

PERANCANGAN *E-LEARNING* MENGGUNAKAN METODE *DESIGN THINKING* DAN *DOMAIN DRIVEN DESIGN*

TESIS

Disusun sebagai salah satu syarat untuk

Memperoleh gelar Magister Komputer

Dari Sekolah Tinggi Manajemen Informatika dan Komputer LIKMI

Oleh :

MUHAMMAD AGUNG RIZKYANA

NPM: 2019210092



**PROGRAM STUDI PASCASARJANA
MAGISTER SISTEM INFORMASI SEKOLAH TINGGI
MANAJEMEN INFORMATIKA DAN KOMPUTER LIKMI
BANDUNG
2021**

**PERANCANGAN *E-LEARNING* MENGGUNAKAN METODE
DESIGN THINKING DAN *DOMAIN DRIVEN DESIGN***

Oleh :
MUHAMMAD AGUNG RIZKYANA
NPM: 2019210092

Bandung, 2 Maret 2021
Menyetujui,

Dr. Hery Heryanto, S.Kom, M.Kom
Pembimbing

**PROGRAM STUDI PASCASARJANA
MAGISTER SISTEM INFORMASI SEKOLAH TINGGI
MANAJEMEN INFORMATIKA DAN KOMPUTER LIKMI
BANDUNG
2021**

ABSTRAK

PERANCANGAN *E-LEARNING* MENGGUNAKAN METODE *DESIGN THINKING* DAN *DOMAIN DRIVEN DESIGN*

Muhammad Agung Rizkyana

NPM: 2019210092

Pandemi COVID19 pada awal tahun 2020 memberikan dampak yang signifikan pada segala sisi kehidupan masyarakat. Termasuk cara masyarakat bekerja dan belajar. Kebijakan work from home yang dilakukan oleh perusahaan atau learn from home yang dilakukan oleh institusi pendidikan adalah langkah beradaptasi dengan wabah ini. Kebijakan – kebijakan tersebut bergantung sangat tinggi pada kehandalan teknologi informasi khususnya pada kegiatan belajar dari rumah.

Perkembangan teknologi informasi pada era ini sudah semakin pesat. Jaringan internet sudah berada pada generasi 4 (4G) bahkan beberapa negara sudah di generasi 5 (5G). Pada generasi tersebut kecepatan transfer data dan lebar pita (bandwidth) semakin besar. Spesifikasi tersebut sangat mendukung aktivitas pendidikan dan pelatihan yang harus dilakukan dari rumah. Konten dan kegiatan belajar mengajar dapat disampaikan melalui media aplikasi yang terhubung pada jaringan ini.

Bisnis pendidikan dan pelatihan secara daring mulai tumbuh subur. Tren ini sebenarnya sudah terjadi sebelum masa pandemi. Namun dengan keadaan ini, bisnis tersebut menjadi sangat diminati. Para insan kreatif yang mempunyai keahlian dapat mengajarkan keahliannya tersebut melalui banyak media yang tersebar di internet. Selain itu, baik individu maupun kelompok yang membutuhkan pengajaran terkait keahlian tertentu dapat mengakses nya dari rumah. Sementara itu, terdapat kendala yang terjadi melalui fenomena ini, seperti materi yang disampaikan tidak sesuai dengan yang ditawarkan, konten terbatas pada on demand video atau live *streaming* atau webinar, pembajakan konten hingga pelacakan kemajuan belajar

Penelitian ini memiliki tujuan umum untuk mengetahui perancangan *e-learning* menggunakan metode *Design Thinking* dan *Domain Driven Design*. Secara khusus tujuan penelitian adalah analisis dan perancangan sistem informasi yang sesuai dengan kebutuhan pengguna (user centric), menentukan domain permasalahan *e-learning*, *Domain Driven Design* (DDD) digunakan untuk perancangan model *e-learning* berdasarkan domain masalah yang dihasilkan dari *Design Thinking* serta implementasi DDD kepada *arsitektur microservice*.

Metode *Design Thinking* dapat digunakan sebagai metode untuk mengidentifikasi kebutuhan pengguna. Tahapan yang dilalui pada penelitian ini adalah Empathise, Define the problem dan Ideate. Empathise mengidentifikasi kebutuhan pengguna melalui teknik observasi pengguna dan *user persona*. Define the problem mendefinisikan kebutuhan *e-learning* menggunakan metode *domain driven design*. Tahapan Ideate menerapkan model *e-learning* pada *arsitektur microservice*. Metode *Domain Driven Design* dapat digunakan sebagai pendekatan untuk membuat model *e-learning*. Teknik yang digunakan pada pemodelan domain antara lain identifikasi problem dan *solution space* pada domain *e-learning*, identifikasi *aggregates*, identifikasi *Entities* dan *value object*, definisikan proses bisnis, identifikasi *Command* dan queries. Selanjutnya, membuat model *event publish* dan *subscribe* untuk setiap *bounded context*.

Kata kunci : *E-learning*, *Arsitektur Microservice*, *Domain-Driven Design*, *Design Thinking*

ABSTRACT
PERANCANGAN *E-LEARNING* MENGGUNAKAN METODE
DESIGN THINKING* DAN *DOMAIN DRIVEN DESIGN

Muhammad Agung Rizkyana

NPM: 2019210092

The COVID-19 pandemic in early 2020 had a significant impact on all sides of people's lives. Including the way, people work and learn. The work from home policy carried out by companies or learn from home carried out by educational institutions is a step to adapt to this outbreak. These policies depend very much on the reliability of information technology, especially in learning from home.

The development of information technology in this era has grown rapidly. The internet network is already in generation 4 (4G) and some countries are already in generation 5 (5G). In this generation, the data transfer speed and bandwidth are getting bigger. These specifications strongly support educational and training activities that must be carried out from home. Content and teaching and learning activities can be conveyed through application media connected to this network.

Online education and training business is starting to thrive. This trend happened before the pandemic. However, under these circumstances, the business is in great demand. Creative people who have expertise can teach their skills through many media spread on the internet. Also, both individuals and groups who need teaching related to certain skills can access it from home. Meanwhile, some obstacles occur through this phenomenon, such as the material presented that is not following what is offered, limited Content to on-demand video or live streaming or webinars, Content piracy to tracking learning progress.

This study has a general objective to determine e-learning design using Design Thinking and Domain Driven Design methods. Specifically, the reSearch objectives are analysis and design of information systems according to user needs (user-centric), determining the domain of e-learning problems, Domain-Driven Design (DDD) is used for designing e-learning models based on problem domains resulting from Design Thinking and implementation. DDD to microservice architecture.

The Design Thinking method can be used as a method to identify user needs. The stages taken in this reSearch are Empathize, Define the problem and Ideate. Identifying user needs through user observation techniques and user personas. Defining the problem of defining e-learning needs using a domain driven design method. The Ideate stage applies the e-learning model to the microservice architecture. The Domain-Driven Design method can be used as an approach to creating an e-learning model. Techniques used in domain modelling include problem channels and space solutions in the E-learning domain, channel aggregates, entity and object values, define business processes, orders and queries. Next, create a publish and subscribe event model for each constrained context.

Keywords: E-learning, Microservice Architecture, Domain-Driven Design, Design Thinking

KATA PENGANTAR

Alhamdulillah puji dan syukur penulis panjatkan ke hadirat Allah Swt atas rahmat dan karunia-Nya yang telah diberikan kepada penulis, sehingga dapat menyelesaikan penyusunan laporan Tesis ini.

Tujuan dari penyusunan laporan Tesis ini adalah untuk mengembangkan kemampuan menyusun suatu karya tulis ilmiah dari hasil penelitian ilmiah dan mengembangkan kemampuan untuk menganalisis serta memecahkan masalah secara sistematis. Selain itu, tujuan dari Tesis ini adalah untuk menyelesaikan kuliah Pasca Sarjana.

Dengan segala kerendahan hati pada kesempatan ini penulis menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. Bapak Dr. Hery Heryanto, S.Kom., M.Kom selaku dosen pembimbing yang telah membimbing dan mengarahkan penulis dalam menyelesaikan laporan Tesis ini.
2. Bapak Dr. Djajasukma Tjahjadi, S.E, M.T selaku dosen wali
3. Keluarga yang telah memberikan semangat dan motivasi hingga terselesaikannya tesis ini serta atas segala dukungan dan doanya selama penulis menjalani pendidikan S2
4. Rekan-rekan Pasca Sarjana STMIK LIKMI, terima kasih atas dukungan kalian.
5. Semua pihak yang telah membantu, memberi semangat dan memberi segala masukan dalam menjalankan penelitian dan penyusunan laporan tesis yang tidak dapat disebutkan satu persatu.

Penulis menyadari Tesis ini masih jauh dari kata sempurna dan masih banyak kekurangannya. Oleh Karenanya kritik dan saran penulis harapkan demi penyusunan yang lebih baik dimasa yang akan datang. Penulis mengharapkan semoga laporan ini dapat bermanfaat bagi penulis dan umumnya bagi pembaca.

Akhir kata penulis berharap semoga tesis ini dapat memberikan manfaat khususnya di dunia ilmu pengetahuan bagi semua pihak. Penulis menyadari bahwa dalam melakukan penelitian Tesis ini masih banyak kekurangan dan jauh dari kata sempurna, kritik dan saran yang membangun sangat diharapkan untuk menyempurnakan Tesis ini.

Bandung, ____ Maret 2021

Penulis

DAFTAR ISI

ABSTRAK.....	i
ABSTRACT	ii
KATA PENGANTAR	iii
DAFTAR ISI.....	iv
DAFTAR GAMBAR	vii
DAFTAR TABEL	ix
DAFTAR LAMPIRAN	x
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah	2
1.3. Tujuan Penelitian	3
1.4. Batasan Penelitian	3
1.5. Sistematika Penulisan	4
BAB II LANDASAN TEORI	2
2.1. E-learning.....	2
2.1.1. E-learning pada organisasi	2
2.1.2. Metode belajar pada e-learning	3
2.1.3. Komponen pada e-learning.....	3
2.1.4. Ukuran kualitas pada E-learning.....	4
2.2. Design Thinking	6
2.2.1. Teknik pada Design Thinking.....	8
2.2.2. Definition	8
2.2.3. Research.....	8
2.2.4. Interpretation	10
2.2.5. Idea Generation	11
2.2.6. Prototyping	12
2.3. Domain Driven Design.....	13

2.3.1.	Konsep Domain Driven Design	13
2.3.2.	Arsitektur Microservice	19
2.3.3.	Integrasi antara Service	22
2.4.	UML (Unified Modelling Language)	25
2.4.1.	Notasi Use-Case Diagram	27
2.4.2.	Notasi Class Diagram	27
2.4.3.	Notasi Activity Diagram	28
2.5.	Penelitian Terkait	29
BAB III	OBJEK DAN METODOLOGI PENELITIAN	32
3.1.	E-learning	32
3.2.	Metode Design Thinking	33
3.3.	Metodologi Penelitian	34
BAB IV	HASIL DAN PEMBAHASAN	37
4.1.	Empathise	37
4.1.1.	Observasi pengguna	37
4.1.2.	User Persona	45
4.2.	Define the problem	46
4.2.1.	Problem and Solution space	47
4.2.2.	Aggregates	48
4.2.3.	Entities and Value object	49
4.2.4.	Saga	51
4.2.5.	Command and Queries	55
4.2.6.	Event	58
4.3.	Ideate	59
4.3.1.	Arsitektur E-learning	59
4.3.2.	Model E-learning	62
BAB V	KESIMPULAN DAN SARAN	69
5.1.	Kesimpulan	69
5.2.	Saran	70

DAFTAR PUSTAKA	x
LAMPIRAN	xi

DAFTAR GAMBAR

Gambar 2.1. Metodologi <i>Design Thinking</i> (What is <i>Design Thinking</i> , 2020)	6
Gambar 2.2. Contoh <i>user persona</i> (Yaici, 23:2016)	9
Gambar 2.3. <i>Empathy Mapping</i> Diagram (Yaici, 33:2016)	11
Gambar 2.4. <i>Prototyping</i> low fidelity	12
Gambar 2.5. <i>Prototyping</i> storyboard	13
Gambar 2.6. Identifikasi bisnis utama sebagai Problem Space, Core Business Domain atau Business Problem (Nair, Vijay. 2019)	14
Gambar 2.7. Identifikasi <i>sub-domain</i> sebagai dekomposisi dari domain utama (bisnis utama) (Nair, Vijay. 2019)	14
Gambar 2.8. Konsep <i>bounded context</i> pada model domain (Nair, Vijay. 2019)	15
Gambar 2.9. Konsep Domain Driven <i>Design</i> (D. Paulius, G. Gintautas, 2018)	16
Gambar 2.10. <i>Arsitektur Microservice</i> hasil dari pemodelan Domain Driven <i>Design</i> (Sam Newman, Building <i>Microservice</i> , 2015).....	19
Gambar 2.11. Perbandingan <i>Arsitektur Monolitik</i> dengan <i>Arsitektur Microservice</i> (Darren Gallipeau and Sara Kudrle, 2018).....	20
Gambar 2.12. Transisi metode DDD kepada <i>Arsitektur Microservice</i> (Roland H. Steinegger, Pascal Giessler, Benjamin Hippchen dan Sebastian Abeck, 2017)	21
Gambar 2.13. Model komunikasi Request/Response (Sam Newman, 91: 2015)	23
Gambar 2.14. Model komunikasi <i>Event-based</i> (Sam Newman, 92: 2015).....	24
Gambar 2.15. Operasi sistematis pada <i>bounded context</i> (Nair, Vijay. 2019).....	24
Gambar 2.16. Hexagonal Architecture (Nair, Vijay. 2019).....	25
Gambar 3.1. Metodologi Penelitian.....	35
Gambar 4.1. Penerimaan <i>e-learning</i> yang sudah ada	39
Gambar 4.2. Persentase responden	39
Gambar 4.3. Rentang usia responden	40
Gambar 4.4. Hasil analisa empathy mapping	44
Gambar 4.5. <i>Problem space E-learning</i>	47
Gambar 4.6. <i>Solution space E-learning</i>	48

Gambar 4.7. <i>E-learning aggregate</i>	49
Gambar 4.8. <i>Entities dan Value object</i>	50
Gambar 4.9. Proses bisnis pada <i>bounded context Course</i>	51
Gambar 4.10. Proses bisnis pada <i>bounded context Exam</i>	52
Gambar 4.11. Proses bisnis pada <i>bounded context Assignment</i>	53
Gambar 4.12. Proses bisnis pada <i>bounded context Payment</i>	54
Gambar 4.13. <i>Publish dan Subscribe event</i> pada domain <i>e-learning</i>	58
Gambar 4.14. <i>Arsitektur microservice E-learning</i>	60
Gambar 4.15. <i>Publish dan Subscribe event</i> pada domain <i>e-learning</i> dengan penambahan User service dan Notification service	61
Gambar 4.16. <i>Arsitektur Microservice E-learning</i> dengan User Service dan Notification Service	62
Gambar 4.17. Business use-case <i>e-learning</i>	63
Gambar 4.18. Use-case <i>Course Service</i>	64
Gambar 4.19. Use-case <i>Assignment Service</i>	65
Gambar 4.20. Use-case <i>Exam Service</i>	66
Gambar 4.21. Use-case <i>Payment Service</i>	67
Gambar 4.22. Package diagram <i>e-learning</i>	68

DAFTAR TABEL

Tabel 2.1. Model pada Domain Driven <i>Design</i>	16
Tabel 2.2. Pattern model domain driven <i>design</i>	17
Tabel 2.3. Artefacts Domain Driven <i>Design</i>	17
Tabel 2.4. Deskripsi simbol use-case	27
Tabel 2.5. Deskripsi simbol <i>class</i> diagram	28
Tabel 2.6. Deskripsi simbol activity diagram	28
Tabel 2.7. Pembahasan tentang penelitian terkait	29
Tabel 3.1. Use-case <i>E-learning</i>	32
Tabel 4.1. Variabel empathy mapping	37
Tabel 4.2. Variabel <i>empathy mapping</i> (skala likert)	38
Tabel 4.3. Bobot skala likert	40
Tabel 4.4. Nilai maksimum dan minimum	41
Tabel 4.5. Persentase variabel <i>empathy mapping</i> TF	41
Tabel 4.6. Persentase variabel <i>empathy mapping</i> SD	41
Tabel 4.7. Persentase variabel <i>empathy mapping</i> HR1 (Rekan)	42
Tabel 4.8. Persentase variabel <i>empathy mapping</i> HR2 (Pimpinan)	42
Tabel 4.9. Persentase variabel <i>empathy mapping</i> SE	43
Tabel 4.10. Persentase variabel <i>empathy mapping</i> GN	43
Tabel 4.11. Tanggapan pengguna <i>e-learning</i>	44
Tabel 4.12. <i>User persona</i>	45
Tabel 4.13. <i>Sub-domain</i> dari domain <i>e-learning</i>	46
Tabel 4.14. <i>Command Queries Course</i>	55
Tabel 4.15. <i>Command Queries Assignment</i>	56
Tabel 4.16. <i>Command Queries Exam</i>	57
Tabel 4.17. <i>Command Queries Payment</i>	57

DAFTAR LAMPIRAN

Lampiran A : Formulir Kuisisioner	xi
Lampiran B : Spreadsheet Response Kuisisioner	xviii

BAB I

PENDAHULUAN

1.1. Latar Belakang

Pada awal tahun 2020 seluruh dunia digemparkan dengan wabah COVID 19. Dimulai sekitar November 2019 di sekitar wilayah Wuhan, Cina hingga menyebar hampir ke seluruh dunia. Pendidikan hingga industri mengharuskan setiap individu untuk bekerja dan beraktivitas dari rumah. Kampanye *stay at home* menyebar di seluruh negeri. Kebijakan working from home diberlakukan hampir di setiap instansi. Tren beraktivitas dari rumah berlaku juga pada dunia pendidikan. Baik pendidikan dasar, menengah, hingga tinggi. Kemudian, pendidikan yang bersifat temporal seperti lembaga kursus dan pelatihan, semua dilakukan dari rumah. Cara untuk mendapatkan akses terhadap pendidikan dan pelatihan ini adalah melalui daring.

Perkembangan teknologi informasi pada era ini sudah semakin pesat. Jaringan internet sudah berada pada generasi 4 (4G) bahkan beberapa negara sudah di generasi 5 (5G). Pada generasi tersebut kecepatan transfer data dan lebar pita (*bandwidth*) semakin besar. Untuk jaringan 4G atau LTE (*Long Term Evolution*) kecepatan *downlink* sekitar 100 Mbps dan *uplink* sekitar 50 Mbps sedangkan untuk jaringan 5G kecepatan *downlink* sekitar 20 Gbit/s dan *uplink* 10 Gbit/s (Dahmen-Lhuissier, Sabine. 2020). Spesifikasi tersebut sangat mendukung aktivitas pendidikan dan pelatihan yang harus dilakukan dari rumah. Konten dan kegiatan belajar mengajar dapat disampaikan melalui media aplikasi yang terhubung pada jaringan ini.

Bisnis pendidikan dan pelatihan secara daring mulai tumbuh subur. Tren ini sebenarnya sudah terjadi sebelum masa pandemi. Namun dengan keadaan ini, bisnis tersebut menjadi sangat diminati. Para insan kreatif yang mempunyai keahlian dapat mengajarkan keahliannya tersebut melalui banyak media yang tersebar di internet. Selain itu, baik individu maupun kelompok yang membutuhkan pengajaran terkait keahlian tertentu dapat mengakses nya dari rumah. Kemudian, bisnis ini menawarkan nilai lebih diantara nya sertifikasi keahlian, lapangan pekerjaan, pengalaman kerja (*portfolio*), materi

yang mudah diakses dengan harga murah. Penyedia layanan pendidikan dan pelatihan secara daring pun bermunculan baik dari swasta bahkan hingga negara melalui BUMN. Sementara itu, terdapat kendala yang terjadi melalui fenomena ini, seperti materi yang disampaikan tidak sesuai dengan yang ditawarkan, konten terbatas pada on demand video atau live *streaming* atau webinar, pembajakan konten hingga pelacakan kemajuan belajar. Penelitian ini berfokus pada perancangan *e-learning* yang dapat menjadi solusi atas permasalahan tersebut. Perancangan sistem akan melalui metode *Design Thinking*. Metode ini melalui tahapan *Empathise*, *Ideate*, *Design*, *Prototype* dan *Test*. Setiap tahapan dapat memastikan rancangan sistem informasi yang sesuai dengan kebutuhan pengguna dan dapat memberikan nilai tambah. Selain itu, perancangan sistem informasi ini pun akan menggunakan pendekatan *Domain Driven Design*. Pendekatan ini digunakan pada tahapan *Design* dan *Prototype* pada metode *Design Thinking*. Dengan fokus serta metode dan pendekatan tersebut, maka judul penelitian ini adalah Perancangan *E-learning* menggunakan metode *Design Thinking* dan *Domain Driven Design*.

Metode *Design Thinking* dapat digunakan sebagai metode untuk mengidentifikasi kebutuhan pengguna. Tahapan yang dilalui pada penelitian ini adalah *Empathise*, *Define the problem* dan *Ideate*. *Empathise* mengidentifikasi kebutuhan pengguna melalui teknik observasi pengguna dan *user persona*. *Define the problem* mendefinisikan kebutuhan *e-learning* menggunakan metode *domain driven design*. Tahapan *Ideate* menerapkan model *e-learning* pada *arsitektur microservice*. Metode *Domain Driven Design* dapat digunakan sebagai pendekatan untuk membuat model *e-learning*. Teknik yang digunakan pada pemodelan domain antara lain identifikasi *problem* dan *solution space* pada domain *e-learning*, identifikasi *aggregates*, identifikasi *Entities* dan *value object*, definisikan proses bisnis, identifikasi *Command* dan *queries*. Selanjutnya, membuat model *event publish* dan *subscribe* untuk setiap *bounded context*.

1.2. Rumusan Masalah

Berdasarkan latar belakang penelitian, maka didapatkan rumusan masalah dari penelitian ini. Rumusan masalah terkait dengan topik utama penelitian yaitu *Design Thinking* dan *Domain Driven Design*. Rumusan tersebut antara lain :

1. Bagaimana analisis dan perancangan *e-learning* menggunakan *design thinking*?
2. Bagaimana mengidentifikasi permasalahan (domain) utama dari *e-learning*?
3. Bagaimana merancang *e-learning* menggunakan domain driven *design*?
4. Apa contoh implementasi dari rancangan *e-learning* menggunakan domain driven *design*?

1.3. Tujuan Penelitian

Penelitian ini memiliki tujuan umum untuk mengetahui perancangan *e-learning* menggunakan metode *Design Thinking* dan Domain Driven *Design*. Secara khusus tujuan penelitian ini antara lain :

1. Analisis dan perancangan
Design Thinking membantu analisis dan perancangan sistem informasi yang sesuai dengan kebutuhan pengguna (*user centric*)
2. Identifikasi domain
Design Thinking membantu untuk menentukan domain permasalahan
3. Perancangan model *e-learning*
Domain Driven Design (DDD) digunakan untuk perancangan model *e-learning* berdasarkan domain masalah yang dihasilkan dari *Design Thinking*
4. Contoh implementasi
DDD diimplementasi kepada salah satu arsitektur yaitu *arsitektur microservice*

1.4. Batasan Penelitian

1. Tidak membahas mengenai strategi dan perencanaan sistem informasi serta proyek sistem informasi
2. Tidak membahas perbandingan metodologi perancangan sistem informasi
3. Menggunakan data berdasarkan hasil studi literatur dan observasi
4. Tidak mencantumkan implementasi dalam bentuk *source code* atau aplikasi atau desain tampilan.
5. *E-learning* yang dibuat adalah *Asynchronous*.
6. Lingkup *e-learning* terdiri dari *Course, Assignment, Examination, Payment*.

1.5. Sistematika Penulisan

BAB I PENDAHULUAN

Uraian dari pembahasan tentang latar belakang, rumusan masalah, tujuan penelitian, batasan penelitian dan sistematika penulisan

BAB II LANDASAN TEORI

Uraian dari pembahasan teori – teori yang digunakan pada penelitian ini yaitu, *E-learning*, Komponen *e-learning*, Metode belajar pada *e-learning*, *Design thinking*, *Domain Driven Design* dan Pengembangan sistem informasi.

BAB III OBJEK DAN METODOLOGI PENELITIAN

Uraian dari pembahasan objek dan metodologi penelitian yang berisi tentang *e-learning*, metode *design thinking* dan proses perancangan *e-learning* menggunakan *domain driven design*.

BAB IV ANALISIS DAN PEMBAHASAN

Uraian dari analisis dan inti dari penelitian *e-learning* menggunakan metode *design thinking* dan *domain driven design*

BAB V KESIMPULAN DAN SARAN

Uraian tentang kesimpulan dan saran – saran berkaitan dengan perancangan *e-learning*.

BAB II LANDASAN TEORI

2.1. *E-learning*

Sistem informasi pembelajaran daring atau yang lebih dikenal sebagai *e-learning* adalah suatu sistem pendidikan yang memanfaatkan teknologi informasi dalam proses belajar mengajar. *E-learning* menjadi sangat populer pada era teknologi informasi dan komunikasi terlebih dengan kehadiran aplikasi berbasis web dan *mobile*. Pertumbuhan yang cepat pada teknologi tersebut berakibat pada kebiasaan atau cara belajar yang berubah. Hal ini sangat mendukung proses kegiatan belajar mengajar. Kemudian, *e-learning* dapat dibagi menjadi dua tipe berdasarkan lingkungan penerapannya (D. Wu, J. Lu and G. Zhang, 2015), antara lain :

1. Formal

Pada tipe ini materi pembelajaran diberikan langsung oleh institusi pendidikan seperti sekolah atau universitas. Materi yang diberikan sudah tersusun berdasarkan kurikulum atau silabus.

2. Informal

Pada tipe ini, biasanya materi yang diberikan disampaikan oleh pihak yang mempunyai keahlian dibidang tertentu. Setiap materi bisa saja dibawa oleh pihak yang berbeda. Kemajuan dalam belajar sangat tergantung pada motivasi peserta.

2.1.1. *E-learning* pada organisasi

Pada sudut pandang organisasi, penggunaan *e-learning* adalah cara yang efektif untuk pembelajaran layaknya belajar langsung dikelas dengan biaya yang relatif lebih murah. Kemampuan yang dapat di kembangkan pada *e-learning* antara lain: kognitif, interpersonal dan psikomotor. Kemampuan kognitif meliputi pengetahuan dan pemahaman terhadap studi kasus, sebagai contoh kemampuan algoritme dan pemrograman atau konsep – konsep ilmiah. Kemampuan interpersonal adalah pengembangan diri yang fokus terhadap perubahan pola pikir dan tingkah laku, sebagai contoh kemampuan ini meliputi

komunikasi, negosiasi, berbicara pada umum (*public speaking*) dsb. Kemampuan psikomotor meliputi peningkatan kemampuan pada fisik, sebagai contoh olahraga.

2.1.2. Metode belajar pada e-learning

Jika dilihat pada metode atau cara belajar di *e-learning*, terdapat dua jenis cara belajar di *e-learning*, antara lain : *self-paced* dan *instructor-led*. *Self-paced* adalah cara peserta belajar mempelajari materi secara mandiri. Materi yang disampaikan sudah disesuaikan target pencapaiannya dan disampaikan melalui beragam media seperti *text*, *audio*, gambar, *audio*, *video* atau gabungan. Dengan peningkatan teknologi jaringan saat ini (4G atau 5G) memungkinkan materi disampaikan secara *streaming* sehingga pembelajar mendapatkan opsi selain mengunduh materi tersebut. *Instructor-led* adalah cara belajar dengan dibimbing langsung oleh instruktur. Materi dapat disampaikan sama seperti pada *self-paced* namun akan ada sesi langsung dengan instruktur. Pada metode ini biasanya jadwal sudah tersedia, selain itu ada tugas bagi peserta belajar dan komunikasi interaktif dengan instruktur. Komunikasi dilaksanakan melalui media forum diskusi atau chat. Selain itu, melalui survey, polling, aplikasi berbagi file (*file sharing*) atau melalui media video conference. Pada akhir sesi, dilaksanakan ujian untuk mengukur kemampuan peserta belajar.

Jika dilihat pada waktu maka *e-learning* dapat dibagi menjadi dua, yaitu : *synchronous* dan *asynchronous*. Pada *synchronous*, proses belajar terjadi pada waktu dan media yang sama, sebagai contoh adalah pembicaraan pada aplikasi chat atau video conference. Kemudian, *asynchronous* adalah proses belajar melalui waktu yang berbeda atau bebas terhadap waktu. Sebagai contoh pada metode belajar mandiri, peserta belajar dapat mengakses konten video belajar kapan saja. (*E-learning methodology*, 2011).

2.1.3. Komponen pada e-learning

Komponen *e-learning* merupakan apa – apa saja yang dapat menjadi media belajar bagi peserta belajar berdasarkan metode belajar yang dapat dilakukan pada *e-learning*. Komponen tersebut antara lain : Konten atau materi ajar, Kelas virtual, *Tutor* atau pengajar dan Kolaborasi. (Ghirardini Beatrice, 2015).

Konten atau materi ajar dapat berupa video, *text (e-book)* atau *audio*. Selain itu, dapat berupa pembelajaran interaktif sebagai contoh, materi *video* dengan pertanyaan pilihan yang dapat dilakukan oleh peserta ajar. Kemudian, materi dalam bentuk simulasi. Sebagai contoh, pada materi pengembangan aplikasi web, materi simulasi dapat disampaikan dalam bentuk proyek aplikasi sehingga peserta ajar dapat melakukan dan mengalami proses – proses pengembangan aplikasi web secara langsung. Komponen selanjutnya yaitu Kelas Virtual, komponen ini mempunyai definisi dari *E-learning Methodologies* yang menyatakan bahwa :

A virtual classroom is an e- learning event where an instructor teaches remotely and in real time to a group of learners using a combination of materials

Kelas virtual sebenarnya sama dengan kelas konvensional pada umumnya, hanya saja kelas ini berlangsung melalui media aplikasi dan internet. Pengajar atau tutor dan peserta ajar dalam satu tempat dan waktu.

Pengajar atau tutor merupakan aspek manusia yang dihadirkan pada *e-learning*. Aspek ini menghadirkan interaksi sosial secara langsung antara peserta ajar dengan tutornya. Interaksi ini dapat berlangsung melalui *e-learning* tersebut. Komponen terakhir adalah Kolaborasi. Komponen ini berisi tentang segala peristiwa yang terjadi pada saat proses belajar mengajar. Sebagai contoh, diskusi dan berbagi pengetahuan (*knowledge-sharing*) antara peserta ajar. Kegiatan ini dapat berlangsung melalui diskusi via aplikasi *chatting* atau *email*. Selain itu, kolaborasi dapat berlangsung dengan pemberian tugas dalam kelompok sehingga dapat dilakukan oleh semua peserta ajar.

2.1.4. Ukuran kualitas pada E-learning

Kualitas dalam pembelajaran baik secara tradisional (tatap muka) ataupun melalui media *online (e-learning)* harus tetap diupayakan dengan baik. Dengan tujuan setiap materi yang disampaikan dapat meningkatkan pemahaman peserta ajar selain itu kemampuan, pengetahuan dan informasi dapat mengalami pembaruan. Kemudian, adapun ukuran kualitas dari *e-learning (E-learning Methodologies, 14: 2011)* antara lain :

1. Materi yang berbasis pada peserta ajar (*learned-centered Content*)

Materi sebaiknya berbasis pada kebutuhan peserta ajar serta sesuai dengan peran dan tanggungjawab secara profesional yang melekat pada peserta ajar. Selanjutnya, diharapkan pada akhir pembelajaran ada perubahan dan peningkatan pada kemampuan, pengetahuan dan informasi yang diterima peserta ajar.

2. Materi yang detail dan spesifik (*granularity*)

Materi pada *e-learning* sebaiknya dibuat spesifik dan dikelompokkan (*segmented*) agar dapat memfasilitasi kesulitan yang dihadapi peserta ajar saat menerima pengetahuan baru. Selain itu, terdapat keleluasaan waktu pembelajaran.

3. Materi yang mengikat peserta ajar (*engage Content*)

Materi pada *e-learning* mengutamakan penyampaian yang kreatif. Kreatif dimaksudkan agar terjadi ikatan antara materi dengan peserta ajar serta memberikan motivasi belajar.

4. Materi yang interaktif (*interactivity*)

Materi pada *e-learning* sebaiknya interaktif agar interaksi peserta ajar dengan materi dapat terus dipertahankan konsentrasinya dan mendukung proses belajar.

5. Materi yang disesuaikan dengan peserta ajar

Materi pada *e-learning* dipelajari baik secara mandiri oleh peserta ajar (*Self-paced*) atau melalui instruktur (*Instructor-led*). Pada *self-paced*, materi sebaiknya disesuaikan dengan minat peserta ajar. Sementara itu, pada *instructor-led*, arahan dari instruktur, materi yang disampaikan serta fasilitas belajar sebaiknya mengikuti perkembangan belajar dari peserta ajar.

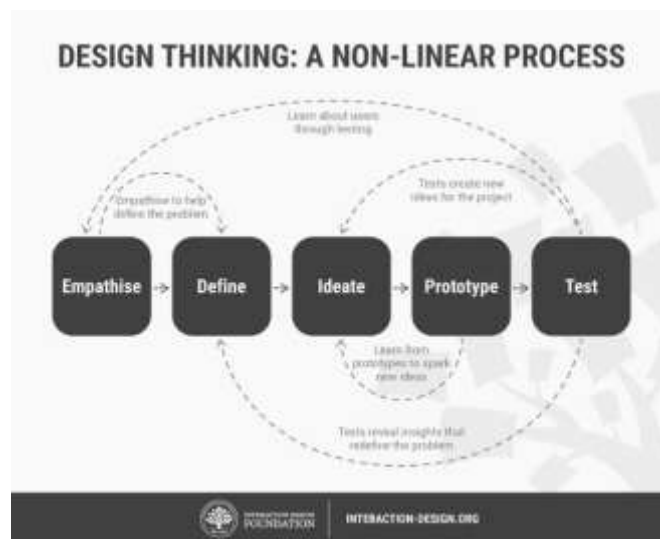
Dari ukuran kualitas *e-learning* yang sudah disampaikan terdapat cara belajar yang mendukung tentang isu kualitas yaitu metode *Blended Learning*. Metode ini menggabungkan proses belajar secara tradisional (tatap muka) dengan kelas *online*. Metode ini diharapkan mampu meningkatkan kualitas pemahaman, menghilangkan kesenjangan informasi dan membentuk komunitas diantara peserta ajar dan instruktur untuk pembahasan materi yang lebih mendalam. Terdapat beberapa pendekatan yang dapat dilakukan untuk menerapkan metode *blended learning* pada *e-learning*. Pendekatan tersebut antara lain :

1. Kelas *online* kepada kelas tatap muka
2. Kelas tatap muka kepada kelas *online*
3. Kelas *online* kepada kelas tatap muka dilanjutkan kembali kepada kelas *online*

2.2. **Design Thinking**

Design Thinking merupakan salah satu metode pengembangan sistem informasi untuk menemukan solusi secara sistematis yang diadopsi dari disiplin ilmu user experience. Metode ini berfokus pada psikologi manusia (*Human Centered*). Adapun pendapat dari artikel dengan judul “*What is Design Thinking*” menyatakan bahwa :

Design Thinking is a design methodology that provides a solution-based approach to solving problems. It's extremely useful in tackling complex problems that are ill-defined or unknown, by understanding the human needs involved, by re-framing the problem in human-centric ways, by creating many ideas in brainstorming sessions, and by adopting a hands-on approach in Prototyping and testing (“What Is Design Thinking?”, 2020)



Gambar 2.1. *Metodologi Design Thinking (What is Design Thinking, 2020)*

Gambar 2.1 menunjukkan metode *Design Thinking* mempunyai 5 tahapan. Sifat dari metode *Design Thinking* adalah non-linear, artinya setiap tahapan dapat dilalui setelah selesai tahapan sebelumnya atau satu tahapan dapat lompat ke beberapa tahapan selanjutnya atau kembali ke tahapan sebelumnya. Tahapan tersebut antara lain :

1. *Empathise*

Tahapan ini menggambarkan karakteristik dari pengguna sistem informasi. Perancangan sistem informasi menggunakan asumsi terlebih dahulu selanjutnya mendalami sifat dan karakteristik dari pengguna. Pendalaman dapat melalui wawancara atau survey (kuisisioner).

2. *Define the problem*

Tahapan ini melakukan analisa masalah dari tahapan sebelumnya. Masalah di definisikan dalam pernyataan yang dapat dimengerti secara umum. Definisi masalah bukan berpusat pada persepsi diri sendiri, sebagai contoh, daripada melalui pernyataan "Kita harus meningkatkan sebaran pasar makanan siap saji dengan sasaran wanita karir sebesar 10%", akan lebih baik dengan pernyataan "Wanita karir sangat membutuhkan makanan siap saji untuk memudahkan membuat sarapan bagi keluarganya".

3. *Ideate*

Tahapan ini melakukan ideasi terhadap masalah yang sudah didefinisikan. Ideasi dilakukan dengan mengumpulkan banyak ide kemudian merumuskannya. Untuk dapat memunculkan ide, dapat digunakan pernyataan "Bagaimana seharusnya kita [masalah]?". Sebagai contoh, "Bagaimana seharusnya kita memasarkan makanan siap saji yang memudahkan para wanita karir saat membuat sarapan bagi keluarganya?"

4. *Prototype*

Tahapan ini melakukan rancangan interaksi antara pengguna dari sistem informasi. Fitur, data, proses bisnis di definisikan juga pada tahapan ini. Akhir dari tahapan ini adalah sistem informasi yang sudah jelas secara solusi atas permasalahan yang sudah didefinisikan pada tahapan sebelumnya

5. *Test*

Tahapan ini melakukan pengujian terhadap rancangan sistem informasi. Pengujian dapat dilakukan dengan memberikan rancangan secara langsung pada pengguna.

Metode *Design Thinking* merupakan metode yang memungkinkan adanya iterasi, maka pada tahapan ini bisa saja kembali lagi ke tahapan pertama atau sebelumnya

2.2.1. Teknik pada *Design Thinking*

Design Thinking mempunyai fokus terhadap masalah pengguna. Masalah tersebut bisa timbul dari pengguna itu sendiri (kompetensi dan pengalaman), produk atau jasa yang digunakan, dan bisnis yang beroperasi. Metode ini bukanlah metode yang dapat digunakan untuk setiap permasalahan. Beberapa permasalahan masih memerlukan pendekatan lain untuk ditemukan solusi yang optimal. Sebagai contoh perencanaan proyek dan strategi implementasi teknologi informasi. Pengembangan sistem informasi yang menggunakan metode *Design Thinking* mencoba untuk membuat solusi nyata dari permasalahan yang dihadapi oleh penggunanya. Untuk dapat menjalankan setiap fase *Design Thinking* dengan baik diperlukan teknik yang tepat guna (Yayici, 2016). Teknik tersebut antara lain : *Definition, Research, Interpretation, Idea Generation, Prototyping*.

2.2.2. Definition

Penentuan definisi dari masalah yang akan dibuatkan solusi adalah hal yang penting. *Design Thinking* akan membutuhkan banyak pandangan untuk dapat menentukan solusi. Definisi dari masalah menjadi kunci kesuksesan *design thinking*. Menyediakan waktu yang lebih panjang untuk membuat definisi menjadi jelas, spesifik, achievable dan purpose-led. Adapun teknik untuk membuat definisi menjadi jelas dengan *HMW Questions Technique*. HMW (*How might we*) adalah pertanyaan yang dapat diajukan kepada tim pengembang untuk membantu definisi masalah dan menentukan lingkupnya. Contoh pertanyaan tersebut adalah “*How might we improve the sales of our dealers at remote locations?*” atau “Bagaimana seharusnya kita dapat meningkatkan penjualan di setiap dealer dari lokasi yang jauh?”

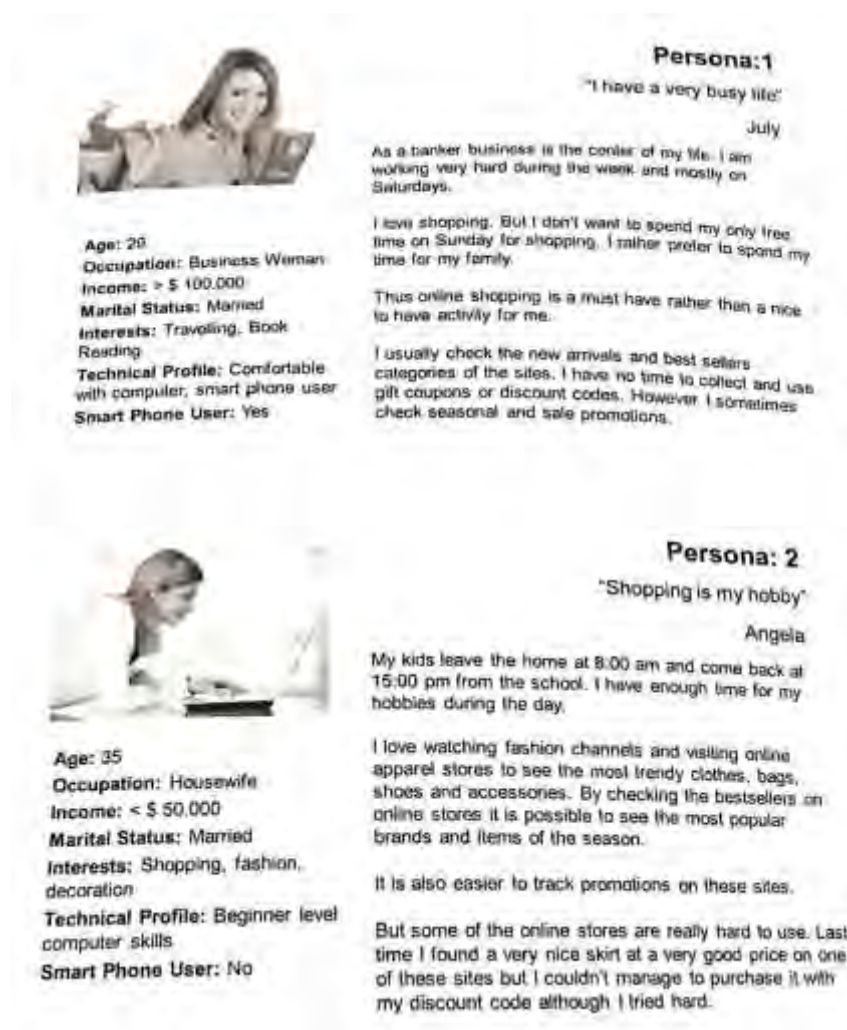
2.2.3. Research

Solusi yang baik bukanlah satu solusi besar atau umum yang dapat menyelesaikan masalah. Solusi yang baik adalah solusi yang spesifik terhadap kebutuhan dan masalah dari target penggunaannya. Oleh karena itu, tim pengembang yang

menjalankan metode *Design Thinking* perlu untuk melakukan riset. Teknik riset yang digunakan antara lain :

1. User Persona

Persona adalah gambaran karakter representative dari pengguna. Teknik *user persona* tidak perlu menggambarkan keseluruhan karakter dan perilaku pengguna. Perlu ada batasan persona. Setiap persona dapat mewakili suatu kasus atau proses bisnis tertentu.



Gambar 2.2. Contoh *user persona* (Yaici, 23:2016)

Sebuah *user persona* harus menyedikan nama, foto dan info demografis seperti usia, jenis kelamin, pendidikan, profesi dsb. Serta terdapat sebuah skenario yang mewakili

perilaku dari pengguna tersebut. Kemudian setelah ditentukan *user persona*, perlu untuk memahami keinginan, kapabilitas dan ekspektasi mereka. Cara efektif untuk mengetahui hal tersebut melalui wawancara (interview).

2. Interview (Wawancara)

Tujuan dari wawancara adalah mendapatkan informasi sebanyak – banyaknya dengan menanyakan beberapa pertanyaan spesifik dan tidak bias kepada pengguna (*user persona*).

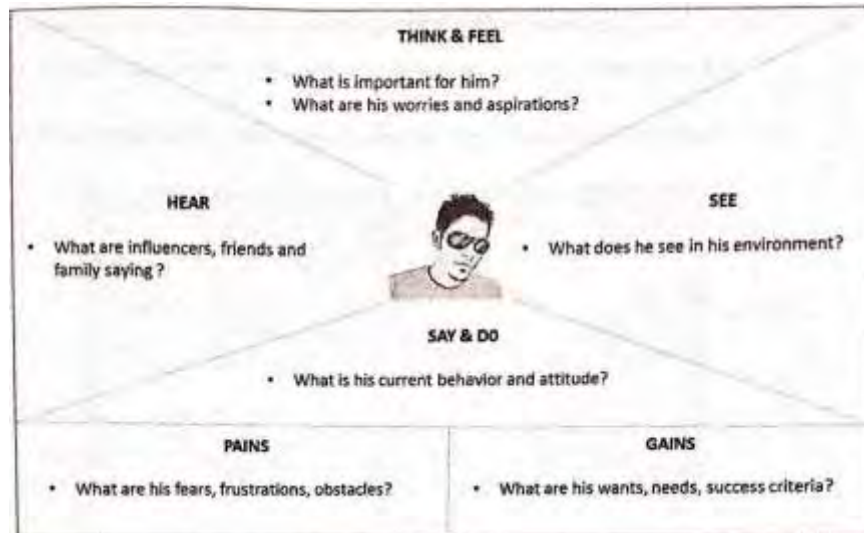
3. Observasi pengguna

Tujuan dari observasi pengguna adalah menentukan konteks atau wawasan lain dari kebutuhan dan masalah yang tidak dapat ditemukan pada saat wawancara.

2.2.4. Interpretation

Interpretation merupakan tahapan untuk menganalisa data hasil riset, identifikasi pola yang ada serta membuat tindak lanjut berdasarkan solusi kreatif (Yayici, 31:2016). Pada tahapan ini diupayakan visualisasi terhadap data hasil riset. Beberapa cara untuk membuat visualisasi tersebut menggunakan diagram. Diagram yang dapat membantu diantaranya *empathy mapping diagram*, *mind-mapping*, *journeys maps* dan *affinity diagram*.

Metode *Design Thinking* pada dasarnya menggunakan pendekatan empati. Pendekatan tersebut digunakan untuk lebih mendapatkan wawasan terhadap aspek manusia (*emotional bonds*). Selanjutnya, produk yang dibuat dapat mempunyai keterikatan secara emosi dengan penggunanya. *Empathy mapping* diagram adalah alat bantu untuk membuat visualisasi terhadap empati tersebut. Sebelum dapat menentukan empati yang tepat, diperlukan data riset dan *user persona*. Adapun komponen dari *empathy mapping* adalah, *think and feel*, *hear*, *see*, *say and do*, *pains and gains*.



Gambar 2.3. *Empathy Mapping Diagram* (Yaici, 33:2016)

2.2.5. Idea Generation

Idea generation adalah tahapan untuk menemukan solusi secara kreatif yang dapat menangani kebutuhan pengguna. Ada beberapa cara atau teknik untuk membuat ide antara lain : *Brainstorming*, *Brain-dump*, *Benchmarking*, *Prioritization*, *Value proposition canvas*. Sementara itu kemampuan pendukung untuk dapat membuat ide secara efektif adalah improvisasi, berfikir tidak biasa (*thinking out-of-the-box*), dan kreatifitas.

Brainstorming dilakukan secara kelompok. Setiap anggota kelompok secara spontan menyampaikan ide – ide untuk kemudia dituliskan ke sebuah daftar catatan ide. Apabila setiap anggota mempunyai keraguan akan ide yang akan disampaikan, dapat dilakukan teknik *brain dump*. Teknik *brain-dump* dilakukan dengan cara setiap anggota menuliskan ide – ide pada kertas tanpa disampaikan secara lisan.

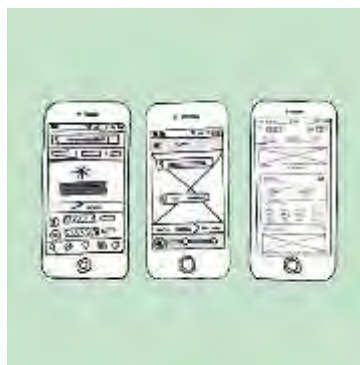
Benchmarking dilakukan untuk mendapatkan ukuran yang sesuai bagi pengguna dari sistem informasi yang sudah ada sebelumnya. Hal ini diperlukan agar perancangan tidak hanya berfokus pada fungsional dan solusi yang praktis namun dapat juga mendapatkan wawasan dari sistem informasi yang sudah ada dan diterima oleh pengguna. Teknik ini dilakukan secara ideal apabila sudah terdapat ide – ide yang didapatkan pada teknik *brainstorming* atau *brain-dump*.

Prioritization dilakukan untuk menentukan skala prioritas dari setiap ide yang dikemukakan. Setiap proyek mempunyai keterbatasan sumber daya sehingga penetapan prioritas menjadi penting. Pertimbangan prioritas sebaiknya bertumpu pada *value proposition* dan tingkat kesulitan implementasi. *Value proposition* adalah kesesuaian ide dengan kebutuhan dari pengguna. Untuk dapat menentukan nilai yang tepat pada suatu ide dapat dilakukan dengan teknik *value proposition canvas*. *Value proposition canvas* menunjukkan bagaimana manfaat dari ide – ide tersebut yang memberikan dampak terhadap ekspektasi yang diharapkan dan mengurangi kesulitan pengguna yang disampaikan pada *empathy mapping*.

2.2.6. Prototyping

Prototyping adalah tahapan untuk membuat iterasi pada suatu ide. Iterasi itu dilakukan untuk menguji dan meningkatkan fungsional dan *usability* dari ide tersebut. Terdapat dua teknik *Prototyping*, yaitu low fidelity dan storyboard. Low fidelity adalah teknik pada iterasi prototype agar dapat dilakukan perubahan secara cepat dan mudah. Prototype disajikan bukan dalam bentuk final, sehingga dapat mengumpulkan sebanyak – banyaknya tanggapan dari pengguna. Kemudian, tim desain thinking dapat dengan efektif beradaptasi dan mengantisipasi kegagalan dari setiap iterasi.

Storyboard dibuat untuk membuat purwarupa (*prototype*) yang sudah berhubungan dengan proses bisnis. Teknik ini di adaptasi dari proses pembuatan film. Solusi prototype dibuat dengan cara memvisualisasikan interaksi pengguna ke dalam bentuk cerita bergambar.



Gambar II.4. *Prototyping low fidelity*
(<https://invisionapp.com/inside-design/low-fi-vs-hi-fi-Prototyping/>)



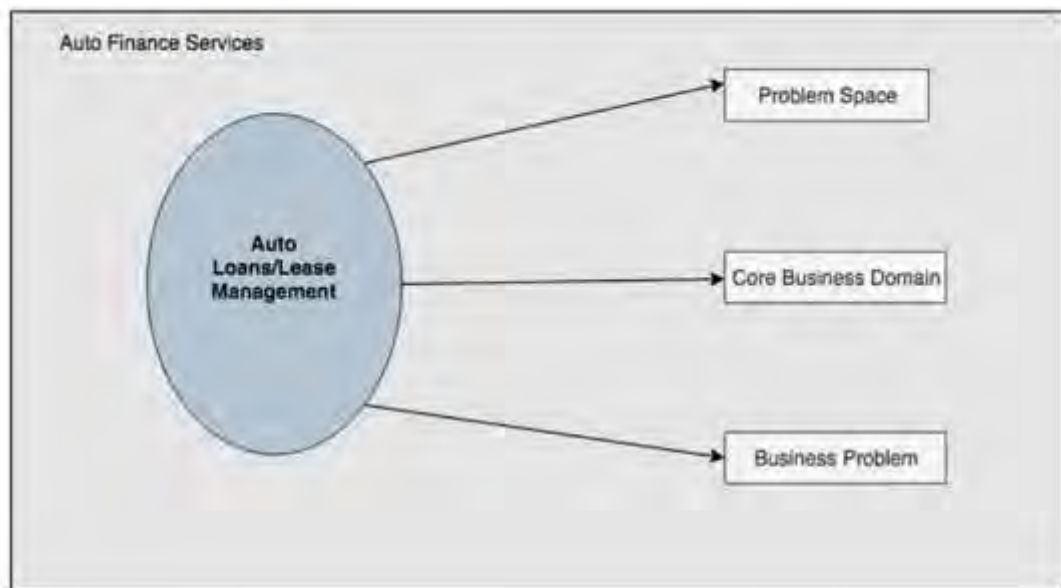
Gambar 2.5. *Prototyping storyboard*
(<https://builduniversity.org/prototype/>)

2.3. *Domain Driven Design*

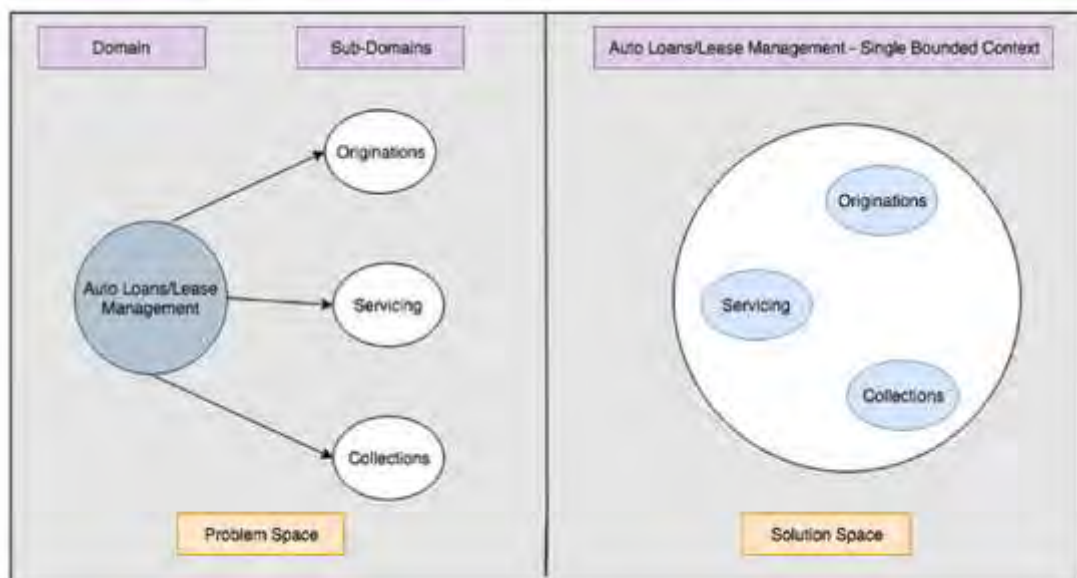
Domain Driven Design atau disingkat dengan *DDD* merupakan salah satu pendekatan yang digunakan untuk merancang sistem informasi. Metode *DDD* ciri utama yaitu dekomposisi domain utama menjadi beberapa *sub-domain*. Selain itu, *DDD* dapat diterapkan ke berbagai arsitektur sistem informasi atau siklus pengembangan. Pada penelitian ini pembahasan tentang *DDD* dimulai dari konsep hingga implementasi *DDD* pada *arsitektur microservices*.

2.3.1. *Konsep Domain Driven Design*

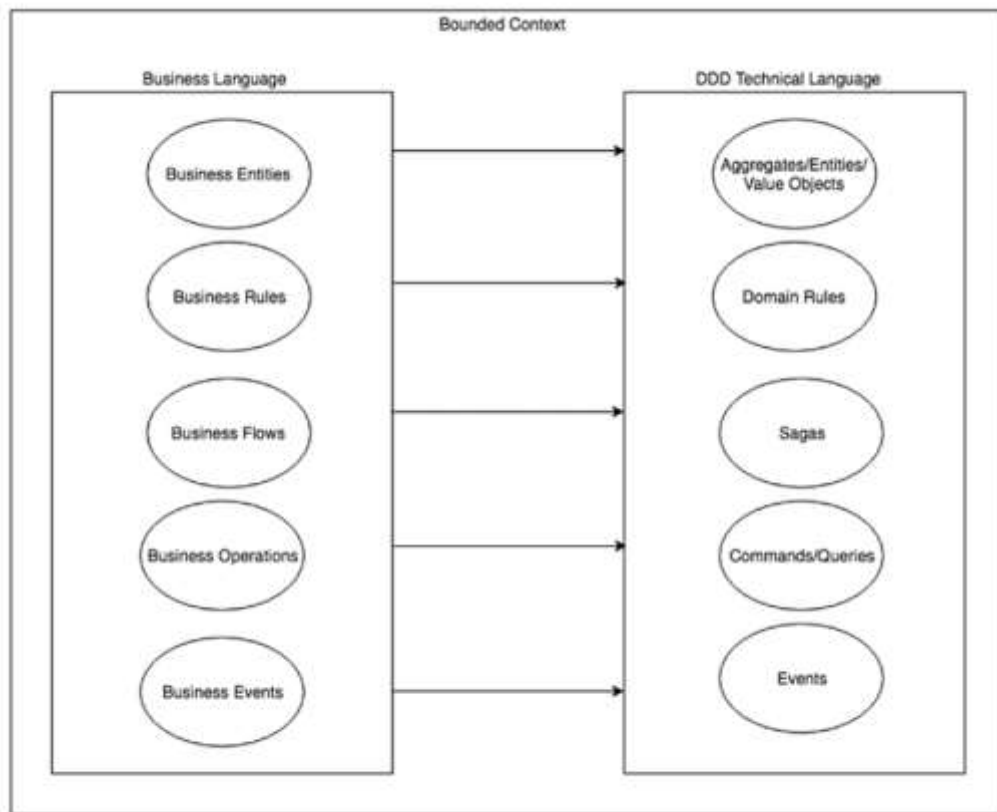
Domain Driven Design atau dapat disingkat menjadi *DDD* adalah pendekatan yang dilakukan pada pengembangan sistem informasi yang berpusat pada domain. *Domain* merupakan permasalahan yang dibuatkan solusi dalam sistem informasi. (Roland H. Steinegger, dkk, 2017). Fokus pada *DDD* terdapat pada mengurai kompleksitas sistem informasi berdasarkan *domain*. *Domain* adalah bisnis utama atau masalah yang perlu dipecahkan. *DDD* mengidentifikasi permasalahan (*problem space*), selanjutnya menentukan *sub-domain* dari *domain* permasalahan yang sudah diidentifikasi sebelumnya. *Sub-domain* merupakan dekomposisi dari *domain* utama (Nair, Vijay. 2019). Pada gambar 2.7 dan 2.8, dijelaskan contoh dari *problem space* menggunakan studi kasus pembiayaan kendaraan (*Auto loans/lease management*). *Domain* utama adalah manajemen pembiayaan kendaraan, di dekomposisi menjadi 3 *sub-domain* yaitu *Originations*, *Servicing* dan *Collections*.



Gambar 2.6. Identifikasi bisnis utama sebagai Problem Space, Core Business Domain atau Business Problem (Nair, Vijay. 2019)



Gambar 2.7. Identifikasi *sub-domain* sebagai dekomposisi dari *domain* utama (bisnis utama) (Nair, Vijay. 2019)

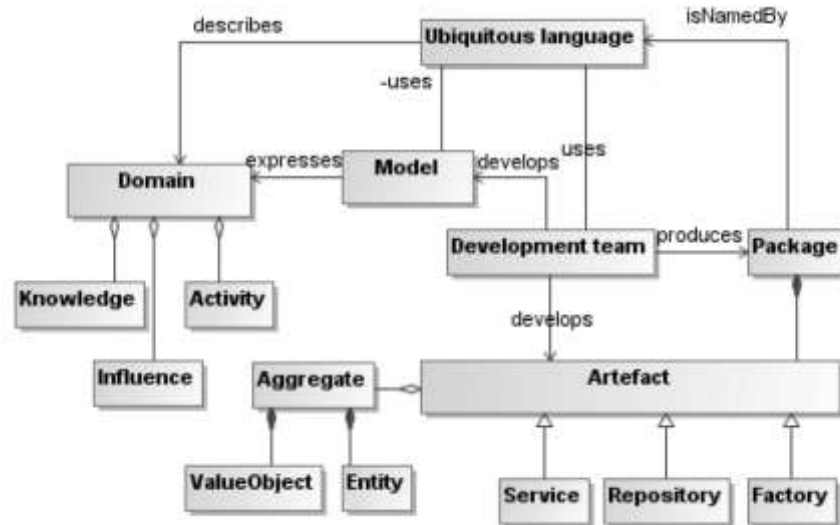


Gambar 2.8. Konsep *bounded context* pada model *domain* (Nair, Vijay. 2019)

Gambar 2.8 menjelaskan *solution space* yang sudah diidentifikasi akan menjadi *bounded context* pada model *domain*. Menurut Vijay Nair, *bounded context* terdiri dari *Business Entities*, *Business rules*, *Business flows*, *Business operations* dan *Business events*. Aspek bisnis tersebut di petakan menjadi bahasa teknis pada DDD menjadi *Aggregates*, *Domain rules*, *Sagas*, *Command/Queries* dan *Events*. *Bounded context* meliputi bisnis logik dari *sub-domain*.

Menurut Paulius dan Gintautas, konsep DDD yang dijelaskan pada gambar 2.9 menggambarkan hubungan antar komponen DDD. *Domain* merupakan subjek dari aplikasi yang mewakili aktivitas, proses bisnis, masalah, pengaruh atau peristiwa. Model merupakan abstraksi dari *domain* tertentu dan dapat digunakan untuk mengembangkan solusi tentang *domain* tersebut. *Ubiquitous language* adalah bahasa atau kata yang digunakan oleh tim pengembang untuk menghubungkan *domain* model dengan aspek

teknis (modul, packages, komponen aplikasi, user interface) (D. Paulius, G. Gintautas, 2018).



Gambar 2.9. Konsep Domain Driven Design (D. Paulius, G. Gintautas, 2018)

Beberapa model pada DDD terdiri dari *Associations*, *Entities*, *Value objects*, *Services* dan *Packages*. Pada tabel 2.1 dijelaskan mengenai model - model tersebut. Kemudian, pengelolaan *domain* pada DDD melalui pola (*pattern*) yang diadaptasi dari pemrograman berorientasi objek. Pola tersebut antara lain *Factories* dan *Repositories*. Sebelum dapat menggunakan pola tersebut, ada tahapan pengelompokan objek (*domain*) untuk dapat memperjelas lingkup dari *domain* tersebut. Tahapan tersebut adalah *Aggregate*. Tahapan *aggregate* sangat penting dilakukan untuk menjaga integritas *domain*. Pada tabel 2.2 dijelaskan mengenai pola *Factories* dan *Repositories* serta penjelasan mengenai *Aggregate* (E, Evans. 83 – 135:2003)

Tabel 2.1. Model pada Domain Driven Design

Model	Deskripsi
<i>Associations</i>	Hubungan antara objek. Pada umumnya dikenali dengan <i>one-to-many</i> , <i>one-to-one</i> dan <i>many-to-many</i> . Pemodelan asosiasi pada perancangan memungkinkan terjadi perbedaan implementasi khususnya pada bahasa pemrograman dan <i>database</i> .
<i>Entities</i>	Abstraksi yang mewakili atau menggambarkan sebuah atau beberapa objek. Objek di kenali melalui perilaku dan atribut yang ada pada objek tersebut. Objek yang mempunyai perilaku secara berkelanjutan dan terbatas pada konteks <i>domain</i> maka

Model	Deskripsi
	dapat dimasukkan sebagai <i>entity</i> . Entitas dapat berupa peran, peristiwa, proses ataupun sistem lain.
<i>Value objects</i>	Perbedaan mendasar dengan entitas, jenis objek ini tidak mempunyai perilaku tetapi hanya mempunyai atribut. Sebagai contoh harga suatu produk, alamat pengguna.
<i>Services</i>	Suatu abstraksi dari aktivitas yang menghubungkan antara entitas. <i>Service</i> dapat dikelompokkan sebagai objek.
<i>Packages</i>	Model ini merepresentasikan <i>domain</i> dan pengelompokannya.

Tabel 2.2. Pattern model *domain driven design*

Patterns	Deskripsi
<i>Aggregates</i>	Tahapan untuk mengelompokan objek (<i>domain</i>) yang mempunyai perilaku dan atribut yang sama. Selain itu, <i>aggregate</i> menentukan <i>sub-domain</i> yang mempunyai kesamaan yang merupakan dekomposisi dari <i>domain</i> utama.
<i>Factories</i>	Pola ini adalah implementasi dari <i>Factory Pattern</i> pada konsep <i>Object Oriented</i> . <i>Factory</i> membuat objek <i>domain</i> yang enkapsulasi proses logik dan mengurangi dependensi terhadap <i>class</i> utama.
<i>Repositories</i>	Suatu objek yang merepresentasikan storage (penyimpanan) dari objek <i>domain</i> . Secara teknis pada implementasi code, repository harus tetap tersedia (<i>persistence</i>) atau dapat diakses langsung

Sumber: E, Evans. 2003

Pada gambar 2.9, fokus utama pada DDD adalah tim pengembangan (*development team*) membuat artefact dari DDD. Artefact tersebut merupakan wujud atau implementasi dari model *domain* yang terdiri dari konsep utama dan pola (*pattern*) objek oriented. Implementasi dapat dijelaskan sebagai konseptual ataupun teknis. Pada tabel 2.3, dijelaskan mengenai *artefacts* dari *domain driven design*.

Tabel 2.3. Artefacts *Domain Driven Design*

Model	Deskripsi
<i>Entity</i>	Abstraksi yang mewakili atau menggambarkan sebuah atau beberapa objek. Objek di kenali melalui perilaku dan atribut yang ada pada objek tersebut. Objek yang mempunyai perilaku secara berkelanjutan dan terbatas pada konteks <i>domain</i> maka dapat dimasukkan sebagai <i>entity</i> . Entitas dapat berupa peran, peristiwa, proses ataupun sistem lain.
<i>Value object</i>	Perbedaan mendasar dengan entitas, jenis objek ini tidak mempunyai perilaku tetapi hanya mempunyai atribut. Sebagai contoh harga suatu produk, alamat pengguna.

Model	Deskripsi
<i>Aggregates</i>	Sekumpulan entitas yang terhubung dengan entitas utama (<i>root entity</i>).
<i>Service</i>	Suatu abstraksi dari aktivitas yang menghubungkan antara entitas. <i>Service</i> dapat dikelompokkan sebagai objek.
<i>Repository</i>	Suatu objek yang merepresentasikan storage (penyimpanan) dari objek <i>domain</i> . Secara teknis pada implementasi <i>code</i> , <i>repository</i> harus tetap tersedia (<i>persistence</i>) atau dapat diakses langsung
<i>Factory</i>	Model ini adalah implementasi dari <i>Factory Pattern</i> pada konsep <i>Object Oriented</i> . <i>Factory</i> membuat objek <i>domain</i> yang mengenkapsulasi proses logik dan mengurangi dependensi terhadap <i>class</i> utama.
<i>Packages (modules)</i>	Model ini merepresentasikan <i>domain</i> dan pengelompokannya.

Sumber: D. Paulius, G. Gintautas, 2018

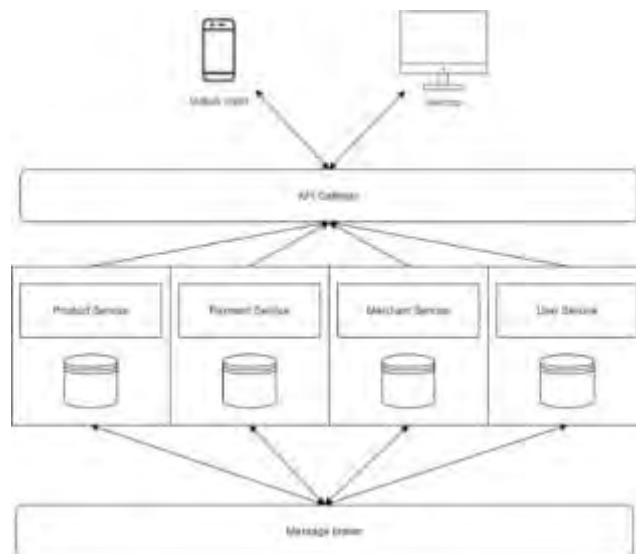
Sementara itu, terdapat konsep pendukung dari DDD antara lain, *Bounded context (BC)*, *Domain events* dan *Inter-BC interaction*. *Bounded context (BC)* adalah strategi untuk memisahkan antara *aggregate* (kelompok *domain*) untuk mengerjakan fungsionalitas yang lebih spesifik. Pada BC, bisa saja terdapat objek atau *domain* yang saling beririsan antar kelompok. *Domain events* adalah proses atau aktivitas – aktivitas yang menjadi penghubung antar *domain*. *Inter-BC interaction* adalah cara setiap *bounded context* berkomunikasi. *Domain event* dan *Inter-BC interaction* selain mendefinisikan model juga sudah menetapkan teknologi yang digunakan untuk menjalin komunikasi antar *domain* atau bounded contex (Roland H. Steinegger, dkk, 2017).

DDD sangat memungkinkan untuk menerapkan siklus pengembangan sistem informasi apapun. Pada umumnya DDD menerapkan pengembangan sistem informasi dengan pola Perencanaan, Analisis, Perancangan dan Implementasi. Pada DDD terdapat spesifikasi pada proses pengembangan sistem informasi, spesifikasi proses tersebut antara lain: klasifikasi, deskripsi aktivitas, analisis kebutuhan sistem informasi, perancangan, implementasi dan pengujian. Sementara itu, DDD pada dasarnya tidak langsung membahas aspek teknis seperti implementasi teknologi, bahasa pemrograman, kode bahkan *design pattern*. DDD akan menjadi lebih bermakna apabila permasalahan bisnis dapat dipahami terlebih dahulu. *Domain* permasalahan akan menjadi bahasa atau

istilah yang sama digunakan pada sisi bisnis dan sisi teknologi (Roland H. Steinegger, dkk, 2017).

2.3.2. *Arsitektur Microservice*

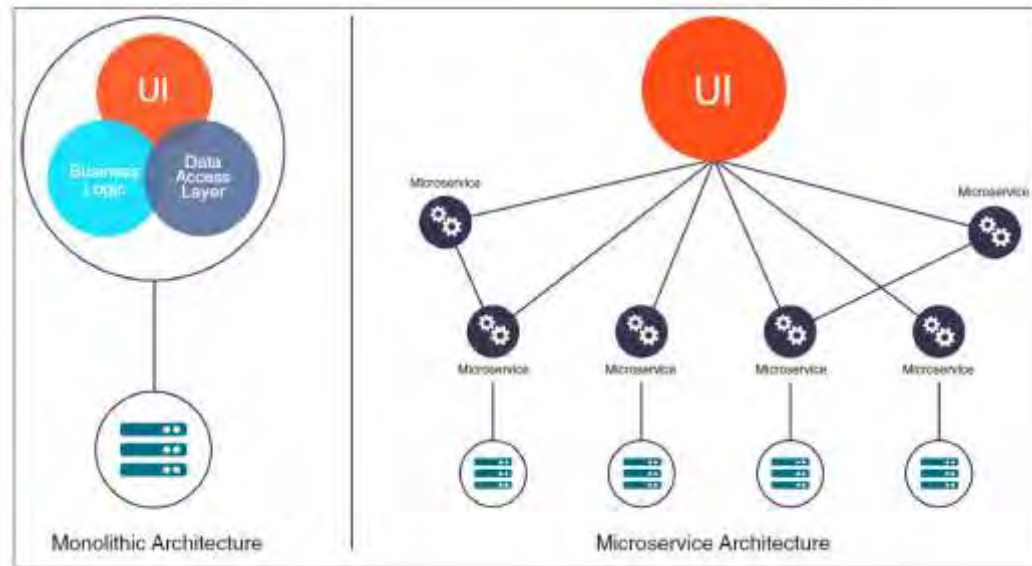
Pembahasan terkini mengenai *Domain Driven Design* berkaitan dengan arsitektur sistem informasi *Microservice*. Definisi *microservice* adalah pendekatan atau paradigma dalam pengembangan suatu aplikasi dimana setiap bagian aplikasi dijadikan unit *service* kecil yang berjalan pada runtime-nya sendiri (Darren Gallipeau dan Sara Kudrle, 2018). Setiap *service* mempunyai *minimum viable* yang berjalan pada jaringan komputer (network) dan mempunyai nilai bisnis yang terukur. *Microservice* adalah upaya merancang suatu sistem informasi dengan pendekatan yang lebih detail dari awal pengembangan. *Domain Driven Design* membantu dalam hal detail tersebut yang terdapat pada komponen domain.



Gambar 2.10. *Arsitektur Microservice* hasil dari pemodelan *Domain Driven Design* (Sam Newman, Building Microservice, 2015)

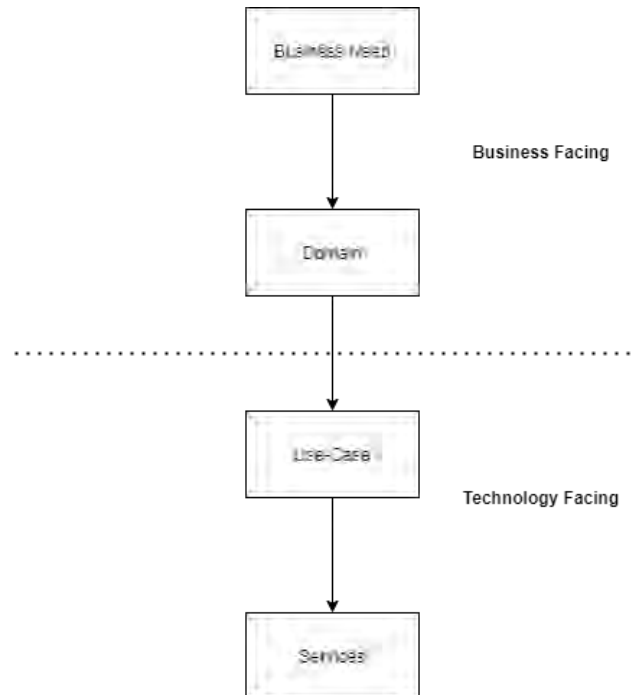
Apabila memperhatikan perkembangan dari arsitektur sistem informasi maka didapatkan arsitektur yang dikenal dengan nama Monolitik. Arsitektur ini bersifat spesifik terhadap bisnis, setiap fungsional saling berkaitan erat (*tightly integrated*) dan terbatas pada kebutuhan (*limited and isolated needs*). Arsitektur ini menjadi kurang tepat digunakan apabila menghadapi perkembangan teknologi dan bisnis yang cepat. Isu tentang adaptasi teknologi, aplikasi yang harus segera digunakan (*ready-to-market*),

kapasitas dan kemampuan sistem menjadi sangat besar. Pada arsitektur monolitik apabila ada kebutuhan untuk perubahan akan membutuhkan pengembangan baru bahkan hingga ubahan pada *source-code*.



Gambar II.11. Perbandingan Arsitektur Monolitik dengan Arsitektur *Microservice* (Darren Gallipeau and Sara Kudrle, 2018)

Arsitektur microservice memberikan solusi dalam memudahkan perancangan (*design*), pengembangan (*develop*), pemeliharaan (*maintain*) dan migrasi (*deploy*). Selain itu, *microservice* memberikan pendekatan baru dalam pengembangan sistem dan jaringan komputer (*network*). Kemudian, isu kapasitas dan kemampuan sistem (*scalability*) dan pemanfaatan kembali fungsi, modul bahkan *source-code* (*reusability*) dapat ditangani. *Domain* bisnis sistem informasi pada arsitektur *microservice* menjadi jelas batasannya mulai dari analisis hingga tahapan implementasi kode (*firm module boundary*).



Gambar 2.12. Transisi metode DDD kepada Arsitektur Microservice(Roland H. Steinegger, Pascal Giessler, Benjamin Hippchen dan Sebastian Abeck, 2017)

Pada gambar 2.12 dijelaskan mengenai transisi dari metode DDD kepada *arsitektur microservice*. Transisi ini terbagi pada 2 (dua) sudut pandang yaitu Bisnis dan Teknologi. Pada sudut pandang bisnis, terdapat 2 aspek utama yaitu kebutuhan bisnis yang diterjemahkan menjadi *domain*. Selanjutnya, *domain* menuju sudut pandang teknologi menjadi use-case sistem informasi selanjutnya menjadi *service – service* pada *arsitektur microservice*. Kriteria ideal dari *service* (Modelling Microservice, 64: 2015) antara lain :

1. *Loose Coupling*

Loose coupling bermakna longgar dari saling ketergantungan antara *service*. Ketergantungan tersebut akan membuat *service* itu sendiri menjadi rumit untuk dikembangkan. *Service* yang sudah terbebas dari ketergantungan tidak akan mengakibatkan perubahan pada *service* lain apabila terjadi perubahan itu sendiri. Sementara itu, apabila diperlukan dapat dibatasi jumlah pemanggilan *service* dari satu *service* ke *service* yang lain.

2. *High Cohesion*

Setiap *service* menjalankan fungsionalitas yang terbatas pada *domain* permasalahan.

Hal ini memastikan proses dalam *service* tersebut dapat berjalan secara terpadu.

2.3.3. Integrasi antara *Service*

Pada pembahasan sebelumnya diketahui bahwa *domain* pada metode DDD merupakan bisnis dari sistem informasi. Pada gambar 2.12 terdapat gambaran mengenai transisi *domain* menjadi *service* pada *arsitektur microservice*. Setiap *service* harus memenuhi kriteria ideal yaitu *loose coupling* dan *high cohesion*. Kriteria ideal tersebut didukung dengan pendekatan integrasi yang diterapkan. Integrasi antara *service* menjadi kunci utama dari proses yang terjadi pada *arsitektur microservice*. Kondisi ideal dari integrasi tersebut (Modelling *Microservice*, 81: 2015) dapat memenuhi kriteria sebagai berikut :

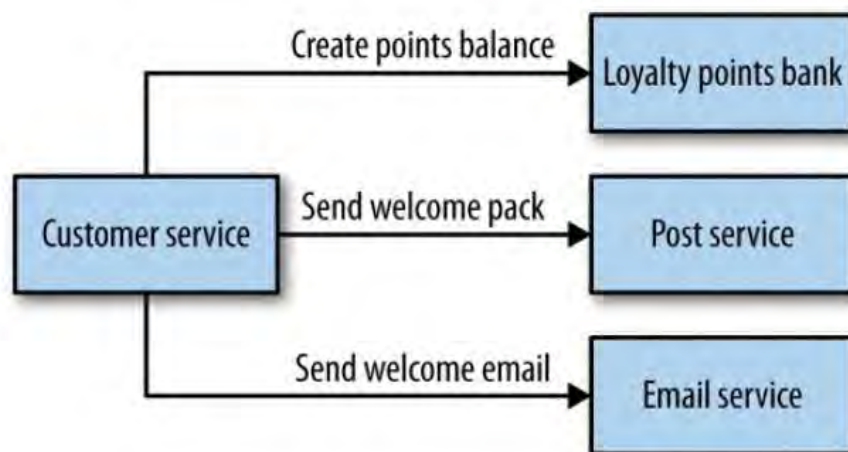
1. Menghindari perubahan yang signifikan terutama pada fungsionalitas *service*
2. Implementasi teknologi pada antarmuka *service* (API) harus berprinsip bebas teknologi
3. Membuat *service* dengan proses yang sederhana
4. Detail dari *service* yang disembunyikan untuk mengurangi kompleksitas pada saat integrasi

Kemudian, kondisi ideal tersebut diwujudkan dengan integrasi antara *service*. Metode atau teknik yang digunakan untuk membuat setiap *service* terintegrasi dikelompokkan menjadi 2 (dua) yaitu *Asynchronous* dan *Synchronous*. Definisi metode *synchronous* adalah apabila terjadi pemanggilan *service* ke *service* lain, maka *service* pemanggil akan menunggu hingga proses selesai dijalankan dan *service* tujuan memberikan response yang dibutuhkan. Pada metode *asynchronous*, apabila terjadi pemanggilan *service* ke *service* lain, maka *service* pemanggil tidak perlu menunggu hingga proses selesai untuk dapat menerima response. Kondisi proses selesai tidak perlu diketahui berhasil atau gagal. Metode *synchronous* atau dapat disebut sebagai model

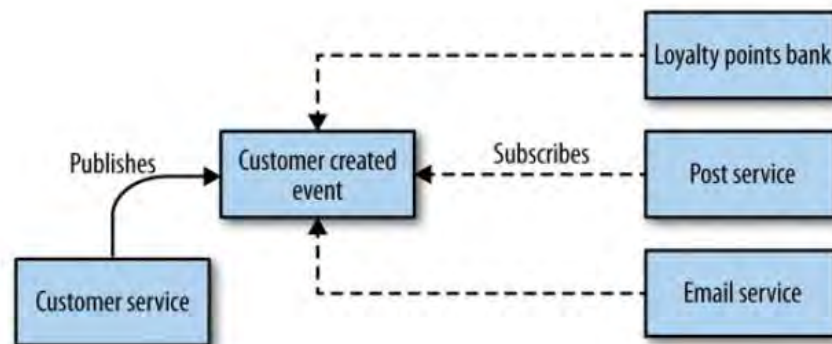
komunikasi request/response. Metode *asynchronous* atau dapat disebut sebagai model komunikasi *event* based.

Model komunikasi request/response merupakan model komunikasi yang paling umum digunakan pada integrasi *service* baik pada *arsitektur Microservice* atau bahkan monolitik. Model komunikasi request/response terdapat peran sebagai client dan *server*. *Service* yang membuat request sebagai client dan *service* yang menerima request dan memberikan response adalah *server*. *Service client* perlu menunggu beberapa waktu hingga *service* memberikan *response*.

Model komunikasi *event*-based merupakan model komunikasi yang memisahkan atau mendistribusikan logik bisnis pada masing – masing *service*. Setiap *service* berkomunikasi pada identitas *event* yang dikenali satu sama lain. Pada model ini, terdapat peran *publisher* dan *subscriber*. *Publisher* merupakan *service* yang memberikan informasi pada identitas *event* yang disepakati, sedangkan *Subscriber* merupakan *service* yang menerima informasi pada identitas *event* yang disepakati. Identitas *event* tersebut merupakan kunci komunikasi dari model komunikasi *event*-based. Pada implementasinya *publisher* dan *subscriber* akan mengakses satu *service* dengan nama *event bus*. *Event bus* ini berisi identitas *event* yang disepakati antara *publisher* dan *subscriber*. Gambar 2.13 dan 2.14 menunjukkan komunikasi *service* melalui *request/response* dan *event publish/subscribe*.

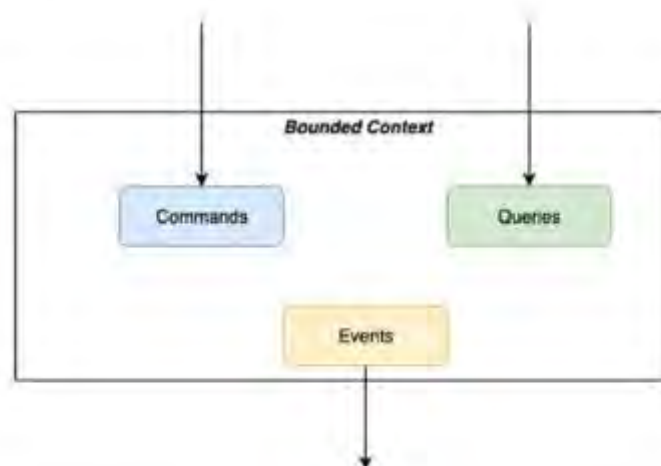


Gambar 2.13. Model komunikasi *Request/Response* (Sam Newman, 91: 2015)

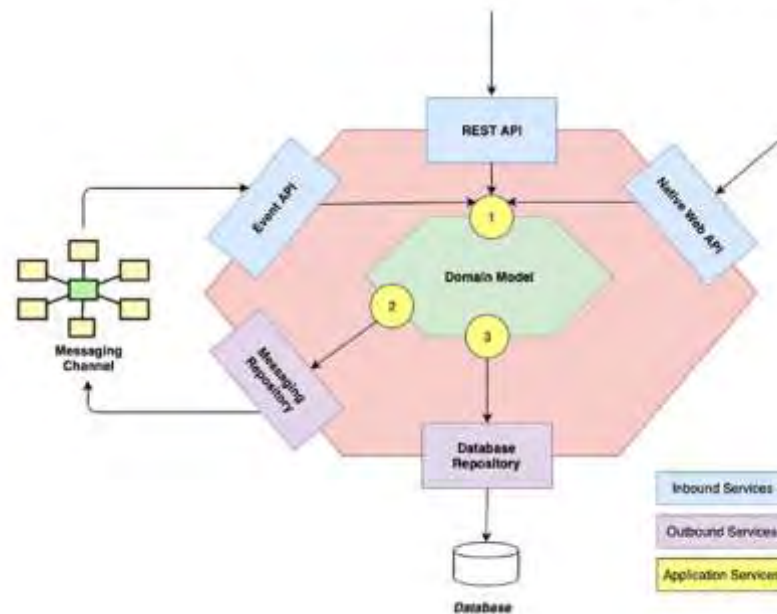


Gambar 2.14. Model komunikasi *Event-based* (Sam Newman, 92: 2015)

Service pada *arsitektur microservice* merupakan implementasi dari *bounded context* yang ada pada *model domain*. Suatu *bounded context* mempunyai masukan *Command* dan *Queries* serta keluarannya adalah *Event* (Nair, Vijay. 29:2019). *Command* adalah proses untuk menambahkan atau mengubah status pada *bounded context*. *Queries* adalah proses untuk meminta status pada *bounded context*. *Event* adalah proses untuk memberitahu perubahan status yang terjadi pada *bounded context*. Apabila sudah dalam *arsitektur Microservice*, maka masukan menjadi *Inbound service*, *bounded context* menjadi *Application service*, keluaran menjadi *Outbound service*. Gambar 2.15 menunjukkan operasi pada *bounded context*.



Gambar 2.15. Operasi sistematis pada *bounded context* (Nair, Vijay. 2019)



Gambar 2.16. *Hexagonal Architecture* (Nair, Vijay. 2019)

Service pada *arsitektur Microservice* dapat dimodelkan dengan *hexagonal architecture*. Gambar 2.16 menunjukkan representasi *hexagonal architecture*. Pada model ini, sisi atas (*inbound port*) menjadi antarmuka untuk *Inbound services*, sisi bawah (*outbound port*) menjadi antarmuka untuk *Outbound services*. *Inbound port* adalah antarmuka untuk operasi - operasi bisnis dari *Application service*. *Outbound port* adalah antarmuka untuk operasi – operasi teknis dari *Application service*. *Inbound port* dapat digunakan oleh pihak lain (*external clients*) untuk mengakses *application service*. Pihak lain dapat mengakses *application service* melalui *REST API*, *Event API* atau *Native Web API*.

2.4. UML (Unified Modelling Language)

UML adalah bagian dari notasi atau simbolasi dari pemodelan sistem informasi berorientasi objek. Pengembangan sistem informasi berorientasi dapat dimodelkan menggunakan notasi apa saja, namun dengan UML akan membuat standar notasi sehingga dapat dengan mudah dipahami oleh setiap peran dalam pengembangan sistem informasi mulai dari sistem analisis, *software developer*, *software tester* bahkan hingga proyek manajer.

UML mempunyai fungsi utama sebagai bahasa pemodelan yang digunakan untuk analisis dan perancangan sistem informasi. UML diperkenalkan oleh Booch, Rumbaugh dan Jacobson pada pertengahan tahun 1990 di Rational Software Corporation. UML dibagi menjadi 3 (tiga) jenis diagram yaitu : *Structure, Behavior dan Interaction* (Dennis A, 25:2015), menyatakan bahwa definisi dari masing – masing diagram tersebut antara lain :

1. Structure Diagram

Digunakan untuk memodelkan struktur fisik dari sistem informasi. Bentuk fisik tersebut berdasarkan organisasi arsitektur sistem, elemen fisik sistem, konfigurasi waktu proses atau *domain* bisnis. Diagram ini terdiri dari : *Package Diagram, Class Diagram, Component Diagram, Deployment Diagram, Object Diagram* dan *Composite Structure Diagram*

2. Behaviour Diagram

Digunakan untuk memodelkan setiap peristiwa atau perilaku setiap objek. Sebagai contoh, objek dapat dibuat atau dihapus, objek dapat mengirim pesan ke objek lain atau bahkan dipicu dari sistem lain untuk mengoperasikan objek dalam sistem. Diagram ini terdiri dari : *Use-case Diagram, Activity Diagram* dan *State Machine Diagram*

3. Interaction Diagram

Bagian dari Behavior Diagram namun dengan spesifikasi pemodelan yang lebih spesifik dengan melibatkan teknologi yang digunakan dalam interaksi antara objek. Diagram ini terdiri dari : *Sequence Diagram, Communication Diagram, Interaction Overview Diagram* dan *Timing Diagram*

Suatu sistem informasi terdiri dari aspek Konsep, Logik dan Fisik. Pada aspek konsep, pemodelan difokuskan pada entitas *domain* (bisnis) yang ada dan keterhubungannya dengan entitas lain. Aspek konsep sebaiknya terbebas dari pengaruh teknologi apa yang harus digunakan karena masih membahas mengenai *domain* bisnis. Selanjutnya, aspek logik, pemodelan dilakukan dengan tujuan membentuk model konsep kedalam bentuk abstraksi yang lebih jelas dan semua mekanismenya hingga menjadi arsitektur sistem informasi. Kemudian, aspek fisik, pemodelan difokuskan pada teknologi

apa saja yang digunakan seperti bahasa pemrograman dan konfigurasi perangkat keras yang akan digunakan pada sistem informasi.

2.4.1. Notasi Use-Case Diagram

Notasi use-case diagram digunakan untuk mendokumentasikan kebutuhan bisnis serta menjadi jembatan antara tim analisis dan tim pengembangan untuk dapat memahami kebutuhan bisnis tersebut sehingga terjadi keselarasan pemahaman baik sudut pandang bisnis maupun teknis. Use-case diagram menggambarkan konteks dari sistem yang akan dibuat serta fungsionalitas sistem-nya. Digambarkan pula siapa saja yang berinteraksi dengan sistem.

Pada use-case diagram terdapat penggambaran aktor dan use-case. Aktor merupakan peran baik jabatan pada organisasi ataupun sistem lain yang berinteraksi dengan sistem pada konteksnya. Sementara itu, use-case menggambarkan apa saja yang dapat dilakukan sistem sesuai harapan dari aktor tersebut.. Use-case dapat berupa proses ataupun peristiwa yang terjadi pada sistem. (Dennis A, 142: 2015). Pada tabel 2.4 dijelaskan tentang simbol – simbol use-case yang digunakan pada penelitian ini.

Tabel 2.4. Deskripsi simbol use-case

Simbol	Deskripsi
 Actor	Aktor merupakan peran dalam sistem yang mengeksekusi use-case dengan cara meminta atau mengirimkan data.
 Use Case	Simbol use-case yang menggambarkan bisnis yang penting dalam sistem
	Simbol asosiasi yang menghubungkan aktor dengan use-case.

2.4.2. Notasi Class Diagram

Notasi *class* diagram digunakan untuk menggambarkan secara spesifik objek yang terlibat pada sistem. Objek merupakan representasi dari sesuatu di dunia nyata. *Class diagram* dapat menggambarkan *domain* (masalah), *user-interfaces*, struktur data, struktur berkas (*file structure*), *operating environment*, dokumen, dan segala macam tipe – tipe multimedia lainnya. Pada *class* diagram didefinisikan atribut dari objek. Atribut ini merupakan penanda dari objek tersebut pada saat berinteraksi dalam suatu proses bisnis.

Selain atribut terdapat behavior atau perilaku dari objek. (Dennis A, 165:2015). Pada tabel 2.5 dijelaskan tentang simbol – simbol *class* diagram yang digunakan pada penelitian ini.

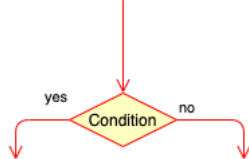
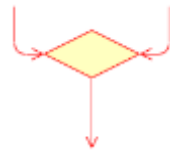
Tabel 2.5. Deskripsi simbol *class* diagram

Simbol	Deskripsi
Classname + field: type + method(type): type	Simbol <i>class</i> diagram yang terdiri dari nama <i>class</i> , atribut dan metod.
	Simbol asosiasi antar <i>class</i> dengan tambahan informasi hubungan kardinalitas.

2.4.3. Notasi Activity Diagram

Notasi activity diagram digunakan untuk menggambarkan proses bisnis pada suatu sistem. Proses bisnis terdiri dari beberapa aktivitas yang dikombinasikan sehingga untuk mengolah suatu input menjadi output yang ditentukan. Activity diagram disesuaikan dengan kebutuhan sistem. Sementara itu, diagram aktivitas digunakan juga untuk memodelkan perilaku pada suatu proses bisnis yang tidak terkait dengan objek. Diagram aktivitas dapat memodelkan mulai dari gambaran bisnis secara umum hingga spesifik pada masing -masing use-case. (Dennis A, 130: 2015). Pada tabel 2.6 dijelaskan tentang simbol – simbol activity diagram yang digunakan pada penelitian ini.

Tabel 2.6. Deskripsi simbol activity diagram

Simbol	Deskripsi
	Simbol untuk menandakan permulaan proses bisnis
	Simbol untuk menandakan akhir proses bisnis
	Simbol untuk mewakili aktivitas yang terjadi pada proses bisnis
	Simbol (decision) untuk menggambarkan kondisi yang memenuhi syarat
	Simbol untuk menggabungkan kembali alur proses yang terpecah dari simbol kondisi (<i>decision</i>)

2.5. Penelitian Terkait

Penelitian ini merupakan bagian dari perkembangan dari penelitian – penelitian sejenis yang sudah ada sebelumnya. Penulis menggunakan beberapa referensi penelitian yang membahas topik mengenai *Domain Driven Design*, *Arsitektur Microservice*, *e-learning* dan perancangan sistem informasi. Selain daripada topik tersebut, penulis mempertimbangkan pendekatan serta metodologi yang dilakukan pada penelitian – penelitian terkait. Adapun penjelasan daripada hal tersebut terdapat pada tabel 2.4.

Tabel 2.7. Pembahasan tentang penelitian terkait

No	Judul	Penulis / Instansi	Tahun	Pembahasan
1.	<i>Does applying Design Thinking result in better new product concepts?</i>	Martin Meinel, dkk	2020	Perbandingan dua pendekatan user preference-driven dan user experience-driven. Membuktikan hipotesis tentang DT menghasilkan product yang lebih baik
2.	<i>Implementasi user experience menggunakan metode Design Thinking pada prototype aplikasi cleanstic</i>	Gusti Karnawan, dkk	2020	Menunjukkan langkah - langkah sistematis dari penerapan <i>Design Thinking</i> pada suatu studi kasus pengembangan aplikasi Cleanstic
3.	Penerapan metode <i>Design Thinking</i> pada model perancangan UI/UX aplikasi penanganan laporan kehilangan dan temuan barang tercecer	Aria Ar Razi, dkk	2018	Membuat model perancangan UI/UX pada aplikasi mobile yang dirancang khusus untuk kasus kehilangan dan temuan barang. Panduan model UX dengan Hooked Model : <i>Triggers, Actions, Rewards</i> .
4.	<i>Microservices: Building Blocks to New Workflows and Virtualization</i>	Darren Gallipaeu dan Sara Kudrle. SMPTE Motion Imaging Journal	2018	Penelitian ini merupakan jenis technical paper. Pada penelitian dibahas mengenai perkembangan arsitektur dari monolitik ke <i>Microservice</i> , definisi <i>Microservice</i> , perbandingan <i>microservice</i> dengan monolitik, manfaat <i>microservice</i> serta implementasi <i>microservice</i> pada infrastruktur virtualisasi.
5.	Implementasi Teknologi <i>Microservice</i> pada Pengembangan Mobile Learning	Maksy Sendiang, Sonny Kasenda, Jerry Purnama	2018	Penelitian ini membahas tentang pengembangan aplikasi <i>e-learning</i> berbasis mobile dengan penerapan <i>microservice</i> . Pada penelitian ini dibahas mengenai definisi mobile <i>e-learning</i> , definisi dan gambaran

No	Judul	Penulis / Instansi	Tahun	Pembahasan
				<i>microservice</i> , pemodelan sistem menggunakan use-case diagram, metode penelitian menggunakan RUP (Rational Unified Process), arsitektur perangkat lunak menggunakan MVC (<i>Model-View-Controller</i>) serta pengujian sistem.
6.	Analisis Model Arsitektur <i>Microservice</i> pada Sistem Informasi DPLK	Ghifari Munawar, Ade Hodijah	2018	Penelitian ini membahas tentang analisis arsitektur <i>microservice</i> terhadap sistem informasi DPLK. Analisis dilakukan dengan metode kandidat <i>microservice</i>. Selain itu penelitian berdasarkan keberhasilan migrasi arsitektur monolitik kepada arsitektur <i>microservice</i> pada sistem informasi sebelumnya. Dijelaskan pula mengenai tahapan migrasi <i>microservice</i>. Tahapan tersebut antara lain : 1. Pemetaan tabel di <i>database</i> 2. Buat grafik ketergantungan <i>domain</i> 3. Identifikasi pasangan <i>domain</i> 4. Pilih pasangan <i>domain</i> 5. Identifikasi kandidat <i>microservice</i> 6. Buat gateway API untuk setiap <i>microservice</i>
7.	<i>Coupling Design Thinking, User Experience and Agile</i>	Galia Novakova, Elena Shoikova	2017	Meneliti perpaduan antara konsep DT, UX dan Agile untuk mendapatkan cara baru yang lebih efektif dan efisien dalam mengembangkan sistem informasi. Penerapan aktivitas di setiap iterasi Agile dipadukan dengan DT Integrasi konsep
8.	<i>Overview of a Domain-Driven Design Approach to Build Microservice-Based Applications</i>	Rolan H. Steinegger, Pascal Giessler, Benjamin Hippchen dan Sebastian Abeck	2017	Penelitian ini membahas tentang adopsi <i>domain driven design</i> pada proses – proses pengembangan perangkat lunak baik pada tahapan umum pengembangan hingga rancangan arsitektur perangkat lunak. Kemudian, pembahasan tentang arsitektur <i>microservice</i> dengan pendekatan <i>backend-for-frontend</i> , model komunikasi antara komponen pada arsitektur <i>microservice</i> , definisi <i>microservice</i> . Selain itu, dibahas pula mengenai proses – proses pengembangan perangkat lunak mulai dari klasifikasi, pengumpulan kebutuhan sistem dan analisis, perancangan sistem hingga implementasi dan pengujian.
9.	<i>Domain-Driven Design of</i>	Pavel P. Oleynik,	2015	Penelitian ini membahas tentang perancangan sistem informasi

No	Judul	Penulis / Instansi	Tahun	Pembahasan
	<i>Information System for Queuing System in Terms of Unified Metamodel of Object System</i>	Nikolay Kuznetsov		antrian menggunakan <i>domain-driven design</i> dan metamodel sistem objek. Metamodel objek dibuat berdasarkan ide dari penulis. Penelitian ini menggunakan tools SharpArchitect RAD Studio. Kemudian, pembahasan selanjutnya tentang definisi <i>domain-driven design</i> , kriteria maksimal untuk sistem informasi antrian. Selain itu, dibahas tentang pemodelan <i>domain</i> berdasarkan metamodel objek. Metomodel objek ditransalasi menjadi <i>metaclasses</i> .

BAB III

OBJEK DAN METODOLOGI PENELITIAN

3.1. *E-learning*

E-learning adalah sistem pendidikan yang memanfaatkan teknologi dan sistem informasi dalam penerapannya. *E-learning* yang akan dirancang pada penelitian ini mengadopsi komponen – komponen *e-learning*. Komponen tersebut merupakan proses bisnis atau fungsional dari sistem. Adapun komponen *e-learning* antara lain :

1. Konten atau materi ajar
2. Kelas virtual
3. Tutor atau pengajar
4. Kolaborasi

Setiap komponen *e-learning* bergantung pada dimensi waktu yang digunakan pada *e-learning* tersebut. Dimensi waktu yang digunakan antara lain *Synchronous* dan *Asynchronous*. Dimensi *synchronous* adalah waktu belajar secara langsung antara peserta ajar dan pengajarnya. Sedangkan dimensi *asynchronous* adalah peserta ajar dapat mengakses materi ajar kapan saja. Dengan constraint terhadap dimensi waktu tersebut, maka komponen *e-learning* harus menyesuaikan. Selanjutnya, setiap komponen akan menentukan proses bisnis atau use-case utama dari sistem ini. Pada tabel 3.1, dipetakan antara komponen dan use-case yang dibentuknya.

Tabel 3.1. Use-case *E-learning*

Komponen	Use-Case	Deskripsi
Konten atau Materi Ajar	Manajemen Konten (<i>Content Management</i>)	Use-case untuk pengelolaan materi ajar. Terdiri dari CRUD (Create-Retrieve-Update-Delete) Data materi ajar, Pencarian hingga Pengelompokan
Kelas Virtual	Manajemen Jadwal (<i>Schedule Management</i>)	Use-case untuk pengelolaan jadwal penyampaian materi ajar. Jenis jadwal tergantung dengan materi dan dimensi waktu yang digunakan.
Tutor atau Pengajar	Manajemen Tutor (<i>Tutor Management</i>)	Use-case untuk pengelolaan pengajar (tutor). Terdiri dari

Komponen	Use-Case	Deskripsi
		pendaftaran tutor, validasi tutor melalui <i>event</i> wawancara, test dan video mengajar, seleksi hingga penerimaan
Kolaborasi	Manajemen Kehadiran (Presence Management)	Use-case untuk pengelolaan kehadiran tutor dan peserta ajar. Use-case ini digunakan untuk materi ajar dan jadwal dengan dimenasi waktu <i>synchronous</i> .
	Pelacakan Kemajuan Belajar (learning progress tracker)	Use-case untuk pengelolaan kemajuan belajar peserta ajar. Use-case ini digunakan pada materi dan jadwal ajar dengan dimensi waktu <i>asynchronous</i> sehingga peserta ajar dapat mengetahui kemajuan belajar
	Manajemen Tugas (Assignment)	Use-case untuk pengelolaan tugas dari tutor kepada peserta ajar. Terdiri dari CRUD (Create-Retrieve-Update-Delete) tugas, <i>Assignment</i> tugas kepada peserta ajar secara langsung atau per jadwal.
	Manajemen Ujian (Examination)	Use-case untuk pengelolaan ujian dari tutor kepada peserta ajar. Terdiri dari CRUD (Create-Retrieve-Update-Delete) tugas, <i>Assignment</i> tugas kepada peserta ajar secara langsung atau per jadwal.

Dalam penelitian sistem informasi *e-learning* ini, penulis menggunakan 2 (dua) metode yaitu *Design Thinking* dan *Domain Driven Design*. Metode *Design Thinking* untuk menemukan user journey dan pain point dari *e-learning* yang sudah ada untuk kemudian ditemukan model sistem informasi yang tepat. Sementara itu, *domain driven design* adalah metode untuk memodelkan sistem informasi tersebut.

3.2. Metode Design Thinking

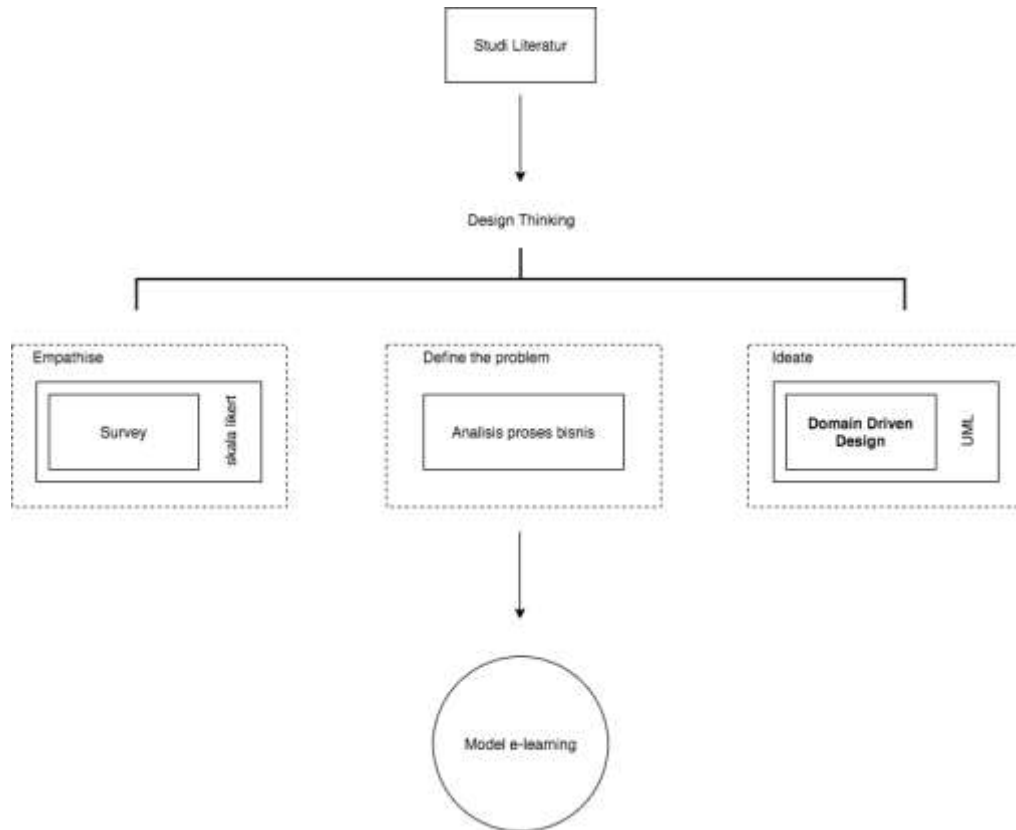
Design Thinking merupakan metode untuk menemukan solusi atas suatu permasalahan yang diadopsi dari beberapa disiplin ilmu seperti perancangan arsitektur, *user experience reSearch* dan psikologi. Fokus terhadap interaksi antara manusia dengan

sistem (*Human Centered*) apabila dibandingkan dengan siklus pengembangan sistem informasi pada umumnya.

Pada dasarnya setiap tahapan tersebut masih mempunyai keterkaitan dengan siklus pengembangan sistem informasi pada umumnya, yaitu Perencanaan, Analisis, Perancangan dan Implementasi. Namun, metode ini lebih dekat dengan fase Analisis dan Perancangan dengan pertimbangan setiap tahapan yang berfokus pada analisis permasalahan, menemukan solusi dan melakukan perancangan sistem. *Design Thinking* mempunyai 5 tahapan dalam pengembangan sistem informasi, yaitu : *Empathise*, *Define the problem*, *Ideate*, *Prototype* dan *Test*.

3.3. Metodologi Penelitian

Dengan tahapan yang ada pada metode ini, penulis menjadikan acuan sebagai metodologi penelitian. Pada metode *Design Thinking*, pengembangan sistem informasi secara menyeluruh melalui 5 tahap yang sudah disebutkan sebelumnya. Penelitian ini berfokus pada perancangan sistem informasi *e-learning*, sehingga menggunakan 3 tahap dari *Design Thinking* yaitu *Emphatise*, *Define the Problem* dan *Ideate*. Gambaran mengenai metodologi penelitian terdapat pada gambar 3.1.



Gambar 3.1. Metodologi Penelitian

Adapun gambaran dari setiap tahapan tersebut adalah sebagai berikut :

1. Fase empathise

Melakukan identifikasi masalah melalui *e-learning* yang sudah ada sebelumnya untuk menemukan user journey dan pain point sebagai acuan solusi. Proses ini dibantu dengan metode survey melalui kuisioner. Kuisioner menggunakan metode Skala Likert.

- a. Fokus terhadap kepuasan pengguna dan nilai fungsionalitas *e-learning* terhadap pengguna
- b. Skala likert, mengukur fungsionalitas dan manfaat *e-learning*

2. Fase define the problem

Melakukan analisis permasalahan, pengelompokan dan klasifikasi kebutuhan *e-learning*.

3. Fase ideate

Melakukan perancangan *e-learning* menggunakan metode *domain driven design*. Metode ini memodelkan sistem informasi yang dibagi menjadi 4 bagian utama yaitu, *Aggregates*, *Bounded context*, *Domain Events* dan *Inter-BC Interaction*. Pemodelan dilakukan dengan notasi UML (*Unified Modelling Language*) dan paradigma berorientasi objek. Setiap *domain* merupakan permasalahan yang dapat menjadi use-case pada *e-learning*. Use-case tersebut dapat mempunyai objek – objek didalamnya yang saling berinteraksi dan dapat diklasifikasi.

BAB IV HASIL DAN PEMBAHASAN

4.1. *Empathise*

Pada tahapan *empathise*, dilakukan identifikasi terhadap pengguna dari *e-learning*. Peneliti menggunakan metode *user persona* dan observasi pengguna. Pada *user persona*, ditentukan perkiraan identitas pengguna. Sementara itu, observasi pengguna meneliti perilaku pengguna berdasarkan hasil kuisioner. Berikut ini dijelaskan secara lebih mendalam mengenai hasil dari dua metode tersebut.

4.1.1. Observasi pengguna

Penelitian ini menggunakan teknik observasi pengguna untuk mendapat wawasan lebih detail. Hasil dari observasi ini digunakan untuk menentukan *domain* pada pemodelan *domain driven design*. Observasi pengguna melalui media kuisioner. Panduan untuk membuat kuisioner menggunakan metode *empathy mapping*.

Sebaran kuisioner dilakukan pada media sosial Telegram dan Whatsapp. Pada Telegram di sebar ke beberapa group dengan topik pembahasan teknologi informasi pengembangan aplikasi web. Pada Whatsapp disebar secara langsung kepada rekan – rekan di kantor. Hasil observasi pengguna didapatkan berdasarkan jawaban dari 21 responden yang dilakukan dari tanggal 2 November 2020 hingga 7 November 2020. Sementara itu, panduan metode *empathy mapping* didukung dengan skala *likert* untuk mengetahui kesepakatan pengguna terhadap komponen *empathy mapping*. Berikut ini adalah kelompok pertanyaan dan pernyataan yang dibuat berdasarkan teknik *empathy mapping*.

Tabel 4.1. Variabel *empathy mapping*

Variabel	Komponen	Pertanyaan
TF	Think and Feel	Mengapa Anda memilih menggunakan <i>e-learning</i> tersebut?
SE	See	Bagaimana pendapat Anda tentang fasilitas disekitar Anda saat menggunakan <i>e-learning</i> tersebut?
HR	Hear	Bagaimana pengaruh rekan atau pimpinan Anda untuk menggunakan <i>e-learning</i> tersebut?
SD	Say and Do	Bagaimana cara Anda menggunakan <i>e-learning</i> tersebut?
PN	Pains	Apa yang Anda takutkan saat menggunakan <i>e-learning</i> tersebut?

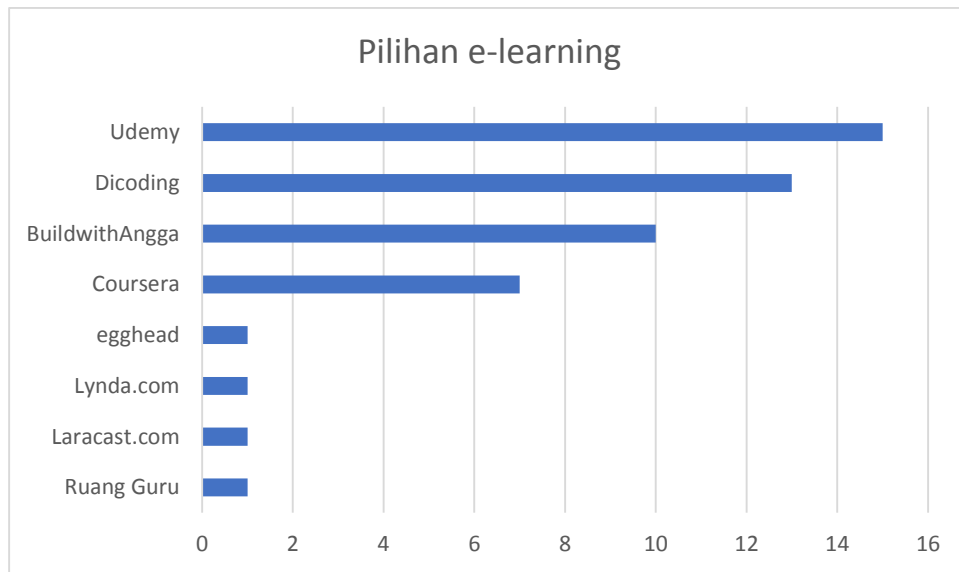
Variabel	Komponen	Pertanyaan
		Bagaimana tanggapan Anda tentang kesulitan yang dihadapi saat menggunakan <i>e-learning</i> tersebut?
GN	Gains	Bagaimana tanggapan Anda tentang <i>e-learning</i> yang baik?

Tabel 4.2. Variabel *empathy mapping* (skala likert)

Variabel	Komponen	Pernyataan
TF	Think and Feel	Aplikasi <i>e-learning</i> tersebut membantu cara belajar saya
SE	See	Fasilitas mendukung penggunaan <i>e-learning</i> tersebut
HR	Hear	Pimpinan saya menganjurkan menggunakan <i>e-learning</i> tersebut Rekan saya menganjurkan menggunakan <i>e-learning</i> tersebut
SD	Say and Do	Aplikasi <i>e-learning</i> tersebut mudah digunakan
PN	Pains	Apa yang Anda takutkan saat menggunakan <i>e-learning</i> tersebut? Bagaimana tanggapan Anda tentang kesulitan yang dihadapi saat menggunakan <i>e-learning</i> tersebut?
GN	Gains	Aplikasi <i>e-learning</i> tersebut sudah sesuai harapan saya

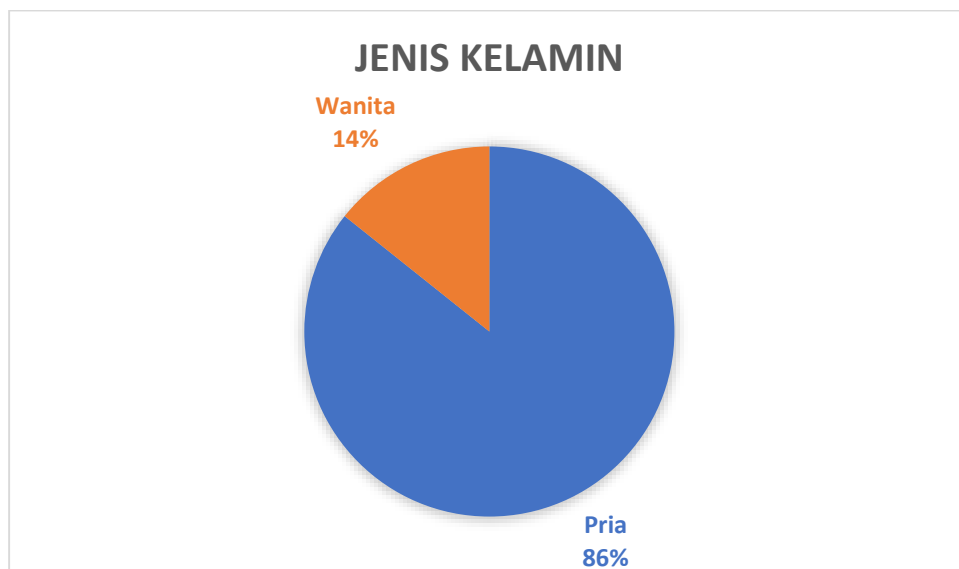
Pada tahapan observasi pengguna, sebelum mengisi komponen *empathy mapping*, diarahkan terlebih dahulu untuk mengisi identitas yang terdiri dari nama, jenis kelamin, usia profesi. Tujuan dari pengisian ini adalah untuk mendapatkan profil pengguna yang sesuai dengan *user persona*. Selanjutnya, ada pengisian pilihan *e-learning* yang sudah ada. Hal ini dilakukan untuk mendapatkan wawasan tentang pengaruh atau penerimaan pengguna terhadap *e-learning* tersebut.

Penelitian ini menggunakan contoh aplikasi *e-learning* yang sudah ada antara lain: *Dicoding*, *Udemy*, *Coursera*, *Buildwithangga*. Pemilihan *e-learning* tersebut disebabkan popularitas penggunaannya pada lingkungan pengembangan kemampuan teknologi informasi. Pada observasi pengguna, diperbolehkan untuk menambahkan *e-learning* lain. Hasil dari penerimaan aplikasi *e-learning* yang sudah ada terdapat pada gambar 4.1 hingga 4.3.



Gambar 4.1. Penerimaan *e-learning* yang sudah ada

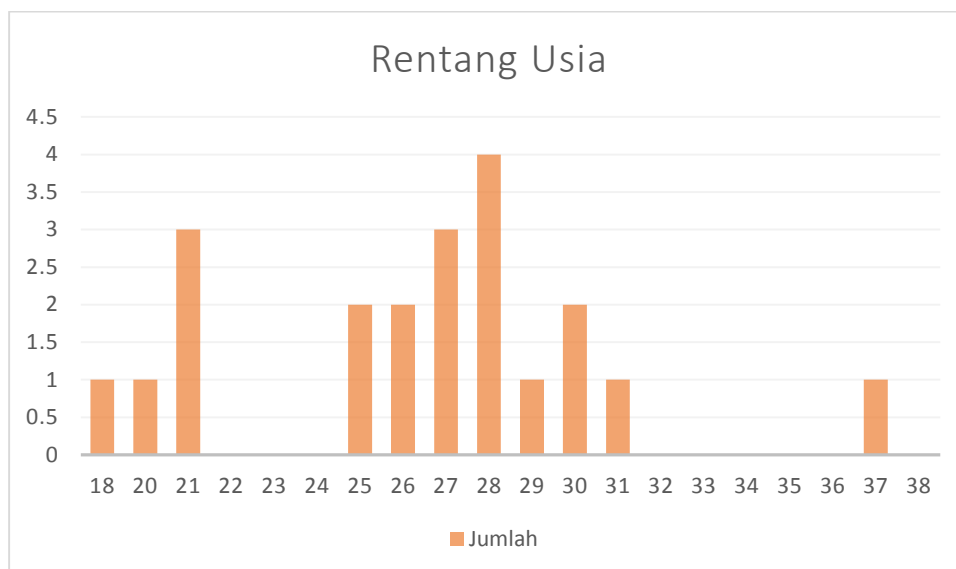
Gambar 4.1 menunjukkan penerimaan *e-learning* dari sejumlah responden. Pada grafis tersebut ditunjukkan *e-learning* Udemy mendapatkan penerimaan tertinggi kemudian diikuti oleh Dicoding dan BuildwithAngga. Selanjutnya pilihan *e-learning* lainnya merupakan opsi yang diberikan oleh responden itu sendiri.



Gambar 4.2. Persentase responden

Gambar 4.2 menunjukkan sebaran jenis kelamin responden. Responden mayoritas adalah Pria dengan persentase 86% sedangkan Wanita sejumlah 14%. Hal ini menunjukkan

pula bahwa *e-learning* untuk kemampuan teknologi informasi masih didominasi peminatnya oleh kaum Pria.



Gambar 4.3. Rentang usia responden

Berdasarkan pada gambar 4.3, disimpulkan bahwa pengguna banyak yang menggunakan *e-learning* Udemmy dengan persentase 71.43%. Pilihan kedua adalah Dicoding dengan persentase 61.90% dan ketiga adalah BuildwithAngga dengan persentase 47.62%. Udemmy, Dicoding dan BuildwithAngga menjadi pilihan utama bagi pengguna yang membutuhkan *e-learning* untuk meningkatkan kemampuan teknologi informasi. Selanjutnya, dari kuisisioner, pada aspek tanggapan digunakan skala *likert* untuk mendapatkan pandangan dari penerimaan Dicoding dan Udemmy. Skala dimulai dari sangat tidak setuju, tidak setuju, biasa saja, setuju, sangat setuju. Hasil dari tanggapan responden terdapat pada tabel 4.5 hingga 4.10. Sementara itu, panduan nilai menggunakan bobot skala *likert* pada tabel 4.3. Pada tabel 4.4 terdapat nilai maksimum dan minimum dari hasil perhitungan.

Tabel 4.3. Bobot skala likert

Tanggapan	Bobot
Sangat Setuju	5
Setuju	4
Biasa Saja	3
Tidak Setuju	2
Sangat Tidak Setuju	1

Pada tabel 4.3 menunjukan nilai bobot pada skala *likert* yang digunakan pada survey yang dilakukan oleh penulis. Bobot dimulai dari tanggapan Sangat Setuju dengan nilai paling besar yaitu 5 hingga tanggapan Sangat Tidak Setuju dengan nilai paling kecil yaitu 1. Penggunaan bobot akan dilanjutkan pada tahapan perhitungan skala likert.

Tabel 4.4. Nilai maksimum dan minimum

Nilai	
Maksimal	105
Minimal	21

Pada tabel 4.4 menunjukan nilai maksimal dan minimal dari skala *likert* yang digunakan. Nilai maksimal tersebut didapatkan dengan jumlah total responden dikali dengan nilai bobot paling tinggi, sehingga didapatkan perhitungan $21 \times 5 = 105$. Sedangkan nilai minimal didapatkan dengan jumlah total responden dikali dengan nilai bobot paling rendah, sehingga didapatkan perhitungan $21 \times 1 = 21$.

Tabel 4.5. Persentase variabel *empathy mapping* TF

Variabel Empathy Mapping	Pernyataan	Jawaban	Hasil	
			F	Nilai
TF	Aplikasi <i>e-learning</i> tersebut membantu cara belajar saya	SS	9	45
		S	6	24
		BS	5	15
		TS	1	2
		STS	0	0
		Total	86	
		Persentase	81.90	

Pada tabel 4.5 menunjukan hasil dari skala *likert* untuk variabel *empathy mapping* TF (*Think and Feel*). Hasil tersebut menunjukan tanggapan dengan persentase 81.90% pada pernyataan “Aplikasi *e-learning* tersebut membantu cara belajar saya”. Nilai dari tanggapan menyatakan bahwa *e-learning* memberikan dampak yang baik untuk cara belajar responden.

Tabel 4.6. Persentase variabel *empathy mapping* SD

Variabel Empathy Mapping	Pernyataan	Jawaban	Hasil	
			F	Nilai
SD	Aplikasi <i>e-learning</i> tersebut mudah digunakan	SS	11	55
		S	4	16
		BS	4	12

Variabel Empathy Mapping	Pernyataan	Jawaban	Hasil	
			F	Nilai
		TS	1	2
		STS	1	1
		Total		86
		Persentase		81.90

Pada tabel 4.6 menunjukkan hasil dari skala *likert* untuk variabel *empathy mapping* SD (Say and Do). Hasil tersebut menunjukkan tanggapan dengan persentase 81.90% pada pernyataan “Aplikasi *e-learning* tersebut mudah digunakan”. Nilai dari tanggapan menyatakan bahwa *e-learning* mudah untuk digunakan oleh para responden.

Tabel 4.7. Persentase variabel *empathy mapping* HR1 (Rekan)

Variabel Empathy Mapping	Pernyataan	Jawaban	Hasil	
			F	Nilai
HR1	Rekan saya mengajurkan menggunakan <i>e-learning</i> tersebut	SS	4	20
		S	4	16
		BS	5	15
		TS	1	2
		STS	7	7
		Total		60
		Persentase		57.14

Pada tabel 4.7 menunjukkan hasil dari skala *likert* untuk variabel *empathy mapping* HR (Hear). Untuk variabel Hear, penulis membagi dua menjadi pengaruh rekan dan pimpinan (HR2). Hasil tersebut menunjukkan tanggapan dengan persentase 57.14% pada pernyataan “Rekan saya mengajurkan menggunakan *e-learning* tersebut”. Nilai dari tanggapan menyatakan bahwa pengaruh rekan untuk penggunaan *e-learning* tidak terlalu signifikan.

Tabel 4.8. Persentase variabel *empathy mapping* HR2 (Pimpinan)

Variabel Empathy Mapping	Pernyataan	Jawaban	Hasil	
			F	Nilai
HR2	Pimpinan saya menganjurkan menggunakan <i>e-learning</i> tersebut	SS	5	25
		S	1	4
		BS	9	27
		TS	3	6
		STS	3	3
		Total		65
		Persentase		61.90

Pada tabel 4.8 menunjukkan hasil dari skala *likert* untuk variabel *empathy mapping* HR (Hear). Hasil tersebut menunjukkan tanggapan dengan persentase 61.90% pada pernyataan “Pimpinan saya menganjurkan menggunakan *e-learning* tersebut”. Nilai dari tanggapan menyatakan bahwa pimpinan memberikan pengaruh yang cukup signifikan terhadap responden untuk menggunakan *e-learning*.

Tabel 4.9. Persentase variabel *empathy mapping* SE

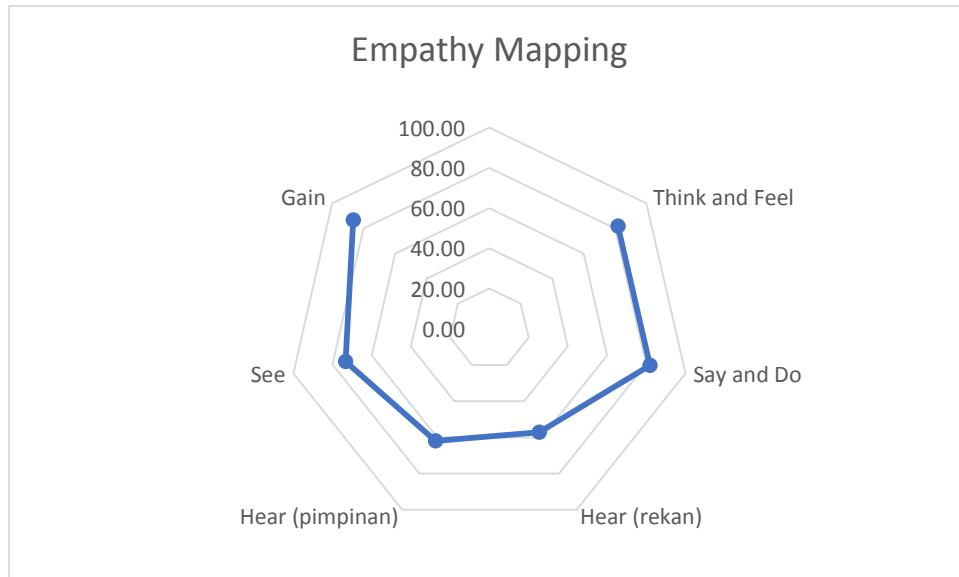
Variabel Empathy Mapping	Pernyataan	Jawaban	Hasil	
			F	Nilai
SE	Fasilitas mendukung penggunaan <i>e-learning</i> tersebut	SS	6	30
		S	5	20
		BS	7	21
		TS	3	6
		STS	0	0
		Total	77	
		Persentase	73.33	

Pada tabel 4.9 menunjukkan hasil dari skala *likert* untuk variabel *empathy mapping* SE (See). Hasil tersebut menunjukkan tanggapan dengan persentase 73.33% pada pernyataan “Fasilitas mendukung penggunaan *e-learning* tersebut”. Nilai dari tanggapan menyatakan bahwa fasilitas yang dimiliki responden memberikan pengaruh pada saat menggunakan *e-learning*.

Tabel 4.10. Persentase variabel *empathy mapping* GN

Variabel Empathy Mapping	Pernyataan	Jawaban	Hasil	
			F	Nilai
GN	Aplikasi <i>e-learning</i> tersebut sudah sesuai harapan saya	SS	9	45
		S	10	40
		BS	2	6
		TS	0	0
		STS	0	0
		Total	91	
		Persentase	86.67	

Pada tabel 4.10 menunjukkan hasil dari skala *likert* untuk variabel *empathy mapping* GN (Gain). Hasil tersebut menunjukkan tanggapan dengan persentase 86.67% pada pernyataan “Aplikasi *e-learning* tersebut sudah sesuai harapan saya”. Nilai dari tanggapan menyatakan bahwa *e-learning* sudah sesuai dengan harapan para responden.



Gambar 4.4. Hasil analisa empathy mapping

Pada gambar 4.4 menunjukkan bahwa pengaruh rekan dan pimpinan tidak terlalu signifikan terhadap keputusan untuk menggunakan *e-learning*. Keputusan tersebut dominan di pengaruhi oleh kehendak pribadi. Hal tersebut dipengaruhi oleh apa yang target pengguna lihat, rasakan dan harapkan. Analisa *empathy mapping* didukung oleh observasi pengguna dengan pertanyaan yang lebih spesifik dan deskriptif. Pada tabel 4.11 ditunjukkan hasil observasi pengguna.

Tabel 4.11. Tanggapan pengguna *e-learning*

No	Pertanyaan	Tanggapan
1.	Mengapa Anda memilih <i>e-learning</i> tersebut?	1. Materi dan Trainer yang baik dan jelas dalam menjelaskan materi 2. Populer 3. kursus yg banyak dan ada yg gratis 4. Belajarnya lebih santai 5. Murah dan mudah
2.	Bagaimana tanggapan Anda tentang kesulitan yang dihadapi saat menggunakan	1. kadang tidak up to date, bahasa / logat dari Trainer (issu komunikasi) 2. sulit bertanya

No	Pertanyaan	Tanggapan
	<i>e-learning</i> tersebut?	3. ada beberapa error dan stag dan tutornya slow respon 4. kadang susah mencari <i>course</i> yang bagus (personalisasi)
3.	Apa yang Anda takutkan saat menggunakan <i>e-learning</i> tersebut?	1. tidak lengkap materi yang diinginkan, tidak jelas penyampaian materi (<i>issu Content</i>) 2. biaya (<i>issu financial</i>) 3. tidak fokus (<i>issu student</i>)
4.	Bagaimana pengaruh rekan atau pimpinan Anda untuk menggunakan <i>e-learning</i> tersebut?	1. Pimpinan mempunyai pengaruh (mengizinkan dan membelikan <i>course</i>) untuk kebutuhan pendalaman teknologi 2. Pengaruh rekan sedikit
5.	Bagaimana cara Anda menggunakan <i>e-learning</i> tersebut	1. cukup lengkap 2. masih kurang
6.	Bagaimana tanggapan tentang <i>e-learning</i> yang baik untuk mempelajari teknologi informasi?	mudah digunakan, ada yang mereview hasil belajar (<i>mentoring</i>)

4.1.2. User Persona

User persona yang digunakan adalah menggunakan persona mahasiswa dan pegawai. Pembatasan terhadap persona ini digunakan agar penelitian ini spesifik terhadap wawasan (*insight*) tentang *e-learning* untuk pengembangan kemampuan teknik informatika. Pada tabel 4.12 menunjukan *user persona* dari *e-learning*.

Tabel 4.12. *User persona*

Nama	: Rayhan
Usia	: 25 tahun
Pekerjaan	: Pegawai swasta
Kemampuan	: Bisa membuat website fullstack hingga hosting
Skenario	
Saya seorang pegawai swasta di suatu perusahaan multinasional. Bekerja sebagai web developer dalam suatu tim pengembangan sistem informasi. Saya sudah terbiasa untuk mengerjakan aplikasi web mulai dari frontend, backend hingga deployment ke vps. Pada	

beberapa waktu, saya sering menghadapi kesulitan teknis. Saya biasa mencari solusinya melalui media sosial atau youtube. Apabila dirasa perlu, saya membeli akses untuk belajar di beberapa *online course*. Saya membutuhkan ragam materi yang sesuai dengan kesulitan yang saya hadapi sehari – hari. Terkadang materi di beberapa *online course* hanya membahas studi kasus saja tanpa bisa diketahui arah pembelajaran materi. Saya juga perlu untuk mengetahui kemajuan saya belajar. Untuk lebih termotivasi target waktu penyelesaian suatu materi dapat menjadi pilihan. Sementara itu, tambahan tugas bahkan ujian akan lebih baik bagi saya. Materi pun akan lebih mudah dipahami apabila saya dapat berkonsultasi langsung dengan pengajar. Apabila terdapat sertifikat atas penyelesaian materi, akan lebih baik untuk portfolio saya. Terakhir, pilihan pembayaran baik secara angsuran atau tunai agar materi dapat lebih mudah diakses tanpa terkendala biaya.

4.2. Define the problem

Secara umum *e-learning* mempunyai kebutuhan bisnis yang sudah dijelaskan pada bab 3 (tiga). Kebutuhan bisnis tersebut merupakan *sub-domain* dari *domain* utama yaitu *e-learning*. Dengan pertimbangan kebutuhan umum dari *e-learning*, hasil dari observasi, dan lingkup penelitian, maka dalam penelitian ini mengidentifikasi *e-learning* kedalam 4 *sub-domain* yaitu *Course*, *Assignment*, *Examination* dan *Payment*. Pada tabel 4.13 dijelaskan pemetaan *sub-domain* dengan kebutuhan pengguna.

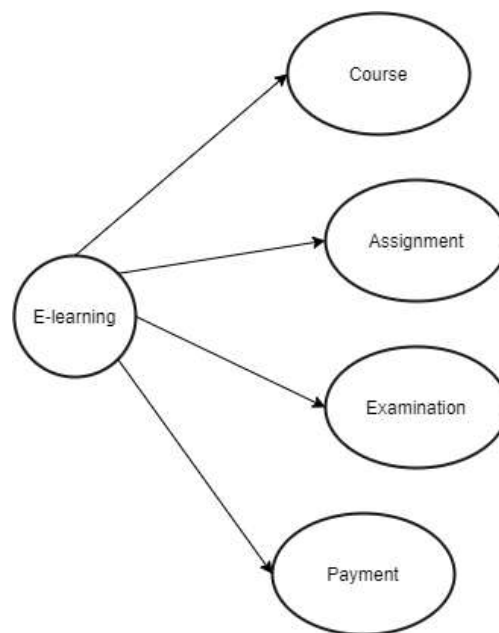
Tabel 4.13. *Sub-domain* dari *domain e-learning*

Domain	Sub-domain	Kebutuhan
E-learning	Course	Tersedia ragam materi ajar yang sesuai dengan kebutuhan
		Peserta ajar dapat melihat jalur belajar (<i>learning path</i>)
		Peserta ajar dapat mengetahui kemajuan belajar (<i>learning progress</i>)
		Cara belajar yang dinamis dapat per konten ajar atau dengan target waktu belajar (<i>enrollment</i>)
	Assignment	Tersedia tugas di setiap konten ajar untuk mengasah kemampuan peserta
		Tersedia review disetiap tugas langsung dari pengajar (<i>tutor</i>)
	Examination	Di setiap jalur belajar (<i>course</i>) terdapat ujian yang berguna untuk validasi kemampuan peserta ajar

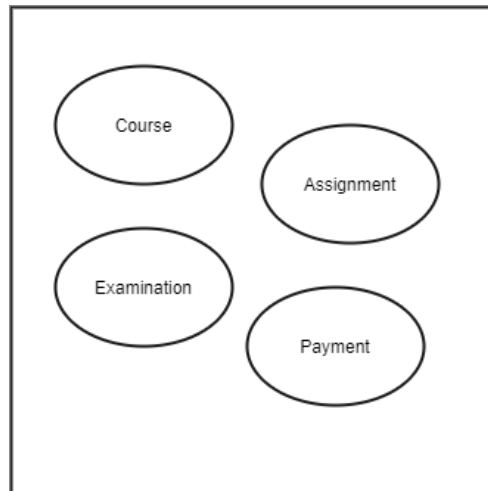
<i>Domain</i>	<i>Sub-domain</i>	Kebutuhan
		Hasil dari ujian adalah sertifikat kompetensi
	<i>Payment</i>	<i>Course</i> dapat diakses dengan cara pembayaran berangsur atau tunai. Cara ini untuk memenuhi kebutuhan harga yang terjangkau

4.2.1. *Problem and Solution space*

Domain diidentifikasi ke dalam kelompok *problem space* dan *solution space*. *Problem space* menunjukkan kelompok *domain* utama dan *sub-domain*. *Solution space* menunjukkan kelompok *sub-domain*. *Solution space* akan menjadi kandidat Bounded-Context pada *domain-driven design*. Pada gambar 4.5 menunjukkan analisis *problem space* dan gambar 4.6 menunjukkan analisis *solution space*.



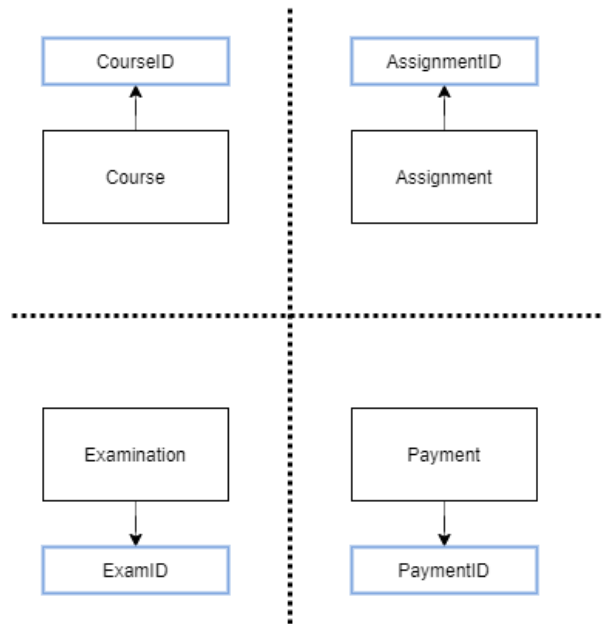
Gambar 4.5. *Problem space E-learning*



Gambar 4.6. *Solution space E-learning*

4.2.2. **Aggregates**

Setelah identifikasi problem dan solution space, selanjutnya mengidentifikasi *aggregate* dari *domain e-learning*. *Aggregate* didapatkan dari solution space. *Aggregate* mendefinisikan lingkup dari *e-learning*. Setiap *aggregate* mempunyai penanda (*identifier*). Identifier tersebut berguna dalam membantu identifikasi *entity* dan *value object* pada tahapan selanjutnya. Identifier juga menunjukkan alur logik dari setiap *entity* yang ada pada *aggregate* tersebut.



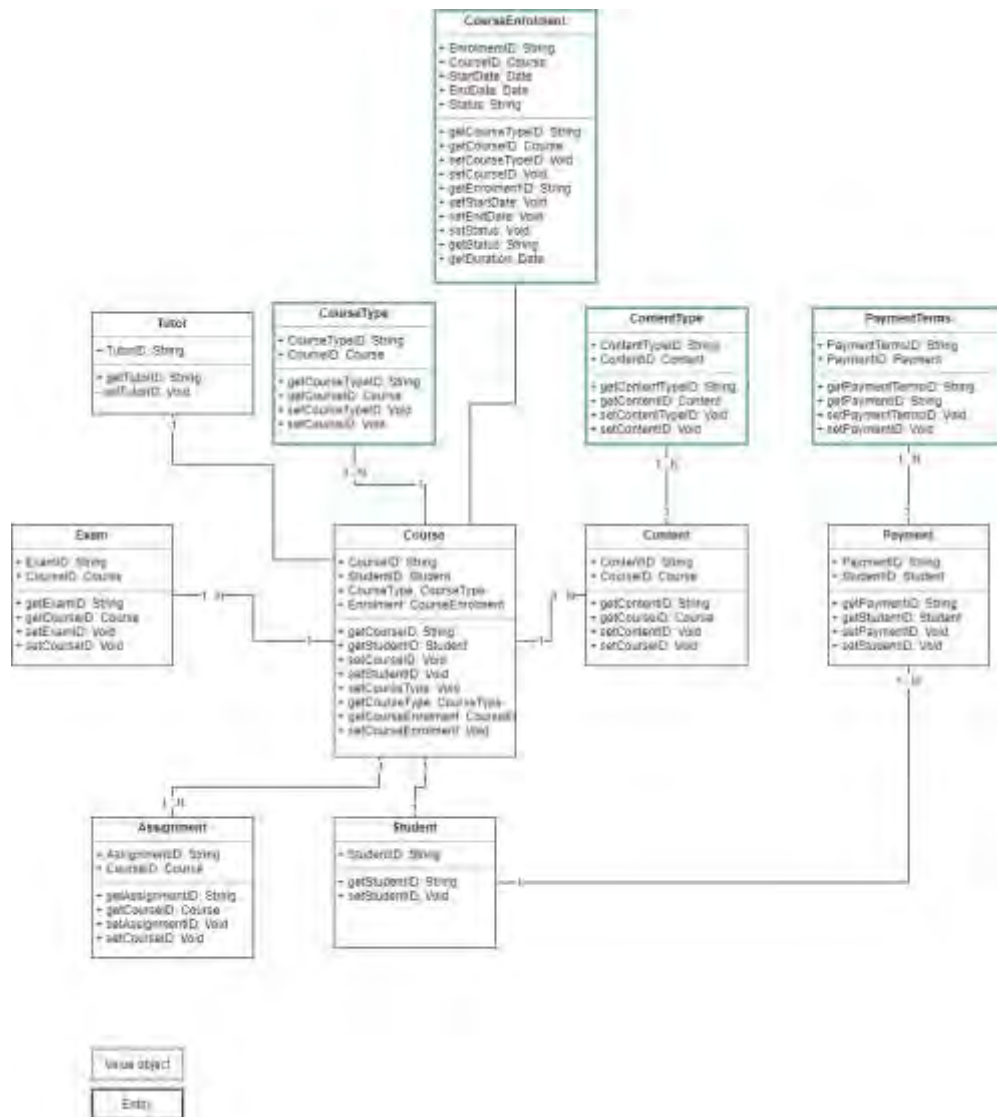
Gambar 4.7. *E-learning aggregate*

. Pada gambar 4.7 menjelaskan tentang *aggregate* dari *domain e-learning*. *Aggregate e-learning* terdiri dari *Course* dengan penanda *CourseID*, *Assignment* dengan penanda *AssignmentID*, *Examination* dengan penanda *ExamID* dan *Payment* dengan penanda *PaymentID*. Setiap *aggregate* akan menjadi *bounded context* pada *domain driven design*. *Bounded context* ini mewakili bisnis utama yang ada pada *domain e-learning*.

4.2.3. *Entities and Value object*

Sebelumnya sudah ditentukan *aggregate* dan *bounded context* dari *e-learning*. Pada *bounded context* dapat diidentifikasi *Entities* dan *value object* yang terlibat. Pada penelitian ini, terdapat entitas – entitas di dalam masing -masing *bounded context* yang diperlukan sebagai pembentuk *e-learning*. Entitas tersebut antara lain *Tutor*, *Course*, *Student*, *Content*, *Payment*, *Exam*, *Assignment*. Sementara itu, *value object* antara lain *CourseType*, *CourseEnrolment*, *ContentType* dan *PaymentTerms*.

Pada gambar 4.8 menunjukkan representasi hubungan antara entites menggunakan model *class diagram*. Untuk membedakan antara entites dan *value object*. Notasi *class diagram* dibedakan dengan warna. *Entities* menggunakan warna hitam dan *value object* menggunakan warna hijau. Makna dari hubungan antara *Entities* tersebut adalah mewakili proses – proses logik pada *bounded context*.



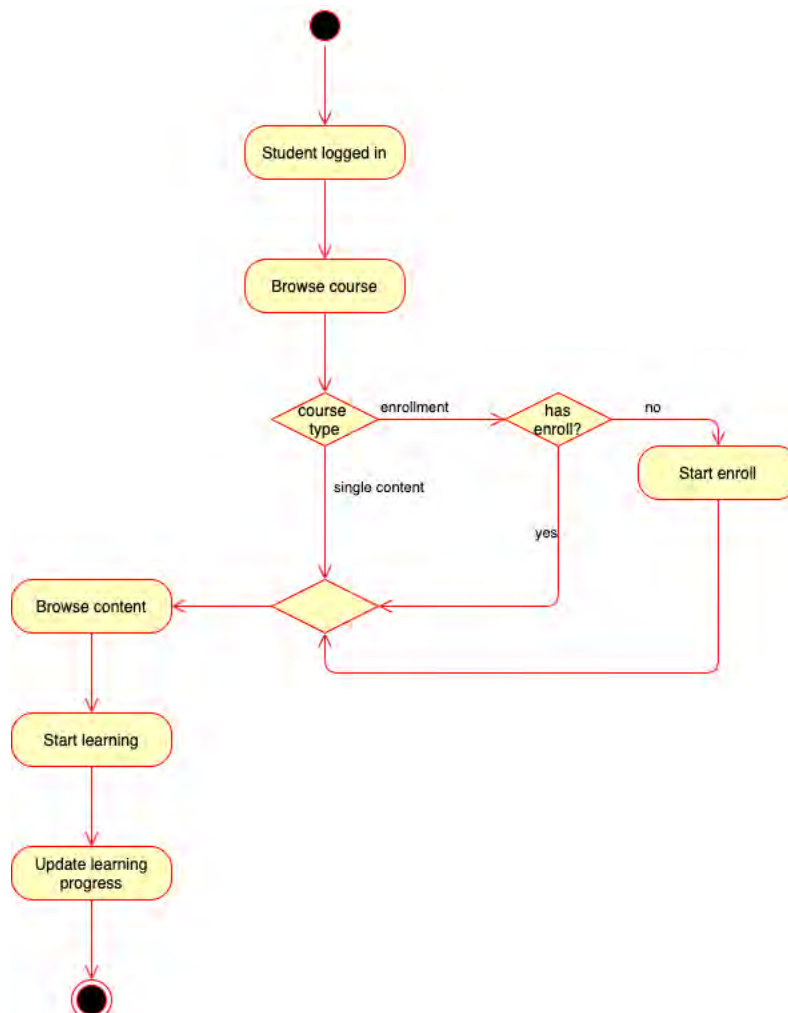
Gambar 4.8. *Entities dan Value object*

Bounded context Course meliputi entitas *Course*, *Content*, *Tutor*, *Student* dan *Payment*. *Bounded context Assignment* meliputi entitas *Assignment*, *Course*, *Content* dan *Student*. *Bounded context Exam* meliputi *Exam*, *Course*, dan *Student*. *Bounded context Payment* meliputi *Payment*, *Student* dan *Course*. Sementara itu, *value object* merupakan entitas yang menjadi nilai dari entitas lain terdiri dari *CourseType*, *ContentType* dan *PaymentTerms*. Sebagai contoh, saat terjadi transaksi penyimpanan data pada entitas *Course*, atribut *CourseType* dapat digantikan oleh *value object* *CourseType* karena objek tersebut membawa nilai yang diperlukan antara lain *SingleContent* dan *Enrollment*. *CourseType* dipertimbangkan dalam bisnis sebagai pembeda antara proses belajar materi

tanpa batas waktu (*SingleContent*) atau proses belajar materi dengan batas waktu yang ditentukan (*Enrollment*).

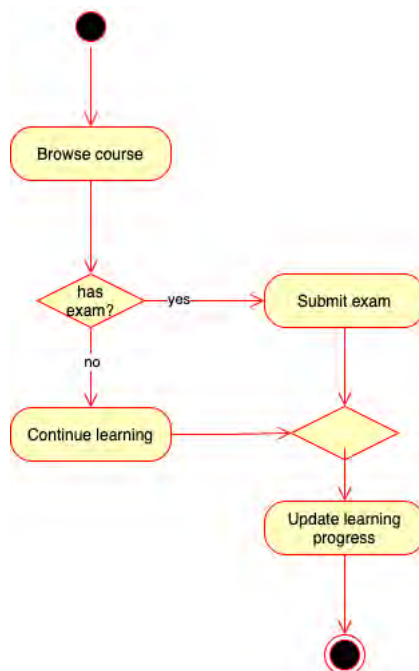
4.2.4. Saga

Saga adalah gambaran proses bisnis di setiap *bounded context*. Pada setiap saga dimodelkan dengan *flowchart*. Kebutuhan pemodelan proses bisnis adalah untuk mengetahui alur proses bisnis dengan lebih detail. Penelitian ini akan mengimplementasikan *domain* kepada *arsitektur microservice*. Tahapan pemodelan saga digunakan untuk identifikasi *event – event* apa saja yang terjadi. *Event* tersebut merupakan output dari setiap *bounded context*. Proses bisnis ini memberikan manfaat kepada pengguna utama yaitu peserta ajar (*Student*).



Gambar 4.9. Proses bisnis pada *bounded context* Course

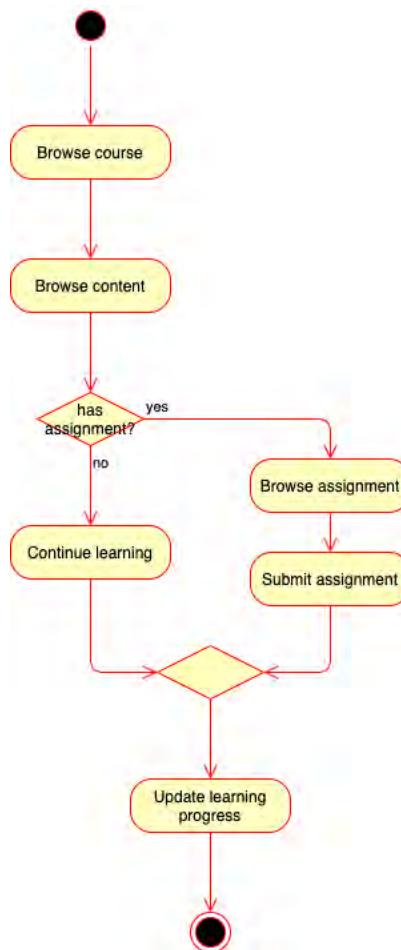
Gambar 4.9 memodelkan proses bisnis pada *bounded context Course*. Dimulai dari keadaan *Student* sudah login, *Student* mengakses *course* (*Browse course*). Apabila *course* tersebut mempunyai jenis *course* (*CourseType*) single *Content* maka *Student* dapat langsung mengakses *Content* (*Browse Content*) untuk selanjutnya langsung belajar. Selanjutnya *e-learning* meng-update status belajar (*update learning progress*). Sementara itu, apabila *Course* yang dipilih adalah jenis *course* enrolment, maka *Student* akan dikonfirmasi oleh sistem apakah sudah pernah enroll atau belum. Apabila sudah pernah maka dapat langsung mengakses *Content* namun apabila belum maka *Student* harus melakukan *Start enroll* terlebih dahulu. Dari enrolment tersebut *Student* dapat langsung belajar dengan mengakses *Content* selanjutnya sistem mengubah status kemajuan belajar.



Gambar 4.10. Proses bisnis pada *bounded context Exam*

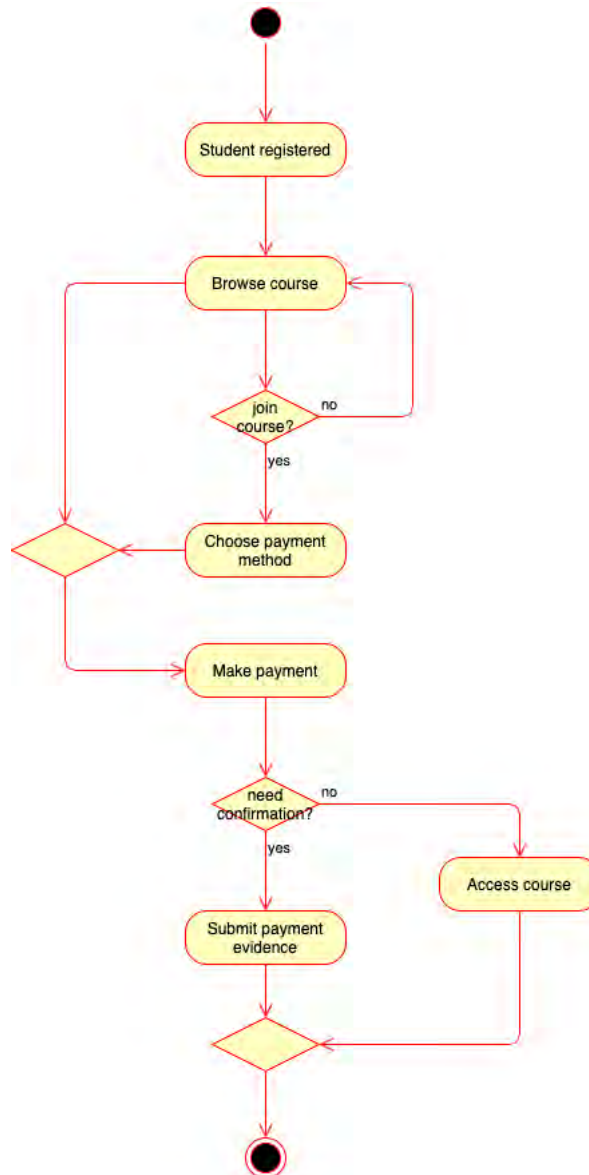
Pada gambar 4.10 memodelkan proses bisnis pada *bounded context Exam*. Model ini mempunyai asumsi *Student* sudah login. Dimulai dari *Student* mengakses *Course* (*Browse course*). Apabila *Course* tersebut mempunyai ujian (*Exam*) maka *Student* harus mengerjakan ujian tersebut (*Submit Exam*) untuk kemudian sistem akan mengubah status

kemajuan belajar (*update learning progress*). Sementara itu, apabila *Course* tidak mempunyai ujian maka *Student* dapat melanjutkan belajar.



Gambar 4.11. Proses bisnis pada *bounded context Assignment*

Pada gambar 4.11 memodelkan proses bisnis pada *bounded context Assignment*. Asumsi yang digunakan sama seperti pada model saga *Exam*. *Student* mengakses *course* dan mengakses *Content*. Apabila terdapat *Content* yang mempunyai tugas (*Assignment*) maka *Student* dapat mengakses *Assignment* tersebut (*Browse Assignment*) dan mengerjakan *Assignment* (*submit Assignment*). Selanjutnya, sistem akan update status belajar. Kemudian, apabila *Content* tersebut tidak mempunyai *Assignment* maka *Student* dapat melanjutkan belajar.



Gambar 4.12. Proses bisnis pada *bounded context Payment*

Pada gambar 4.12 memodelkan proses bisnis pada *bounded context Payment*. Dimulai dari *Student* yang sudah terdaftar pada *e-learning*. *Student* mengakses *Course* (*Browse course*). Sistem menampilkan konfirmasi kepada *Student* apakah akan mengikuti *Course* tersebut. Apabila *Student* memutuskan untuk mengikuti *Course*, maka Sistem menawarkan metode pembayaran (*Choose payment method*) yang dapat dilakukan oleh *Student*.

Kemudian, apabila *Student* membutuhkan pembayaran secara angsuran (*payment terms*) maka *Student* dapat memilih termin pembayaran. Selanjutnya, *Student*

melakukan pembayaran. Sistem akan mengirimkan informasi pembayaran melalui email. Pada email tersebut, apabila pembayaran tersebut diperlukan konfirmasi pembayaran maka *Student* harus unggah bukti pembayaran (*submit payment proof*) untuk kemudian dapat mengakses *Course*. Apabila tidak diperlukan konfirmasi maka *Student* dapat langsung mengakses *Course*.

4.2.5. **Command and Queries**

Command and Queries meliputi segala proses yang terjadi pada setiap entitas di dalam *bounded context*. Proses untuk menyimpan atau mengubah data disebut sebagai *Command*. Proses untuk mengambil atau menampilkan data disebut sebagai *query*. Pada tabel 4.14 hingga 4.17 menampilkan *Command* dan *queries* dari masing – masing *bounded context* pada domain *e-learning*.

1. *Course bounded context*

Tabel 4.14. *Command Queries Course*

<i>Command</i>	<i>Queries</i>
<i>Start enroll</i>	<i>Browse course</i>
<i>Update enroll</i>	<i>Browse Content</i>
<i>Update learning progress</i>	<i>Watch Content</i>
	<i>Download Content</i>
	<i>Read Content</i>
	<i>Search course</i>
	<i>Search Content</i>

Command pada *bounded context Course* terdiri dari *Start enroll*, *Update enroll* dan *Update learning progress*. Sedangkan *Queries* terdiri dari *Browse Course*, *Browse Content*, *Watch Content*, *Download Content*, *Read Content*, *Search Course* dan *Search Content*. *Start enrol* terjadi saat entitas *Student* memulai suatu *Course* dengan *value* dari *CourseType* adalah *Enrolment*. *Start enrol* melakukan insert data terhadap *value object CourseEnrolment*.

Update enrol terjadi saat entitas *Student* mengakses suatu *course* yang sudah berjalan waktu *enrolment* namun sudah melewati target waktu belajar sehingga perlu di set ulang *StartDate* dan *EndDate* pada *value object CourseEnrolment*. *Update learning progress* terjadi saat *Student* sudah menyelesaikan suatu materi (*Content*).

Browse Course terjadi saat entitas *Student* melakukan akses data entitas *Course*. *Browse course* memungkinkan untuk selanjutnya terjadi *Browse Content*, *Watch Content*, *Download Content* dan *Read Content*. *Watch Content* terjadi saat entitas *Student* mengakses salah satu *Content* dengan jenis *Content* (*ContentType*) video. *Download Content* terjadi saat entitas *Student* mengakses salah satu *Content* dengan jenis *Content* *Downloadable*, format data bisa dokumen (doc atau pdf) dan video. *Read Content* terjadi saat entitas *Student* mengakses salah satu *Content* dengan jenis konten *Text*. *Search Course* dan *Search Content* terjadi saat entitas *Student* melakukan pencarian data *Course* dan *Content*.

2. *Assignment bounded context*

Tabel 4.15. *Command Queries Assignment*

<i>Command</i>	<i>Queries</i>
<i>Submit Assignment</i>	<i>Browse Content</i>
<i>Update learning progress</i>	<i>Browse course</i>
	<i>Browse Assignment</i>
	<i>Search course</i>
	<i>Search Content</i>

Command pada *bounded context Assignment* terdiri dari *Submit Assignment* dan *Update learning progress*. *Queries* pada *bounded context Assignment* terdiri dari *Browse Content*, *Browse Course*, *Browse Assignment*, *Search Course* dan *Search Content*. *Submit Assignment* terjadi saat entitas *Student* mengakses suatu *Content* yang memiliki tugas didalamnya. *Student* selanjutnya harus mengerjakan tugas tersebut dan submit di *e-learning*. *Update learning progress* terjadi saat entitas *Student* selesai mengerjakan tugas maka *Course* tersebut mengalami update data kemajuan belajar (*learning progress*). *Browse Course*, *Browse Content* dan *Browse Assignment* terjadi saat entitas *Student* melakukan akses terhadap *Course*, *Content* dan *Assignment*. *E-learning* menampilkan masing – masing data dari entitas – entitas tersebut. Sementara itu, *Search Course* dan *Search Content* terjadi saat entitas *Student* melakukan pencarian terhadap data *Course* dan *Content*.

3. Exam bounded context

Tabel 4.16. Command Queries Exam

Command	Queries
Submit Exam	Browse course
Update learning progress	Browse Exam
	Search course

Command pada *bounded context Exam* terdiri dari *Submit Exam* dan *Update learning progress*. Queries pada *bounded context Exam* terdiri dari *Browse Course*, *Search Course* dan *Browse Exam*. *Submit Exam* terjadi saat entitas *Student* mengakses *course* yang mempunyai ujian. *Student* mengerjakan ujian tersebut selanjutnya data ujian di submit ke *e-learning*. Setelah data berhasil di simpan, maka sistem melakukan ubahan data kemajuan belajar (*update learning progress*). *Browse Course* dan *Search Course* sama seperti pada *bounded context Assignment* dan *Course*. *Browse Exam* terjadi saat entitas *Student* mengakses daftar ujian yang tersedia pada suatu *Course*.

4. Payment bounded context

Tabel 4.17. Command Queries Payment

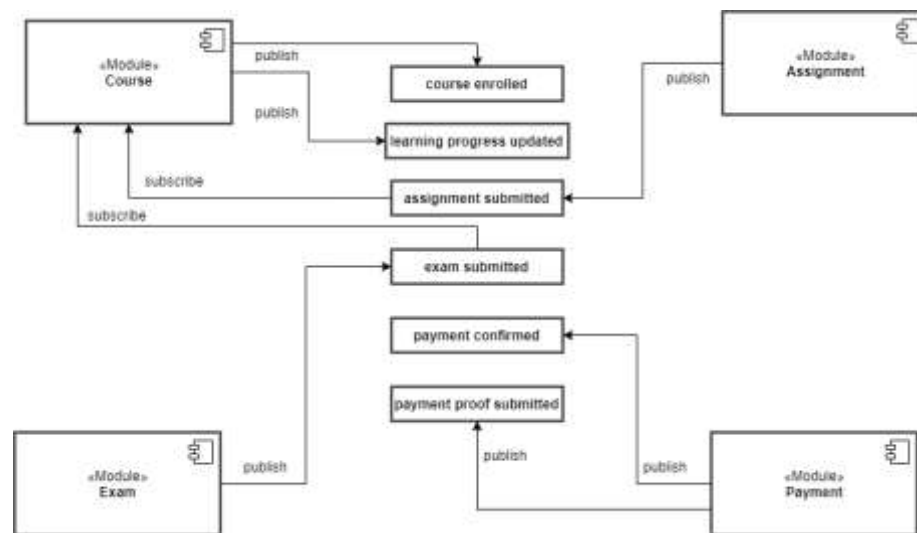
Command	Queries
Choose course	Browse course
Choose payment	Search course
Make payment	Browse payment method
Send payment information	
Submit payment proof	

Command pada *bounded context Payment* terdiri dari *Choose Course*, *Choose Payment*, *Make Payment*, *Send Payment Information*, *Submit Payment Proof*. Queries pada *bounded context Payment* terdiri dari *Browse Course*, *Search Course* dan *Browse Payment Method*. *Choose course* terjadi saat entitas *Student* memilih salah satu *Course* untuk diikuti selanjutnya sistem memunculkan daftar metode pembayaran (*Browse payment method*). *Make payment* terjadi saat *Student* melakukan pembayaran selanjutnya sistem menyimpan pembayaran tersebut. Setelah pembayaran disimpan, dikirim informasi pembayaran (*Send Payment Information*) melalui email. *Submit payment proof* terjadi saat *Student* menerima informasi pembayaran yang membutuhkan konfirmasi pembayaran.

Browse Course dan *Search Course* masih sama seperti pada *bounded context Course*, *Assignment* atau *Exam*.

4.2.6. Event

Hasil dari suatu *bounded context* adalah *event*. *Event* tersebut apabila dihasilkan dari suatu *bounded context* disebut sebagai *publish event*. Sedangkan *event* yang sudah di-publish kemudian digunakan oleh *bounded context* yang lain maka disebut sebagai *subscribe event*. Gambar 4.13 menunjukkan *event* yang terjadi pada model *domain e-learning* dari penelitian ini, *event* tersebut antara lain : *Course Enrolled*, *Learning Progress Updated*, *Assignment Submitted*, *Exam Submitted*, *Payment Confirmed*, *Payment Proof Submitted*. *Course Enrolled* dan *Learning Progress Updated* dihasilkan dari *bounded context Course*.



Gambar 4.13. Publish dan Subscribe event pada domain e-learning

Assignment Submitted dihasilkan dari *bounded context Assignment*. *Assignment Submitted* juga di-subscribe oleh *bounded context Course* untuk kebutuhan update learning progress. *Exam Submitted* dihasilkan dari *bounded context Exam*. *Event* ini juga di-subscribe oleh *bounded context Course* untuk kebutuhan update learning progress. Selanjutnya, *event Payment Confirmed* dan *Payment Proof Submitted* dihasilkan oleh *bounded context Payment*.

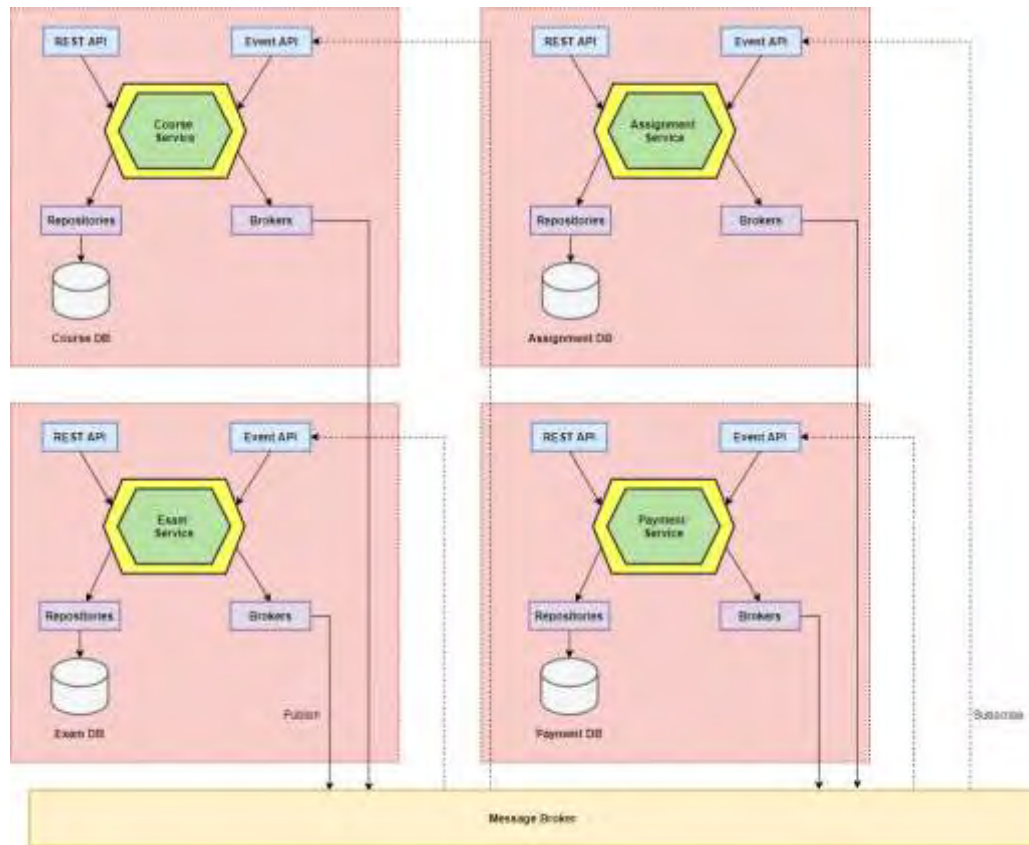
4.3. Ideate

Pada tahapan Ideate diajukan gagasan tentang arsitektur *e-learning* yang menggunakan *arsitektur microservice* dan model *e-learning* yang di representasikan menggunakan UML. *Arsitektur microservice* mengadopsi model hexagonal architecture dan model *e-learning* menggunakan use-case diagram, package diagram dan deployment diagram.

4.3.1. Arsitektur *E-learning*

Tahapan Ideate akan menerapkan model *domain* ke dalam bentuk arsitektur sistem informasi. Penelitian ini menggunakan *arsitektur microservices*. Setiap *bounded context* menjadi kandidat modul *service* pada *arsitektur microservices*. Sehingga akan terdapat 4 modul *service* yaitu *Course Service*, *Assignment Service*, *Exam Service* dan *Payment Service*. Gambar 4.14 menunjukkan rancangan *arsitektur microservices* pada *e-learning*.

Arsitektur microservices yang digunakan mengadopsi hexagonal architecture. Pada arsitektur tersebut setiap modul terdiri dari inbound *services*, *Application services* dan outbound *services*. Inbound *services* dapat sebagai REST API atau *Event API*. REST API adalah salah satu metode dataflow melalui protokol HTTP. *Event API* adalah salah satu metode dataflow melalui protokol *socket* (pub/subs).



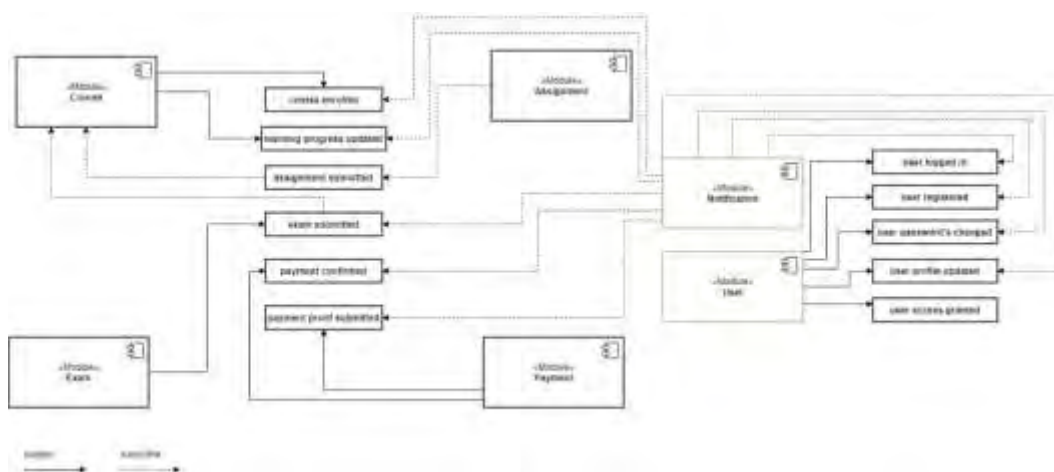
Gambar 4.14. Arsitektur Microservice E-learning

Application services pada arsitektur ini merupakan program yang terdiri dari beberapa *class* atau *functions* yang berguna untuk melakukan pengolahan data dari inbound *services* untuk kemudian di alirkan ke outbound *services*. *Application services* pada arsitektur ini mewakili *bounded context* dari domain *e-learning*. *Course Service* mempunyai inbound *services* dari jenis *Event API* yang terdiri dari *Assignment Submitted* dan *Exam Submitted*. Sementara itu, outbound *services* dari jenis *Brokers* akan mem-publish event *Course Enrolled* dan *Learning Progress Updated*.

Assignment Service, *Exam Service* dan *Payment Service* pada gambar 4.14 hanya mem-publish event. Oleh karena itu, *services* tersebut mempunyai outbound *services* pada masing – masing *Brokers* yang terdiri dari *Assignment Submitted*, *Exam Submitted*, *Payment Confirmed* dan *Payment Proof Submitted*. Outbound *services* pada arsitektur ini dibagi menjadi dua yaitu *Repositories* dan *Brokers*. *Repositories* merupakan kumpulan *class* atau *functions* yang mewakili akses data ke *database*. Sedangkan *Brokers*

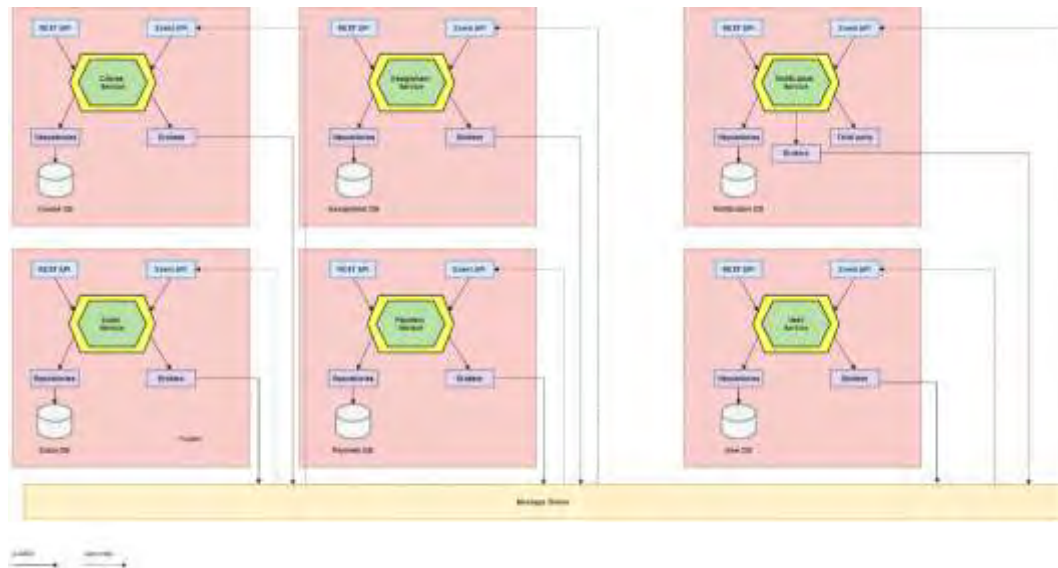
merupakan kumpulan *class* atau *functions* yang berguna untuk mem-*publish* data ke *service message broker*.

Untuk performansi *e-learning*, peneliti menambahkan dua *service* diantara nya adalah *User Service* dan *Notification Service*. Penambahan *service – service* ini untuk meningkatkan operasi bisnis dan teknis dari *e-learning*. *User service* digunakan untuk *authentication* dan *authorization* pengguna *e-learning* serta untuk proses bisnis pendaftaran dan validasi peserta ajar (*Student*). *Notification Service* digunakan untuk mengirim informasi kemajuan belajar, tugas, ujian dan pembayaran kepada peserta ajar. Informasi yang dikirim dapat melalui media email.



Gambar 4.15. *Publish dan Subscribe event pada domain e-learning dengan penambahan User service dan Notification service*

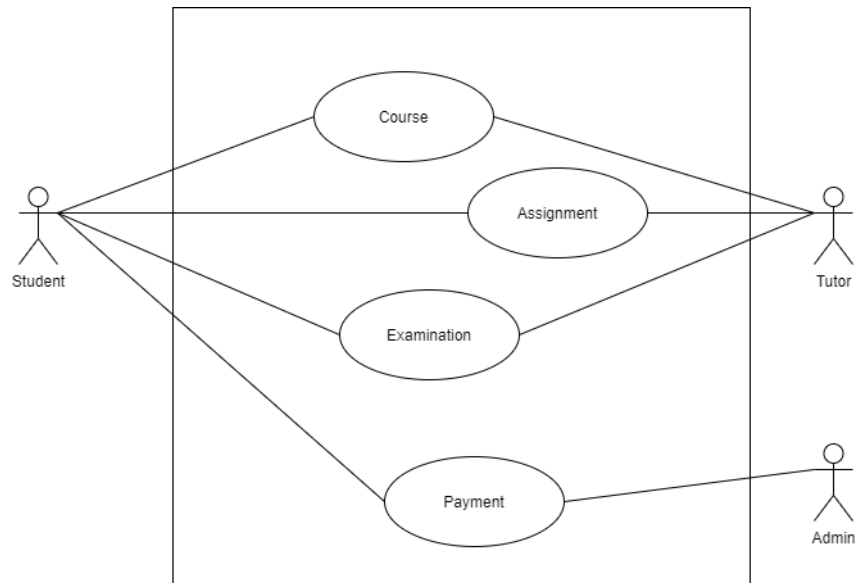
Gambar 4.15 menunjukan *event course enrolled*, *learning progress updated*, *Assignment Submitted*, *Exam Submitted*, *payment confirmed* dan *payment proof* di-subscribe oleh *Notification Service*. Tujuannya adalah apabila *event – event* tersebut terjadi maka pengguna *e-learning* (*Student*) dapat mengetahui informasi tersebut melalui email. Sementara itu, *User Service* menangani proses login pengguna, pendaftaran, serta otorisasi pengguna. Untuk itu, *user service* mem-*publish event* antara lain : *user logged in*, *user registered*, *user password's changed*, *user profile updated* dan *user access granted*. Pada gambar 4.16 menunjukan *arsitektur microservice e-learning* yang sudah disertakan *User Service* dan *Notification Service*.



Gambar 4.16. Arsitektur Microservice E-learning dengan User Service dan Notification Service

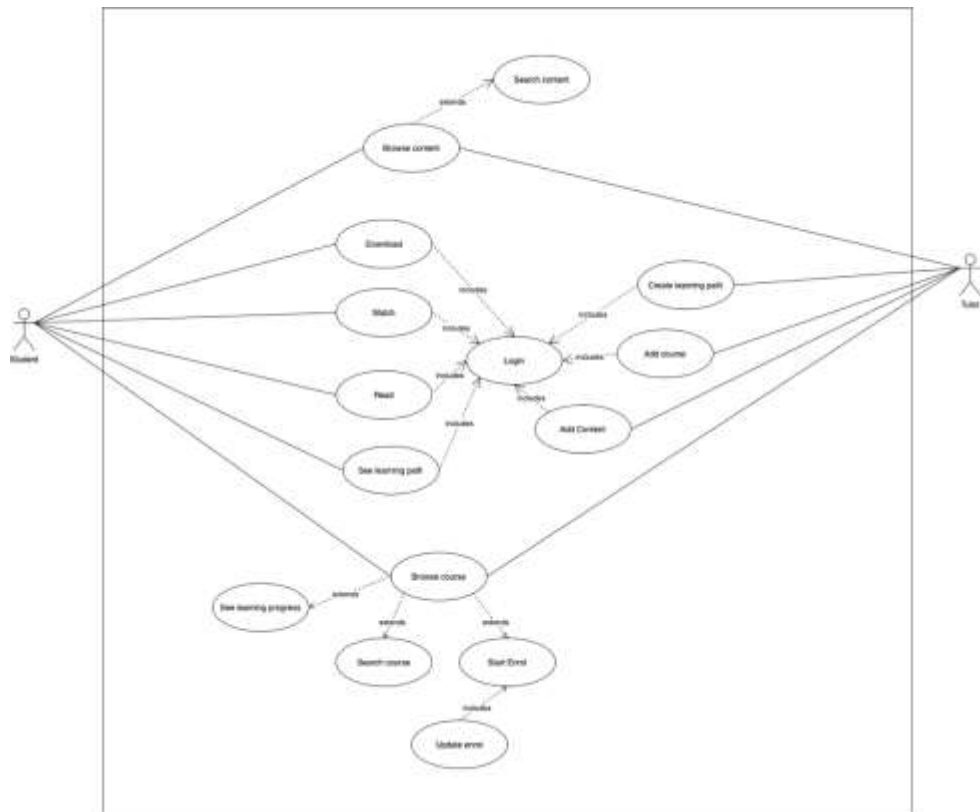
4.3.2. Model E-learning

E-learning dari penelitian ini mempunyai 4 (empat) business use-case, yaitu Course, Assignment, Examination dan Payment. Business use-case diambil dari sub-domain pada model domain. Adapun aktor yang terlibat adalah peserta ajar (Student), pengajar (Tutor) dan Admin. Interaksi antara use-case dan aktor terdapat pada gambar 4.17.



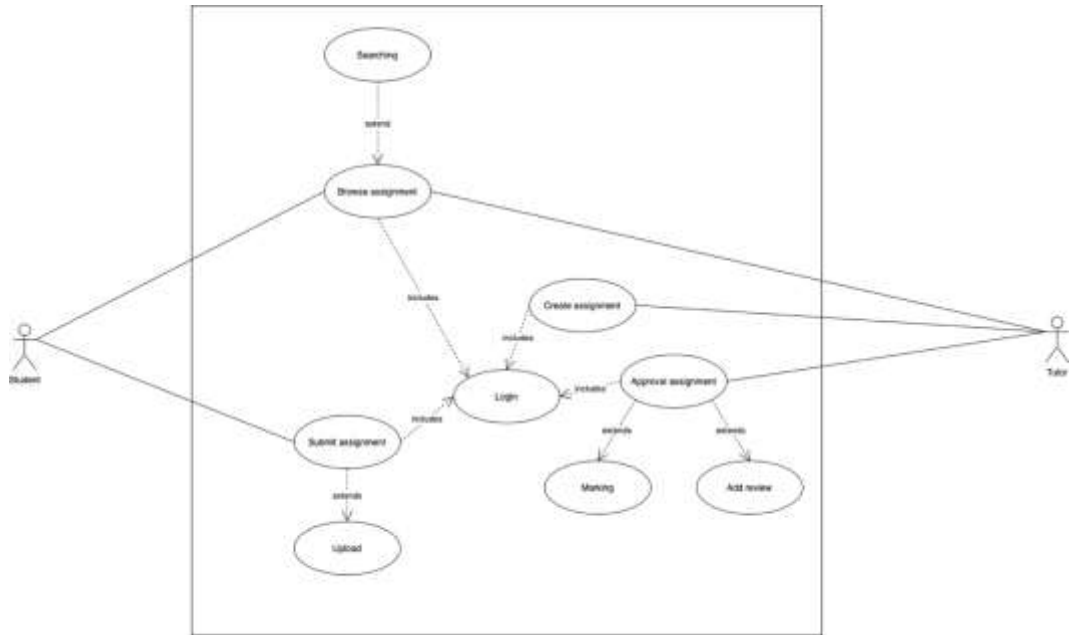
Gambar 4.17. *Business use-case e-learning*

Use-case *Course*, *Assignment*, *Examination* berinteraksi dengan *Student* dan Tutor sebagai bagian dari proses belajar di *e-learning*. Use-case *Payment* berinteraksi dengan *Student* dan Admin sebagai bagian dari proses pendaftaran dan hak akses konten di *e-learning*. Sementara itu, dari pemodelan *domain* diketahui bahwa masing – masing business use-case adalah *sub-domain*. *Sub-domain* mempunyai proses bisnis yang berinteraksi dengan aktor *Student* dan Tutor. Untuk semua use-case pada masing – masing *service* mempunyai ketergantungan terhadap use-case Login.



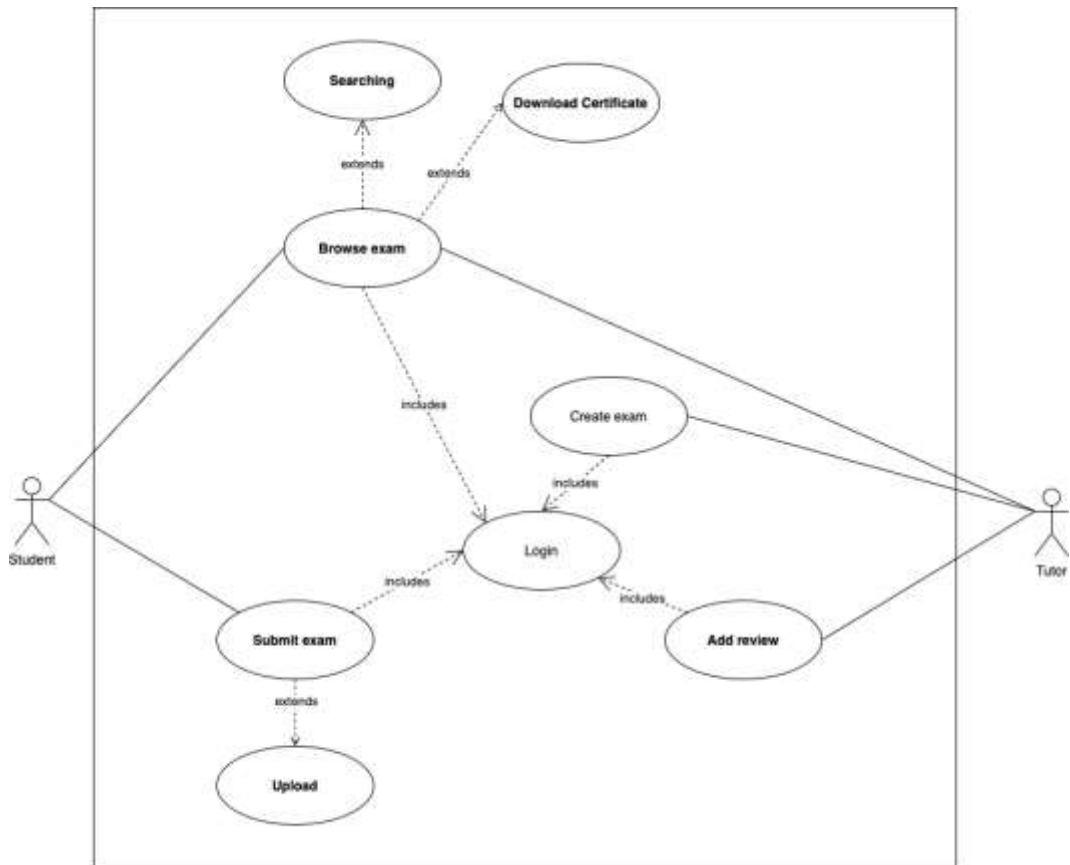
Gambar 4.18. Use-case Course Service

Gambar 4.18 adalah use-case dari *Course Service*. *Course Service* mempunyai 2 aktor yaitu *Student* dan Tutor. *Student* berinteraksi dengan use-case *Browse Content*, *Search Content*, *Download*, *Watch*, *Read*, *See learning path*, *Browse Course*, *See learning progress*, *Search course*, *Start enrol*, *Update enrol*. Tutor berinteraksi dengan use-case *Add Content*, *Add Course*, *Create learning path*, *Browse Content* dan *Browse course*.



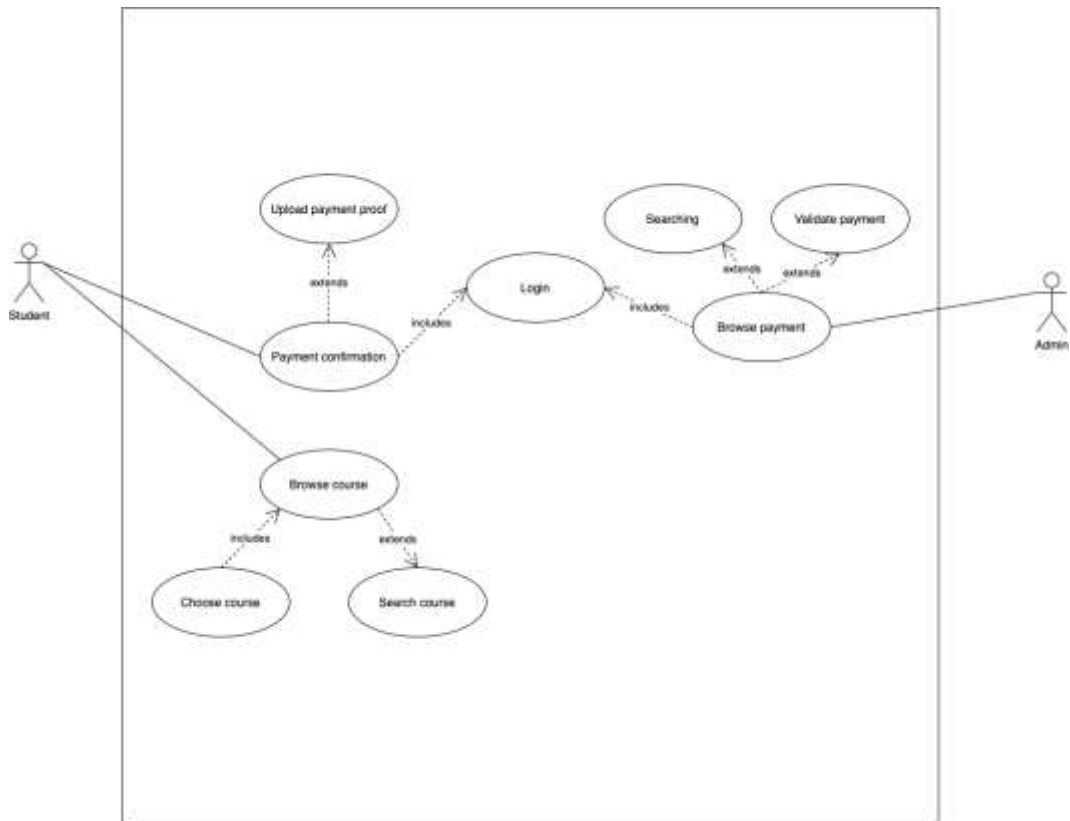
Gambar 4.19. Use-case *Assignment Service*

Gambar 4.19 adalah use-case dari *Assignment Service*. *Assignment Service* mempunyai 2 aktor yaitu *Student* dan Tutor. *Student* berinteraksi dengan use-case *Browse Assignment*, *Searching*, *Submit Assignment*, *Upload*. Tutor berinteraksi dengan use-case *Create Assignment*, *Approval Assignment*, *Marking* dan *Add review*.



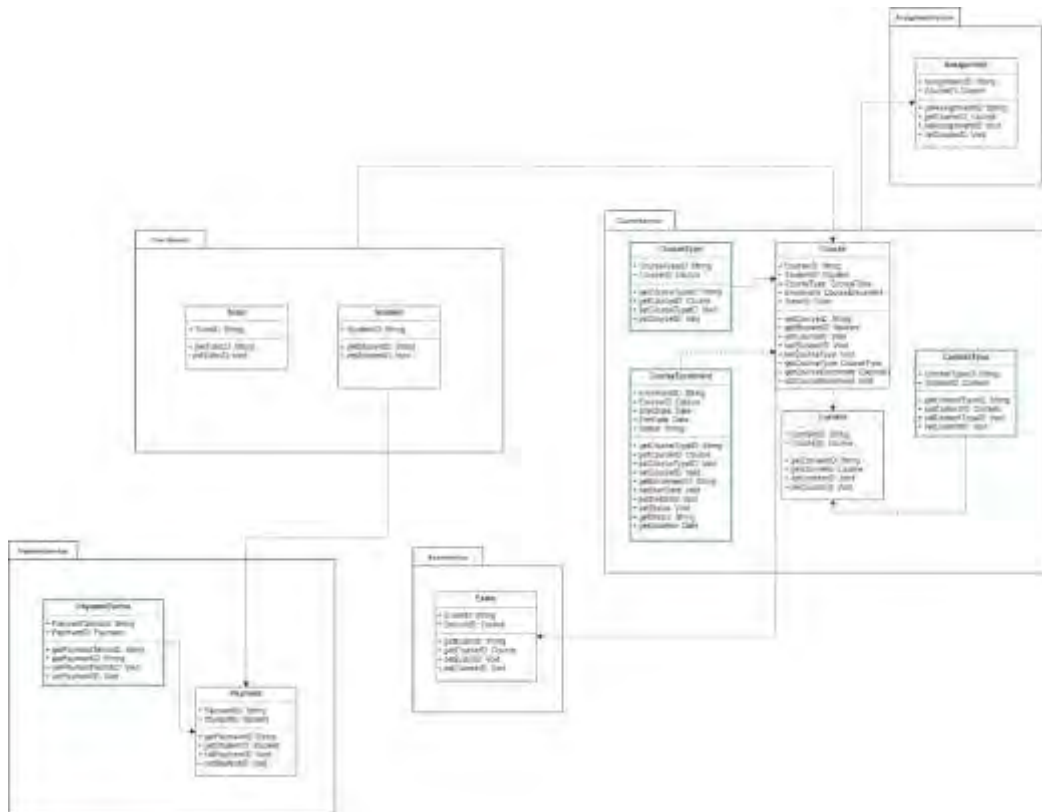
Gambar 4.20. Use-case *Exam Service*

Gambar 4.20 adalah use-case dari *Exam Service*. *Exam Service* mempunyai 2 aktor yaitu *Student* dan Tutor. *Student* berinteraksi dengan use-case *Browse Exam*, *Searching*, *Download certificate*, *Submit Exam*, *Upload*. Tutor berinteraksi dengan use-case *Create Exam* dan *Add review*.



Gambar 4.21. Use-case *Payment Service*

Gambar 4.21 adalah use-case dari *Payment Service*. *Payment Service* mempunyai 2 aktor yaitu *Student* dan *Admin*. *Student* berinteraksi dengan use-case *Browse course*, *Choose course*, *Search course*, *Payment confirmation*, *Upload payment proof*. *Admin* berinteraksi dengan use-case *Browse payment*, *Searching* dan *Validate payment*.



Gambar 4.22. Package diagram *e-learning*

Masing – masing *service* menjalankan bisnis logik tertentu dalam lingkup sebagai modul *microservice*. Modul *microservice* merupakan *package* dalam *e-learning*. Selain bisnis logik, pada *package service* terdapat entitas yang terlibat serta hubungannya. Pada gambar 4.22 di representasikan hubungan antara *package* dan entitasnya.

BAB V KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan uraian diatas, dapat disimpulkan hal – hal sebagai berikut :

1. Metode *Design Thinking* dapat digunakan sebagai metode untuk mengidentifikasi kebutuhan pengguna. Tahapan yang dilalui pada penelitian ini adalah *Empathise*, *Define the problem* dan *Ideate*. *Empathise* mengidentifikasi kebutuhan pengguna melalui teknik observasi pengguna dan *user persona*. *Define the problem* mendefinisikan kebutuhan *e-learning* menggunakan metode *domain driven design*. Tahapan *Ideate* menerapkan model *e-learning* pada *arsitektur microservice*.
2. Metode *Domain Driven Design* dapat digunakan sebagai pendekatan untuk membuat model *e-learning*. Teknik yang digunakan pada pemodelan *domain* antara lain identifikasi problem dan *solution space* pada *domain e-learning*, identifikasi *aggregates*, identifikasi *Entities* dan *value object*, definisikan proses bisnis, identifikasi *Command* dan *queries*. Selanjutnya, membuat model *event publish* dan *subscribe* untuk setiap *bounded context*.
3. *E-learning* dengan pendekatan *Asynchronous* dapat dibangun dengan *arsitektur microservices*.
4. Dari hasil observasi pengguna menggunakan metode *Design Thinking* ditemukan bahwa penerimaan *e-learning* saat ini sudah cukup baik. Terdapat beberapa kebutuhan tambahan seperti materi yang beragam, kebutuhan mengetahui jalur belajar (*learning path*), mengetahui kemajuan belajar (*learning progress*), cara belajar yang dinamis dapat per konten atau dengan target waktu belajar, tersedia tugas di setiap konten ajar untuk mengasah kemampuan, tersedia *review* dengan tutor, tersedia ujian untuk validasi kemampuan, sertifikasi kompetensi dan cara pembayaran yang leluasa serta harga yang terjangkau.
5. Pengembangan dari observasi tersebut menggunakan *Domain Driven Design* di identifikasi *sub-domain* dari antara lain *Course*, *Assignment*, *Examination* dan *Payment*. Implementasi dari model *domain e-learning* menggunakan *arsitektur microservices*.

5.2. Saran

Pengembangan selanjutnya dari penelitian ini adalah memadukan pendekatan *Asynchronous* dan *synchronous*. Sehingga, dapat memberikan alternatif rancangan *e-learning* yang lebih komprehensif. Selain itu, metode *Design Thinking* dapat dimaksimalkan penggunaannya dengan menerapkan langkah *Prototyping* dan *Testing*. Langkah tersebut dilakukan agar teridentifikasi *e-learning* yang lebih tepat guna bagi peserta ajar. Kemudian *e-learning* yang dibangun dapat berfokus pada pendidikan dasar, menengah atau tinggi. Fokus tersebut sebagai alternatif dari *e-learning* yang digunakan pada penelitian ini yaitu *e-learning* untuk peningkatan kompetensi keahlian di bidang teknologi informasi.

Kendala yang dihadapi dalam penelitian ini adalah menetapkan hasil yang efektif terhadap kebutuhan pengguna. Metode *Design Thinking* yang dilakukan belum secara menyeluruh dilaksanakan. Iterasi dalam pengembangan sistem menggunakan *Design Thinking* sangat diperlukan. Namun, peneliti mencoba untuk menggali sejauh mana *Design Thinking* dapat membantu perancangan *e-learning* secara mendasar.

Kemudian, metode *Design Thinking* dan *Domain Driven Design* dapat diaplikasikan oleh pihak – pihak yang berkepentingan di dunia pendidikan, baik institusi pendidikan tinggi, menengah, ataupun lembaga independen lainnya. Hal ini akan bermanfaat dalam perancangan arsitektur *e-learning* yang dapat secara efektif dan efisien dikembangkan serta dapat beradaptasi dengan perkembangan teknologi dan bisnis di masa depan.

DAFTAR PUSTAKA

- Dennis, A., Wixom, B. H., & Tegarden, D. P. (2015). *System analysis & design*, an object-oriented approach with UML. Hoboken, NJ: Wiley.
- Dennis, A., Roth, R. M., & Wixom, B. H. (2012). *System Analysis and Design*, Fifth Edition. John Wiley & Sons.
- E, Evans. (2003). *Domain-Driven Design: Tackling Complexity in the Heart of Software*. Addison Wesley
- Emrah Yayici, (2016). *Design Thinking Methodology Book*. ArtBizTech
- Ghirardini, B. (2011). *E-learning methodologies: A guide for designing and developing e-learning courses*. Rome: Food and Agriculture Organization of the United Nations.
- Gallipeau D and Kurdle S. (2018), "Microservices: Building blocks to New Workflows and Virtualization", SMPTE Motion Imaging Journal
- Karnawan, Gusti & Andryana, Septi & Titi, Ratih. (2020). Implementasi User Experience Menggunakan Metode *Design Thinking* Pada Prototype Aplikasi Cleanstic. Jurnal Teknologi dan Manajemen Informatika. 6. 10.26905/jtmi.v6i1.3785.
- Munawar, Ghifari & Hodijah, Ade. (2018). Analisis Model *Arsitektur Microservice* Pada Sistem Informasi DPLK. Sinkron. 3.
- Meinel, Martin & Eismann, Tobias & Baccarella, Christian & Fixson, Sebastian & Voigt, Kai-Ingo. (2020). Does applying *Design Thinking* result in better new product concepts than a traditional innovation approach? An experimental comparison study. European Management Journal. 38. 10.1016/j.emj.2020.02.002.
- Newman Sam, (2015). *Building Microservice: Designing Fine Grained System*, O'Reilly Media.
- Nedeltcheva, Galia & Shoikova, Elena. (2017). Coupling *Design Thinking*, User Experience *Design* and Agile. 10.1145/3175684.3175711.
- Oleynik, P.P. & Kuznetsov, N.V. & Galiaskarov, Edward & Kozlova, K.O.. (2015). *Domain-driven design* of information system for queuing system in terms of unified metamodel of object system. International Journal of Applied Engineering Research. 10. 35229-35238.
- Razi, Aria & Mutiaz, Intan & Setiawan, Pindi. (2018). PENERAPAN METODE *DESIGN THINKING* PADA MODEL PERANCANGAN UI/UX APLIKASI PENANGANAN LAPORAN KEHILANGAN DAN TEMUAN BARANG TERCECER. Desain Komunikasi Visual, Manajemen Desain dan Periklanan (Demandia). 3. 75. 10.25124/demandia.v3i02.1549.
- Sendiang, Maksy. Kasenda, Sonny. Purnama Jerry, "Implementasi Teknologi *Microservice* pada Pengembangan Mobile Learning", 2018, Journal of Applied Informatics and Computing (JAIC), Vol.2 No.2, Desember 2018, 63-66.
- Steinegger, Roland & Giessler, Pascal & Hippchen, Benjamin & Abeck, Sebastian. (2017). Overview of a *Domain-Driven Design* Approach to Build *Microservice*-Based Applications.
- Vijay Nair. (2019). *Practical Domain-Driven Design in Enterprise Java*. Appress.
- "What Is *Design Thinking*?" The Interaction *Design* Foundation, www.interaction-design.org/literature/topics/design-thinking. Di akses pada 30 Juni 2020 pukul 14.00 WIB

LAMPIRAN

Lampiran A : Formulir Kuisioner

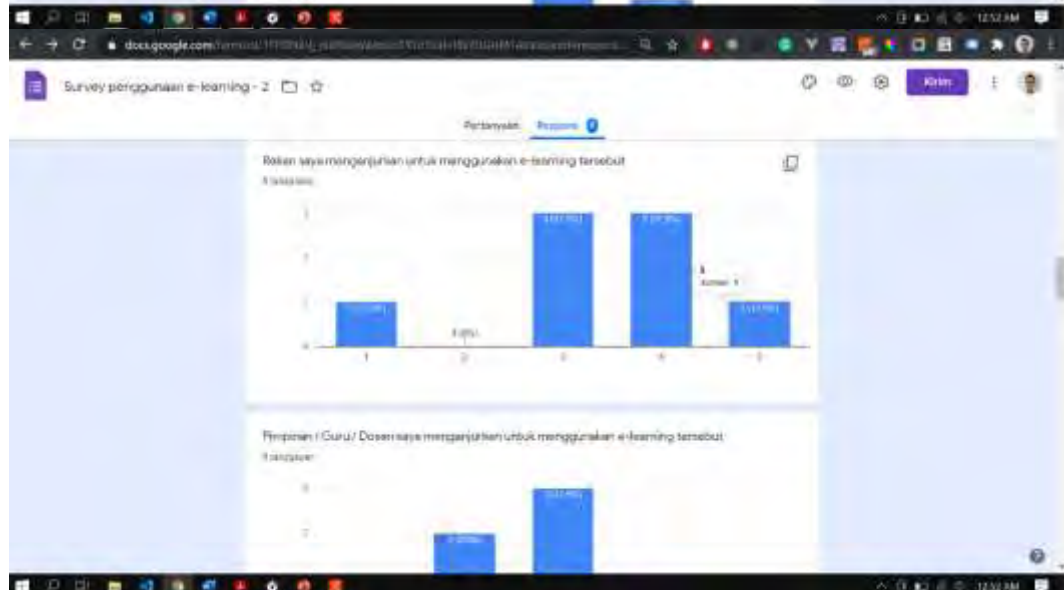
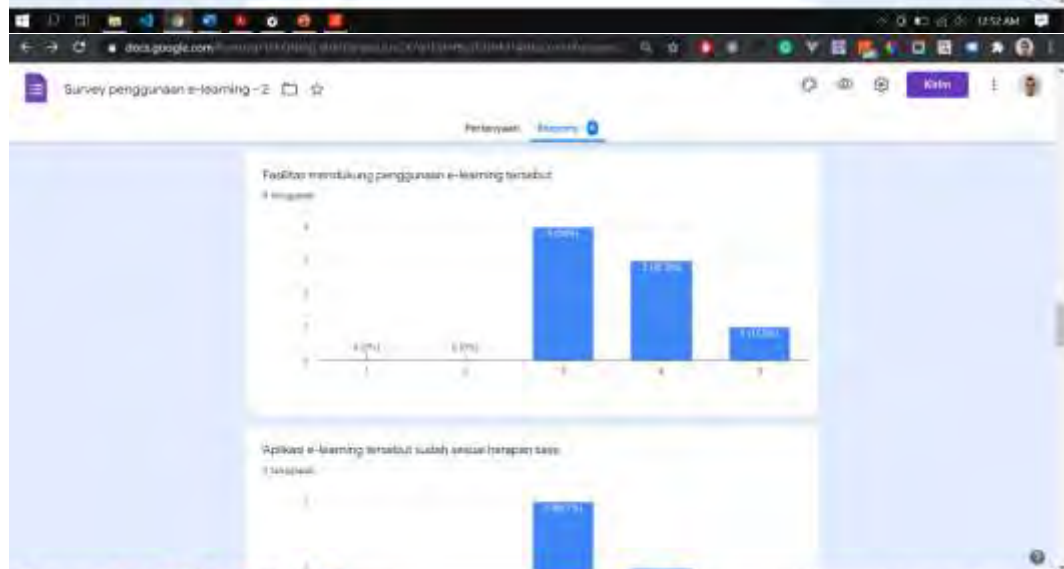
The image displays two screenshots of a Google Forms survey titled "Survey penggunaan e-learning".

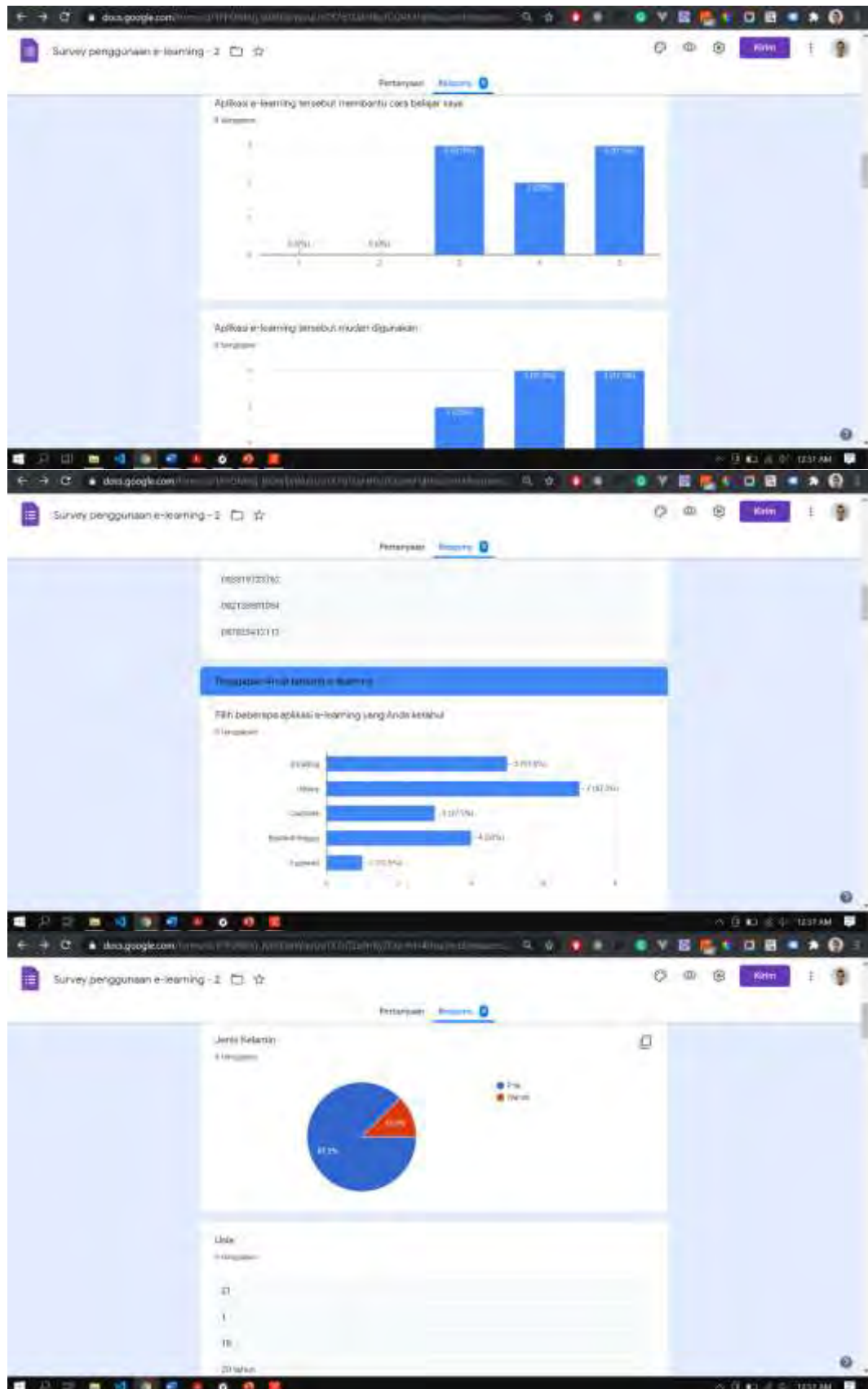
Top Screenshot (Pertanyaan - Questions):

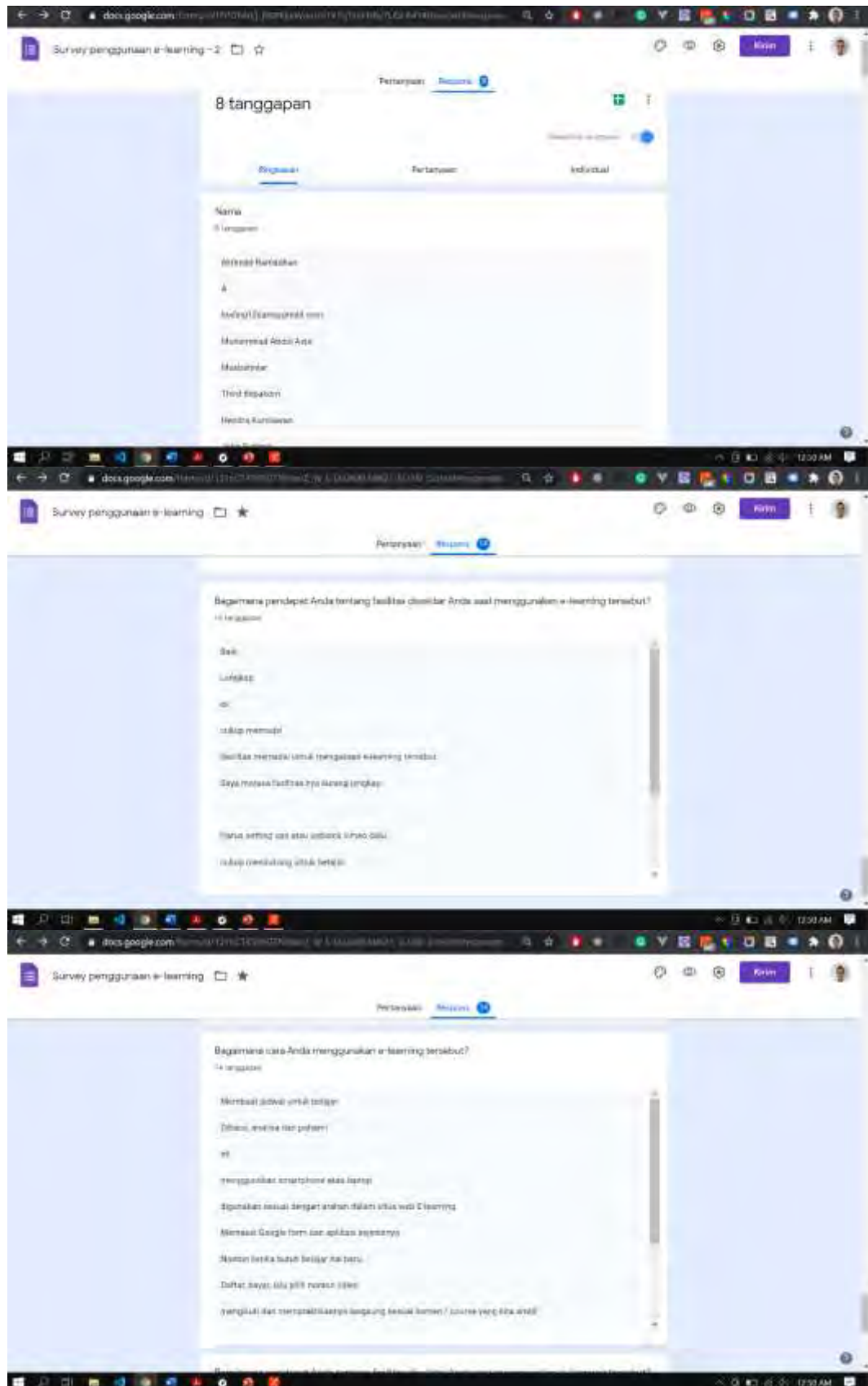
- Question 1: "Apakah Anda lakukan saat menggunakan e-learning tersebut?" (Do you do when using e-learning?).
 - Response options: Tidak ada (None), Bayar (Pay), A, Tidak ada (None), Pengantar dan paham (Introduction and understanding), Tidak lengkap materi yang diberikan, dan tidak ada penyampaian materi (Incomplete material given and no material delivery), Dapat akses (Can access).
- Question 2: "Bagaimana pengaruh rekan atau pimpinan Anda untuk menggunakan e-learning tersebut?" (How is the influence of your colleague or supervisor to use e-learning?).
 - Response options: Tidak ada (None).

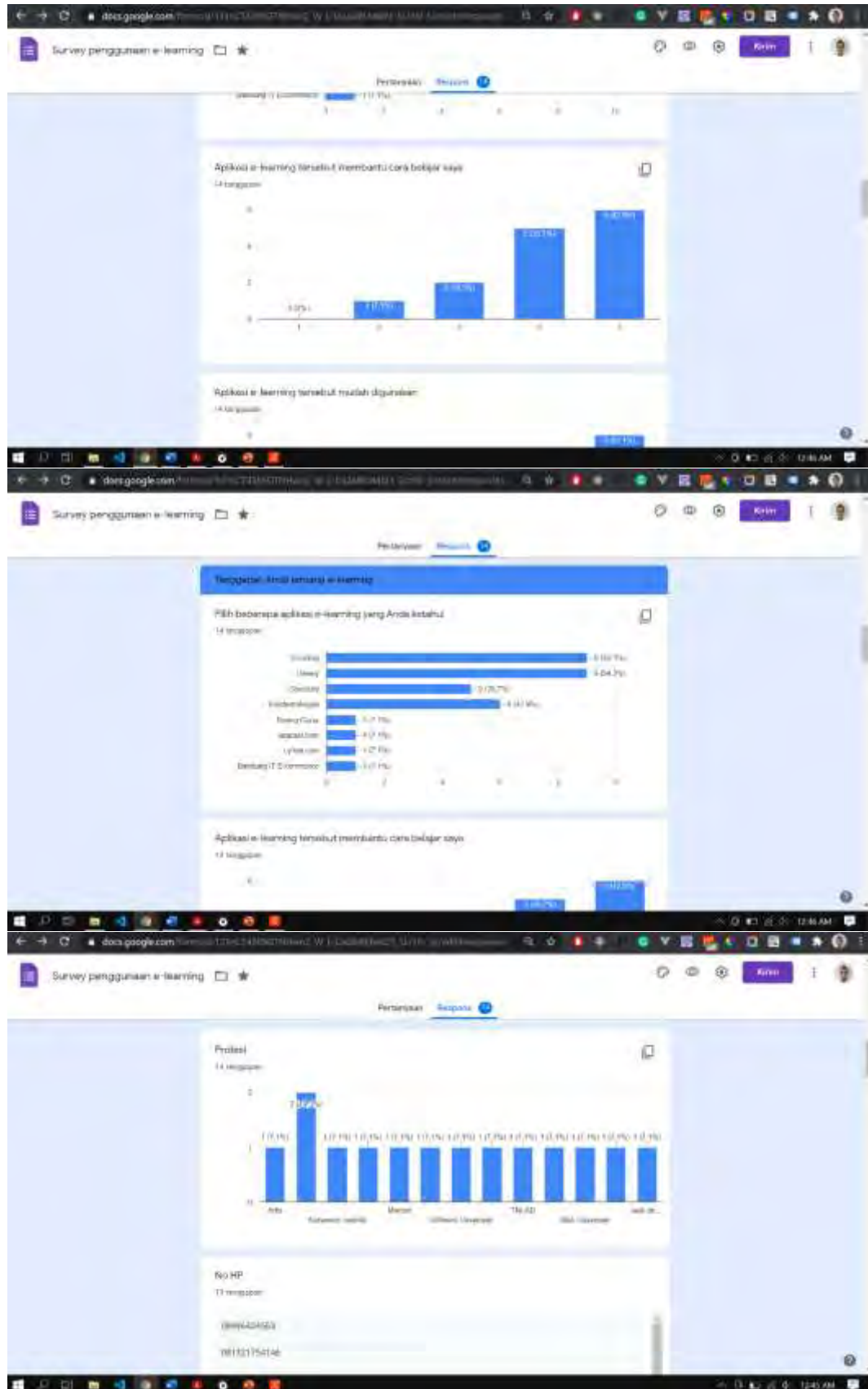
Bottom Screenshot (Jawaban - Answers):

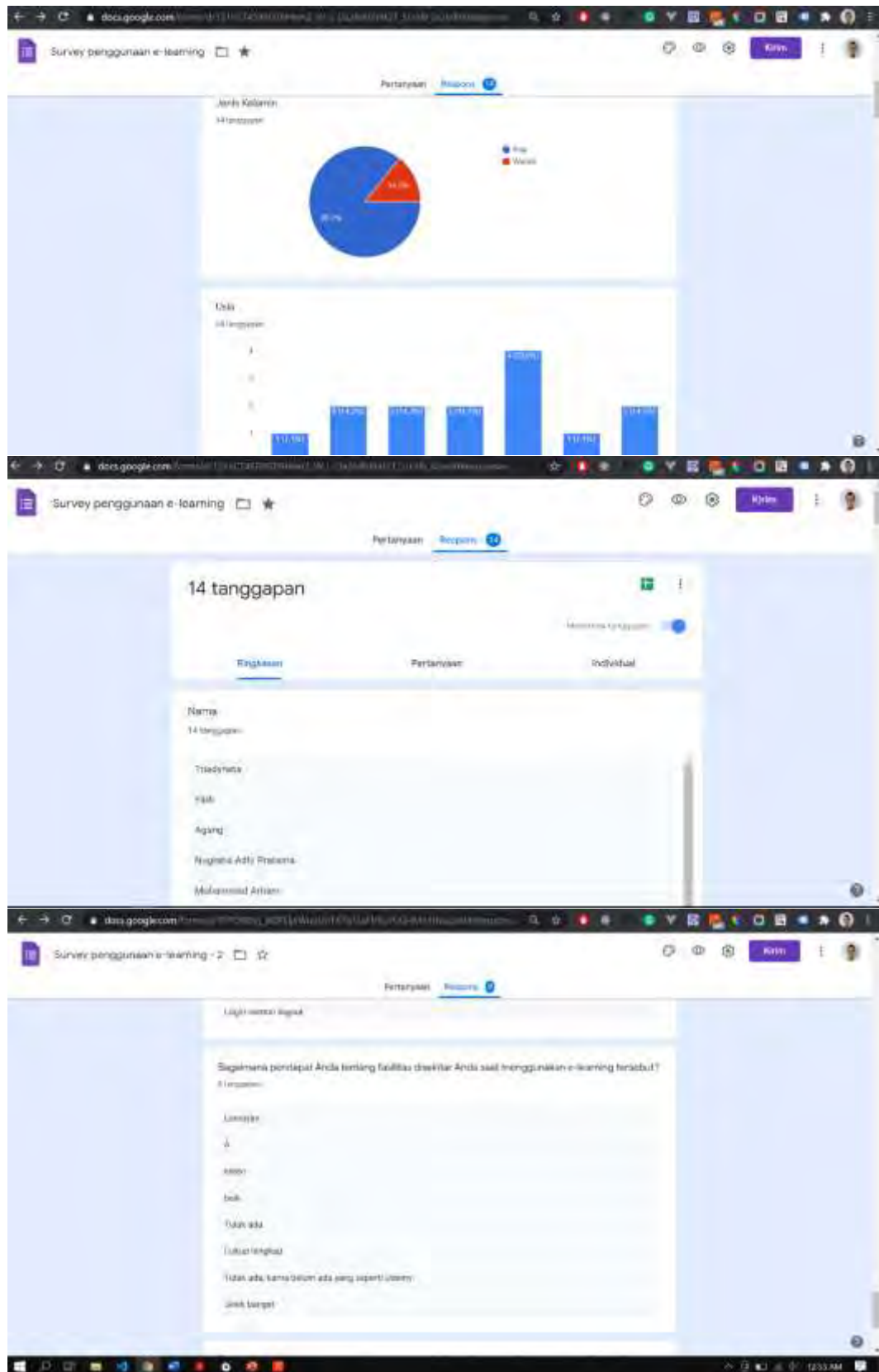
- Question 1: "Bagaimana tanggapan Anda tentang kesulitan yang dihadapi saat menggunakan e-learning tersebut?" (How is your response to the difficulties encountered when using e-learning?).
 - Response options: Kesulitan Bayar (Difficulty Paying), A, Menakutkan saja (Just scary), Terkadang saat ada beberapa yang enter dan ngetik dan ternyata seperti slow loading (Sometimes when there are a few who enter and type and it turns out like slow loading), Tidak ada kesulitan (No difficulty), Kadang sudah merasa nyaman yang banyak (Sometimes already feel comfortable with a lot), Kadang tidak up to date, kalaupun login dan tidak (Sometimes not up to date, even if login and not), Sifat lainnya (Other characteristics).
- Question 2: "Apakah Anda lakukan saat menggunakan e-learning tersebut?" (Do you do when using e-learning?).











Lampiran B : Spreadsheet Response Kusioner

https://docs.google.com/spreadsheets/d/1i0IGszKb_gSzwk1RJ680JbhZYZVZxLSSnQoiO_vlol/edit?usp=sharing

https://docs.google.com/spreadsheets/d/19G36ot_Kgw27s2l4ARsN91eJYxv2fAYVyHIm_r1e8V0/edit?usp=sharing