

## PERANCANGAN PERANGKAT LUNAK UNTUK MENGGAMBAR DIAGRAM BERBASIS ANDROID

**Wilianti Aliman**

Sekolah Tinggi Manajemen Informatika dan Komputer LIKMI (STMIK LIKMI)  
Bandung, Indonesia

Email: wiliantialiman@gmail.com

### **Abstract**

*UML diagrams misreside developer apps for men to design systems. Research which way android-based UML drawing diagram software to improve the mobility of developer applications from time to time. The purpose of this research is to know every uml diagram especially using case diagrams, activity diagrams, diagrams and classes to know so analysts in building a information system. Assisting users to draw diagrams mobilely, assisting analysis and design analysis of the design of an information system, and at that time the application and development of computer graphics and information systems that have been tasted and during the lecture. Which research force is with the System Development Life Cycle (SDLC) Prototype model. The results of which the results show that the analysis, diagram of the application design and creation in the application "Vistchart" in this application, the user's menu diagram that chooses to be drawn is a usage case diagram, activity diagram or class diagram.*

**Keywords:** computer graphics; UML diagram drawing tools; android canvas

### **Abstrak**

Diagram UML merupakan sebuah alat bantu bagi seorang pengembang aplikasi untuk mendesain sistem. Penelitian yang dilakukan adalah perancangan perangkat lunak menggambar diagram UML berbasis android untuk meningkatkan mobilitas seorang pengembang aplikasi. Tujuan dari penelitian ini adalah untuk mengetahui informasi ketentuan setiap diagram UML khususnya *use case diagram*, *activity diagram*, dan *class diagram* untuk kebutuhan seorang analis dalam membangun sebuah sistem informasi. Membantu pengguna untuk menggambar diagram secara mobile, membantu pembelajaran analisis dan desain perancangan sebuah sistem informasi, dan sebagai penerapan dan pengembangan dari ilmu grafika komputer dan sistem informasi yang sudah dipelajari dan ditempuh selama perkuliahan. Metode penelitian yang digunakan adalah dengan menggunakan *model System Development Life Cycle (SDLC) Prototype*. Hasil penelitian yang didapatkan menunjukkan bahwa analisis, perancangan dan pengujian aplikasi menggambar diagram "Vistchart" pada aplikasi ini, pengguna memilih menu diagram yang akan digambar yaitu *use case diagram*, *activity diagram* atau *class diagram*.

**Kata Kunci:** grafika komputer; alat gambar UML diagram; android canvas

<b>How to cite:</b>	Aliman, Wilianti (2021) Perancangan Perangkat Lunak untuk Menggambar Diagram Berbasis Android. <i>Syntax Literate</i> . 6(6). <a href="http://dx.doi.org/10.36418/syntax-literate.v6i6.1404">http://dx.doi.org/10.36418/syntax-literate.v6i6.1404</a>
<b>E-ISSN:</b>	2548-1398
<b>Published by:</b>	Ridwan Institute

## Pendahuluan

Sistem informasi yang baik dibangun dengan kerjasama antara seorang analis dengan tim *programmer*. Seorang analis akan merancang sistem dengan sebuah siklus yang biasa disebut dengan SDLC atau *System Development Life Cycle*. Salah satu tahapan dalam SDLC adalah *design*, tahap ini bertujuan agar sistem informasi yang dirancang dapat memenuhi kebutuhan *user*. Diagram menjadi cara untuk mengkomunikasikan setiap kebutuhan user kepada *programmer* (Lee, 2012).

Kemudahan yang hadir melalui Android seperti tingkat mobilitas tinggi, penginstalan yang praktis, dan pemakaian yang lebih sederhana dan mudah dimengerti, membantu pengguna yang akan menggambar diagram dengan mobilitas yang tinggi dan juga memberikan alternatif yang baik untuk seorang analis yang tidak memiliki PC (*Personal Computer*) (Sommerville, 2011). Kehadiran sebuah aplikasi perancang diagram dalam OS Android akan mempermudah pengguna dalam mempelajari dan menggambar sebuah diagram dengan tingkat mobilitas yang cukup tinggi. Aplikasi menggambar diagram dirancang dengan fungsi untuk merancang diagram-diagram yang dibutuhkan dalam membangun sebuah sistem informasi (Eck, 2016). Aplikasi ini menyediakan template-template komponen untuk menggambar diagram yang mudah digunakan oleh user dengan cara melakukan *pick and drop* pada kanvas yang telah tersedia pada aplikasi. Aplikasi ini menyediakan sistem yang dapat memvalidasi *relational* yang digambar oleh pengguna sesuai dengan aturan dasar dalam penggambaran diagram UML seperti *use case diagram*, *activity diagram*, dan *class diagram*. Aplikasi ini juga dapat melakukan penyimpanan dalam bentuk format JPG yang tersimpan pada penyimpanan foto (Mufadhhol & Suprayogi, 2014).

Aplikasi menggambar diagram UML dapat dikembangkan untuk bidang lainnya seperti untuk bidang Pendidikan. Perancangan Aplikasi dengan fitur pengecekan tersebut akan membantu pengguna meminimalisir kesalahan dari aturan menggambar diagram UML yang sudah ada, terutama bagi seseorang yang masih dalam proses belajar menggambar diagram UML.

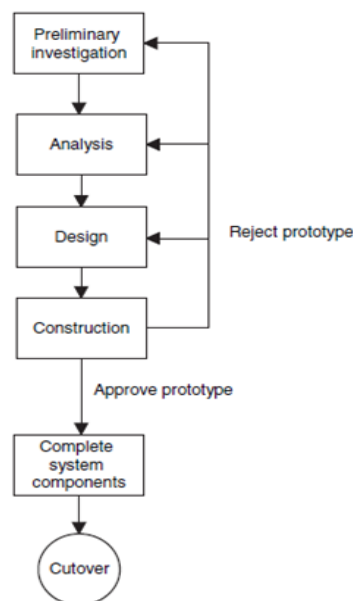
*Unified Modeling Language* (UML) sudah diterima luas sebagai standar untuk pemodelan objek. *Object Management Group* (OMG), badan standar industri, mengadopsi UML pada bulan November 1997 (Profesi & Henderi, 2018). Penggunaan notasi dan garis penghubung yang tidak sesuai dengan standar yang ada akan membuat kebingungan bagi seorang pengembang berkomunikasi, maka perlu adanya sebuah validasi untuk memeriksa kesesuaian penggunaan bentuk notasi dan garis penghubung. Aplikasi untuk menggambar diagram seperti yang sudah ada saat ini, belum ada yang menawarkan fitur untuk melakukan pengecekan sesuai dengan aturan internasional UML.

Berdasarkan latar belakang yang telah dijelaskan di atas akan dirancang sebuah aplikasi Android sebagai media untuk penggambaran diagram dalam membangun sebuah sistem dengan judul “Perancangan Perangkat Lunak Untuk Menggambar Diagram UML Berbasis Android”.

Tujuan dari penelitian ini adalah untuk mengetahui informasi ketentuan setiap diagram UML khususnya *Use Case Diagram*, *Activity Diagram*, dan *Class diagram* untuk kebutuhan seorang analis dalam membangun sebuah sistem informasi. Membantu pengguna untuk menggambar diagram secara mobile, membantu pembelajaran analisis dan desain perancangan sebuah sistem informasi, dan sebagai penerapan dan pengembangan dari ilmu grafika komputer dan sistem informasi yang sudah dipelajari dan ditempuh selama perkuliahan.

### Metode Penelitian

Metode penelitian yang digunakan adalah dengan menggunakan model *System Development Life Cycle (SDLC) Prototype*. Model SDLC yang klasik memiliki beberapa kelemahan salah satunya mengenai waktu pengerjaan yang cukup lama sekitar 3 sampai 5 tahun untuk melewati tahap-tahapnya (Sabale & Dani, 2012). Pengembangan sistem untuk mengatasi hal ini disediakan oleh sebuah model yaitu *prototyping* karena dinilai efektif dan lebih cepat. Tahapan *prototyping* diatur dengan salah satu dari empat tahap pertama diulang sampai pengguna menyetujui *prototype* (Pressman, 2010). Setelah prototipe disetujui maka sistem akan menjadi sebuah *blue print* yang akan diimplementasikan, untuk lebih jelas dapat dilihat pada gambar 1.



**Gambar 1**  
**Tahapan *prototyping***

Sumber: (Everett & McLeod Jr, 2007)

Pada perancangannya digunakan konsep *object oriented* dengan memanfaatkan Android Canvas pada Android studio sebagai alat pembuatannya (DiMarzio, 2016). Definisi *object-oriented* menurut John Horton dalam buku “*Android Programming for Beginners*” adalah sebagai berikut:

“OOP is a way of programming that involves breaking our requirements down into chunks that are more manageable than the whole.” (Horton, 2015). Penelitian mengenai aturan penggambaran juga dilakukan dengan melakukan studi pustaka pada website resmi UML diagram. UML atau *Unified Modeling language* merupakan sebuah bahasa pemodelan yang sudah banyak dipakai dalam metode *object-oriented* (Iriyanto & Zaini, 2014). Pengertian UML menurut C. Thomas Wu dalam buku “An Introduction To Object-Oriented Programming with java, Fifth Edition” adalah sebagai berikut:

“The unified modelling language (UML) provides graphical notation to model computer systems that are developed by using object-oriented software engineering (OOSE)” (Wu, 2006).

UML bukan hanya sekedar sebuah metodologi, tetapi merupakan sebuah gambaran menggunakan notasi-notasi yang dapat menggambarkan desain dari sebuah program untuk dijadikan sebuah bahasa komunikasi untuk berdiskusi mengenai sistem yang sedang dirancang (Kendal, 2009).

Berdasarkan pengertian dari berbagai sumber buku dapat ditarik kesimpulan bahwa UML merupakan sebuah bahasa untuk memodelkan sebuah sistem yang akan dirancang melalui notasi-notasi yang sudah disepakati (Sukamto & Shalahuddin, 2013). Pemodelan dibagi beberapa diagram pemodelan menurut C. Thomas Wu dalam buku “An Introduction To Object-Oriented Programming with java, Fifth Edition” adalah sebagai berikut:

### 1. Use Case Diagram

*Use Case* merupakan langkah pertama dalam memodelkan sebuah sistem. *Use Case* merupakan pemodelan untuk kebutuhan sebuah sistem fungsional, setiap *Use Case* digambarkan sebagai kunci dari suatu skenario yang dilakukan oleh aktor dan diringkas dalam sebuah batas sistem, setiap *Use Case* dihubungkan dengan sebuah garis notasi (Kurniawan, 2018).

### 2. Activity Diagram

Setelah membuat model *Use Case*, maka setiap scenario yang ada di *Use Case* akan dideskripsikan lebih jelas di dalam *activity diagram*. *Activity diagram* merupakan pemodelan yang menggambarkan sebuah sistem kerja dari sebuah objek atau sebuah sistem, sebuah *activity diagram* digambarkan dengan sebuah alur secara terstruktur proses kerja dari *use case* yang sedang diproses dari titik awal sampai titik akhir, setiap aktivitas digambarkan dengan notasi-notasi sesuai fungsinya (Kusumanto & Tomponu, 2011).

### 3. Class Diagram

*Class diagram* adalah salah satu pemodelan yang cukup penting dalam UML, fungsinya adalah untuk membuat sebuah logical models dari sebuah sistem (Wu, 2006). Sebuah *class diagram* akan menunjukkan bagaimana skema dari arsitektur sebuah sistem yang sedang dirancang (Kendal, 2009). *Class diagram* digambarkan dengan *class* yang berisi atribut dan *method*, setiap *class* akan dihubungkan dengan sebuah garis disebut Asosiasi.

**Hasil dan Pembahasan**

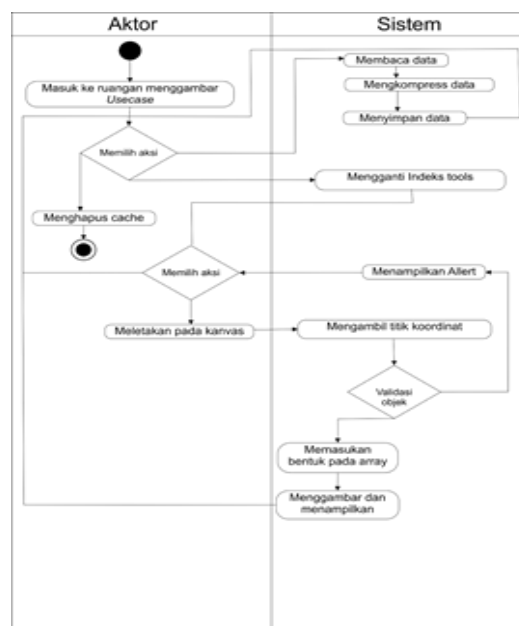
Perancangan perangkat lunak dijalankan pada Android dengan sistem operasi minimum versi 7.0 (Nougat). Spesifikasi perangkat keras yang dibutuhkan adalah sebagai berikut:

**Tabel 1**  
**Spesifikasi perangkat keras**

Spesifikasi Perangkat Keras	
Merk	Asus
Tipe	Zenfone 3 max ZC520TL
Ram	2 GB
Display	5.2 inchi
Internal Storage	16 GB

Sumber: Hasil Penelitian Tahun 2021

*Activity diagram* menggambarkan sebuah alur kerja dari sebuah sistem perangkat lunak yang sedang dibangun. Berikut merupakan *activity diagram* utama dari setiap *use case* yang ada pada *use case diagram* yang telah dirancang (Binanto, 2010).



**Gambar 2**  
*Activity Diagram Use Case Menggambar diagram "Use Case"*

Pada aplikasi tersebut dirancang hasil ekspor dari UML tersebut berupa gambar berformat JPG atau PNG. Selain itu, hasil penggambaran dapat disimpan dalam bentuk vektor agar dapat digambar kembali. Salah satu keunggulan memanfaatkan Android Canva adalah proses penerapan pemahaman mengenai gambar vektor dimana gambar

vektor merupakan susunan dari pixel-pixel dengan perhitungan atribut-atribut vektor, sehingga memungkinkan untuk tetap melakukan perubahan bentuk, warna atau atribut lainnya.

Diagram UML tentunya merupakan sebuah rancangan yang dinamis karena penuh dengan perhitungan dan penyesuaian, maka gambar vektor dapat menjadi salah satu cara untuk menyimpan hasil penggambaran diagram awal yang memungkinkan perubahan di waktu yang akan datang apabila dibutuhkan. Penyediaan dan pengembangan aplikasi menggambar diagram dengan Android akan menjawab permasalahan mobilitas, sehingga setiap *analyst* dapat melakukan penyesuaian gambaran diagram dimanapun dan kapanpun yaitu melalui media Smartphone dengan sistem operasi Android yang pastinya sudah tidak asing dan sudah digunakan oleh banyak orang saat ini.

### **Kesimpulan**

Berdasarkan hasil penelitian, analisis, perancangan dan pengujian aplikasi menggambar diagram “Vistchart” ini dapat ditarik kesimpulan bahwa pada aplikasi ini pengguna memilih *menu diagram* yang akan digambar yaitu *Use Case diagram*, *activity diagram* atau *class diagram*. Setelah itu pengguna dapat melakukan penggambaran dengan memilih template notasi yang sudah tersedia dan meletakkan objek pada kanvas sehingga sistem akan membaca titik koordinat untuk menggambar bentuk sesuai dengan notasi yang dipilih. Penerapan ilmu menggambar diagram sangatlah luas, dengan aplikasi ini membantu para pengguna untuk mengerti bentuk-bentuk notasi yang ada dalam membangun sebuah diagram dan juga fungsi validasi sederhana untuk penggambaran yang dapat membantu pengguna mengerti dalam memberikan garis penghubung atau *association*.

## BIBLIOGRAFI

- Binanto, Iwan. (2010). *Multimedia Digital-Dasar Teori Dan Pengembangannya*. Penerbit Andi. [Google Scholar](#)
- Dimarzio, Jerome. (2016). *Beginning Android Programming With Android Studio*. John Wiley & Sons. [Google Scholar](#)
- Eck, David J. (2016). *Introduction To Computer Graphics*. David J. Eck. [Google Scholar](#)
- Everett, Gerald D., & Mcleod Jr, Raymond. (2007). *Software Testing*. Testing Across The Entire. [Google Scholar](#)
- Horton, John. (2015). *Android Programming For Beginners*. Packt Publishing Ltd. [Google Scholar](#)
- Iriyanto, S. Y., & Zaini, T. M. (2014). *Pengolahan Citra Digital*. Lampung: Anugrah Utama Raharja. [Google Scholar](#)
- Kendal, Simon. (2009). *Object Oriented Programming Using Java*. Bookboon. [Google Scholar](#)
- Kurniawan, Tri A. (2018). Pemodelan Use Case (UML): Evaluasi Terhadap Beberapa Kesalahan Dalam Praktik. *J. Teknol. Inf. Dan Ilmu Komput*, 5(1), 77. [Google Scholar](#)
- Kusumanto, R. D., & Tompunu, Alan Novi. (2011). Pengolahan Citra Digital Untuk Mendeteksi Obyek Menggunakan Pengolahan Warna Model Normalisasi RGB. *Semantik*, 1(1), 83-87. [Google Scholar](#)
- Lee, Wei Meng. (2012). *Beginning Android 4 Application Development*. John Wiley & Sons. [Google Scholar](#)
- Mufadhol, Mufadhol, & Suprayogi, Mohammad Sani. (2014). Penentuan Koordinat Titik Pada Teknologi Garis Dalam Grafika Komputer Dengan Menggunakan Algoritma Line Equation. *Jurnal Transformatika*, 12(1), 24–26. [Google Scholar](#)
- Pressman, Roger S. (2010). A Practitioner’s Approach. *Software Engineering*, 2, 41–42. [Google Scholar](#)
- Profesi, Dwindia Etika, & Henderi, Henderi. (2018). Analisis Dan Perancangan Sistem Informasi Kepegawaian Menggunakan Unified Modeling Language (Uml). *E-Jurnal Jusiti: Jurnal Sistem Informasi Dan Teknologi Informasi*, 7(1), 22–33. [Google Scholar](#)
- Sabale, Rajendra Ganpatrao, & Dani, A. R. (2012). Comparative Study Of Prototype Model For Software Engineering With System Development Life Cycle. *IOSR Journal Of Engineering*, 2(7), 21–24. [Google Scholar](#)

Wilianti Aliman

Sommerville, Ian. (2011). *Software Engineering. America*. Pearson Education, Inc. [Google Scholar](#)

Sukanto, Rosa Ariani, & Shalahuddin, Muhammad. (2013). *Rekayasa Perangkat Lunak Terstruktur Dan Berorientasi Objek*. Bandung: Informatika, 3. [Google Scholar](#)

Wu, C. Thomas. (2006). *An Introduction To Object-Oriented Programming With Java TM*. Mcgraw-Hill Incorporated. [Google Scholar](#)

---

**Copyright holder:**

Wilianti Aliman (2021)

**First publication right:**

Jurnal Syntax Literate

**This article is licensed under:**

