :Ice Cream Sandwich)
9. Android versi 4.1 (Jelly Bean)

AVD (*Android Virtual Device*)

Safa'at (2011:9) menyebutkan bahwa "AVD merupakan *emulator* untuk menjalankan program aplikasi Android". AVD ini nantinya yang dijadikan sebagai tempat tes dan menjalankan aplikasi android yang dibuat.

Eclipse

Safaat (2011:16) menjelaskan bahwa "Eclipse adalah IDE untuk pengembangan java atau android", dapat dijalankan di semua platform (*platform-independent*). Berikut ini adalah sifat dari Eclipse:

a. *Multi-platform*: Target sistem operasi Eclipse adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.
b. *Mulit-language*: Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++ Cobol, Python, Perl PHP, dan lain sebagainya.
c. *Multi-role*: Selain sebagai IDE untuk pengembangan aplikasi, Eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, *test* perangkat lunak, pengembangan *web*, dan lain sebagainya.

Eclipse pada saat ini merupakan salah satu IDE favorit dikarenakan gratis dan *open source*, yang berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan dari Eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan *plug-in*.

Android SDK

Safaat (2011:15) menjelaskan bahwa "Android SDK adalah tool API (*Application Programming Interface*) yang diperlukan untuk memulai mengembangkan aplikasi pada platform android menggunakan bahasa pemprograman java". Untuk source SDK android ini dapat dilihat dan diunduh langsung dari situs resmi penembang android di http//:www.developer.android.com.

Metode Algoritma

Ahmad Zaelani, dkk (2006:73) mengatakan bahwa "Persamaan Linear Dua Variabel adalah persamaan yang memiliki dua variabel dengan pangkat masing-masing variabel sama dengan satu". Bentuk umum Persamaan Linear Dua Variabel adalah ax + by +c = 0, dengan a, b tidak nol dan a, b, c merupakan bilangan riil. Sedangkan x dan y disebut variabel, a dan b disebut koefisien dan c disebut kostanta.

Pengujian *Software*

Efendi (2011:22) menjelaskan bahwa "Pengujian *software* dilakukan untuk mencegah terjadinya kesalahan dalam program dengan metode-metode tertentu".

Pengujian *software* adalah proses menganalisis suatu entitas *software* untuk mendeteksi perbedaan antara kondisi yang ada dengan kondisi yang diinginkan (*detects/error/bugs*) dan mengevaluasi fitur-fitur dari entitas *software*.

Di dalam pengujian *software* terdapat beberapa metode diantaranya adalah:

1. Metode Pengujian *White Box*

Menurut Rizky (2011:264) kadang disebut pula pengujian *glass box*, adalah metode desain *test case* yang menggunakan struktur kontrol desain prosedural untuk memperoleh *test case* atau dengan kata lain bahwa pengujian dilakukan untuk memastikan bahwa operasi internal bekerja sesuai dengan spesifikasi dan semua komponen internal telah diamati dengan baik. Dengan menggunakan metode pengujian ini perekayasa sistem dapat melakukan *test case* yaitu:

a. Memberi jaminan bahwa semua jalur *independent* pada suatu modul telah digunakan paling sedikit satu sekali.
b. Menggunakan semua keputusan logis pada sisi *true* dan *false*.
c. Mengeksekusi semua *loop* sesuai dengan batasan.

2. Metode Pengujian *Black Box*

Menurut Rizky (2011:261) pengujian ini berfokus pada persyaratan fungsional perangkat lunak. Pengujian ini memungkinkan perekayasa sistem mendapatkan serangkaian kondisi *input* yang sepenuhnya menggunakan semua persyaratan perangkat fungsional untuk suatu program. Pengujian ini berusaha menemukan kesalahan dalam kategori sebagai berikut:

a. Fungsi-fungsi yang tidak benar atau hilang.
b. Kesalahan *interface*.
c. Kesalahan dalam struktur data atau akses *database* eksternal.
d. Kesalahan kinerja.

Inisialisasi dan kesalahan terminasi.

Peralatan Pendukung

Membuat suatu aplikasi android diperlukan berbagai macam pendukung yang dapat membantu terbentuknya suatu aplikasi mulai dari logika hingga konsep pemrogramannya.

a. OOP (*Object Oriented Program*)

Shalahuddin (2011:82) menyebutkan bahwa "Metologi berorientasi objek adalah suatu strategi pembangnan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya".

Metode berorientasi objek meliputi rangkaian aktivitas analisis berorientasi objek, perancangan berorientasi objek, pemrograman berorientasi objek dan pengujian berorientasi objek, metode berorientasi objek lebih banyak dipilih karena metode lama banyak menimbulkan masalah seperti adanya kesulitan pada saat mentransformasikan hasil dari satu tahapan pengembangan ke tahap berikutnya.

b. UML (*Unifield Modeling Language*)

Pada perkembangan teknik pemrograman berorientasi objek, muncullah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman yang berbasi objek yaitu *Unifield Modeling Language* (UML). UML muncul karena adanya kebutuhan pemodelan visual untuk mengimplementasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung.

1. *Class Diagram*
   Shalahuddin (2011:2012) menjelaskan bahwa "Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem". Kelas memiliki apa yang disebut atribut dan metode atau operasi.
2. *Use Case Diagram*
   Shalahuddin (2011:130) menjelaskan bahwa "*Use Case Diagram* merupakan pemodelan untuk kelakuan sistem informasi yang akan dibuat".
3. *Activity Diagram*
   Shalahuddin (2011:134) menjelaskan bahwa "Diagram aktivitas atau *Activity Diagram* mengambarkan *work flow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis".
   Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor jadi aktivitas yang dapat dilakukan sistem. Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut.
4. *Sequence Diagram*
   Shalahuddin (2011:137) menjelaskan bahwa "Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *messege* yang dikirimkan dan diterima antar objek". Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam *use case* beserta metode-metode yang dimiliki kelas yang diimplementasikan menjadi objek itu. Banyaknya diagram sekuen yang harus digambarkan adalah sebanyak pendefinisian *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak.

III. METODE PENELITIAN

Analisis Penelitian

Untuk mengimplementasikan aplikasi yang akan dibuat penulis memerlukan analisis untuk kebutuhan dalam perancangannya diantaranya:

a. Analisis kebutuhan
   Adapun perancangan aplikasi ini membutuhkan suatu perangkat yang diantaranya adalah *Hardware* berupa laptop dengan spesifikasi prosesor Pentium(R) *dual core T4400 @ 2.20 GHZ, RAM 2 GB, video card 1 GB, hardisk 320 GB* dan *Software* yaitu *windows 7 32 bit, Eclips Indigo, ADT (Android Development Tools) 18.0.0 Plugin, Java JDK (Java Development kit)* dan *Android SDK (Standart Development Kit).*
b. Desain
   Dalam mendesain program penulis menggunakan *Eclipse indigo*. Diantara fitur Eclipse indigo disediakan konsep GUI (*Graphical User Interface*) sehingga memudahkan seorang *programmer* dalam mendesain. Adapun jika ingin mendesain secara manual pun bisa, sedangkan konsep *OOP (Objeck Oriented Program)* adalah konsep pemrograman *java.*
   - *Software architecture*
     Software architecture yaitu proses yang mendefinisikan solusi yang terstruktur yang memenuhi kebutuhan teknis dan operasional, disisi lain mengoptimasi qualitas dari sebuah aplikasi yang meliputi: *performance*, *security*, dan *manageability*. Adapun program yang dibuat penulis termaksud *software architecture* berbasis OOP.
   - *User Interface*
     Adapun *User Interface* yang digunakan untuk membangun aplikasi ini, bisa menggunakan android *draw* atau bisa langsung di-*eclipse* indigo itu sendiri yang sudah disediakan fungsi GUI.

c. Testing
Untuk melakukan uji coba program yang sudah dibuat, penulis menggunakan *Android Virtual Device (AVD)* yang merupakan emulator untuk menjalankan program aplikasi android, *AVD* ini nantinya yang dijadikan sebagai tempat *test* dan menjalankan aplikasi android yang dibuat, *AVD* berjalan di *Virtual Machine*. Adapun pengujian lainnya dilakukan dengan *White Box Testing* yaitu pengujian dengan melihat ke dalam modul untuk meneliti kode-kode program yang ada, dan menganalisis apakah ada kesalahan atau tidak serta *Black-box testing* adalah metode pengujian perangkat lunak yang tes fungsionalitas dari aplikasi yang bertentangan dengan struktur internal.

d. Implementasi
   Jika pengujian di *AVD* tidak ada kesalahan maka aplikasi android di-*install* ke dalam *Mobile* dengan Operasi Sistem *Android.*

Metode Pengumpulan Data

Dalam aplikasi pembuatan aplikasi ini penulis menggunakan beberapa metode diantaranya:

a. Observasi
   yaitu perbandingan terhadap aplikasi elektronika yang sudah dibuat sebelumnya.
b. Studi Kepustakaan
   yaitu pengambilan data dengan cara mengambil materi-materi yang berhubungan dengan judul karya ilmiah melalui buku-buku dan jurnal.

IV. HASIL DAN PEMBAHASAN

Analisis Kebutuhan *Software*

Pada bab ini akan disampaikan tahapan analisis kebutuhan perangkat lunak yang merupakan langkah awal dalam pembuatan aplikasi perhitungan kapasitor SMD berbasis android.

a. Identifikasi Masalah

Identifikasi Permasalahan adalah salah satu proses penelitian yang boleh dikatakan paling penting diantara proses lain. Masalah penelitian secara umum bisa ditemukan lewat studi literatur atau lewat pengamatan lapangan (observasi, survei, dan sebagainya). Pada penulisan skripsi permasalahan yang akan diteliti dalam pembuatan aplikasi perhitungan kapasitor SMD dengan perangkat lunak android berbasis *mobile* atau *tablet* yang mampu memberikan informasi tentang pengetahuan bagaimana caranya menghitung kapasitor SMD.

b. Analisis Kebutuhan
   Tahap Analisis kebutuhan
   Tahap analisis kebutuhan mencakup *hardware*, *software*, aplikasi, dan output yang digunakan adalah sebagai berikut.
   a. Komponen *Hardware.*
      Komputer yang digunakan mempunyai spesifikasi sebagai berikut.
      1) Tipe :Notebook acer 4732Z
      2) HDD :160 GB
      3) RAM : 2 GB
      4) *Proccesor*: Pentium® *Dual core T4400 @ 2.20 GHZ*
      5) *Graphic*: *Radeon (tm) HD Graphic 1.0 GHz*

      Dalam pembuatan aplikasi android minimal *Dual core* karena akan berpengaruh terhadap pembuatan virtual android yang akan digunakan.
   b. Komponen *Software*
      Komponen perangkat lunak yang digunakan untuk membuat sistem tersebut adalah.
      1) *Eclipse*
         Eclipse merupakan tempat untuk membuat projek aplikasi *android* dan

ada beberapa *device* yang harus di-*install* di eclips diantaranya:
a) *Android* SDK
b) *Android* ADT

2) *Java* JDK
Java JDK digunakan untuk plugin bahasa pemrograman java.
3) *Adobe PhotoShop 7.0*
Software yang diguakan untuk mendesain tampilan berupa gambar-gambar yang akan digunakan dalam program *android*.

c. Aplikasi
Aplikasi yang digunakan adalah aplikasi yang berbasis android sehingga program tersebut dapat digunakan untuk menjalankan fungsinya.

d. *Input/output*
Input/output yang digunakan adalah input penggunaan dari *interface android* itu sendiri yang menghasilkan output berdasarkan dari input yang dimasukkan.

Desain

Sebelum aplikasi android ini di implementasikan dalam bentuk package .apk, maka perlu dirancang terlebih dahulu. Tahap perancangan bertujuan untuk memenuhi kebutuhan pengguna dan memberikan gambaran yang jelas mengenai aplikasi yang akan dibuat. Keseluruhan dari perancangan ini akan diimplementasikan dalam *gadget* berbasis android dengan menggunakan perangkat lunak Eclipse.

a. Rancangan Algoritma

Berhubung aplikasi perhitungan kapasitor SMD berbasis android terdapat banyak kalkulator komponen, maka dengan ini penulis mengambil 1 sampel saja dari sekian fitur aplikasi yang penulis buat. Contohnya adalah kapasitor SMD ventax standar. Adapun algoritma yang digunakan untuk menyelesaikan masalah perhitungan kapasitor SMD ventax standar adalah algoritma linear. Di dalam algoritma linier terdapat variabel relatif (x dan y). Dalam perhitungan kapasitor SMD ventax standar terdapat beberapa variabel yang dijadikan sebagai variabel relatif.

Variabel relatif:
X = Nilai digit pertama
Y = Nilai digit kedua
Z = hasil nilai kapasitansi

Untuk menghitung kapasitor maka dilakukan perhitungan dengan data-data sebagai berikut.

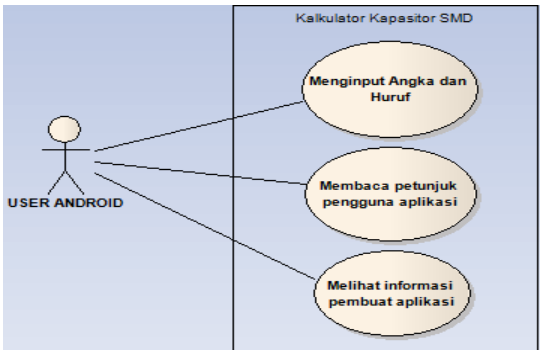
X = 60
Y = 10^2
Z = ?
(X.Y) – Z = 0
(X.Y) = Z
(60x10^2) = Z
(6000) = Z

b. *Software Architecture*

Rekayasa perangkat lunak merupakan pembangunan dengan menggunakan prinsip dan konsep rekayasa dengan tujuan menghasilkan perangkat lunak yang bernilai ekonomi yang dipercaya dan bekerja secara efisien menggunakan mesin. Perangkat lunak banyak dibuat dan pada akhirnya sering tidak digunakan karena masalah-masalah non teknis.

1. Diagram *Use Case* Aplikasi Kapasitor SMD

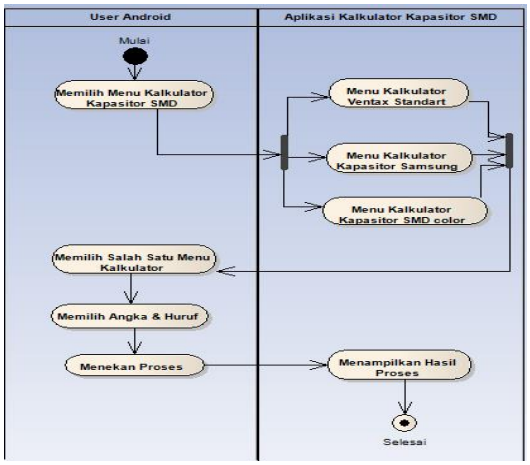
Use Case merupakan model diagram UML yang digunakan untuk menggambarkan *requirement* fungsional yang diharapkan dari sebuah sistem. Diagram *Use Case* menekan pada "siapa" melakukan "apa" dalam lingkungan perangkat lunak yang akan dibangun.



Gambar 1. Diagram *Use Case* Aplikasi Kapasitor SMD

Menggambarkan dimana menu awal aplikasi dimana pertama kali pengguna akan menemui menu utama yang terdiri dari tiga *button* menu yaitu *button* menu kalkulator, bantuan, dan tentang.

2. Diagram *Activity* memilih menu bantuan

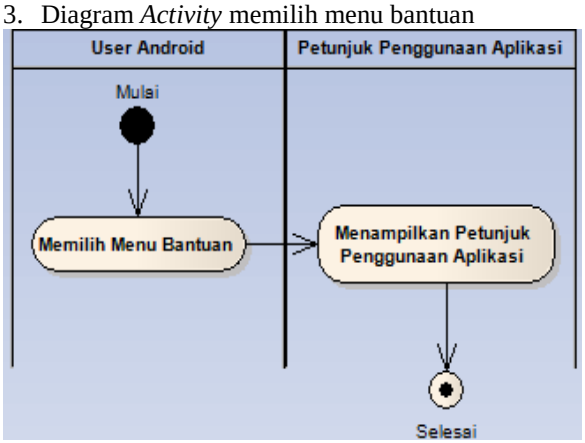


Gambar 2. Diagram *Activity* Memilih Menu Kalkulator

Pada diagram ini digambarkan mengenai *activity* dari aplikasi kalkulator kapasitor secara keseluruhan dimulai pada saat pengguna membuka

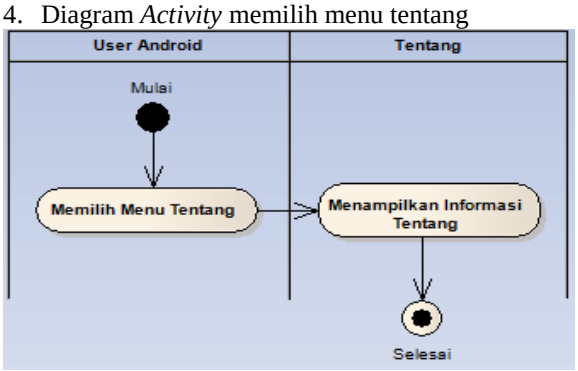
aplikasi sampai dengan pengguna melakukan pilihan untuk keluar atau tidak.

3. Diagram *Activity* memilih menu bantuan



Gambar 3. Diagram *Activity* Memilih Menu Bantuan

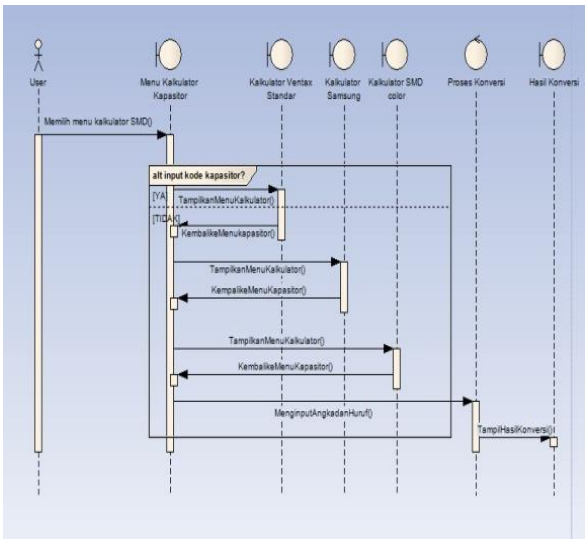
4. Diagram *Activity* memilih menu tentang



Gambar 4. Diagram Activity Memilih Menu Tentang

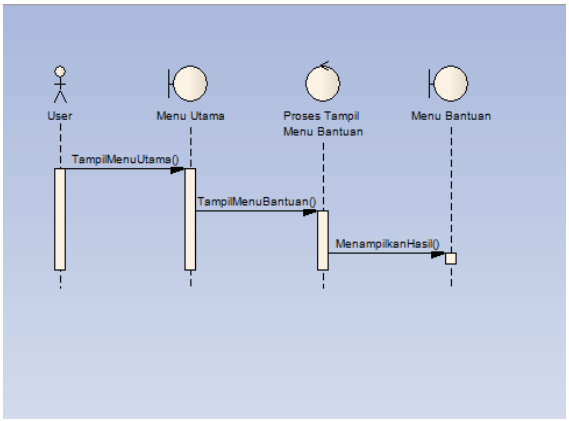
Diagram ini menggambarkan *activity* untuk menampilkan informasi pembuat aplikasi kalkulator kapasitor SMD.

5. Diagram *Sequence* menilih menu kalkulator



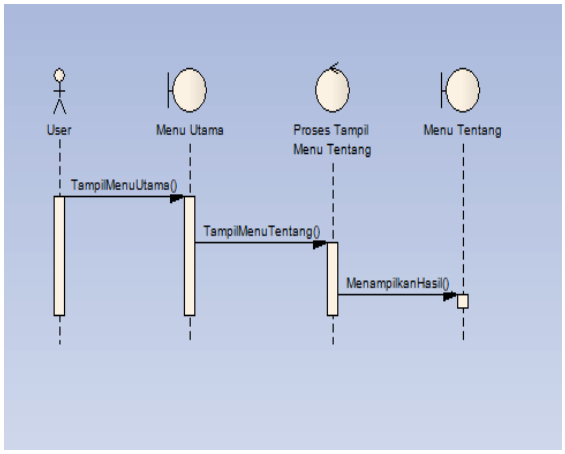
Gambar 5. Diagram Sequence Memilih Menu Kalkulator

6. Diagram *Sequence* memilih menu bantuan



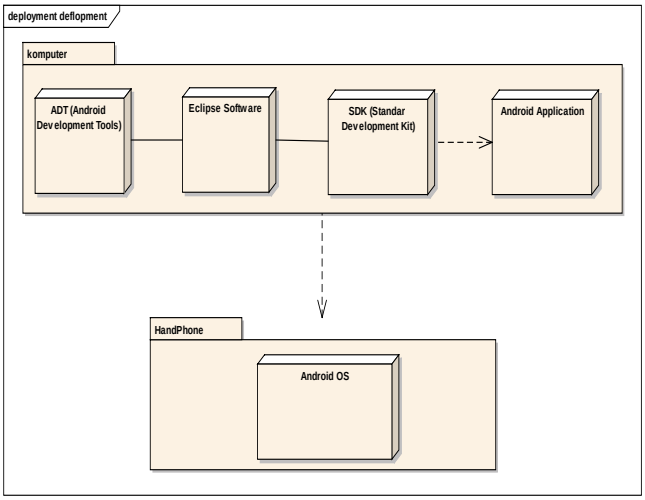
Gambar 6. Diagram *Sequence* Memilih Menu Bantuan

7. Diagram *Sequence* memilih menu tentang



Gambar 7. Diagram Sequence Memilih Menu Tentang

8. Diagram *Deployment*



Gambar 8. Diagram *Deployment*

User Interface

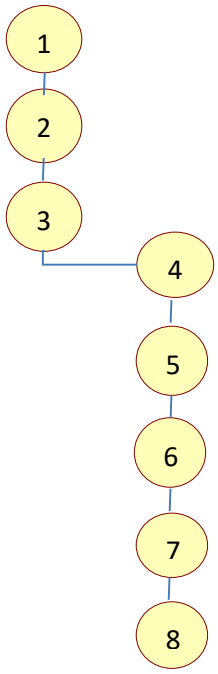
Arsitek *User Interface* pada aplikasi android adalah *user interface* yang meliputi *Activity* dan *User Interface* yang terdiri dari komponen. Semuanya yang berhubungan dengan *user interface* pada aplikasi android biasanya berada pada lokasi *res/layout/filename.xml*. Adapun dalam aplikasi ini menggunakan interface sebagai berikut.

Gambar 9. Tampilan Menu Utama

Testing

Testing menngunakan *white box* dan *black box* untuk pengujian *whitebox*, dengan menggunakan skema diagram alir, berikut merupakan diagram alir dari aplikasi perhitungan kapasitor smd.

1. White Box



Gambar 10. Skema Diagram Alir

Kompleksitas siklomatis dari grafik alir dapat diperoleh dengan perhitungan:

V(G) = E – N + 2

Dimana :
E = jumlah *Edge* yang ditentukan gambar panah.
N = jumlah simpul grafik alir ditentukan dengan gambar lingkaran.

V(G) = 7 – 8 + 2 = 1
V(G) < 8 berarti memenuhi syarat kekomplesitas siklomatisnya.

Baris set yang dihasilkan dari jalur *independent* adalah sebagai berikut:

a. 1-2-3-4-5-6-7-8
b. Ketika aplikasi dijalankan, maka terlihat bahwa satu set baris yang dihasilkan adalah 1-2-3-4-5-6-7-8 dan terlibat bahwa simpul telah dieksekusi satu kali.

2. Black Box

Pengujian selanjutnya dilakukan untuk memastikan bahwa suatu *event* atau masukan menjelaskan proses yang tepat dan menghasilkan output yang sesuai dengan rancangan.

Tabel 1. Pengujian *Black Box*

Input	Proses	Output	Validasi
Tombol Gambar digit 1	Memberikan input sesuai dengan gambar yang di input ke 1	Tampil Gambar sesuai yang dipilih user	Sesuai
Tombol Gambar digit 2	Memberikan input sesuai dengan gambar yang di input ke 2	Tampil Gambar sesuai yang dipilih user	Sesuai
Tombol Proses	Melakukan proses perhitungan	Menghasilkan kapasitansi	Sesuai
Tombol Keluar	Menghentikan aplikasi	Keluar dari aplikasi	Sesuai

V. PENUTUP

Setelah aplikasi android ini dibuat dapat disimpulkan bahwa:

1. Membantu masyarakat umum untuk mempelajari dalam perihal ruang lingkup kapasitor *Surface Mounted Device* (SMD).
2. Android merupakan sistem operasi yang mendukung penulisan *script* dalam bahasa *java*.

3. Android merupakan penggabungan antara script *XML* dan *Java*.
4. Memanfaatkan media komulikasi interaksi berkembang sekarang sebagai media pelajaran
5. Menghitung kapasitor bisa dengan lebih mudah dan praktis
6. Aplikasi yang dibuat penulis membantu pembelajaran tentang perhitungan kapasitor smd bagi masyarakat umum khususnya para teknisi *handphone*.

DAFTAR PUSTAKA

Ahmad Zaelani. 2006. Pendalaman Kompetensi Matematika dan Uji Latihan Mandiri Untuk Kelas VIII SMP. Bandung: Yrama Widya Tama

A.s, Rosa dan M.Shalahuddin. 2011. Modul Pembelajaran Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek). Bandung: Modula.

Khairurrijal, Mikrajuddin Abdullah, Neni Surtiyeni, Widayani, dan Euis Sustini.2009.Konsep Komponen Listrik(kapasitor, Induktor, dan Memristor) Menggunakan Analogi Konsep Resistor untuk Pengajaran di Sekolah Menengah Atas.ISSN: 1979-4959.Bandung: Jurnal Pengajaran Fisika Sekolah Menengah Vol. 1, No.4, November 2009. (17 Mei 2015)

Noor Syamsudin dan Noor Saputera. 2014. EFISIENSI PEMAKAIAN DAYA LISTRIK MENGGUNAKAN KAPASITOR BANK. Banjarmasin: Jurnal POROS TEKNIK, Volume 6, No. 2, Desember 2014 : 55 – 102. (12 Mei 2015)

Rudie. 2014. CARA MENGUKUR CAPACITOR SMD.Diambil dari: www.ps3oon.com/2014/05/cara-mengukur-komponen-smd-pada-ps3.html. (09 Juni 2015)

Safaat, Nazrudin. 2012. Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android. Bandung: Informatika Bandung.

Wicaksono Deny Satria da fitro Nur Hakim. 2012. MEDIA PEMBELAJARAN FISIKA INTERAKTIV BAHASAN KAPASITOR BERBASIS FLASE DAN XML. Semarang: *Indonesian Jurnal on Computer Science Speed* - FTI UNSA Vol 9 No 3 - Desember 2012.(10 Juli 2015)