JURNAL TEKNOLOGI INFORMASI
Jurnal Keilmuan dan Aplikasi Bidang Teknik Informatika

[E-ISSN 2656-0321]
[Vol 15 No 2]
[Agustus 2021]

BENCHMARKING METODE RANCANG BANGUN WATERFALL DAN PEMODELAN BERBASIS OBJEK

Moch. Khusien Bagaskoro [a,1], M. Amirul Chakim [b,2], Mohammad Nurul Hilal [c,3], One Thowimma[d,4]
[a] Universitas Islam Negeri Sunan Ampel, Surabaya
[b]Universitas Islam Negeri Sunan Ampel, Surabaya
[c]Universitas Islam Negeri Sunan Ampel, Surabaya
[d]Universitas Islam Negeri Sunan Ampel, Surabaya
[1] 09020620032@student.uinsby.ac.id; [2] 09020620030@student.uinsby.ac.id; [3] 09020620033@student.uinsby.ac.id;
[4]H76219031@student.uinsby.ac.id

ARTICLE INFO

Keywords
Waterfall Method
Unified Modeling Language
Object Oriented Modelling
Benchmarking

ABSTRACT

This research is a research Benchmarking Waterfall Design Method and Object-Based Modeling. Software development has many models, software methodologies have evolved from time to time, including the Waterfall approach and the UML (Unified Modeling Language) approach. Waterfall is a software development model that is carried out sequentially and very systematically. Meanwhile, UML is a visual modeling method that serves as a means of designing object-oriented systems. The difference between the two models is that Waterfall is a structured method, while UML is a system analysis method, a representation of the programming method. This study aims to perform a comparison or benchmarking of each of these software models, which can assist in determining the most suitable model for the current software development process. The method used in this research is descriptive quantitative method, which analyzes data by describing or describing the data that has been collected as it is. From the results of this study, it can be concluded that the UML method is most suitable for software development, because UML in terms of development, is a method that has undergone evolution from previous methods, so this method can follow the use of users in designing projects on a medium to large scale. That are relevant to the current developments.

1. Pendahuluan

1.1. Latar Belakang

Dalam era globalisasi, perkembangan teknologi informasi di Indonesia berjalan cukup pesat. Globalisasi diartikan sebagai proses menyatunya dunia yang meliputi berbagai bidang tata kehidupan dunia mengandung karakteristik adanya perubahan keterbukaan, kreativitas, kecanggihan, kecepatan, keterikatan, keunggulan, kekuatan, dan kompetisi bebas (Turban, 2005). Hal itu berdampak pula pada perkembangan yang sedang masif terjadi pada bidang industri teknologi dan informasi saat ini.

Industri TI telah menjadi salah satu industri terkemuka di dunia, berkontribusi dengan perangkat dan program perangkat lunak inovatif, yang mendukung semua hidang aktivitas saat ini, seperti kedokteran, bisnis, pendidikan, dan, jejaring sosial. Pasar industri TI melebihi $ 4,5 triliun pada tahun 2017, Amerika Serikat menjadi pasar teknologi terbesar di dunia dengan pangsa 31% atau $ 1,5 triliun pada tahun 2018 saja (Comptia, 2018). Seiring dengan perkembangan yang terjadi, kebutuhan akan inovasi yang lebih baru juga menjadi lebih dinamis, dimana kebutuhan pelanggan pun berubah-ubah lebih cepat dari pada sebelumnya.

Dengan demikian siklus hidup TI semakin pendek, produk baru atau versi baru diperlukan untuk memenuhi kebutuhan pelanggan yang kian hari kian meningkat. Hal ini lah yang mengarahkan pada pengembangan lebih pada sektor model pengembangan perangkat lunak atau software. Banyak model pengembangan perangkat lunak yang ada, metodologi pengembangan perangkat lunak pun berevolusi dari masa ke masa, mulai pendekatan waterfall yang sangat kaku hingga pada pendekatan UML.

Unified Modelling Language (UML) adalah suatu alat untuk memvisualisasikan dan mendokumentasikan hasil analisa dan desain yang berisi sintak dalam memodelkan sistem secara visual (Braun, et. al. 2001). Juga merupakan satu kumpulan konvensi pemodelan yang digunakan untuk

https://doi.org/10.47111/JTI

JURNAL TEKNOLOGI INFORMASI
Jurnal Keilmuan dan Aplikasi Bidang Teknik Informatika
[E-ISSN 2656-0321]
[Vol 15 No 2]
[Agustus 2021]

menentukan atau menggambarkan sebuah sistem software yang terkait dengan objek (Whitten, et. al. 2004).

Model Waterfall adalah salah satu model System Development Life Cycle atau classic life cycle. Model ini menggunakan pendekatan sistematis dan urut mulai dari level kebutuhan sistem lalu menuju ke tahapan analysis, design, coding, testing / verification and maintenance.

Baik Model Waterfall dan pemodelan berbasis objek yang dalam hal ini adalah UML, keduanya memiliki perbedaan yang cukup signifikan sehingga dalam pengembangan sistem perlu diketahui kasus atau sistem yang memiliki karakteristik seperti apa yang cocok menggunakan masing-masing pemodelan tersebut.

Berdasarkan permasalahan tersebut maka tujuan dari penulisan jurnal ini adalah melakukan perbandingan atau benchmarking dari masing-masing model perangkat lunak tersebut yang dapat membantu dalam menentukan model yang paling sesuai untuk proses pengembangan perangkat lunak pada saat ini.

1.2. Tinjauan Teori

Benchmarking menurut Gregory H. Watson adalah sebagai pencarian secara berkesinambungan dan penerapan secara nyata praktik-praktik yang lebih baik yang mengarah pada kinerja kompetitif unggul. Sedangkan menurut David Kearns adalah suatu proses pengukuran terus-menerus atas produk, jasa dan tata cara kita terhadap pesaing kita yang terkuat atau badan usaha lain yang dikenal sebagai yang terbaik

Menurut (Pressman, 2001), model waterfall adalah model klasik yang bersifat sistematis, berurutan dan membangun software. Nama model ini sebenarnya adalah “Linear Sequential Model”. Model ini sering disebut juga dengan “classic life cycle” atau metode waterfall. Model ini termasuk model generic dimana pengaplikasiannya dilakukan pada rekayasa perangkat lunak dan pertama kali diperkenalkan oleh Winston Royce sekitar tahun 1970 sehingga sering dianggap kuno, tetapi merupakan model yang paling banyak digunakan dalam hal Software Engineering bahkan hingga saat ini. Model ini melakukan pendekatan secara sistematis dan berurutan. Model ini disebut waterfall karena tahap demi tahap yang dilalui harus menunggu selesainya tahap yang dilakukan sebelumnya dan berjalan berurutan.

Menurut (Whitten & Bentley, 2007), Unified Modeling Language (UML) versi 2.0 adalah sekumpulan konversi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem software yang terkait dengan objek.

Menurut (Whitten & Bentley, 2007), UML menyediakan tiga belas macam diagram untuk memodelkan aplikasi berorientasi objek, yaitu:

1. *Use Case Diagram* menggambarkan interaksi antara sistem internal, sistem eksternal, dan user. Dengan kata lain, secara grafik menjelaskan siapa yang akan menggunakan sistem, dan dengan cara apa user akan berinteraksi dengan sistem.
2. *Activity Diagram* menggambarkan alur sequential dari aktivitas sebuah proses bisnis atau Use Case. Bahkan bisa juga dipergunakan untuk memodelkan logika yang digunakan pada sistem.
3. *Class Diagram* menggambarkan struktur objek sistem. Menunjukkan kelas yang menjadi komponen dari sistem, serta hubungan antar kelas tersebut.
4. *Object Diagram* serupa dengan Class Diagram, memodelkan instansi objek yang sebenarnya beserta nilai atributnya.
5. *State Machine Diagram* untuk memodelkan perilaku objek di dalam sistem terhadap kejadian (event) selama masa hidupnya.
6. *Composite Structure Diagram* menguraikan struktur internal, komponen, atau Use Case dari suatu kelas tertentu.
7. *Sequence Diagram* Menggambarkan bagaimana objek berinteraksi melalui pengiriman pesan (message) dalam pengeksekusian sebuah Use Case, atau operasi tertentu.
8. *Communication Diagram* disebut juga dengan Collaboration Diagram, mirip dengan Sequence Diagram. Namun, Sequence Diagram lebih berfokus pada penilaian waktu dan urutan pesan.

https://doi.org/10.47111/JTI

JURNAL TEKNOLOGI INFORMASI
Jurnal Keilmuan dan Aplikasi Bidang Teknik Informatika
[E-ISSN 2656-0321]
[Vol 15 No 2]
[Agustus 2021]

Sedangkan Communication Diagram berfokus pada penyusunan struktur objek dalam bentuk jaringan

9. *Interaction Overview* Diagram mengombinasikan Activity Diagram dengan Sequence Diagram untuk menunjukkan bagaimana objek berinteraksi dalam tiap aktivitas Use Case.
10. *Timing Diagram* adalah diagram interaksi lain yang berfokus pada batasan penilaian waktu dalam keadaan satu objek atau kumpulan objek yang berubah. Diagram ini sangat berguna ketika mendesain embedded software untuk banyak perangkat.
11. *Component Diagram* menggambarkan penyusunan kode programming yang terbagi menjadi beberapa komponen dan menjelaskan bagaimana komponen tersebut berinteraksi.
12. Deployment Diagram menggambarkan konfigurasi dari komponen software dalam arsitektur fisik dari "simpul-simpul" sistem hardware.
13. Package Diagram menggambarkan bagaimana kelas atau konstruksi dari UML lain disusun dalam bentuk paket (berkaitan dengan paket Java atau C++ dan namespaces dari .Net) dan ketergantungannya antar paket.

2. Metodologi Penelitian

Pendekatan yang digunakan pada penelitian ini adalah pendekatan kuantitatif dengan menggunakan metode deskriptif, yaitu melakukan studi komparatif untuk membandingkan fenomena-fenomena yang ditemukan dan membuat klasifikasi yang bersumber pada suatu standar. Adapun langkah-langkah penelitian meliputi : memilih dan merumuskan masalah, menelusuri sumber-sumber kepustakaan, melakukan analisis data dari beberapa sumber, membuat perbandingan model yang diinginkan sesuai dengan karakteristik model pengembangan dan sistem yang sedang berjalan.

3. Hasil dan Pembahasan

Artikel ini hendak membandingkan Metode Waterfall dengan Metode Berbasis Objek dalam konteks rancang bangun sistem informasi. Pada Metode tersebut terdapat alur pentahapan yang berbeda.

Menurut (Rusdiana & Irfan, 2014), tahap-tahap dalam Model Waterfall, yaitu sebagai berikut:

a) Requirement Analysis, yaitu kebutuhan software harus bisa didapatkan dalam fase ini, termasuk didalamnya kegunaan software yang diharapkan pengguna dan batasan software. Informasi ini biasanya dapat diperoleh melalui wawancara, survei atau diskusi. Informasi tersebut dianalisis untuk mendapatkan dokumentasi kebutuhan pengguna untuk digunakan pada tahap selanjutnya.
b) System Design, tahap ini dilakukan sebelum melakukan coding. Tahap ini bertujuan untuk memberikan gambaran yang seharusnya dikerjakan dan tampilannya. Tahap ini membantu dalam menspesifikasikan kebutuhan hardware dan sistem serta mendefinisikan arsitektur secara sistem secara keseluruhan.
c) Implementation, pada tahap ini dilakukan pemrograman, pembuatan software dipecah menjadi modul-modul kecil yang akan digabungkan dalam tahap berikutnya. Selain itu, dalam tahap ini juga dilakukan pemeriksaan terhadap modul yang dibuat, apakah telah memenuhi fungsi yang diinginkan atau belum.
d) Integration & Testing, pada tahap ini dilakukan penggabungan modul-modul yang telah dibuat dan dilakukan. Pengujian ini dilakukan untuk mengetahui apakah software yang dibuat telah sesuai dengan desainnya dan masih terdapat kesalahan atau tidak.
e) Operating & Maintenance, tahap ini merupakan tahap terakhir dalam model waterfall. Dimana software yang sudah jadi dijalankan serta dilakukan pemeliharaan. Pemeliharaan termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya. Perbaikan implementasi unit sistem dan peningkatan jasa sistem sebagai kebutuhan baru.

https://doi.org/10.47111/JTI

JURNAL TEKNOLOGI INFORMASI
Jurnal Keilmuan dan Aplikasi Bidang Teknik Informatika
[E-ISSN 2656-0321]
[Vol 15 No 2]
[Agustus 2021]

Menurut (Rosenberg & Stephens, 2007) terdapat beberapa analisa dan perancangan sistem dengan menggunakan model UML yang dirangkum dari buku Use Case Driven Object Modelling With UML: Theory and Practice sebagai berikut:

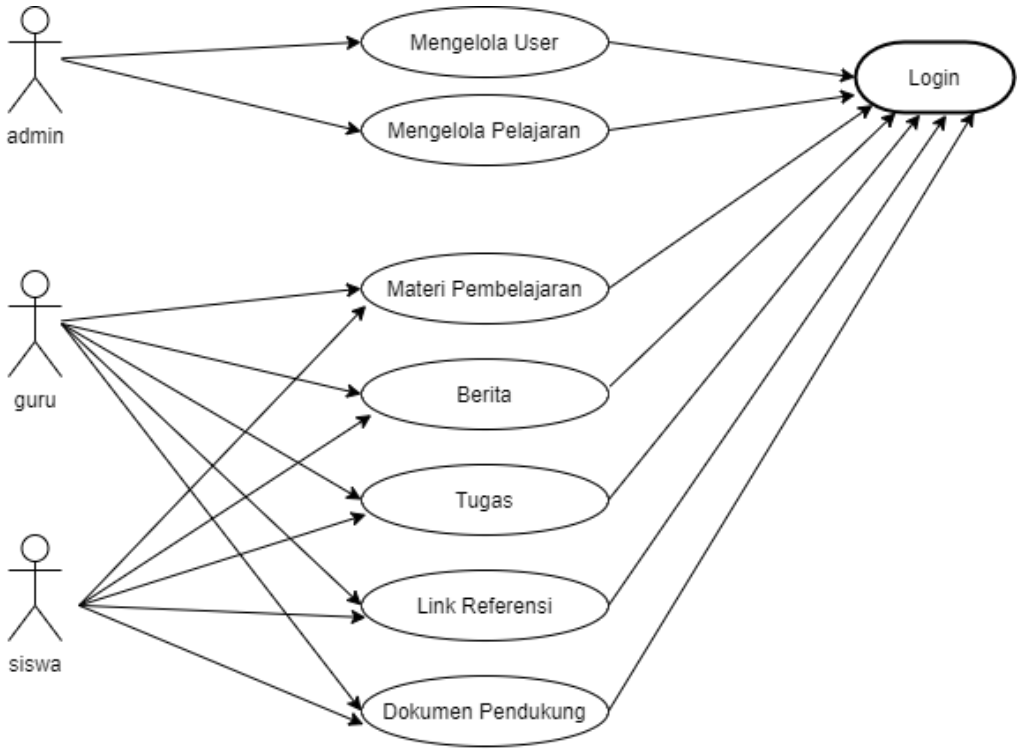
a) Functional Requirement, persyaratan fungsional dapat berupa dokumen penting yang dalam pembuatannya melibatkan business analyst, customer, end user, and project stakeholders. Persyaratan Fungsional ini bersifat tidak terstruktur dan tidak dapat digunakan dalam perancangan secara langsung.
b) Domain Modeling, permodelan domain memiliki tugas membangun glosarium proyek, atau kamus istilah yang digunakan dalam proyek yang tengah kita kerjakan (misalnya, proyek toko buku Internet akan menyertakan objek domain seperti Buku, Pelanggan, Pesanan, dan Item Pesanan). Tujuannya adalah untuk memastikan setiap orang dalam proyek dapat memahami masalah dalam hal istilah sehingga tidak ambigu. Model Domain juga menyediakan kosakata umum untuk memungkinkan komunikasi yang jelas diantara anggota tim.
c) Use Case Modeling, Use Case sendiri menggambarkan bagaimana cara pengguna akan berinteraksi dengan sistem dan sebaliknya pula, bagaimana sistem akan merespon. Kalimat yang digunakan dalam use case harus berupa kalimat aktif, misalnya “user men-klik tombol login”. Use Case harus mengandung nama pada model domain. Dengan demikian kita dapat menghubungkan class-class yang akan dirancang dengan use case.
d) Requirement Review, hal ini dilakukan untuk memastikan bahawa use case dan domain model dibuat dengan baik. Customer perlu dilibatkan dalam hal ini, agar mereka dapat memastikan use case (behavioral requirement) dan functional requirement berjalan dengan lancar sesuai dengan yang diharapkan.
e) Robustness Analysis, untuk beralih dari use case ke design bahkan ke proses coding, kita perlu menautkan use case itu ke objek. Nah dalam hal ini Robustness Analysis digunakan untuk menjembatani kesenjangan yang terjadi pada proses analisis dan perancangan.
f) Sequence Diagram, sequence diagram dibuat untuk setiap use case yang ada, berdasarkan hasil dari tahap sebelumnya yaitu Robustness Analysis. Dalam desain yang berorientasi pada objek, sebagian besar membangun hak sistem berkaitan dengan menemukan alokasi fungsi yang optimal ke kelas dimana dalam hal ini yang dimaksud adalah alokasi perilaku. Intinya adalah menggambar panah pesan pada diagram urutan yang memungkinkan alat permodelan untuk secara otomatis menetapkan operasi ke kelas objek target yang menerima pesan runtime.
g) Critical Design Review, critical design review atau biasa disingkat CDR dilakukan untuk memastikan bahwa tidak ada yang kurang dari tahap sequence model. Dan tak lupa pula untuk memastikan bahwa setiap kelas yang ada telah memiliki atribut dan operasi yang didefinisikan secara lengkap seperti, memiliki nama, tipe, data parameter dan lain sebagainya.
h) Coding, pada tahap ini developer berperan penting dalam mengubah rancangan yang awalnya berbentuk design kemudian diubah kedalam bahasa atau kode program. Karena semua telah direncanakan, maka proses pengkodingan dapat dianggap sebagai pembuktian dari sebuah rancangan tersebut.

Berikut ini merupakan contoh dari masing-masing metode dalam penerapannya pada perancangan apliaksi E-learning. Dimulai dari penerapan metode UML terlebih dahulu:

https://doi.org/10.47111/JTI

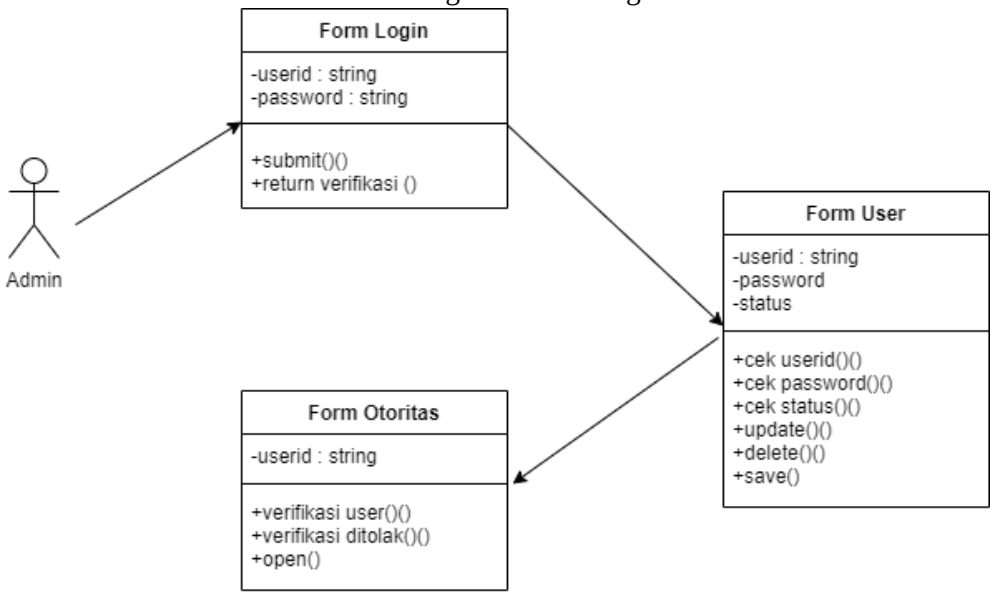
JURNAL TEKNOLOGI INFORMASI
Jurnal Keilmuan dan Aplikasi Bidang Teknik Informatika
[E-ISSN 2656-0321]
[Vol 15 No 2]
[Agustus 2021]

Diagram Use Case Untuk Sistem



Gambar 1. Diagram Use Case Untuk Sistem (Rinaldi, 2019)

Class Diagram Untuk Login

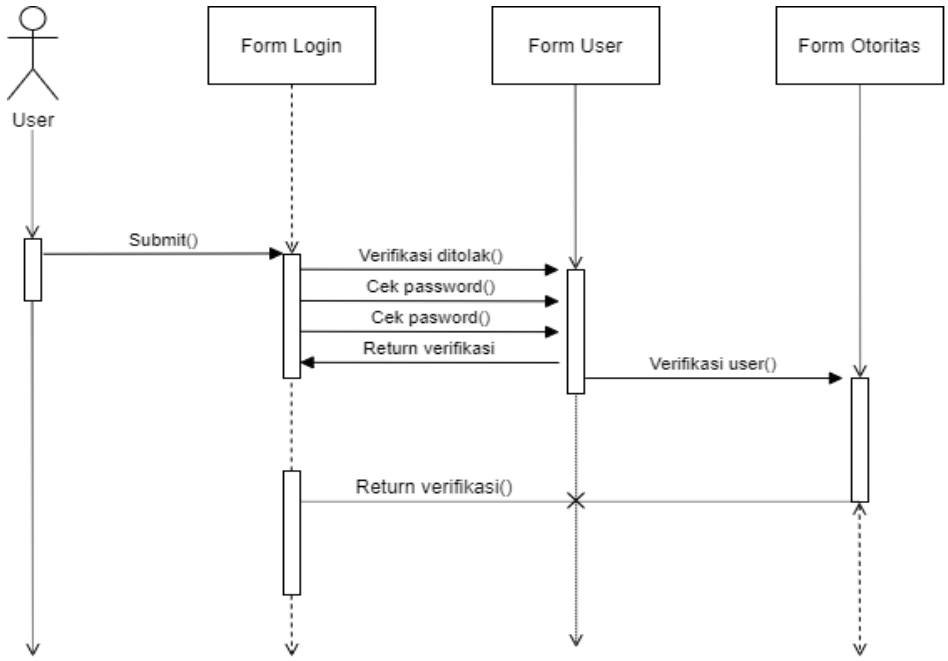


Gambar 2. Class Diagram Untuk Login (Rinaldi, 2019)

https://doi.org/10.47111/JTI

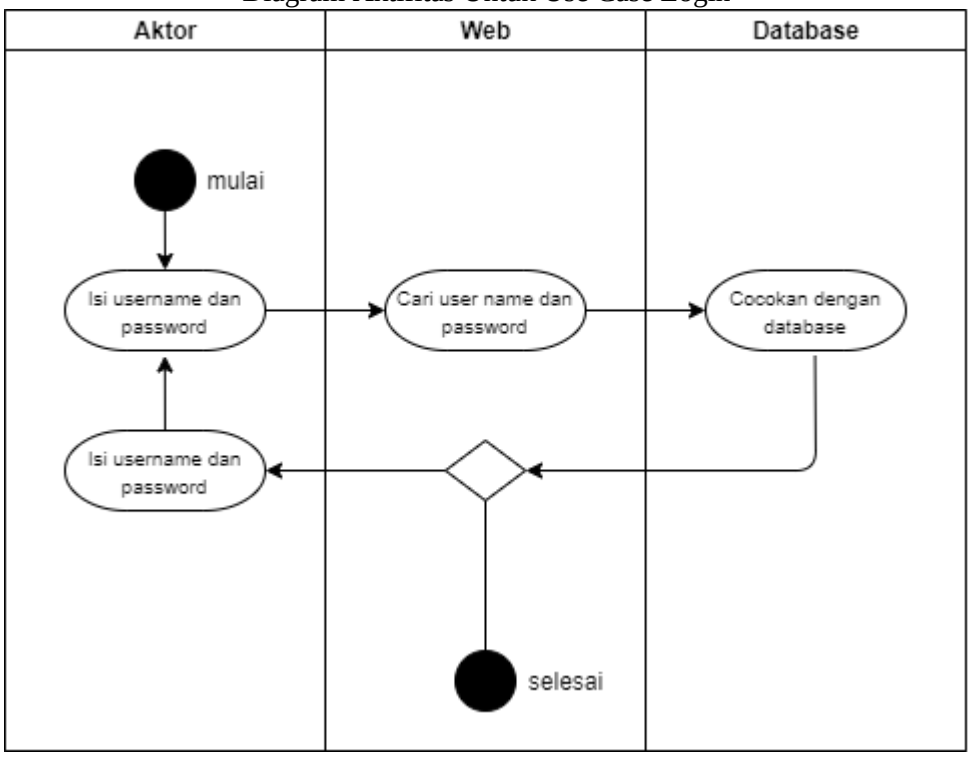
JURNAL TEKNOLOGI INFORMASI
Jurnal Keilmuan dan Aplikasi Bidang Teknik Informatika
[E-ISSN 2656-0321]
[Vol 15 No 2]
[Agustus 2021]

Diagram Sequence Untuk Login



Gambar 3. Diagram Sequence Untuk Login (Rinaldi, 2019)

Diagram Aktifitas Untuk Use Case Login



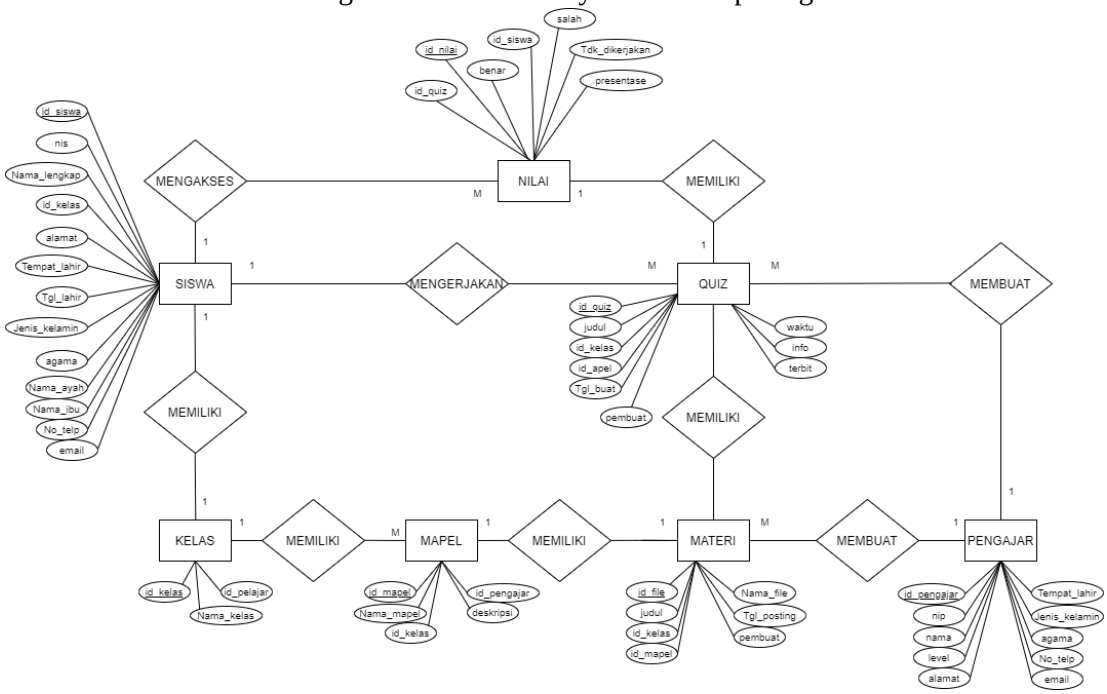
Gambar 4. Diagram Aktifitas Untuk Use Case Login (Rinaldi, 2019)

https://doi.org/10.47111/JTI

JURNAL TEKNOLOGI INFORMASI
Jurnal Keilmuan dan Aplikasi Bidang Teknik Informatika
[E-ISSN 2656-0321]
[Vol 15 No 2]
[Agustus 2021]

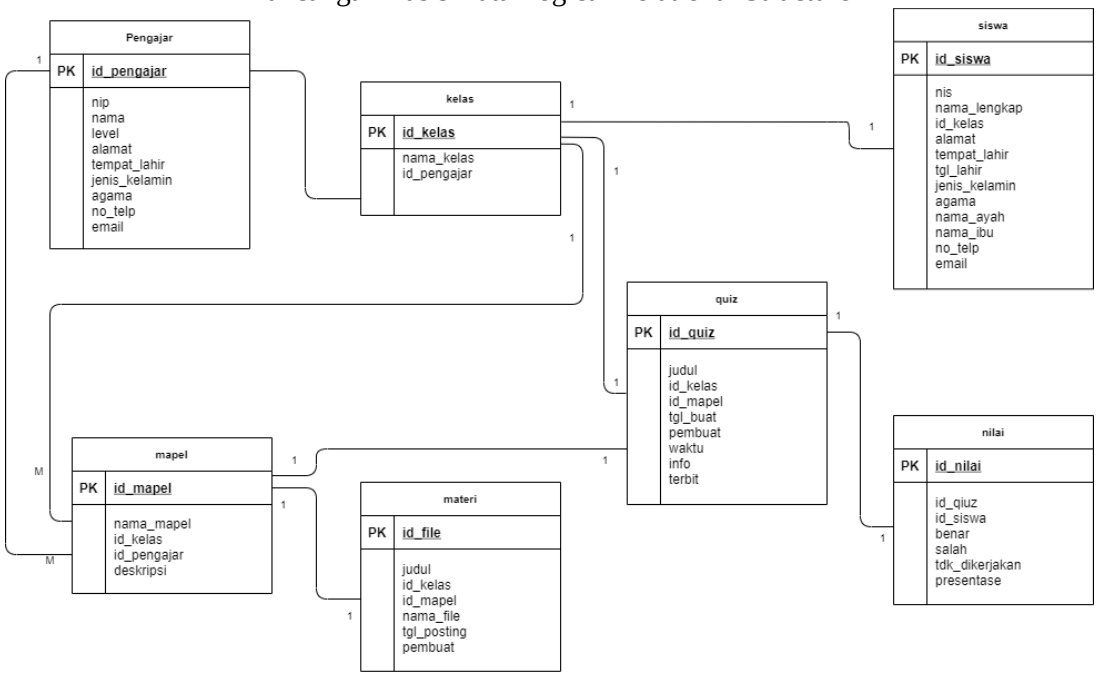
Sementara, penerapan metode Waterfall pada perancangan aplikasi E-Learning adalah sebagai berikut: tahap pertama yaitu melakukan beberapa analisa yang mencakup baik analisa kebutuhan siswa, analisa kebutuhan guru, hingga analisa kebutuhan admin. Setelah semua analisa tersebut didapatkan maka proses berlanjut pada tahap selanjutnya.

Rancangan Basis Data Entity Relationship Diagram



Gambar 5. Entity Relationship Diagram (Sastra, 2017)

Rancangan Basis Data Logical Relational Structure

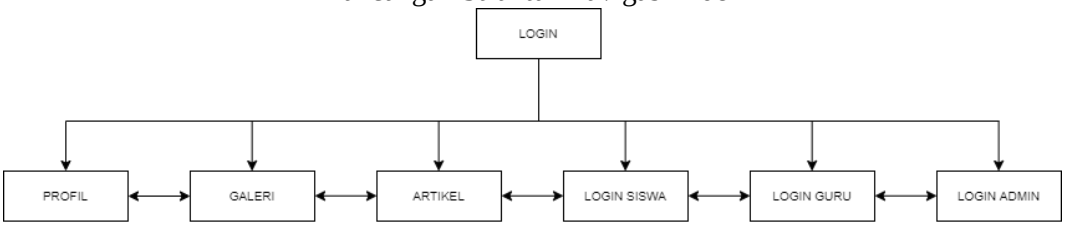


Gambar 6. Logical Relationship Structure (Sastra, 2017)

https://doi.org/10.47111/JTI

JURNAL TEKNOLOGI INFORMASI
Jurnal Keilmuan dan Aplikasi Bidang Teknik Informatika
[E-ISSN 2656-0321]
[Vol 15 No 2]
[Agustus 2021]

Rancangan Struktur Navigasi Index



Gambar 7. Struktur Navigasi Index (Sastra, 2017)

4. Kesimpulan

Menurut kami, penggunaan metode UML ataupun Waterfall dalam segi perancangan suatu software baik dalam bentuk aplikasi atau web sudah cukup baik. Akan tetapi, mungkin ada yang harus dipertimbangkan mengenai penggunaan kedua metode tersebut. Terlebih metode Waterfall, yang dimana terdapat satu hal yang harus di pertimbangkan yaitu dari segi metode sequentalnya. Sequental disini dimaksudkan pada proses yang berurutan dalam pengerjaan suatu aplikasi atau web, mungkin untuk sekarang itu masih dapat relevan dengan zamannya.

Namun, seiring dengan perkembangan software, model ini belum tentu bisa mengikutinya, dan akhirnya yang menjadi kelebihan lama kelamaan akan menjadi kelemahan dari model ini. Oleh karena itu, meskipun secara tahapan masih menggunakan standard tahapan metode waterfall, metode ini cocok untuk pengerjaan suatu project dengan skalanya yang sedang dan mengarah kecil. Sedangkan metode UML dilihat dari segi pengembangannya, merupakan suatu metode yang sudah mengalami evolusioner dari metode-metode sebelumnya, sehingga metode ini dapat mengikuti penggunaan user dalam merancang project dalam skala sedang ke besar yang relevan dengan perkembangan zaman saat ini.

Daftar Pustaka

[1] Comptia. (2018). *IT industry outlook.* https://www.comptia.org/resources/it-industry-trends-analysis. Kaur, R., & Sengupta, J. (2011). Software Process Models and Analysis on Failure of Software Development Projects. *International Journal of Scientific & Engineering Research*, *2*(2), 1-4.

[2] Braun D., Sivils J., Shapiro A., Versteegh J. (2001). Object Oriented Analysis and Design Team. Kennesaw State University CSIS 4650 - Spring 2001

[3] Jeffrey L. Whitten, Lonnie D. Bentley, and Kevin C. Dittman. (2000). Systems Analysis and Design Methods 5e (5th. ed.). McGraw-Hill Higher Education.

[4] Haviluddin. (2011). Memahami Penggunaan UML (Unified Modelling Language). *Memahami Penggunaan UML (Unified Modelling Language)*, *6*(1), 1–15. Retrieved from https://informatikamulawarman.files.wordpress.com/2011/10/01-jurnal-informatika-mulawarman-feb-2011.pdf

[5] Sidharta, L. (1995). Pustaka Setia Pustaka Setia. *Sistem Informasi Manajemen*, 1–387.

[6] Dima, A. M., & Maassen, M. A. (2018). From waterfall to agile software: Development models in the IT sector, 2006 to 2018. impacts on company management. *Journal of International Studies*, *11*(2), 315–326. https://doi.org/10.14254/2071-8330.2018/11-2/21

https://doi.org/10.47111/JTI

[7] Surya Madaan. (2015). Five Models of Software Development Engineering. *International Journal of Scientific & Engineering Research, Volume 6, Issue 11, 6*(11), 1331–1334.

[8] Eason, O. K. (2016). Information Systems Development Methodologies Transitions: An Analysis of Waterfall to Agile Methodology. *University of New Hampshire*, 1–23. Retrieved from https://scholars.unh.edu/honors/286%0Ahttp://scholars.unh.edu/honors%0Ahttp://scholars.unh.edu/honors

[9] Pressman, R. S. (2001). *Sofware Engineering: a Practitioner's Approach-6th ed*. (B. Jones, Ed.) (5 th). Americas, New York: McGraw-Hill. Retrieved from http://www.mhhe.com

[10] Rinaldi, R. (2019). Penerapan Unifield Modelling Language (UML) Dalam Analisis Dan Perancangan Aplikasi E-Learning. *Jurnal SIMTIKA*, *2*(1), 43–50.

[11] Sastra, R. (2017). Metode Pengembangan Perangkat Lunak Waterfall Dalam Perancangan Sistem Informasi E-Learning . IJSE - Indonesian Journal on Software Engineering, 3(1), 27–31. ijse.bsi.ac.id.

[12] Susanto, R., & Andriana, A. D. (2016). Perbandingan Model Waterfall Dan Prototyping Untuk Pengembangan Sistem Informasi. *Majalah Ilmiah UNIKOM*, *14*(1). https://doi.org/10.34010/miu.v14i1.174

https://doi.org/10.47111/JTI