

Analisis dan Praktik Pemodelan Sistem menggunakan *Draw.io* (*Diagrams.net*)

Achmad Maulana^{1*}, Muhammad Fadhil Dwisaputra², Fierren Al-Hilal Saepul Bahri³,
Amar Subagja Firdaus⁴, Nazwa Althafah Athalia⁵,
Muhfiz Zauzi⁶, Danish Alifian Safaraz⁷,
Wawan Setiawan⁸, Ibnu Mas'ud⁹, M. Iman Wahyudi¹⁰

^{1,2,3,4,5,6,7,8,9,10} Program Studi Informatika, UIN Sultan Maulana Hasanuddin Banten

* E-mail: achmadmaulana1504@gmail.com

ABSTRAK

Perkembangan perangkat lunak di era digital saat ini, membuat diagram menjadi salah satu sarana penting untuk menerjemahkan ide, kebutuhan pengguna, dan rancangan teknis ke dalam bentuk yang lebih mudah dipahami. *Draw.io* atau *diagrams.net* merupakan aplikasi yang dapat digunakan untuk membuat berbagai jenis diagram, baik melalui *platform web* maupun aplikasi desktop. Melalui fitur yang disediakan, pengguna dapat menyusun dokumentasi visual secara terstruktur sehingga informasi mengenai sistem dapat disajikan dengan lebih rapi, sistematis, dan mudah dipahami oleh seluruh pihak yang terlibat dalam proses pengembangan. Berdasarkan praktik penyusunan *use case diagram*, *class diagram*, dan *activity diagram*, dapat diketahui bahwa *Draw.io* tidak hanya berperan sebagai alat bantu visualisasi, tetapi juga mendukung proses analisis dan perancangan sistem secara logis. *Use case diagram* dimanfaatkan untuk mengidentifikasi kebutuhan serta interaksi pengguna dengan sistem, *class diagram* digunakan untuk memodelkan struktur objek dan hubungan antar objek, sedangkan *activity diagram* membantu menggambarkan urutan aktivitas yang terjadi dalam suatu proses. Dengan demikian, penggunaan *Draw.io* memberikan kontribusi dalam meningkatkan kualitas perancangan, komunikasi, dan dokumentasi sistem perangkat lunak.

Kata Kunci : *Diagram, Draw.io, UML*

ABSTRACT

The development of software in today's digital era has made diagrams an essential tool for translating ideas, user needs, and technical designs into more easily understood forms. Draw.io, or diagrams.net, is an application that can be used to create various types of diagrams, both through web platforms and desktop applications. Through its features, users can create structured visual documentation so that information about the system can be presented more neatly, systematically, and easily understood by all parties involved in the development process. Based on the practice of creating use case diagrams, class diagrams, and activity diagrams, it can be seen that Draw.io not only acts as a visualization tool but also supports the process of logical system analysis and design. Use case diagrams are used to identify user needs and interactions with the system, class diagrams are used to model object structures and relationships between objects, while activity diagrams help illustrate the sequence of activities that occur in a process. Thus, the use of Draw.io contributes to improving the quality of software system design, communication, and documentation.

Keywords : *Diagram, Draw.io, UML*

PENDAHULUAN

Dalam proses pengembangan perangkat lunak, diagram memiliki peran penting sebagai sarana penghubung antara gagasan, kebutuhan pengguna, dan desain teknis sistem. Penjelasan kebutuhan yang hanya disampaikan dalam bentuk teks sering kali menyulitkan tim pengembang untuk memahami keterkaitan antaraktor, alur proses, data, maupun komponen sistem secara menyeluruh. Karena itu, penggunaan alat pemodelan visual menjadi sangat diperlukan pada tahap analisis dan perancangan. Salah satu tools yang umum digunakan adalah *Draw.io* atau *diagrams.net*. Aplikasi tersebut membantu pengembang membuat berbagai jenis diagram secara lebih mudah dan terstruktur, sehingga komunikasi antaranggota tim menjadi lebih efektif [1]. Dengan bantuan visualisasi, proses memahami kebutuhan sistem, merancang solusi, serta mendokumentasikan rancangan dapat dilakukan dengan lebih jelas dan efisien.

Draw.io atau *diagrams.net* merupakan aplikasi perancangan diagram yang tersedia dalam versi berbasis *web* maupun *desktop*. Aplikasi ini dirancang untuk membantu pengguna membuat berbagai jenis diagram secara visual dan terstruktur, mulai dari *flowchart*, *UML*, *Entity Relationship Diagram (ERD)*, *mind map*, *network diagram*, hingga *C4 model* yang umum digunakan dalam pengembangan sistem dan dokumentasi teknis. Kehadiran *Draw.io* menjadi solusi yang efektif bagi pengembang perangkat lunak, analisis sistem, maupun tim proyek karena mampu menyederhanakan proses penyampaian ide dan rancangan yang kompleks ke dalam bentuk visual yang lebih mudah dipahami. Berdasarkan dokumentasi resminya, *Draw.io* dapat digunakan secara online melalui layanan *app.diagrams.net* maupun secara offline melalui aplikasi *Draw.io Desktop*. Fleksibilitas ini memberikan kemudahan bagi pengguna untuk bekerja di berbagai kondisi, baik ketika terhubung dengan internet maupun saat bekerja secara lokal tanpa koneksi jaringan. Selain itu, aplikasi ini mendukung integrasi dengan berbagai layanan penyimpanan seperti *Google Drive*, *OneDrive*, *Dropbox*, *GitHub*, *GitLab*, *browser storage*, serta penyimpanan lokal pada perangkat komputer. Dukungan tersebut memungkinkan proses kolaborasi, penyimpanan, dan pengelolaan dokumen diagram dilakukan dengan lebih efisien dan aman.

Jurnal ilmiah ini merupakan pengembangan dari pembahasan proyek sebelumnya yang kemudian disesuaikan dengan fokus proyek terbaru, yaitu menganalisis nilai guna serta praktik penggunaan *Draw.io* dalam proses pengembangan perangkat lunak. Pembahasan tidak hanya berfokus pada definisi atau pengenalan aplikasi semata, tetapi juga menguraikan bagaimana *Draw.io* diterapkan secara nyata dalam proses analisis dan perancangan sistem. Penjelasan akan mencakup pembuatan *use case diagram* untuk menggambarkan interaksi pengguna dengan sistem, *class diagram* untuk menunjukkan struktur kelas dan relasi antarobjek [2], serta *activity diagram* untuk memvisualisasikan alur proses dan logika kerja sistem. Dengan memanfaatkan *Draw.io*, proses dokumentasi dapat dilakukan secara lebih sistematis, terorganisasi, dan mudah dipahami oleh seluruh anggota tim pengembang. Diagram yang dihasilkan tidak hanya membantu dalam tahap desain, tetapi juga mempermudah komunikasi antaranggota tim, mengurangi risiko kesalahan interpretasi kebutuhan sistem, serta mendukung proses pemeliharaan dan pengembangan aplikasi di masa mendatang [3]. Dengan demikian, penggunaan *Draw.io* memiliki peran penting dalam meningkatkan kualitas perancangan dan efektivitas proses pengembangan perangkat lunak secara keseluruhan.

METODE

2.1 Draw.IO (Diagrams.Net)

Draw.IO atau *diagrams.net* adalah aplikasi berbasis *JavaScript* yang digunakan untuk membuat diagram dan media *whiteboard* secara visual. Pada repositori resminya di *GitHub jgraph/drawio*, aplikasi ini dijelaskan sebagai platform diagramming yang fleksibel dan dapat dikonfigurasi sesuai kebutuhan pengguna, dengan implementasi utama pada layanan *app.diagrams.net*. Selain itu, dokumentasi resminya menerangkan bahwa *draw.io* dapat dijalankan baik melalui internet maupun secara *offline*, sehingga mendukung aktivitas pembelajaran, penyusunan dokumentasi sistem, serta kolaborasi dalam tim. Dalam bidang analisis dan perancangan perangkat lunak, *draw.io* dimanfaatkan sebagai sarana visualisasi untuk membantu menggambarkan rancangan system [4]. Pengguna dapat membuat berbagai bentuk objek, menghubungkannya menggunakan garis konektor, menambahkan keterangan, serta menyesuaikan tampilan diagram sesuai kebutuhan. Hasil pekerjaan dapat disimpan dalam format *.drawio* maupun diekspor ke bentuk gambar atau dokumen lainnya. Diagram tersebut umumnya digunakan untuk mempresentasikan kebutuhan sistem, desain database, struktur arsitektur aplikasi, alur proses bisnis, hingga dokumentasi akhir suatu proyek pengembangan perangkat lunak [5].

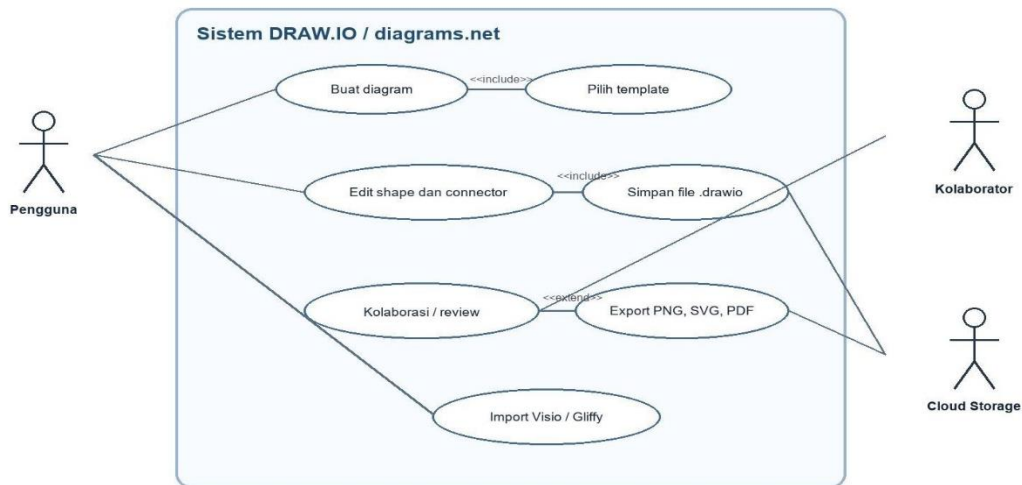
2.2 UML (Unified Modeling Language) dalam Analisis Sistem

Unified Modeling Language atau *UML* merupakan bahasa pemodelan standar yang digunakan untuk menggambarkan cara kerja dan susunan suatu sistem perangkat lunak [6]. Pada *draw.io* (*diagrams.net*), tersedia *library UML* dan *UML 2.5* yang memungkinkan pengguna membuat beragam jenis diagram, seperti *use case*, *class*, *activity*, *sequence*, *component*, *deployment*, hingga *state machine diagram*. *Use case diagram* biasanya digunakan untuk menunjukkan interaksi antara aktor dengan fitur yang disediakan system [7]. Diagram ini membantu pengembang dalam mengidentifikasi kebutuhan pengguna sekaligus menjadi referensi dalam penyusunan dokumentasi sistem. Selanjutnya, *class diagram* dimanfaatkan untuk menampilkan struktur dasar program, meliputi *class*, *atribut*, *method*, serta hubungan antarbagian di dalam sistem. Selain itu, *activity diagram* berfungsi untuk menggambarkan urutan aktivitas atau alur proses kerja secara visual [8]. Dengan adanya diagram tersebut, proses bisnis maupun logika sistem dapat dipahami dengan lebih jelas sebelum tahap implementasi dilakukan.

HASIL DAN PEMBAHASAN

3.1 Analisis dan Praktikum Use Case Diagram

Use case diagram berfungsi untuk menunjukkan berbagai aktivitas atau fitur yang dapat diakses pengguna dalam sebuah sistem. Diagram ini memudahkan pemahaman mengenai hubungan antara pengguna dengan fungsi yang disediakan aplikasi [9]. Pada *draw.io* atau *diagrams.net*, pihak yang terlibat biasanya terdiri dari pengguna yang membuat diagram, rekan kolaborasi yang memberikan saran atau umpan balik, serta media penyimpanan yang menyimpan file diagram yang telah dibuat.



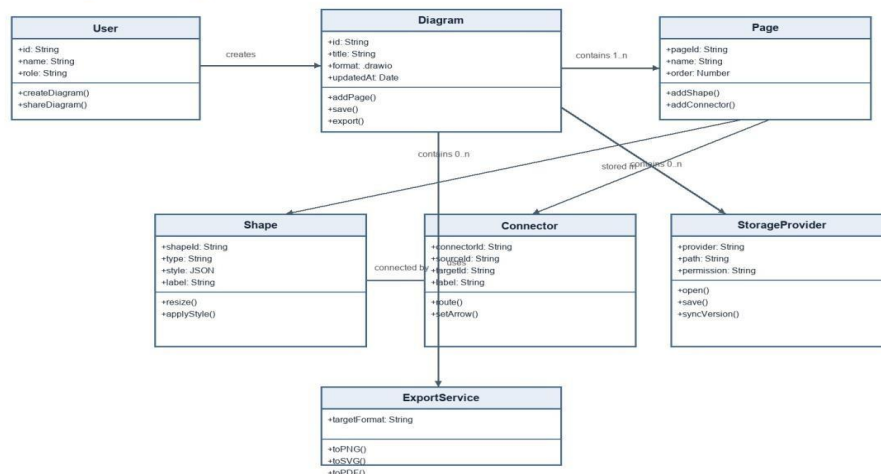
Gambar 1. Use case Diagram konseptual draw.IO/diagrams.net

Gambar 1 menunjukkan interaksi antara *Pengguna*, *Kolaborator*, dan *Cloud Storage* dalam proses pembuatan diagram. Pengguna dapat membuat diagram dengan memilih *template*, kemudian mengedit *shape* dan *connector* sesuai kebutuhan serta menyimpan hasilnya dalam format *.drawio*. Sistem juga mendukung **kolaborasi/review** bersama kolaborator, penyimpanan melalui *Cloud Storage*, serta fitur **export** ke format PNG, SVG, dan PDF untuk kebutuhan dokumentasi atau presentasi. Selain itu, tersedia fitur **import Visio/Gliffy** yang memungkinkan pengguna mengolah diagram dari aplikasi lain. Relasi *include* menunjukkan fungsi yang menjadi bagian dari proses utama, sedangkan *extend* menunjukkan fitur tambahan yang dapat digunakan setelah proses utama dilakukan.

3.2 Analisis dan Praktikum Class Diagram

Class diagram berfungsi untuk menggambarkan model konseptual aplikasi *diagramming*. *Class* utama adalah *User*, *Diagram*, *Page*, *Shape*, *Connector*, *StorageProvider*, dan *ExportService* [10]. Model ini menunjukkan bahwa sebuah diagram dapat memiliki banyak halaman, setiap halaman dapat berisi banyak *shape* dan *connector*, serta proses penyimpanan dan ekspor dilakukan melalui komponen pendukung.

Class Diagram Konseptual DRAW.IO



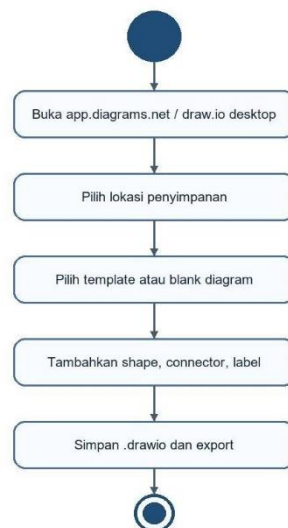
Gambar 2. *Class Diagram* konseptual *draw.IO/diagrams.net*

Class Diagram membantu pengembang memperkirakan struktur data sebelum membuat kode. Misalnya, objek *shape* perlu menyimpan tipe bentuk, *style* dan label. *Connector* perlu menyimpan *sourceId* dan *targetId* agar hubungan antar *shape* dapat digambar ulang saat file dibuka kembali. *StorageProvider* memisahkan logika penyimpanan sehingga diagram dapat disimpan di lokal, *cloud* atau *repository*.

3.3 Analisis dan Praktikum *Activity Diagram*

Activity diagram menggambarkan urutan langkah aktivitas yang dilakukan oleh pengguna. Pada praktik ini, proses dimulai dengan membuka editor, kemudian memilih lokasi penyimpanan [11]. Setelah itu, pengguna dapat menentukan apakah akan menggunakan template atau kanvas kosong. Selanjutnya, pengguna menambahkan elemen-elemen pada diagram, lalu menyimpan hasil pekerjaan dan mengekspornya.

Activity Diagram: Membuat dan Mengekspor Diagram



Gambar 3. Activity Diagram Alur membuat dan mengekspor Diagram

Alur tersebut memperlihatkan bahwa draw.io dapat digunakan sebagai media dokumentasi yang sederhana namun tetap terstruktur. Pengguna tidak perlu langsung membuat diagram yang rumit; proses dapat diawali dari penggunaan template, kemudian dilanjutkan dengan penyusunan elemen-elemen dasar, dan akhirnya ditinjau bersama sebelum hasil akhirnya diekspor.

PENUTUP

4.1 Kesimpulan

Draw.IO atau *diagrams.net* adalah aplikasi pembuatan diagram yang sangat berguna untuk analisis dan perancangan perangkat lunak. Aplikasi ini mendukung berbagai jenis diagram, dapat digunakan baik secara daring maupun luring, menyediakan banyak opsi integrasi penyimpanan, serta memberikan kontrol yang

lebih besar terhadap lokasi penyimpanan data diagram. Dalam pengembangan perangkat lunak, *Draw.IO* membantu tim dalam menjelaskan kebutuhan sistem, menyusun rancangan teknis, serta menghasilkan dokumentasi yang mudah diperbarui. Praktik *use case diagram*, *class diagram* dan *activity diagram* dalam makalah ini menunjukkan bahwa aplikasi tersebut tidak hanya berfungsi sebagai alat menggambar, tetapi juga sebagai sarana untuk berpikir secara sistematis. *Use case diagram* berperan dalam memahami kebutuhan pengguna, *class diagram* membantu dalam mengorganisasi objek beserta relasinya, sedangkan *activity diagram* digunakan untuk memvalidasi alur proses kerja sistem.

4.2 Saran

Tim proyek disarankan untuk menyimpan berkas sumber *Draw.IO* (.drawio) agar diagram dapat dengan mudah diperbarui ketika terjadi perubahan pada sistem. Pada proyek berskala besar, penggunaan draw.io perlu didukung dengan standar dokumentasi, sistem penamaan file yang konsisten, serta proses peninjauan (*review*) agar diagram tetap rapi dan seragam.

DAFTAR PUSTAKA

- [1] N. Pradita, W. Setyawan, R. P. Setiawan, H. L. Kheishi, and U. P. Bangsa, "Perancangan Sistem Informasi Pos Berbasis Mobile pada Toko Plered Indah," vol. 2, no. 1, 2023.
- [2] C. A. Binangkit, A. Voutama, N. Heryana, F. I. Komputer, U. S. Karawang, and A. Musik, "PEMANFAATAN UML (UNIFIED MODELING LANGUAGE) DALAM PERENCANAAN SISTEM PENGELOLAAN SEWA ALAT MUSIK BERBASIS," vol. 7, no. 2, pp. 1429–1436, 2023.
- [3] A. Aurellia, S. Nooriansyah, and Y. Amrozi, "PEMANFAATAN UML DALAM PERANCANGAN SISTEM INFORMASI PRODUK KREATIF DAUR ULANG SAMPAH BERBASIS WEB," *JITET*, vol. 13, no. 3, 2024.
- [4] L. Halifah and J. Subrata, "PERANCANGAN SISTEM INFORMASI PEMINJAMAN DAN PENGEMBALIAN BUKU BERBASIS WEBSITE DI SMK AL-AMIRIYAH LEBAKSIU," vol. 9, no. 5, pp. 7423–7428, 2025.
- [5] R. Oktafiandi *et al.*, "Volume 1 , no . 6 April 2024 ISSN 3025-0889 (media online) PENGENALAN KONSEP PERANCANGAN PERANGKAT LUNAK MENGGUNAKAN DRAW . IO BAGI SISWA SMK SEBAGAI APPA: Jurnal Pengabdian kepada Masyarakat Volume 1 , no . 6 April 2024 ISSN 3025-0889 (media online) Hal 410-416," vol. 1, no. 6, pp. 410–416, 2024.
- [6] S. W. Ramdany, S. A. Kaidar, B. Aguchino, C. Amelia, and A. Putri, "Penerapan UML Class Diagram dalam Perancangan Sistem Informasi Perpustakaan Berbasis Web," *JIES*, vol. 5, no. 1, 2024.
- [7] N. Yunita and R. Rosmawati, "Sistem Informasi Rental Mobil Berbasis Web Pada PT Karya Mobil," *Simpatik J. Sist. Inf. dan Inform.*, vol. 1, no. 1, pp. 53–62, 2021, doi: 10.31294/simpatik.v1i1.410.
- [8] J. Manggara, "RANCANG BANGUN APLIKASI LAYANAN PENGAJUAN SURAT TUGAS DOSEN BERBASIS WEB PADA LP2M," *JITET*, vol. 13, no. 2, 2025.
- [9] S. Narulita, A. Nugroho, and M. Z. Abdillah, "Diagram Unified Modelling Language (UML) untuk Perancangan Sistem Informasi Manajemen Penelitian

- dan Pengabdian Masyarakat (SIMLITABMAS) Universitas Nasional Karangturi Semarang , Indonesia (deskripsi) dan perancangan sistem , khususnya pada pemrograman berorientasi objek (Nistrina,” no. 3, pp. 244–256, 2024.
- [10] T. Masyarakat *et al.*, “Pelatihan Pembuatan UML (Unified Modelling Language) Menggunakan Aplikasi Draw . io Pada Prodi Sistem Informasi Universitas Muhammadiyah Jambi” vol. 1, no. 2, 2024.
- [11] W. Setiawan and N. Fajriyah, “Analisis dan Perancangan Sistem Informasi Pengelolaan Retain Sample QC pada PT. XYZ,” *J. Teknol. Terpadu*, vol. 7, no. 1, pp. 15–22, 2021.