NAMA : MOCHAMMAD RAKANDIYA SHAFWAN GARNIWA

NIM : 2203956

RPL 3-B

PERANCANGAN DAN PEMODELAN PERANGKAT LUNAK

Structure Diagrams

Diagram	Fungsi	Element
Class Diagram	-Menampilkan struktur statis pengklasifikasi pada system -Diagram memberikan notasi dasar untuk diagram struktur lain yang ditentukan UML -Business Analysts dapat menggunakan Class Diagram untuk membuat system model dari perspektif bisnis	Komponen: Classes, Objects, packages, Signals, enumerations, data types, artifacts, Interface Simbol: Class, Relation, Generalization, Nary association, Collaboration, Realization, Dependency, Association
Object Diagram	- Menggambarkan objek tertentu yang saling berhubungan - Pemodelan struktur data yang kompleks - Verifikasi Class diagram untuk kelengkapan dan akurasi - Memodelkan desain statis atau struktur dari suatu system - Memahami fungsionalitas yang system tawarkan kepada pengguna	Object, Attributes, Class, Link, Dependency Relationships, Association, Aggregation, Composition
Package Diagram	- Mengelompokkan elemen-elemen seperti Class Diagram atau Use case kedalam kelompok atau paket yang seusai. - Memberikan informasi tentang elemen dalam system diorganisasikan atau dikelompokkan - Dapat diterapkan pada berbagai jenis diagram UML - Membuat diagram menjadi sederhana dan rapih - Digunakan untuk menyederhakan class diagram yang kompleks dengan mengelompokkannya kedalam package	Package, Packageable element, Dependencies (Import, Access), Element Import, Package Import, Package Merge
Model Diagram	Diagram struktur bantu atau tambahan UML yang menunjukkan beberapa abstraksi atau pandangan spesifik di system, dan juga untuk menjelaskan beberapa aspek arsitektur, logis atau perilaku system itu sendiri.	Model, Package, Packageable element, dependency.
Composite Structure Diagram	- Membantu pengguna memahami keadaan system mereka - Menyediakan detail dari arsitektur run-time dan pola penggunaan yang tidak dijelaskan pada static diagram	Terminator, Node (Circular, Rectangular), Actor, Class (Nama, Atribut, operasi), Part, Port, Interface, Connector (Association, Aggregation,

	- Memecah struktur internal dari berbagai class, interface, atau component, serta cara interaksinya - Memberikan informasi untuk melakukan optimasi dan Troubleshooting dalam system. - Menampilkan elemen-elemen internal pada sebuah class, agar class tersebut terlihat terstruktur ketika pengguna melihatnya	Composition, Dependency), Property, Collaboration
Internal Structure Diagram	- Memahami struktur dan perilaku pada system - Membantu dalam desain dan pengembangan system - Membantu dalam dokumentasi system - Membantu dalam komunikasi antar tim atau anggota	Structured class, part, port, connector, usage
Collaboration Use Diagram	Menggambarkan bagaimana object-object bekerja sama (berkolaborasi) dalam kasus penggunaan tertentu dimana berfokus pada penggunaan kolaborasi pada sistem	Collaboration, Connector, Part, dependency, Role, Role Binding, Role Type (Classifier)
Component diagram	- Digunakan dalam Component-Based-Development untuk menggambarkan system dengan Service-Oriented-Architecture - Menggabambarkan struktur fisik dari system berbasis objek yang terdiri dari komponen-komponen. - Untuk menentukan spesifikasi dan dokumentasi system berbasis komponen. - Dapat menghasilkan system yang bisa di eksekusi melalui proses forward engineering - Untuk memudahkan manajemen dan perawatan system dengan mengidentifikasi komponen-komponen yang perlu diperbaiki. - memodelkan struktur komponen dalam sebuah framework system - menampilkan struktur dari kode itu sendiri	Component, Node, Interface, Port, Package, Note, Link, Dependency, provided interface, required interface, class, connector, artifact, component realization, usage.
Manifestation diagram	Untuk menunjukkan manifestation (implementasi) oleh artifacts dan struktur internal artifacts. Diagram ini juga digunakan untuk menunjukkan bagaimana interface diimplementasikan dan digunakan dalam system.	Interface, Manifestation, Artifact, Implementation, Stereotype, Tagged Value, Comment
Deployment diagram	- menunjukkan struktur dari system run-time - menunjukkan software dan hardware yang akan di install atau digunakan - memodelkan jalur komunikasi antara elemen-elemen hardware, untuk mengetahui bagaimana hardware saling berinteraksi untuk mendukung berjalannya aplikasi - mendokumentasikan komponen software atau node yang berbeda akan dimplementasikan pada hardware yang ada.	Component, Artifact, Link, Interface, Node, Package, Manifestation, Deployment target, Communication path, Deployment, Dependency, Deployment specifications, Deployment specification dependency, Deployment specification association, device, Stereotype

	- Menggambarkan topologi hardware dari system, seperti bagaimana hardware tersebut terhubung dan saling berkomunikasi	
Network architecture diagram	Untuk menunjukkan bagaimana arsitektur jaringan pada sistem dirancang, jaringan tersebut bekerja, serta bagaimana jaringan dapat ditingkatkan, seperti meningkatkan keamanan dan koneksi jaringan.	Node, Hub, Switch, Router, Load Balancer, Firewall, Backbone, Multihoming, communication path, software, network segment, hardware
Profile Diagram	- Untuk membuat Metamodel UML baru - Untuk menerapkan extension atau modifikasi ke metamodel UML yang sudah ada - Untuk mengadaptasi metamodel UML ke berbagai platform dan domain	Profile, Metaclass, Stereotypes, Tagged values, Constraints

Behavior Diagrams

Diagram	Fungsi	Element
Use Case Diagram	- Memperlihatkan proses aktivitas secara terurut pada dalam system - Sebagai prnghubung antara produsen (developer) dengan konsumen untuk mendeskripsikan sebuah system. - Melakukan validasi terhadap arsitektur sistem - melakukan implemntasi dan menghasilkan tes cases - mendifiniskan functional requirements dalam suatu system - menentukan konteks dan persyaratan suatu sistem	Actor, Use case, Extend, Include, Association, System, Dependency, Generalization, Realization, Collaboration, Note, Anchor
Information Flow Diagram	- Memvisualisasikan penerusan informasi dan menganalisis situasi yang berbeda, serta untuk menggambarkan pertukaran informasi dalam system. - Memberikan informasi dengan detail bagaimana informasi mengalir dalam tugas-tugas individu - Memahami kendala proses bisnis dalam Sequential, Deferred, Real-Time, Parallel, Wheel, One-To-Many, Many-To-Many And Many-To-One-To-Many information flow	Actor, Information Flow, Information Item, External Entity, Class
Activity Diagram	- Menjelaskan urutan aktivitas dalam suatu proses pada system - Digunakan pada business modelling untuk menampilkan urutan aktifitas proses bisnis - Mudah dalam memahami proses secara keseluruhan pada system - Mengetahui aktivitas actor berdasarkan use case diagram yang sudah ada	Start Node (Initial Node), Activity, Action, Connector, Join/Synchronization bar, Fork, Decision, Merge, Note, Send Signal, Receive Signal, Shallow history pseudostate, Option loop, Flow final,

	- Merepresentasikan logika dari suatu algoritma - Menyederhakan dan meningkatkan proses apapun dengan mengklasifikasi use case yang rumit	Condition Text, End Node, Swimlane
State Machine Diagram	- Menggambarkan perilaku suatu object atau entitas yang bergantung pada keaadaan atau statusnya dengan membantu memodelkan bagaimana objek berinteraksi dengan sekitarnya - diterapkan pada elemen lain yang mempunyai perilaku terhadaoat entitas lain secara keseluruhan. - Mengilustrasikan use case dalam konteks bisnis agar dapat memahami berbagai aksi dan perubahan keadaan pada proses bisnis - Menunjukkan perilaku keseluruhan dari satu state machine atau sekelompok state machine yang terkait - Merekam perilaku dinamis system, untuk memahami bagaimana system berperilaku dalam berbagai situasi dan kondisi.	State, Transition, Event, Initial State, Final State, Decision, Composite state, Choice pseudo-state, Guard, Actions, Substate, Terminator, Activity, Transitional behavior, Trigger, Exit point
Behavioral State Machine Diagram	- Untuk menggambarkan perilaku suatu object atau entitas yang ada pada system berdasarkan berbagai keadaan (state) yang dialami serta transisi atau perubahan antara states tersebut sebagai respon terhadapat event - Untuk memahami aturan perilaku yang lebih rinci ketika perilaku (behavior) object menjadi kompleks dengan banyaknya transition dan juga states. - Membantu memahami perilaku keseluruhan dari system ketika banyak object yang saling berinteraksi dan berubah keadaannya	behavioral state, behavioral transition, pseudostate, trigger, action, Guard condition.
Protocol State Machine Diagram	- Untuk menggambarkan protocol penggunaan atau lifecycle suatu classfier. Dalam diagram ini, diperlihatkan operasi-operasi yang dapat dipanggil dari classifier pada setiap state, dengan syarat-syarat tertentu, dan memenuhi beberapa *optional postcondition* setelah classifier berpindah ke state tujuan. - untuk menggambarkan perilaku objek atau entitas dalam sistem, terutama dalam konteks komunikasi dan protokol. Diagram ini membantu dalam memodelkan bagaimana objek atau entitas berinteraksi satu sama lain dan mengikuti protokol komunikasi tertentu. Serta dapat menggambarkan bagaimana sistem-sistem berkomunikasi dengan bertukar pesan dan event untuk mencapai tujuan komunikasi yang ditentukan.	protocol state, protocol transition, pseudostate, trigger, message, System, Event.

	- Menjelaskan protocol komunikasi, mendeteksi kesalahan dalam desain atau implementasi, membantu dalam proses testing, dan mendokumentasikan kode program.	
Interaction Diagram	- Menggambarkan system sebagai urutan peristiwa atau kejadian yang berurutan - Untuk melakukan *reverse-engineering* (Menganalisis system yang sudah ada), atau forward-engineering (Merancang system baru) - visualisasi data real-time dan merepresentasikan arsitektur dari sistem berbasis objek. - Merekam perilaku dinamis system - menjelaskan interaksi antara objek-objek dalam sistem, termasuk pesan yang dikirim dan diterima. - Mengkomunikasikan Perilaku Pesan dan Lifeline dalam Sistem - Menggambarkan Interaksi dan Aliran Pesan dalam Sistem.	
Sequence Diagram	- Memodelkan interaksi tingkat tinggi antara objek-objek aktif dalam system, agar mengetahui bagaimana objek tersebut saling berinteraksi dalam lingkungan yang lebih luas - Untuk mengetahui urutan kejadian yang bisa menghasilkan output yang diinginkan, agar mengetahui alur kerja dari aktivitas atau interaksi tertentu - Menggambarkan aliran data dengan lebih terperinci, termasuk data atau perilaku yang diterima atau dikirimkan antara object-objek dalam interaksi tersebut. - untuk memodelkan interaksi generik antara objek-objek (menunjukkan semua kemungkinan jalur melalui interaksi) maupun untuk memodelkan instansi khusus dari interaksi (menunjukkan hanya satu jalur melalui interaksi).	Actor, Activation Box, Lifeline, Object, Messages (Call Message, Return Message, Self Message, Recursive Message, Create Message, Destroy Message, Duration Message), Note
Communication Diagram	- memodelkan bagaimana objek-objek atau elemen dalam system saling berinteraksi satu sama lain untuk mencapai tujuan tertentu. - Untuk memodelkan proses bisnis atau alur kerja yang melibatkan interaksi antara berbagai objek atau elemen dalam system - Untuk menjelaskan bagaimana berbagai tim atau kelompok berinteraksi dan berkolaborasi dalam mengembangkan system - Untuk menggambarkan interaksi tingkat tinggi dalam system, karena cenderung lebih sederhana daripada sequence diagram yang	Objects (Supplier objects, Client objects), Links, Messages, Lifeline

	membuatnya lebih dipahami oleh orang non-teknis. - Menggambarkan Bagaimana Pesan-Pesan Dikirimkan antara Objek dalam Sistem atau Perangkat Lunak - Mendukung Identifikasi Objek, Atribut, dan Operasi yang Terlibat dalam Kasus Penggunaan	
Timing Diagram	- Untuk menggambarkan interaksi antar objek dalam system dan bagaimana interaksi tersebut berlangsung seiring berjalannya waktu, agar memahami urutan waktu pesan-pesan dan peristiwa dalam system - Membantu dalam menganalisis waktu respons objek-objek terhadap pesan atau peristiwa terntentu, agar dapat mengidentifikasi kinerja system dan memastikan objek-objek tersebut merespons dalam waktu yang sesuai - Dapat memodelkan komunikasi synchronous dan asynchronous antara objek-objek, agar dapat memvisualisasikan pesan yang dikirim dan diterima secara langsung atau tanpa menunggu respons langsung. - Memahami ketergantungan waktu antara berbagai peristiwa dalam system, untuk melihat bagaimana satu peristiwa dapat memengaruhi peristiwa lainnya dalam konteks waktu. - Dapat diintegrasikan dengan diagram UML lainnya untuk memberikan pemahaman yang lebih komprehensif tentang interaksi dalam system. - Untuk merencanakan pengujian system dengan memahami urutan waktu yang diharapkan dari pesan-pesan dan peristiwa, agar perencanaan pengujian dan validasi system menjadi lebih baik. - Sebagai alat dokumentasi untuk menjelaskan interaksi waktu dalam system kepada orang lain.	Lifeline, State or Condition Timeline, Duration Constraint, Time Constraint, Destruction Occurrence
Interaction Overview Diagram	- Untuk memvisualisasikan interaksi yang kompleks, terutama ketika banyak objek atau aktivitas, karena bisa menyederhanakan representasi interaksi yang rumit dengan menggunakan subaktivitas atau fragmen aktivitas. - Penggunaan elemen *activity* untuk mewakili Langkah-langkah interaksi yang berbeda, karena digunakan untuk menggambarkan Langkah-langkah interaksi. - Fragmen aktivitas untuk menggambarkan berbagai alternatif atau opsi dalam interaksi,	Interaction, Interaction Use, Initial, Note, Activity/Action, decision, merge, branch, action/control flow, fork, join, flow final, end, duration constraint, time constraint.

	fragmen aktivitas digunakan untuk menggambarkan aliran control yang berbeda dalam *interaction*. - Dapat mengintegrasikan fragmen dari Activity diagram yang lebih rinci, agar menjelaskan lebih detail mengenai *activity* individu dijalankan sebagai bagian dari interaction keseluruhan.	

https://superapp.id/blog/uncategorized/class-diagram/

https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-package-diagram/

https://www.lucidchart.com/pages/uml-package-diagram

https://www.ruanglab.id/apa-itu-package-diagram.jsp

https://www.toketpedia.eu.org/2020/04/pengertian-object-diagram-fungsi-simbol-dan-contohnya.html

https://www.geeksforgeeks.org/unified-modeling-language-uml-object-diagrams/

https://www.dicoding.com/blog/memahami-class-diagram-lebih-baik/

https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-class-diagram/

https://www.ibm.com/docs/en/rsm/7.5.0?topic=structure-class-diagrams

https://aslitekno.com/pengertian-composite-structure-diagram-simbol-dan-contohnya/

https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-composite-structure-diagram/

https://www.lucidchart.com/pages/uml-composite-structure-diagram

https://flylib.com/books/en/2.926.1.86/1/

https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-component-diagram/

https://www.javatpoint.com/uml-component-diagram

https://creately.com/blog/software-teams/component-diagram-tutorial/

https://www.pinhome.id/blog/contoh-component-diagram/

https://www.lucidchart.com/pages/uml-component-diagram

https://www.javatpoint.com/uml-deployment-diagram

https://www.pinhome.id/blog/contoh-deployment-diagram/#komponen-deployment-diagram

https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-deployment-diagram/

https://www.edrawmax.com/article/profile-diagram-explained.html

https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-profile-diagram/

https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/

https://www.niagahoster.co.id/blog/use-case-diagram-adalah/

https://www.dicoding.com/blog/contoh-use-case-diagram/

https://online.visual-paradigm.com/knowledge/business-design/what-is-information-flow-diagram/

https://www.lucidchart.com/pages/uml-activity-diagram

https://www.jojonomic.com/blog/activity-diagram/

https://www.dicoding.com/blog/apa-itu-activity-diagram/

https://www.javatpoint.com/uml-state-machine-diagram

https://www.lucidchart.com/pages/uml-state-machine-diagram

https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-state-machine-diagram/

https://www.lucidchart.com/pages/uml-interaction-diagram

https://www.tutorialspoint.com/uml/uml_interaction_diagram.htm

https://www.javatpoint.com/uml-interaction-diagram

https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/

https://www.dicoding.com/blog/apa-itu-sequence-diagram/

https://www.javatpoint.com/uml-sequence-diagram

https://www.edrawmax.com/article/communication-diagram-uml.html

https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-communication-diagram/

https://www.javatpoint.com/uml-timing-diagram

https://www.edrawmax.com/article/interaction-overview-diagram-uml.html

https://www.uml-diagrams.org/uml-25-diagrams.html