

## BAB III

### PELAKSANAAN KERJA PROFESI

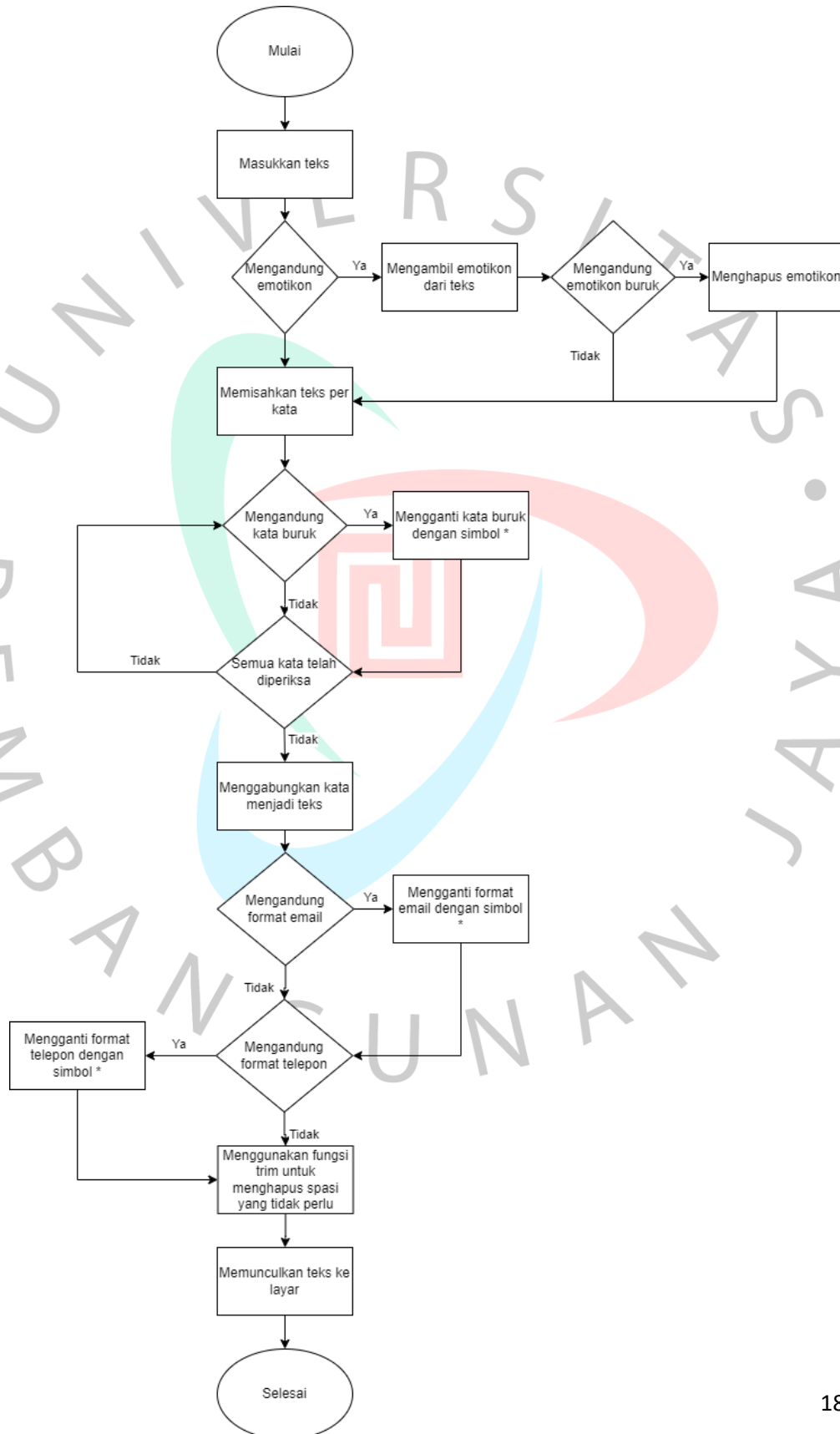
#### 3.1 Bidang Kerja

Selama masa Kerja Profesi Praktikan bertugas di PT.Code Development Indonesia. Sementara divisi yang memberikan penulis tanggung jawab pekerjaan adalah Divisi *Maintenance*. Selama masa kerja, praktikan diberikan tanggung jawab untuk mengerjakan sebuah prototipe aplikasi *live streaming* untuk konser musik dari *client* Telkomsel. Prototipe tersebut terdiri dari 2 aplikasi, yaitu aplikasi *mobile (android dan ios)*, dan website. Untuk membangun aplikasi tersebut dibutuhkanlah juga *backend* untuk menyediakan dan mengolah data dari database yang akan ditampilkan di dalam aplikasi. Praktikan ditugaskan dalam membangun *backend* untuk aplikasi tersebut, pekerjaan *backend* antara lain, membuat design database yang diperlukan untuk aplikasi, membuat API (*Application Programming Interface*) sebagai jalur komunikasi antara *backend* dengan *frontend*, melakukan *research* mengenai teknologi yang akan digunakan di dalam aplikasi serta mengimplementasikannya.

#### 3.2 Pelaksanaan Kerja

Pelaksanaan kerja profesi dilakukan selama selama 50 hari kerja dan 400 jam kerja yang dimulai dari tanggal 1 Agustus 2022 sampai dengan 9 Oktober 2022. Waktu dan hari kerja mengikuti peraturan jam kerja dari perusahaan yaitu, dengan hari kerja Senin sampai Jumat pada pukul 08:30 WIB sampai dengan pukul 17:30 WIB. Selama pelaksanaan kerja profesi berlangsung penulis melakukan pengembangan aplikasi *live chat* di sisi *backend* untuk dijadikan sebagai acuan dalam pengembangan aplikasi selanjutnya yang diinginkan oleh *client*. Pengembangan di *backend* ini menggunakan bahasa pemrograman java, karena proyek yang sedang berjalan saat ini menggunakan bahasa pemrograman java juga, agar ketika dilakukan pengembangan selanjutnya tidak terjadi kesulitan. Untuk basis data, aplikasi ini menggunakan PostgreSQL, sistem manajemen database relasional (RDBMS) yang bersifat *open source*. Dalam pengembangan *backend* ini juga menggunakan *framework* dari java yaitu spring boot, untuk memudahkan dalam pengembangan serta membuat sintaks program menjadi lebih teratur.

### 3.2.1 Flowchart Diagram



Gambar 3. 1 Flowchart Diagram Aplikasi Live Chat

Gambar 3.1 diatas menggambarkan bagaimana alur kerja prototipe aplikasi *live chat* dengan filter *bad word* dan *bad emoticon*, yang jika diuraikan seperti berikut :

- a. Memasukkan input berupa teks ke dalam kolom chat.
- b. Melakukan validasi apakah terdapat *emoticon* di dalam teks, jika ada, *emoticon* akan di validasi apakah mengandung *bad emoticon*, jika mengandung *bad emoticon*, *bad emoticon* akan dihapus, jika tidak mengandung *bad emoticon*, *emoticon* akan dibiarkan.
- c. Memisahkan text menjadi perkata.
- d. Kata tersebut akan di validasi apakah mengandung *bad word*, jika mengandung *bad word* akan kata tersebut akan diganti menjadi simbol \*, validasi dilakukan sampai semua kata telah di validasi.
- e. Menggabungkan kembali kata ke dalam text utuh.
- f. Melakukan pengecekan apakah terdapat format email di dalam teks, jika ada format email tersebut akan diganti menjadi simbol \*.
- g. Melakukan pengecekan apakah terdapat format nomor telepon di dalam teks, jika ada format nomor telepon tersebut akan diganti menjadi simbol \*.
- h. Melakukan *trim function* yaitu menghapus spasi yang tidak diperlukan di dalam teks.
- i. Menampilkan teks yang sudah di filter.

### 3.2.2 Teknologi dan Alat

Teknologi dan *tools* yang dibutuhkan untuk membuat prototipe aplikasi ini yaitu :

- a. Java

Java adalah sebuah bahasa pemrograman yang diciptakan oleh James Gosling. Java mempunyai fleksibilitas dalam pengekseskuan programnya karena Java mempunyai sebuah mesin yang dapat dijalankan di device manapun yang bernama *Java Runtime Environment* (JRE). Sesuai dengan slogan Java yaitu, "*write once, run anywhere*". Java memiliki konsep pemrograman berorientasi objek dan biasanya dipakai untuk mengembangkan aplikasi – aplikasi untuk *enterprise*, mulai dari aplikasi *mobile*, *desktop*, ataupun *website*.



Gambar 3. 2 Logo Java

b. Spring Boot

Spring Boot adalah sebuah *framework* java yang dapat membantu seorang *programmer* java lebih mudah menulis kode program, dan membuat kode program menjadi lebih struktur.(Webb et al., n.d.) Dengan memakai *framework* juga akan memudahkan untuk memelihara kode dan mengembangkan kode jika ada perubahan atau penambahan berikutnya.



Gambar 3. 3 Logo Spring Boot

c. Application Programming Interface (API)

*Application Programming Interface* (API) atau Antarmuka Pemrograman Aplikasi adalah sebuah fungsi atau protokol yang digunakan untuk berkomunikasi antara sistem satu dengan sistem lainnya. Antarmuka Pemrograman Aplikasi mengekspos layanan atau data yang disediakan oleh aplikasi perangkat lunak melalui sumber yang telah ditentukan, seperti metode, objek, atau *URL*(Sylos, Faulring, Yang & Myers,2009). Dengan menggunakan API, aplikasi lain dapat mengakses data atau layanan tanpa harus

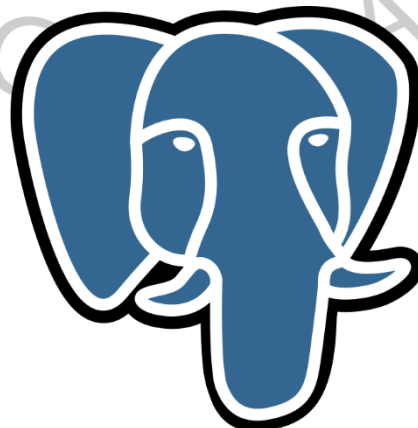
mengimplementasikan objek dan prosedur yang mendasarinya. (Meng et al., 2018)

d. Websocket

Websocket adalah sebuah teknologi yang memungkinkan terjadinya komunikasi *full-duplex* dengan menggunakan satu socket yang beroperasi melalui web. Biasanya teknologi ini digunakan untuk membangun sistem komunikasi *web real-time* yang terukur. Koneksi *Web Socket* menggunakan port HTTP standar (80 dan 443). Koneksi *Web Socket* dapat dibuat selama klien dan server mendukung *Web Socket protocol*. Untuk membangun sebuah aplikasi *live chat* berbasis *website* maka *web socket* adalah teknologi yang dibutuhkan. (Liu & Sun, 2012)

e. PostgreSQL

PostgreSQL merupakan salah satu sistem basis data yang terkenal, dikenal sebagai *open source* objek *relational database system* yang memiliki keandalan, ketangguhan fitur, dan kinerja. PostgreSQL termasuk jenis sistem database ORDBMS (Object Relational Database Management System) yang merupakan pengembangan dari Sistem Management Basis Data Relasional (RDMS) yang berupaya mencoba untuk menjembatani antar paradigma relasional serta berorientasi objek. PostgreSQL dapat dijalankan di banyak sistem operasi besar, telah menerapkan ACID (*atomicity, consistency, isolation, durability*) sejak 2001. Sehingga PostgreSQL sudah menjamin validitas data meskipun terjadi kesalahan, kegagalan daya, dan ketidaksesuaian lainnya. Fitur PostgreSQL sangat lengkap mulai dari tipe data, integritas data, keamanan, pemulihan bencana, standar internasional, pencarian text, dan lainnya. (The PostgreSQL Global Development Group, 2022)



Gambar 3. 4 Logo PostgreSQL

f. Gitlab

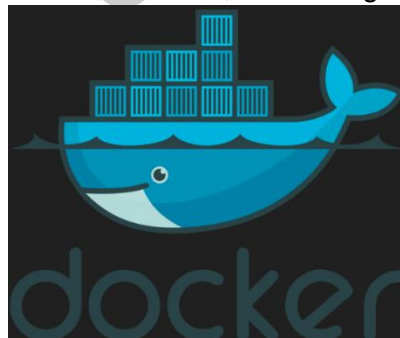
Gitlab adalah sebuah *service* yang menggunakan git. Git adalah sebuah *software* yang memudahkan developer. *Software* ini berguna sebagai *Version Control System (VCS)* yang memiliki tugas melacak perubahan pada suatu file, bagi developer file tersebut adalah code. Hal ini tentu sangat membantu developer untuk berkolaborasi sesama developer untuk menyelesaikan proyek atau pekerjaannya.



Gambar 3. 5 Logo Gitlab

g. Docker

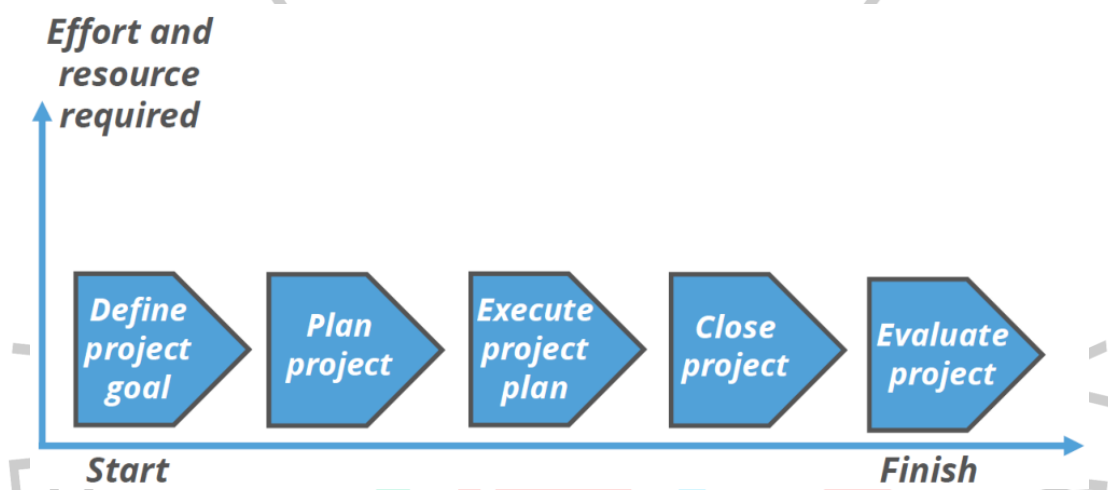
Docker adalah sebuah layanan yang membantu developer untuk membuat, menguji, dan menerapkan aplikasi yang dikembangkan. Docker mengemas perangkat lunak ke dalam unit standar yang disebut *container*. Dengan adanya *container* developer bisa mengemas aplikasi dan menjalankan aplikasi secara terpisah. layanan ini bekerja dalam sebuah container yang tidak dapat dipengaruhi oleh lingkungan dari luarnya. Di dalam container tersebut, docker berperan untuk menjadi sebuah sistem operasi yang digunakan untuk membuat, memulai, atau menghentikan *container*.



Gambar 3. 6 Logo Docker

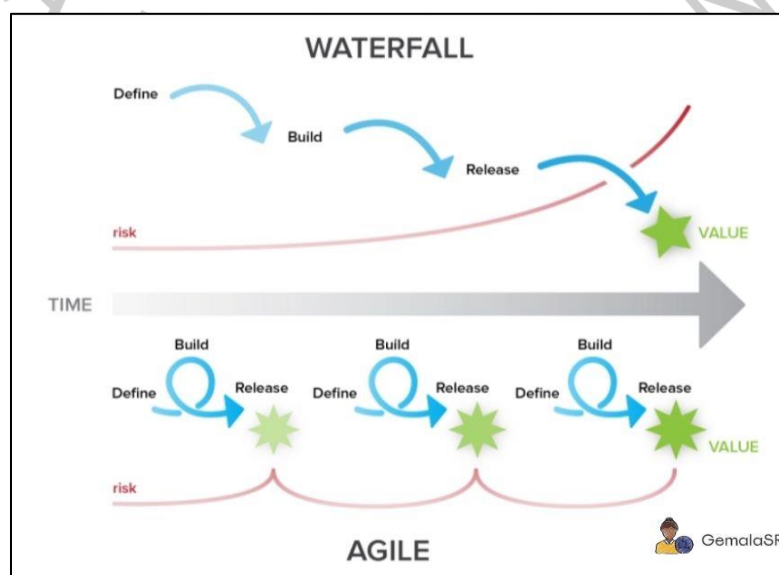
### 3.2.3 Metode Dalam Pengembangan Aplikasi

Dalam manajemen proyek pengembangan aplikasi di CODE ID atau biasa disebut Software Development Life Cycle (SDLC) beberapa proyek menggunakan metode Waterfall dan beberapa proyek lainnya menggunakan Agile Scrum. Berikut ini adalah ilustrasi tahapan *project life cycle*.



Gambar 3. 7 Software Development Life Cycle

Metode Agile adalah ditandai dengan kemampuan siap untuk bergerak dengan anggun dan cepat,” seperti seorang penari yang gesit (Merriam-Webster, 2021). Metode ini meringkaskan SDLC menjadi iterasi atau biasa disebut *sprint*. Seperti yang terlihat pada Gambar 3.8 yang merupakan perbedaan Agile dan Waterfall.



Gambar 3. 8 Waterfall dan Agile

Pengguna dan pengembang bekerja sama untuk menentukan fitur yang diprioritaskan. Dalam Scrum terdiri dari beberapa peran yaitu :

1. Scrum Master

Adalah seseorang yang bertanggung jawab proyek berjalan dengan semestinya

2. Product Owner

Adalah seseorang yang bertanggung jawab untuk memaksimalkan nilai yang dihasilkan oleh tim berdasarkan *backlog* yang sudah disepakati dengan kebutuhan pelanggan/*client*.

3. Development Team

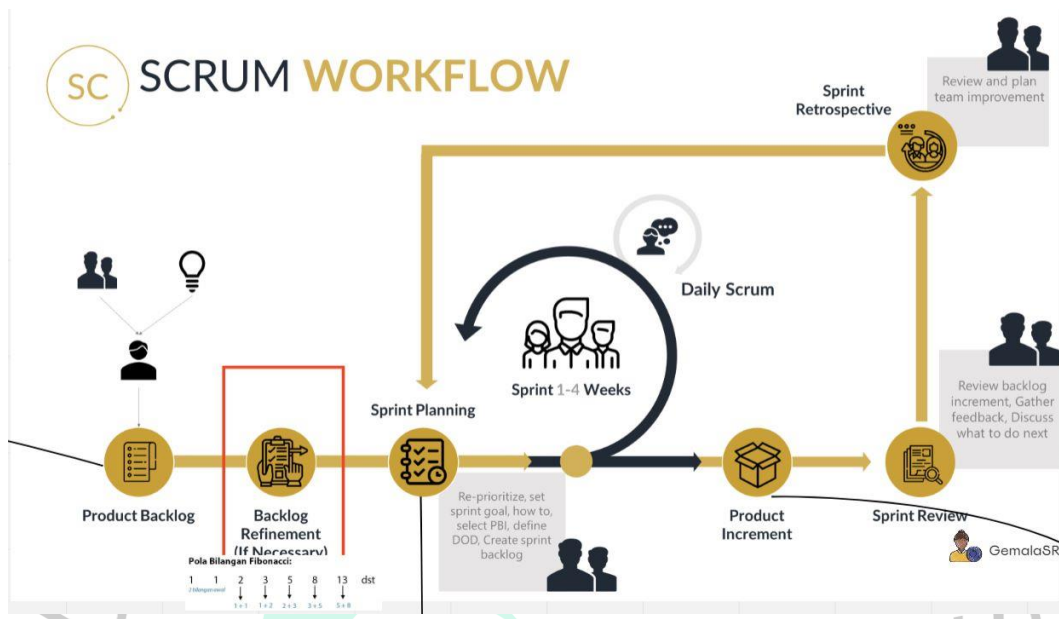
Adalah orang yang bertanggung jawab membuat program software yang sesuai dengan kebutuhan *client* berjalan lancar dengan semestinya, meliputi tim developer, automation, testing, dan lain-lain.



Gambar 3. 9 Scrum Roles

Seperti yang terlihat pada Gambar 3.9 setiap peran dalam scrum memiliki fungsi dan tanggung jawabnya masing - masing





Gambar 3. 10 Scrum Workflow

Istilah – Istilah pada Scrum. Seperti yang terlihat pada Gambar 3.10 Scrum memiliki Workflow yang berbeda dari Waterfall, karena scrum bersifat iterasi maka workflownya adalah sebagai berikut :

1. Product Backlog

Daftar mengenai tugas yang akan dikerjakan untuk membuat produk di dalam sprint yang dilakukan. Product Backlog berasal dari Product Owner yang menggabungkan ide dan kebutuhan pengguna di lapangan. Product Backlog adalah sumber pekerjaan yang digunakan oleh tim Scrum.

2. Backlog Refinement (Pendetailan Backlog)

Tahap ini adalah tahap yang boleh dilakukan dan boleh tidak. Pada tahap ini Backlog yang sudah diberikan oleh Product Owner akan di bahas dan di detailkan satu persatu oleh tim scrum.

3. Sprint Planning

Pada tahap ini backlog di list dan di prioritaskan mana yang akan dibawa ke dalam sprint yang akan dijalani. Selain itu pada tahap ini juga menambahkan tujuan di sprint yang akan dijalankan serta menambahkan *Definition Of Done (DOD)* dalam setiap backlog. Durasi dari Sprint Planning adalah dua kalinya Panjang sprint, jika Panjang sprint adalah 2 minggu, maka durasi Sprint planning adalah 4 jam.

4. Sprint

Sprint adalah iterasi yang dilakukan dengan batas waktu 1 – 4 minggu. Sprint ini ditentukan oleh tim scrum dan tidak bisa diubah, jika tim scrum menyepakati sprint 2 minggu, maka sampai seterusnya sprint akan berjalan selama 2 minggu. Ini dimaksudkan untuk menjaga kestabilan serta pengukuran tim dalam mengerjakan product atau proyek sehingga dapat dilakukan peningkatan untuk pengembangan selanjutnya.

## 5. Daily Scrum

*Daily Scrum* adalah *Meeting* harian selama 15 menit oleh tim scrum untuk membahas *progress* pekerjaan yang telah dilakukan ataupun kendala yang dialami dalam mengerjakan pekerjaan tersebut.

## 6. Sprint Review

Sprint *Review* adalah tahap dimana tim scrum mereview backlog yang dikerjakan selama di sprint kali ini. Selain itu tim juga bisa memberikan tanggapan mengenai sprint serta mendiskusikan apa yang perlu dilakukan selanjutnya. Pelaksanaan Sprint *Review* berada di akhir sprint.

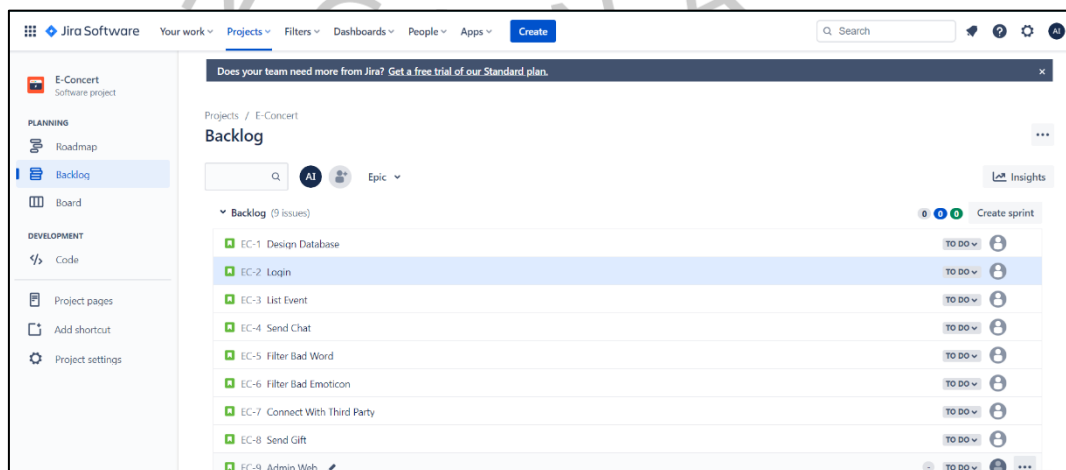
## 7. Sprint Retrospective

Sprint Restrospective dilakukan setelah Sprint *Review*, tahap ini diperlukan untuk mereview dan merencanakan bagaimana agar tim bisa melakukan peningkatan dalam sprint selanjutnya

### a. Dokumentasi Penerapan Scrum dalam Pengembangan Aplikasi di Code Id

#### Product Backlog

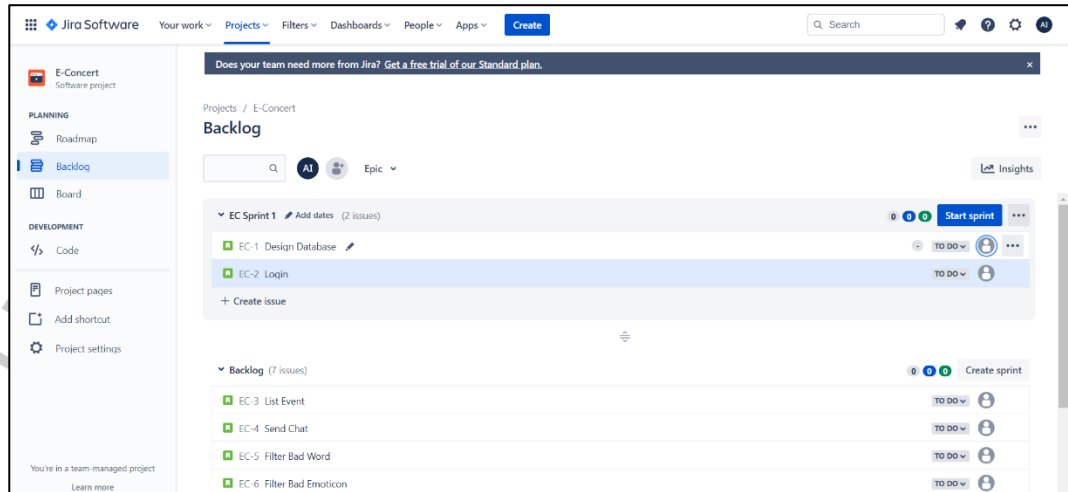
Product Backlog disusun oleh *head of maintenance* dengan tujuan akhir produk berupa aplikasi *e-concert*, yaitu aplikasi untuk menonton konser dengan dalam jaringan (daring). Berikut adalah product backlog yang disusun



Gambar 3. 11 Product Backlog Aplikasi

## Sprint planning

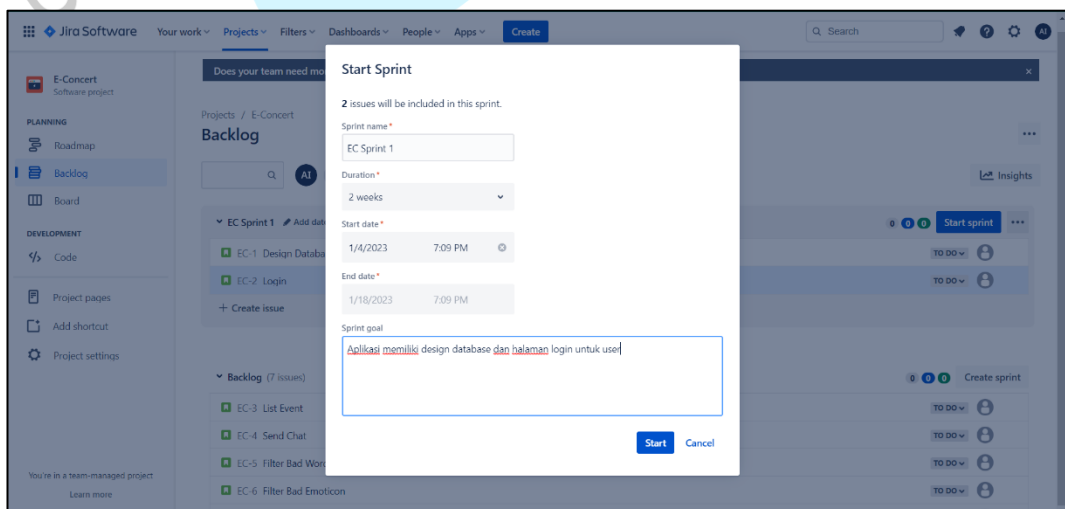
Pada saat Sprint planning, tim scrum akan memasukkan backlog yang sudah disusun ke dalam sprint yang akan dijalankan, berikut adalah backlog yang akan dikerjakan di sprint 1 yaitu Design database dan Login seperti gambar berikut



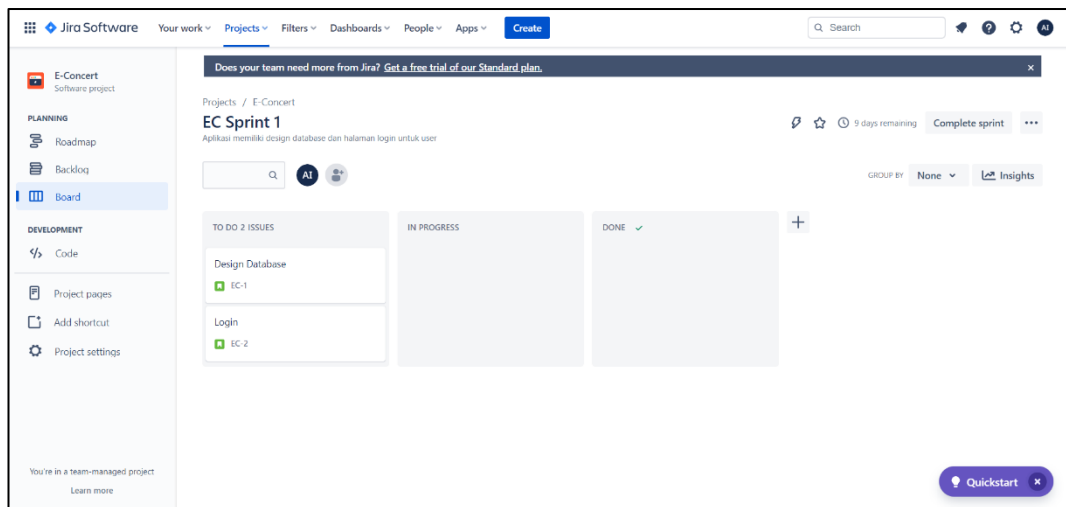
Gambar 3. 12 Task Pada Sprint 1

## Sprint

Setelah sprint planning dilakukan tim akan melakukan start sprint yang merupakan tanda dimulainya sprint, dan tim harus mengerjakan tugas yang ada pada sprint tersebut sampai sprint selesai, berikut ini tampilan ketika sprint dimulai.



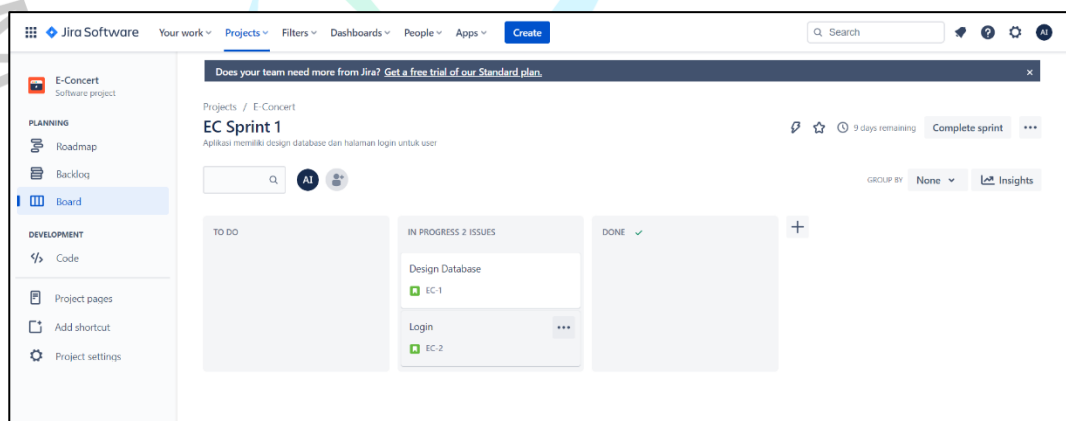
Gambar 3. 13 Start Sprint



Gambar 3. 14 Tampilan Awal Setelah Start Sprint

### Daily Scrum

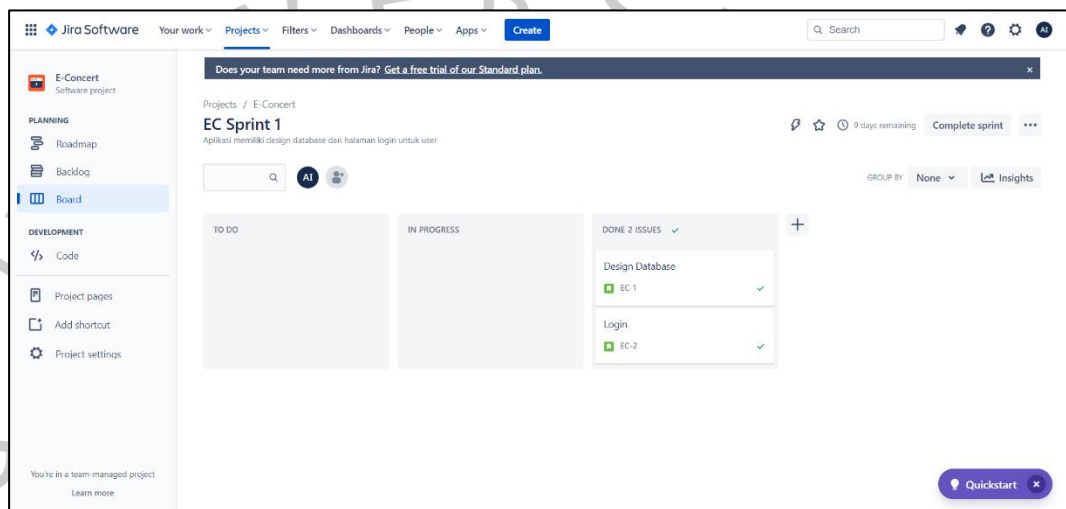
Daily scrum adalah proses dimana tim scrum melakukan meeting setiap 15 menit setiap hari untuk membahas pekerjaan pada sprint yang dijalankan, tahap ini menjelaskan kedua backlog pada gambar diatas sudah sejauh mana prosesnya, apakah ada kendala atau tidak. Pada tahap ini seharusnya task sudah berada di status in progress.



Gambar 3. 15 Task Status In Progress

## Sprint Review dan Sprint Retrospective

Tahap ini adalah tahap akhir sprint, pada tahap ini tim scrum akan melakukan review terhadap pekerjaan yang di jalankan di sprint 1. Serta melakukan diskusi untuk menghadapi sprint berikutnya sehingga berjalan lebih baik. Pada tahap ini seharusnya task sudah berada di status Done.



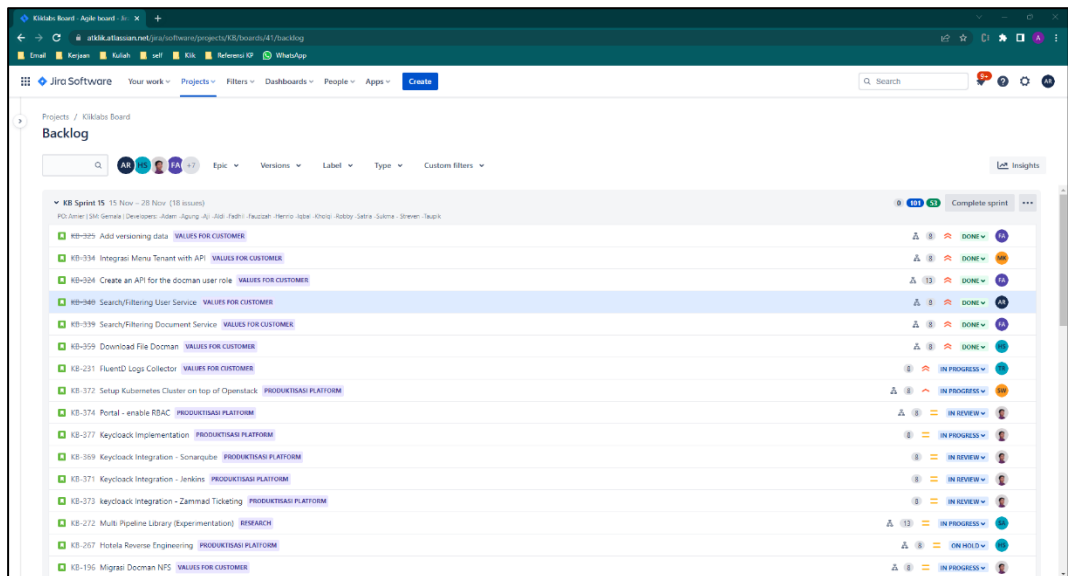
Gambar 3. 16 Task Status Done

### b. Tools Pengembangan Aplikasi

#### 1. Jira

Jira merupakan aplikasi manajemen proyek dari Microsoft. Jira lebih mendukung manajemen proyek secara agile scrum, karena fitur jira sudah dilengkapi dengan keperluan metode scrum mulai dengan backlog, sprint, dan lain-lainnya. Jira memiliki fitur dashboard serta report yang cukup memuaskan, sehingga dapat membantu tim untuk lebih melihat sejauh mana proyek berjalan, selain itu jira juga terintegrasi dengan software – software lainnya untuk pengembangan proyek khususnya IT antara lain, gitlab, Microsoft teams, dan lain-lain. Beberapa fitur yang disediakan jira antara lain :

- Manajemen Proyek Agile
- Custom Dashboards
- Skala Prioritas dan juga ruangan untuk berdiskusi



Gambar 3. 17 Tampilan Jira

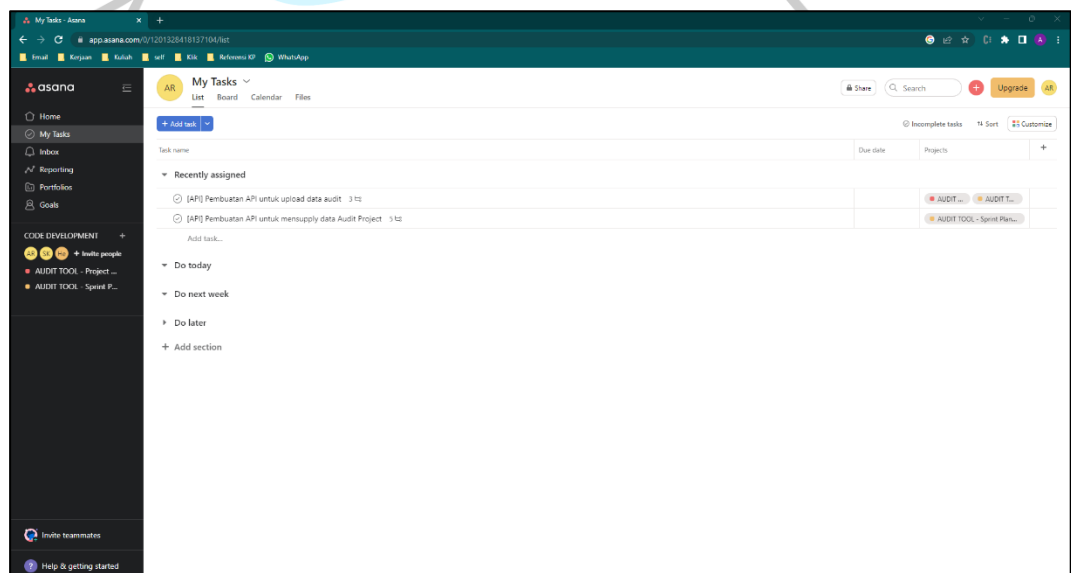
Tampilan Jira pada Gambar 3.17

## 2. Asana

Asana merupakan aplikasi manajemen proyek yang memiliki fitur integrasi dengan software lainnya seperti slack, google calendar, dan juga dropbox. Beberapa fitur yang disediakan asana antara lain :

- Manajemen dokumen,
- Manajemen proyek,
- *time & expense tracking*,
- CRM,
- Manajemen portofolio,
- pengelolaan sumber daya

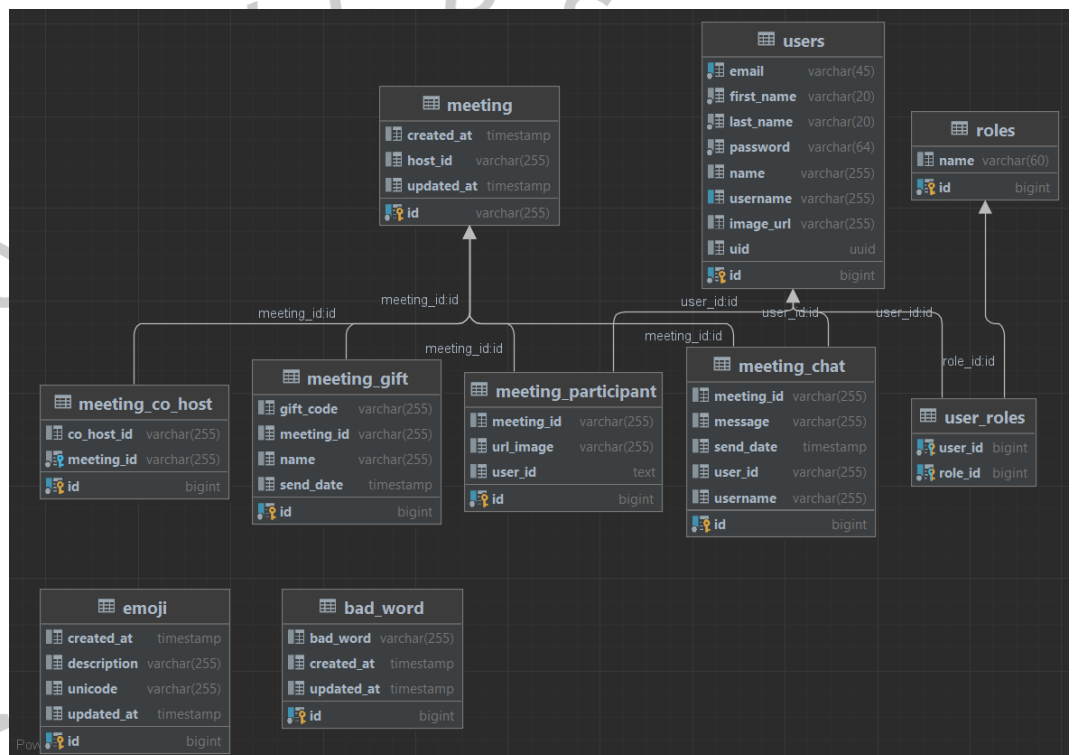
Tampilan Asana pada Gambar 3.18



Gambar 3. 18 Tampilan Asana

### 3.2.4 Skema Database

Pada tahap ini Praktikan mendesain database yang akan digunakan untuk menyimpan data-data pada aplikasi.



Gambar 3. 19 Skema Database

Gambar 3.19 diatas merupakan gambar skema database yang digunakan dalam pembuatan aplikasi yang meliputi tabel – tabel beserta kolomnya di database. Tabel – tabel memiliki relasi satu sama lain kecuali untuk tabel master data. Dan beberapa tabel masih belum terpakai, karena disiapkan untuk pengembangan selanjutnya. Untuk penjelasan detail mengenai tabel – tabel akan dijelaskan di poin berikutnya.

- Tabel Emoji

	id	created_at	description	unicode	updated_at
1	1	2022-04-18 18:23:19.000000	pile of poo	💩	<null>
2	2	2022-04-18 18:23:21.000000	middle finger	🖕	<null>
3	3	2022-04-18 18:23:21.000000	oncoming fist	👊	<null>
4	15	<null>	family: man, man, boy	👨👨👦	<null>
5	96	<null>	Frog	🐸	<null>
6	51	<null>	crossed swords	⚔️	<null>
7	56	<null>	boomerang	🥋	<null>
8	39	<null>	moai	🗿	<null>
9	74	<null>	tornado	🌪️	<null>
10	41	<null>	coffin	🪦	<null>
11	40	<null>	headstone	🪦	<null>
12	88	<null>	scorpion	🦂	<null>
13	106	<null>	angry face with horns	👹	<null>
14	22	<null>	family: woman, woman, girl, boy	👩👩👧👦	<null>
15	28	<null>	jack-o-lantern	🎃	<null>
16	54	<null>	pick	🪓	<null>
17	90	<null>	shark	🦈	<null>
18	75	<null>	wind face	🌪️	<null>
19	101	<null>	bear	🐻	<null>
20	38	<null>	water closet	🚽	<null>
21	98	<null>	rat	🐭	<null>
22	68	<null>	boxing glove	🥊	<null>
23	16	<null>	family: man, man, girl	👨👨👧	<null>
24	113	<null>	zombie	🧟	<null>
25	77	<null>	kitchen knife	🔪	<null>
26	80	<null>	beer mug	🍺	<null>
27	57	<null>	Monkey Face	🐵	<null>

Gambar 3. 20 Tabel Emoji

Tabel Emoji adalah tabel yang berfungsi untuk menampung master data *bad* emoticon. Tabel ini berisi kumpulan *bad* emoticon dengan format UTF- 8 versi web. Semua data di tabel ini diminta dari *client*. Gambar 3.20 menunjukkan data – data pada Tabel Emoji. Tabel ini berisi *column-column* :

- id sebagai primary key
- created\_at tanggal data dibuat
- description deskripsi *bad* emoticon
- Unicode, format bad emoticon standar yang dipakai oleh web
- Dan, Updated\_at tanggal diupdatenya data

- Tabel Bad\_Word

	id	bad_word	created_at	updated_at
1	379	Anjing	<null>	<null>
2	380	Babi	<null>	<null>
3	381	Monyet	<null>	<null>
4	382	Kunyuk	<null>	<null>
5	383	Bajingan	<null>	<null>
6	384	Asu	<null>	<null>
7	385	Bangsat	<null>	<null>
8	386	Kampret	<null>	<null>

Gambar 3. 21 Tabel Bad\_Word



Tabel `bad_word` adalah tabel yang berfungsi untuk menampung master data *bad word*. Tabel ini berisi kumpulan *bad word* dengan bahasa Indonesia maupun bahasa Inggris. Semua data di tabel ini diminta dari *client*. Gambar 3.21 menunjukkan data – data pada Tabel `bad_word`. Tabel ini berisi *column-column* :

- `id` sebagai primary key
- `created_at` tanggal data dibuat
- `bad_word`, kata kata buruk
- Dan, `updated_at` tanggal diupdatenya data

- Tabel Meeting

	<code>id</code>	<code>created_at</code>	<code>host_id</code>	<code>updated_at</code>
72	LM-1659526424623	2022-08-03 11:33:44.760391	<null>	2022-08-03 11:33:44.760449
73	LM-1659526595127	2022-08-03 11:36:35.233579	<null>	2022-08-03 11:36:35.233641
74	LM-1659538321896	2022-08-03 14:52:02.104107	<null>	2022-08-03 14:52:02.104194
75	LM-1659568552956	2022-08-03 23:15:53.978828	<null>	2022-08-03 23:15:53.978878
76	LM-1659573985651	2022-08-04 00:46:25.809215	<null>	2022-08-04 00:46:25.809281
77	LM-1659574034067	2022-08-04 00:47:14.221209	<null>	2022-08-04 00:47:14.221257
78	LM-1659596146920	2022-08-04 06:55:47.121482	<null>	2022-08-04 06:55:47.121509
79	LM-1659596751098	2022-08-04 07:05:51.550086	<null>	2022-08-04 07:05:51.550175
80	LM-1659597003711	2022-08-04 07:10:03.836884	<null>	2022-08-04 07:10:03.836938
81	LM-1659669532350	2022-08-05 03:18:53.375907	<null>	2022-08-05 03:18:53.375938
82	LM-1660221737088	2022-08-11 12:42:23.918555	<null>	2022-08-11 12:42:23.918616
83	LM-1660221819199	2022-08-11 12:43:39.303488	<null>	2022-08-11 12:43:39.303550
84	LM-1660623687084	2022-08-16 04:21:27.305253	<null>	2022-08-16 04:21:27.305369
85	LM-1660792814957	2022-08-18 03:20:15.198073	<null>	2022-08-18 03:20:15.198155
86	LM-1660793258607	2022-08-18 03:27:38.798279	<null>	2022-08-18 03:27:38.798415
87	LM-1661145078906	2022-08-22 05:11:19.627756	<null>	2022-08-22 05:11:19.627806
88	LM-1661487632288	2022-08-26 04:20:33.123467	<null>	2022-08-26 04:20:33.123518
89	LM-1661487680663	2022-08-26 04:21:20.805314	<null>	2022-08-26 04:21:20.805367
90	LM-1661487859112	2022-08-26 04:24:19.245353	<null>	2022-08-26 04:24:19.245422
91	LM-1662191093148	2022-09-03 07:44:53.288615	<null>	2022-09-03 07:44:53.288641
92	LM-1662204619247	2022-09-03 11:30:19.426548	<null>	2022-09-03 11:30:19.426591
93	LM-1662510994813	2022-09-07 00:36:35.566382	<null>	2022-09-07 00:36:35.566417
94	LM-1662559128324	2022-09-07 13:58:48.776239	<null>	2022-09-07 13:58:48.776277
95	94494141	2022-10-11 09:37:29.733743	<null>	2022-10-11 09:37:29.734112
96	94503369	2022-10-11 11:20:22.801123	<null>	2022-10-11 11:20:22.802221
97	94503412	2022-10-11 11:20:49.634293	<null>	2022-10-11 11:20:49.635270
98	94595452	2022-10-12 07:43:34.345791	<null>	2022-10-12 07:43:34.345860

Gambar 3. 22 Tabel Meeting

Tabel `meeting` adalah tabel yang berfungsi untuk menampung data event / meeting yang sedang, akan, dan telah berlangsung. Tabel ini berisi kumpulan *event* yang nantinya bisa dimasuki oleh user dengan `id` sebagai nama event. Gambar 3.22 menunjukkan data-data pada Tabel `meeting`. Tabel ini berisi *column-column* :

- `id` sebagai primary key, juga nama event
- `created_at` tanggal data dibuat
- `host_id`, id host disebuah event

- Dan, updated\_at tanggal diupdatenya data

- Tabel User

id	email	first_name	last_name	password	name	username	image_url	uid
1	2 adamikhsan69@gmail.com	adam	rizaldi	\$2a\$10\$1ZA16oKD2Gt6pKlXtrcbP.rL2ouEtuqKGuNNV..	adam second	damnzaldi	<null>	93678556-c341-4b9e-b
2	1 adamikhsan89@gmail.com	adam	rizaldi	\$2a\$10\$1ZA16oKD2Gt6pKlXtrcbP.rL2ouEtuqKGuNNV..	adam ichsan rizaldi	adamichzal		55de6c53-343d-4f7e-b

Gambar 3. 23 Tabel User

Tabel user adalah tabel yang berfungsi untuk menampung data user yang terdaftar dan akan memasuki event. Tabel ini berisi kumpulan user yang nantinya bisa memasuki sebuah event serta mengirimkan chat di dalam event tersebut. Gambar 3.23 menunjukkan data-data pada Tabel user. Tabel ini berisi *column-column* :

- id sebagai primary key, juga nama event
- email, email user yang terdaftar
- first\_name, nama depan user
- last\_name, nama belakang user
- password, password user untuk memasuki aplikasi
- name, nama user
- username, username untuk memasuki aplikasi
- image\_url, url image untuk menyimpan photo profile user
- uid, id unik sebagai penanda user

- Tabel Meeting\_Chat

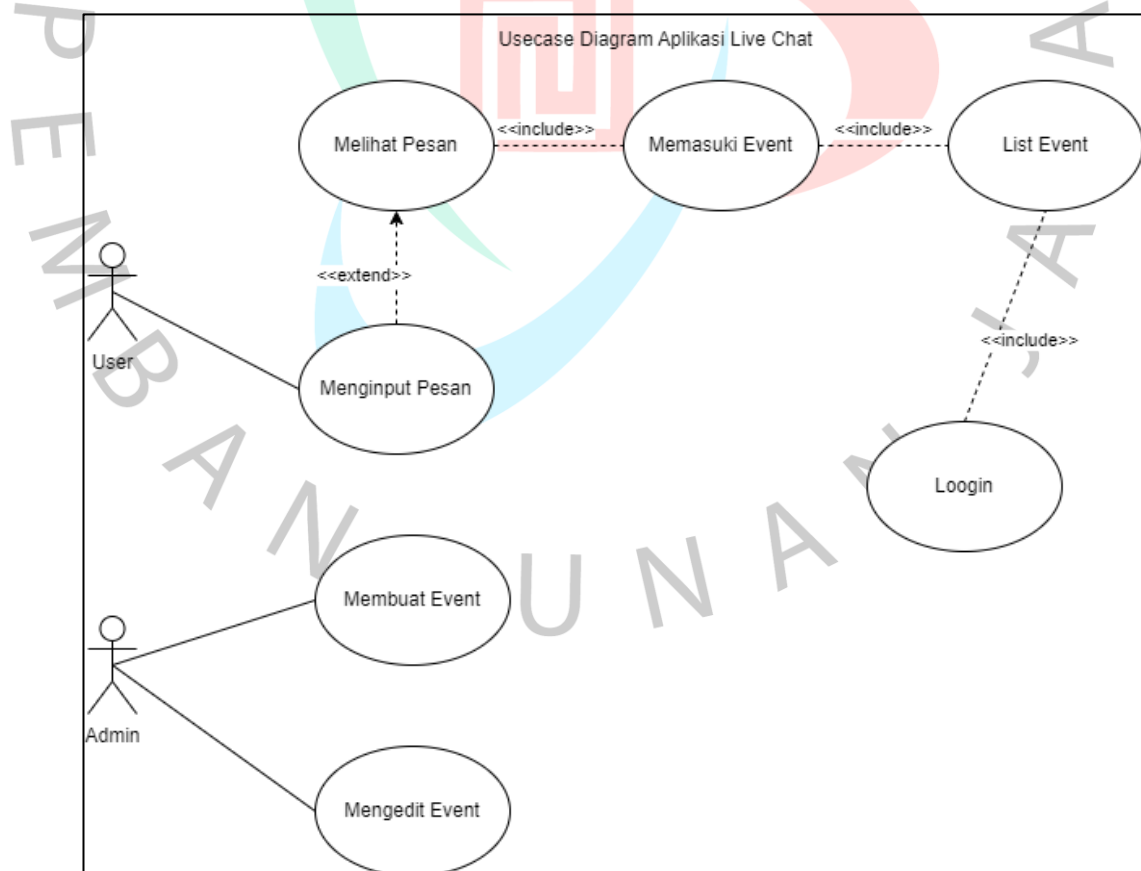
id	meeting_id	message	send_date	user_id	username
1	107 97150319137	****	2022-05-25 02:23:44.919582	16780288	Sigit LM
2	108 95455861525	**** test 123 🤔	2022-05-25 02:32:14.163905	112233	adam
3	109 98022348185	hallo test 123 🤔	2022-05-25 05:26:38.873061	112233	adam
4	110 98022348185	hallo	2022-05-25 05:27:18.103444	112233	adam
5	111 99279761615	test	2022-05-25 09:53:33.375047	16781312	0822 9923 2266
6	112 97150319137	****	2022-05-25 09:59:41.590919	16778240	Sigit LM
7	113 97150319137	asd	2022-05-25 09:59:48.864122	16779264	zjan sebastian
8	114 99279761615	test	2022-05-25 10:08:53.977693	16783360	0822 9923 2266
9	115 99279761615	android send	2022-05-25 10:09:19.527180	16783360	0822 9923 2266
10	116 99279761615	test	2022-05-25 10:12:30.172003	16779264	herru code.id
11	117 99279761615	saya	2022-05-25 10:12:38.999840	16779264	herru code.id
12	118 99279761615	ios send	2022-05-25 10:16:23.341430	16779264	herru code.id
13	119 99279761615	ios kirim	2022-05-25 10:40:16.230406	16779264	herru code.id
14	120 99279761615	android kirim	2022-05-25 10:43:03.957882	16783360	0822 9923 2266
15	121 95455861525	tes	2022-05-25 10:49:21.909624	16779264	cha
16	122 95455861525	oy	2022-05-25 10:52:20.597949	16780288	cha
17	123 95455861525	oy	2022-05-25 10:56:40.750900	16781312	cha
18	124 99279761615	test host kirimi	2022-05-25 11:36:59.672997	16792576	Adam Ichsan Rizaldi
19	125 97150319137	ha	2022-05-27 03:49:05.202988	16778240	Sigit LM
20	126 97150319137	haha	2022-05-27 08:25:41.438112	16778240	Sigit LM
21	127 97150319137	vbbbnjj	2022-05-27 08:26:39.981889	16779264	Sigit LM
22	128 95455861525	oy	2022-05-27 08:27:55.536110	16778240	cha
23	129 97150319137	vbbbnjj	2022-05-27 08:29:36.223588	16779264	Sigit LM
24	130 95455861525	oy	2022-05-27 08:48:50.581678	16778240	cha
25	131 95455861525	ay	2022-05-27 08:59:12.390073	16778240	cha
26	132 95455861525	ay	2022-05-27 09:06:04.418927	16778240	cha
27	133 99279761615	Host kirim chat	2022-05-27 09:24:51.401245	16778240	Host

Gambar 3. 24 Tabel Meeting\_Chat

Tabel Meeting\_Chat adalah tabel yang berfungsi untuk menampung data chat per user pada suatu event. Tabel ini berisi kumpulan chat - chat yang ada pada suatu event dan dikirimkan oleh user. Chat – chat yang masuk ke dalam tabel ini adalah chat yang sudah melalui proses filter bad word dan bad emoticon. Gambar 3.24 menunjukkan data-data pada Tabel Meeting\_Chat. Tabel ini berisi *column-column* :

- id sebagai primary key, juga nama event
- email, email user yang terdaftar
- first\_name, nama depan user
- last\_name, nama belakang user
- password, password user untuk memasuki aplikasi
- name, nama user
- username, username untuk memasuki aplikasi
- image\_url, url image untuk menyimpan photo profile user
- uid, id unik sebagai penanda user

### 3.2.5 Usecase Diagram



Gambar 3. 25 Use Case Diagram Aplikasi Live Chat

Gambar 3.25 diatas menggambarkan User dapat melakukan menginput pesan teks dan melihat pesan teks. Untuk dapat melakukan hal tersebut, user harus melalui tahap memasuki event dengan memilih list event yang tersedia dan juga login. User dapat melihat pesan teks yang dia kirimkan maupun user lain kirimkan. Serta admin yang dapat membuat dan mengedit *event* langsung dari *database*, karena belum ada *user interfacenya*.

Tabel 3. 1 Skenario *Usecase Diagram* Menginput Pesan Teks

Nama Use Case	Menginput Pesan Teks
Aktor	User
Deskripsi Use Case	User dapat membuat/menginput pesan ke dalam database
<i>Precondition</i>	Untuk dapat menginput pesan, masuk ke dalam event terlebih dahulu
Tahapan	<ol style="list-style-type: none"> <li>1.User Login</li> <li>2.User masuk ke halaman list event</li> <li>3.User masuk ke dalam event</li> <li>4.User menginput dan mengirimkan pesan teks</li> </ol>
Post Condition	Data yang dimasukkan akan masuk ke dalam database dan muncul di halaman website

Tabel 3. 2 Skenario *Usecase Diagram* Melihat Pesan Teks

Nama Use Case	Melihat Pesan Teks
Aktor	User
Deskripsi Use Case	User dapat Melihat Pesan Teks dirinya maupun user lain
<i>Precondition</i>	Untuk dapat melihat pesan, masuk ke dalam event terlebih dahulu
Tahapan	<ol style="list-style-type: none"> <li>1.User Login</li> <li>2.User masuk ke halaman list event</li> <li>3.User masuk ke dalam event</li> <li>4.User dapat melihat pesan teks</li> </ol>

	dirinya maupun user lain
Post Condition	List Pesan akan muncul pada halaman <i>Live Chat</i>

**Tabel 3. 3 Skenario Usecase Memasuki Event**

Nama Use Case	Memasuki Event
Aktor	User
Deskripsi Use Case	User dapat Memasuki Event yang tersedia di List Event
<i>Precondition</i>	Untuk dapat Memasuki Event, user harus Memilih Event dari List Event dan login terlebih dahulu
Tahapan	<ol style="list-style-type: none"> <li>1. User Login</li> <li>2. User masuk ke halaman list event</li> <li>3. User masuk ke dalam event</li> </ol>
Post Condition	User berhasil memasuki event dan dapat mengirimkan dan melihat <i>chat</i> yang sedang berlangsung

**Tabel 3. 4 Skenario Usecase List Event**

Nama Use Case	List Event
Aktor	User
Deskripsi Use Case	User dapat Melihat Event yang sedang berlangsung di List Event
<i>Precondition</i>	Untuk dapat Melihat Event, user harus login terlebih dahulu
Tahapan	<ol style="list-style-type: none"> <li>1. User Login</li> <li>2. User masuk ke halaman list event</li> <li>3. User dapat melihat List Event</li> </ol>
Post Condition	List Event akan muncul pada halaman Event

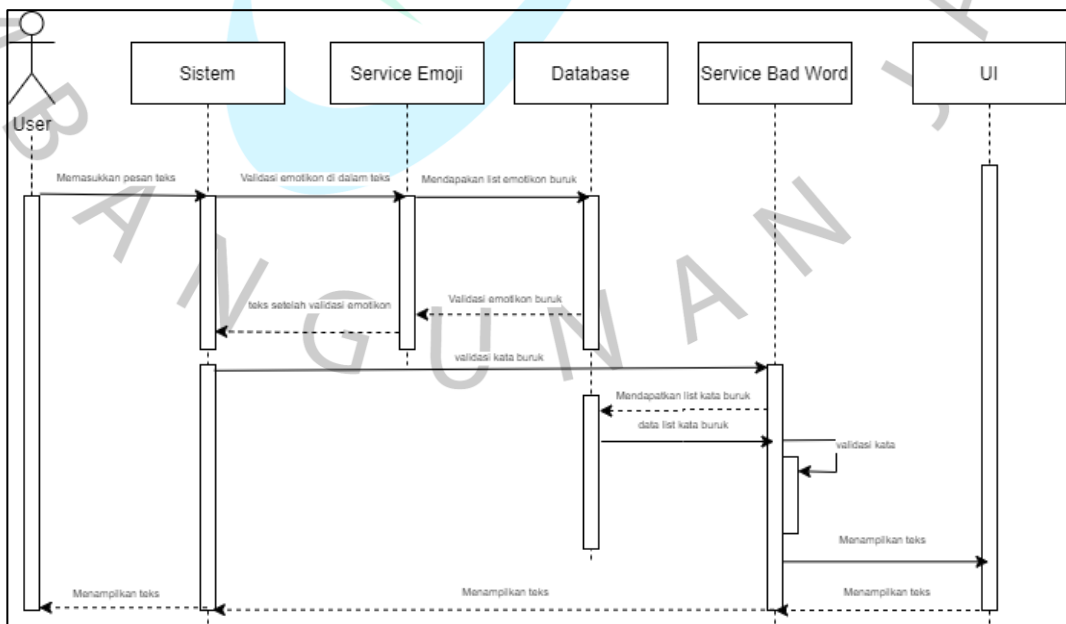
Tabel 3. 5 Skenario Usecase Login

Nama Use Case	Login
Aktor	User
Deskripsi Use Case	User Login dan Memasuki Halaman Event
Tahapan	<ol style="list-style-type: none"> <li>1. User memasuki halaman login</li> <li>2. User memasukkan Username dan Password</li> <li>3. User berhasil Masuk ke Halaman Event</li> </ol>
Post Condition	User berhasil memasuki halaman Event setelah login berhasil

### 3.2.6 Sequence Diagram

Setelah Praktikan mendesain Basis Data, Praktikan melanjutkan pekerjaan ke tahap berikutnya yaitu pembuatan Sequence Diagram untuk menjelaskan secara visual alur kerja yang berisi interaksi antara objek dalam sebuah sistem, juga menampilkan pesan atau perintah yang dikirim, beserta waktu pelaksanaannya.

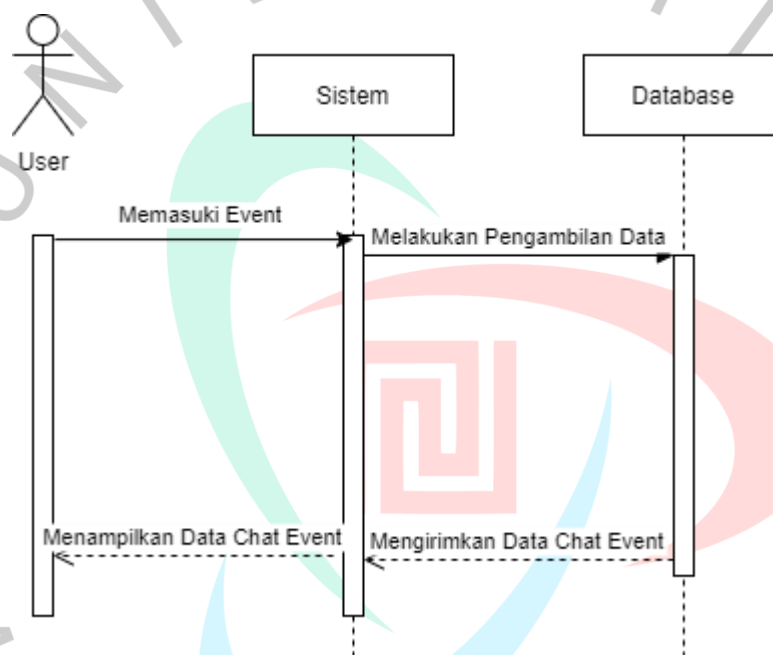
#### 1. Sequence Diagram Menginput Pesan Teks



Gambar 3. 26 Sequence Diagram Menginput Pesan Teks

Penjelasan dari Gambar 3.26 diatas yaitu saat user menginput pesan teks ke dalam sistem lalu sistem akan melakukan validasi di setiap service dibantu dengan database master data, validasi di emoticon service untuk mengetahui adanya *bad emoticon* atau tidak lalu validasi di *badword service* untuk mengetahui adanya *bad word* atau tidak. Selanjutnya data yang sudah di validasi akan dimasukkan ke dalam database setelah itu sistem akan mengirimkan data yang sudah di validasi ke halaman website atau *User Interface* (UI).

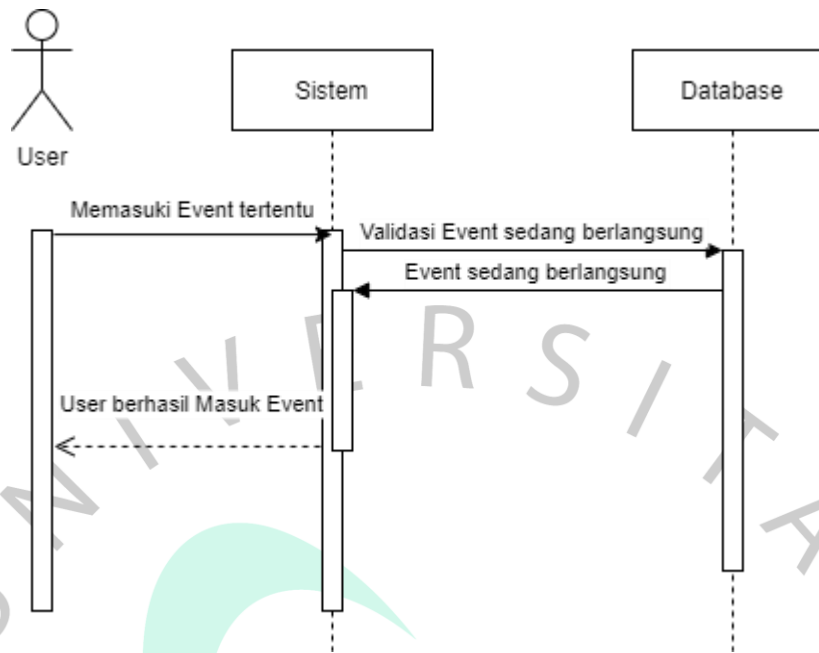
## 2. Sequence Diagram Melihat Pesan Teks



Gambar 3. 27 Sequence Diagram Melihat Pesan Teks

Penjelasan dari Gambar 3.27 diatas yaitu saat user memasuki event user akan merequest ke dalam sistem, lalu sistem akan melakukan penarikan data ke database untuk mengambil data chat yang berada pada event tersebut. Setelah database berhasil mendapatkan data, data diberikan ke sistem untuk kemudian dikirimkan ke user melalui halaman website.

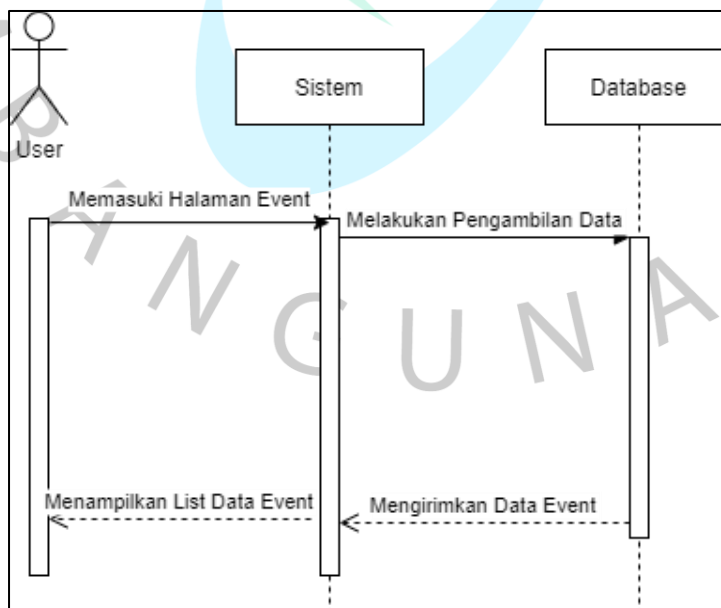
## 3. Sequence Diagram Memasuki Event



Gambar 3. 28 Sequence Diagram Memasuki Event

Penjelasan dari Gambar 3.28 diatas yaitu saat user memasuki event tertentu sistem akan melakukan validasi ke database untuk mengetahui apakah event sedang berlangsung dan bisa dimasuki oleh user atau tidak, jika bisa sistem akan meresponse user dengan memindahkan user ke halaman event spesifik.

#### 4. Sequence Diagram Melihat List Event

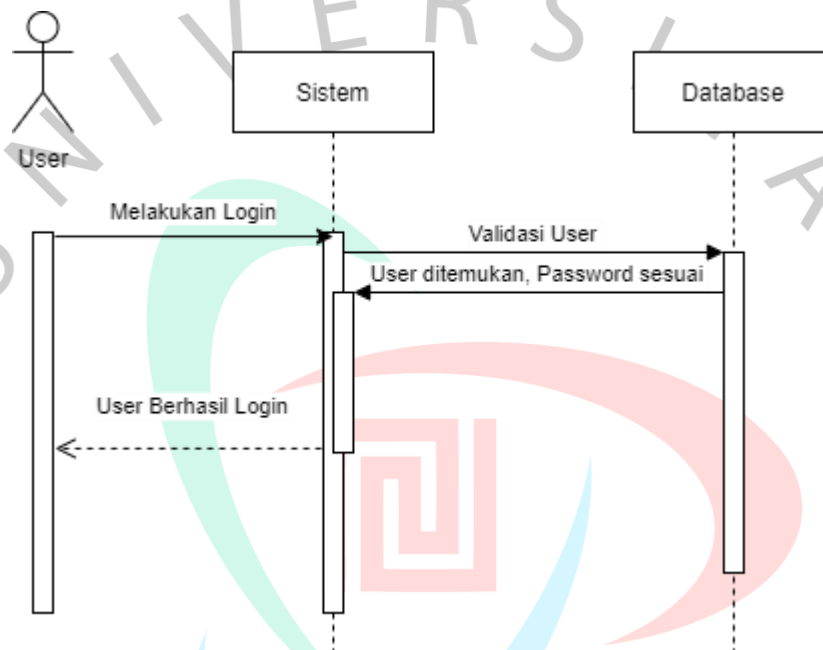


Gambar 3. 29 Sequence Diagram Melihat List Event



Penjelasan dari Gambar 3.29 diatas yaitu saat user memasuki halaman *event*, user akan *merequest* ke dalam sistem, lalu sistem akan melakukan penarikan data ke database untuk mengambil data event yang tersedia di database. Setelah database berhasil mendapatkan data, data diberikan ke sistem untuk kemudian dikirimkan ke user melalui halaman website.

#### 5. Sequence Diagram Login



Gambar 3. 30 Sequence Diagram Login

Penjelasan dari Gambar 3.30 diatas yaitu saat user melakukan login sistem akan melakukan validasi user ke database untuk mengetahui apakah user tersebut terdaftar dan apakah password dan usernamenya sesuai, jika valid maka sistem akan meresponse user dengan memindahkan user ke halaman Event dan menginfokan user telah berhasil login.

#### 3.2.7 Desain Model, View, Controller

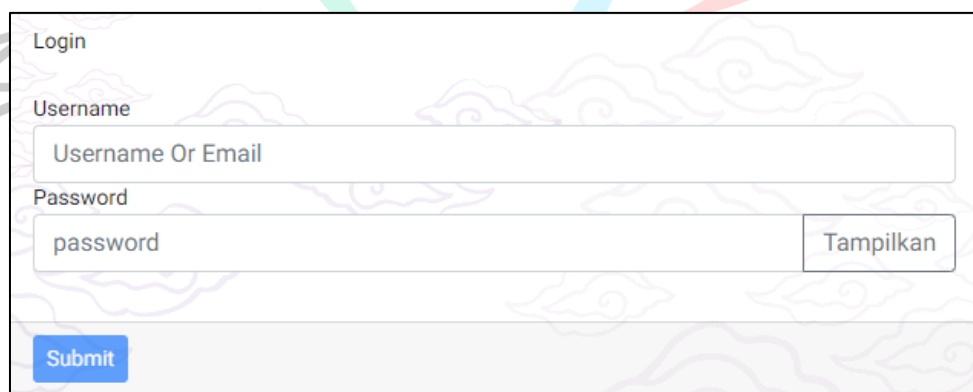
Setelah melakukan perancangan website dilakukan. Praktikan melanjutkan pekerjaan dengan mendesain model dan controller dari framework yang digunakan yaitu Sprint Boot. Model merupakan bagian yang mengelola dan memiliki keterkaitan dengan basis data. Sedangkan untuk Controller adalah bagian yang berfungsi sebagai jembatan yang menghubungkan antara View dan Model dalam setiap proses yang terjadi ketika ada request dari user. Untuk membuat model, praktikan hanya

perlu membuat file baru dengan format “namamodel.php” sesuai dengan data di basis data pada lokasi app/Models. Untuk membuat Controller juga memiliki tahapan yang sama seperti dalam membuat Model namun perbedaannya pada letak folder untuk menyimpan file Controller. File Controller disimpan pada lokasi app/Controllers. Berikut fungsi dari package – package yang dibuat

- a) Controller, sebagai tempat menuliskan API atau penghubung model dan view yang akan di akses oleh client.
- b) Service, semua fungsi yang berkaitan dengan *logic business*.
- c) Config, sebagai tempat untuk melakukan config websocket.
- d) Repository, sebagai tempat untuk menuliskan kode yang berhubungan dengan basis data, seperti untuk menulis *query* ke basis data untuk *insert, update, delete, view data*.
- e) Model, Representasi dari tabel basis data.

### 3.2.8 Implementasi Pembuatan Aplikasi Web

Pada implementasi pembuatan aplikasi web, praktikan menggunakan bahasa pemrograman Java dengan framework Spring Boot. Untuk penulisan kode program, praktikan menggunakan software IntelliJ IDEA dari jet brains sebagai teks editor dan PostgreSQL sebagai sistem manajemen basis data. Dalam urusan server atau pendepleyan praktikan menggunakan server dari PT.Code Development Indonesia. Berikut penjelasan detail halaman pada Web.



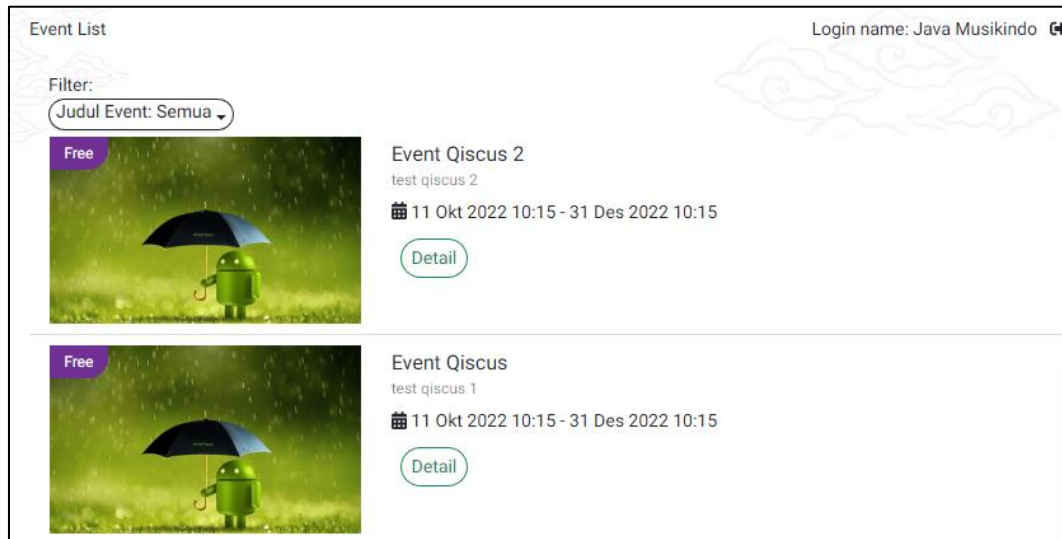
The image shows a web login form with the following elements:

- Title: Login
- Section: Username
- Input field: Username Or Email
- Section: Password
- Input field: password
- Button: Tampilkan
- Button: Submit

Gambar 3. 31 Tampilan Halaman Login

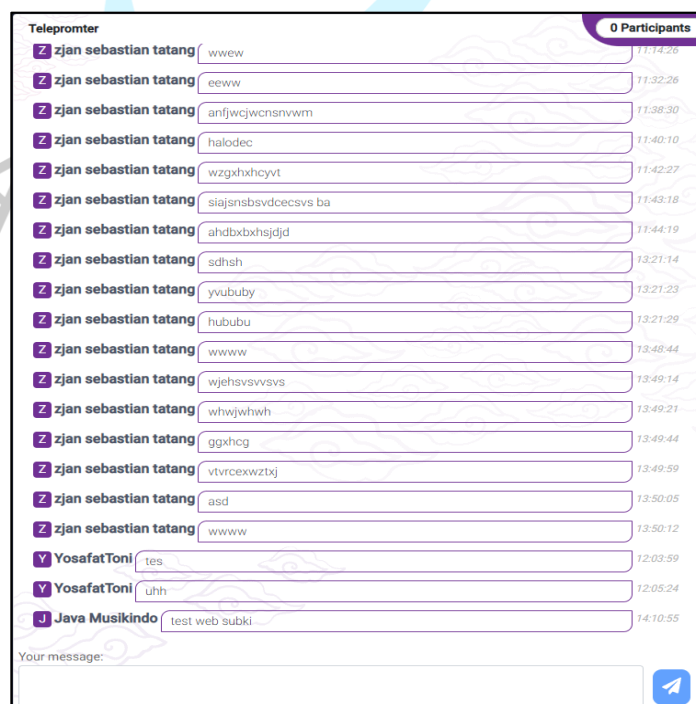
Pada Gambar 3.31 diatas menampilkan halaman login pada aplikasi, User dapat login menggunakan Username ataupun email, dan user dapat melihat password yang dia ketikkan, untuk *defaultnya* password akan ditutupi

oleh simbol titik. *Field* username dan juga password harus diisi sebelum melakukan submit. Tombol submit akan mengarahkan ke sistem dengan API Login yang sudah dibuat di backend. Jika berhasil login maka user akan diarahkan ke halaman list event.



Gambar 3. 32 Tampilan Halaman List Event

Pada Gambar 3.32 diatas merupakan halaman List Event, pada halaman ini user dapat melihat judul Event, deskripsi event, serta tanggal mulai dan tanggal berakhirnya event, jika user ingin memasuki Event, user bisa mengklik tombol Detail. User juga dapat melihat gambar event. Lalu jika User mengklik tombol Detail user akan diarahkan ke dalam live chat event tersebut.

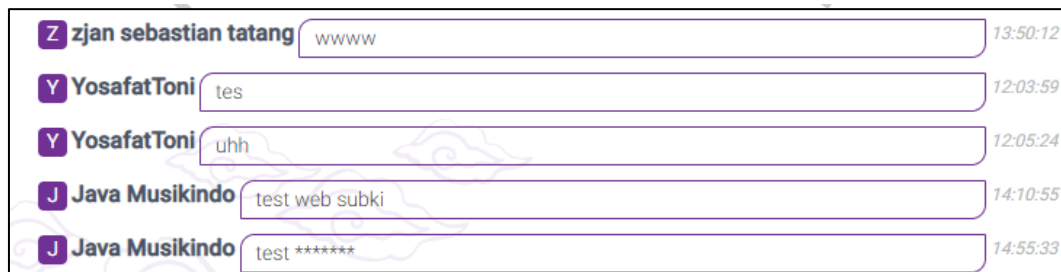


Gambar 3. 33 Tampilan Halaman Live Chat Event

Pada Gambar 3.33 diatas merupakan halaman live chat event yang berisi chat yang berasal dari user – user yang telah memasuki event. Chat tersebut bersifat langsung sehingga user dapat berinteraksi satu sama lain. Selain itu chat yang ditampilkan juga sudah melalui filter *bad word* dan filter *bad emoticon*. Dalam halaman ini user dapat melihat chat user lain dan juga chat dirinya sendiri. Selain itu user juga dapat mengirimkan chat dengan menuliskan teks di kolom *Your message* dan mengirimkannya dengan tombol pesawat kertas biru.

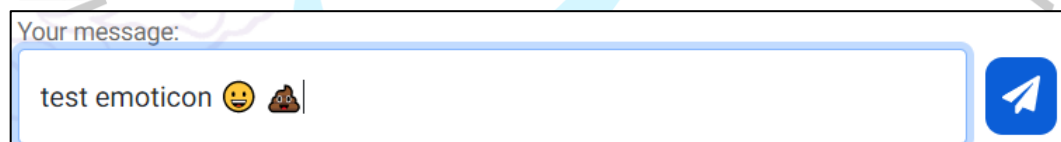


Gambar 3. 34 Tampilan Mengirim *Bad Word*

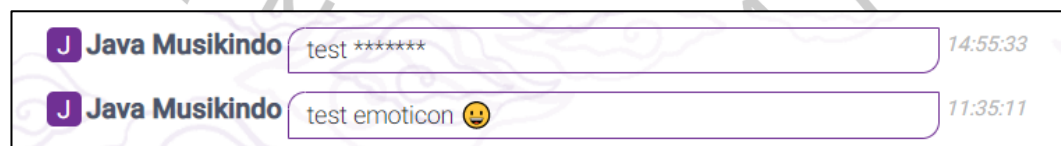


Gambar 3. 35 Tampilan Halaman Live Chat Setelah Mengirim *Bad Word*

Pada Gambar 3.34 praktikan mencoba memasukkan kata *bad word* dan pada gambar 3.35 setelah kata tersebut dikirim maka di halaman live chat kata "kampret" terdeteksi *bad word* dan diganti menjadi simbol \*. Sedangkan kata "test" tidak terdeteksi *bad word* maka kata test tetap di tampilkan seutuhnya.

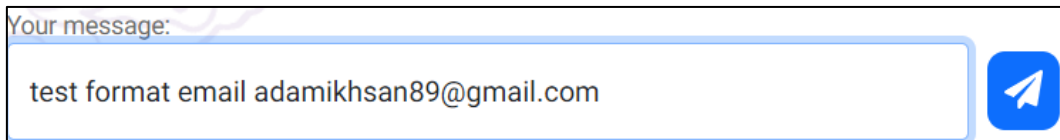


Gambar 3. 36 Tampilan Mengirim *Bad Emoticon*

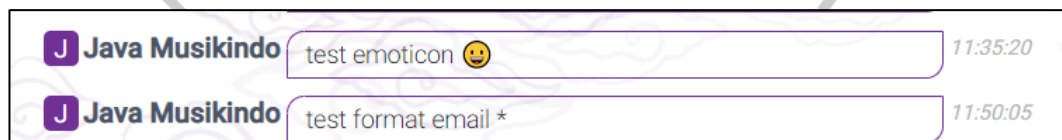


Gambar 3. 37 Tampilan Halaman Live Chat Setelah Mengirim *Bad Emoticon*

Pada Gambar 3.36 praktikan mencoba memasukkan emoticon normal dan *bad* emoticon. gambar 3.37 setelah emoticon tersebut dikirim maka di halaman live chat emoticon “pile of poo” terdeteksi *bad* emoticon dan dihapus. Sedangkan emoticon “smiley” tidak terdeteksi *bad* emoticon maka emoticon smiley tetap di tampilkan seutuhnya



Gambar 3. 38 Tampilan Mengirim format email

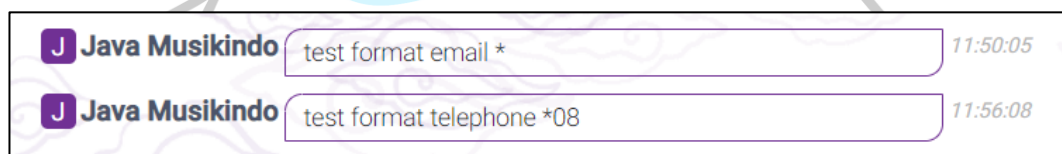


Gambar 3. 39 Tampilan Halaman Live Chat Setelah Mengirim Format Email

Pada Gambar 3.38 praktikan mencoba memasukkan format email. Pada gambar 3.39 setelah teks format email tersebut dikirim maka di halaman live chat kata format email terdeteksi *bad word* dan diganti dengan tanda \*. Sedangkan kata “test format email” tidak terdeteksi *bad word* maka kata test format email tetap di tampilkan seutuhnya.



Gambar 3. 40 Tampilan Mengirimkan Format Telepon



Gambar 3. 41 Tampilan Halaman Live Chat Setelah Mengirimkan Format Telepon

Pada Gambar 3.40 praktikan mencoba memasukkan format nomor telepon. Pada gambar 3.41 setelah teks format nomor telepon tersebut dikirim maka di halaman live chat kata format nomor telepon terdeteksi *bad word* dan Sebagian nomornya diganti dengan tanda \*. Sedangkan kata “test format telephone” tidak terdeteksi *bad word* maka kata test format telephone tetap di tampilkan seutuhnya.

### 3.2.9 Pengujian Aplikasi

Dalam pengembangan aplikasi diperlukannya sebuah proses berupa testing sebagai proses terakhir sebelum aplikasi diserahkan kepada *client*. Proses testing ini merupakan proses untuk mengevaluasi aplikasi perangkat lunak atau system untuk memastikan bahwa sesuai dengan persyaratan yang ditentukan dan menghasilkan hasil yang diharapkan. Pengujian sangat penting karena membantu mengidentifikasi kesalahan bisnis, *errors*, dan *bugs* pada perangkat lunak sebelum dikirimkan ke *client*. Hal ini dapat menghemat waktu, uang, dan sumber daya yang dihabiskan untuk memperbaiki masalah yang ditemukan setelah perangkat lunak dirilis.

Ada berbagai jenis pengujian yang dilakukan selama SDLC, yaitu diantaranya, *unit test*, *integration test*, *system testing*, dan *acceptance testing*.

- Unit Test

Unit test adalah suatu teknik pengujian yang dilakukan pada unit-unit terkecil dari suatu program komputer, biasanya pada level per-fungsi atau per-modul. Tujuan dari unit test ini adalah untuk memastikan bahwa setiap unit dari program tersebut bekerja dengan benar dan memenuhi spesifikasi yang telah ditetapkan. Unit test biasanya dilakukan oleh pengembang program itu sendiri sebelum program tersebut diserahkan untuk diuji secara keseluruhan.

- Integration Test

Integration test adalah suatu teknik pengujian yang dilakukan untuk memeriksa bagaimana unit-unit dari suatu program komputer bekerja bersama-sama. Tujuan dari integration test ini adalah untuk memastikan bahwa setiap unit terintegrasi dengan baik dan dapat bekerja sesuai dengan spesifikasi yang telah ditetapkan. Integration test biasanya dilakukan setelah unit test telah selesai dilakukan, dan biasanya dilakukan oleh tim pengujian yang terpisah dari tim pengembangan program.

- System Test

System test adalah suatu teknik pengujian yang dilakukan pada suatu sistem komputer secara keseluruhan, untuk memastikan bahwa sistem tersebut dapat bekerja sesuai dengan spesifikasi yang telah ditetapkan dan memenuhi kebutuhan pengguna. System test biasanya dilakukan setelah integration test selesai dilakukan, dan biasanya dilakukan oleh tim pengujian yang terpisah dari tim pengembangan

program. System test bertujuan untuk memeriksa bagaimana sistem tersebut dapat bekerja secara efektif dan menangani berbagai kondisi yang mungkin terjadi selama operasi normal.

- Acceptance Test

Acceptance test adalah suatu teknik pengujian yang dilakukan pada suatu sistem komputer untuk memastikan bahwa sistem tersebut memenuhi spesifikasi yang telah ditetapkan dan dapat digunakan oleh pengguna akhir sesuai dengan kebutuhan. Acceptance test biasanya dilakukan setelah system test selesai dilakukan, dan biasanya dilakukan oleh tim pengujian yang terpisah dari tim pengembangan program. Acceptance test bertujuan untuk memastikan bahwa sistem tersebut dapat dioperasikan dengan benar dan sesuai dengan kebutuhan pengguna akhir.

Dalam mengembangkan aplikasi ini pengujian yang dilakukan adalah pengujian *system testing* yaitu pengujian secara *black box testing* serta pengujian *performance testing*

- Black Box Testing

Tujuan dari black box testing adalah untuk memeriksa apakah program komputer atau sistem dapat menghasilkan output yang sesuai dengan spesifikasi yang telah ditetapkan, tanpa memeriksa bagaimana program tersebut menghasilkan output tersebut. Black box testing biasanya dilakukan setelah unit test dan integration test selesai dilakukan, dan bertujuan untuk memastikan bahwa program atau sistem dapat bekerja sesuai dengan kebutuhan pengguna akhir.

- Performance Testing

Tujuan dari performance testing adalah untuk memeriksa bagaimana sistem komputer bertindak dibawah beban kerja yang berbeda, seperti jumlah pengguna yang bersamaan atau jumlah data yang ditangani oleh sistem. Performance testing biasanya dilakukan setelah integration test dan system test selesai dilakukan, dan bertujuan untuk memastikan bahwa sistem dapat bekerja dengan baik dan cepat dibawah berbagai kondisi yang mungkin terjadi selama operasi normal.

Pengujian ini diperlukan untuk memastikan aplikasi dapat berjalan dengan semestinya dan tidak ada hambatan saat aplikasi digunakan oleh *public*.

## 1. Performance Testing

Dalam pengujian *performance test* aplikasi yang dikembangkan ini diuji dengan menggunakan sebuah *software*, yaitu Jmeter. Jmeter adalah aplikasi pengetesan yang berbasis java untuk melakukan *performance* dan *stress test*. Jmeter ini kompatibel dengan sistem operasi Linux, Windows, ataupun Mac OS.



Gambar 3. 42 Logo JMeter

Berikut ini adalah Langkah – langkah menginstall jmeter.

#### 1. Install Java

Karena Jmeter adalah aplikasi berbasis java, maka langkah pertama yang harus dilakukan adalah menginstall java. Untuk dapat menjalankan Jmeter, dibutuhkan Java Virtual Machine (JVM) dengan versi 6 atau di atasnya. Jika sudah menginstall java, dapat mengecek command prompt dengan perintah 'java -version' yang akan memberikan tampilan seperti dibawah ini.

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.17134.590]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\sundush_n>java -version
java version "10.0.2" 2018-07-17
Java(TM) SE Runtime Environment 18.3 (build 10.0.2+13)
Java HotSpot(TM) 64-Bit Server VM 18.3 (build 10.0.2+13, mixed mode)

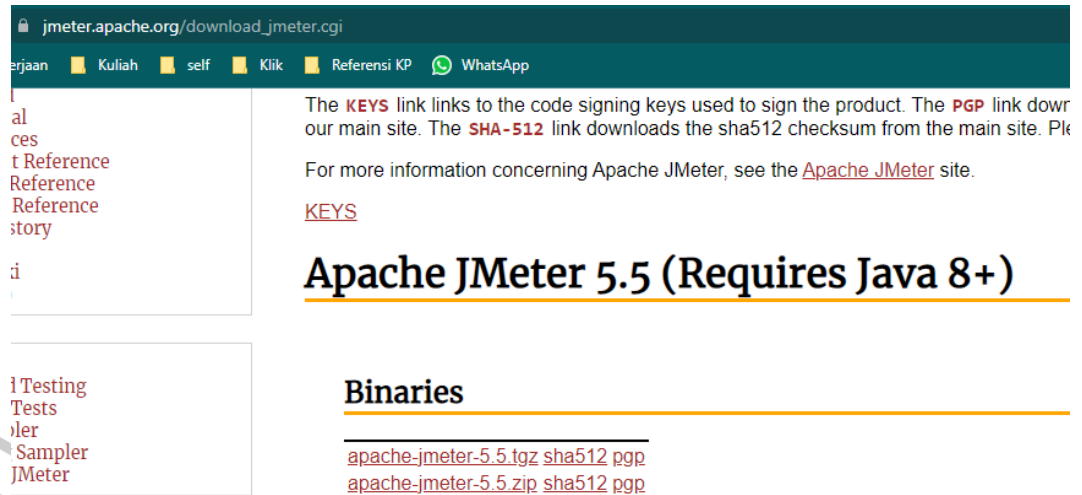
C:\Users\sundush_n>
```

Gambar 3. 43 Java Version di Command Prompt

#### 2. Download Jmeter



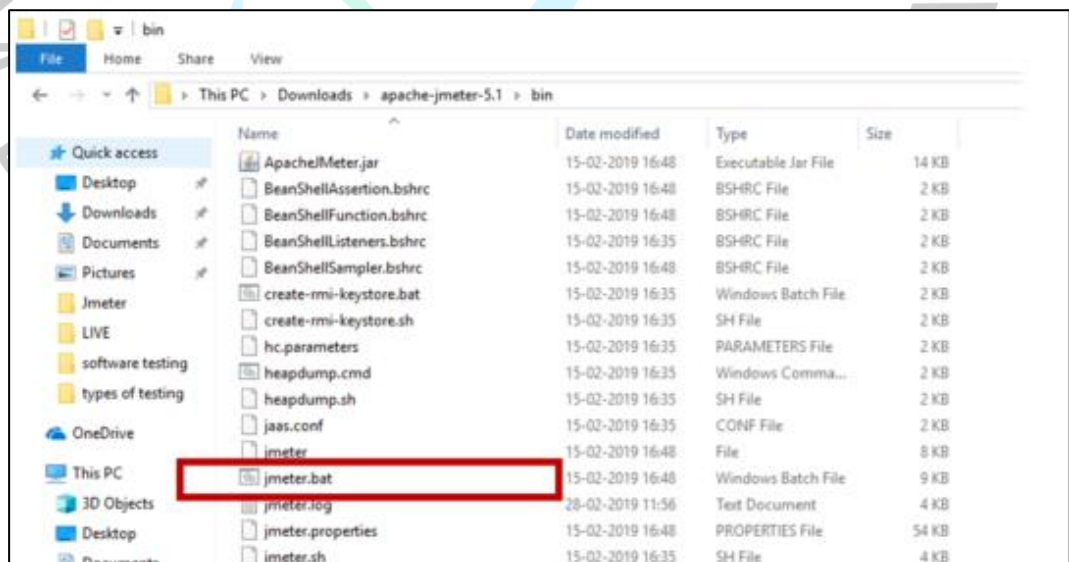
Download Jmeter di website [jmeter.apache.org](http://jmeter.apache.org)



Gambar 3. 44 Tampilan Halaman JMeter

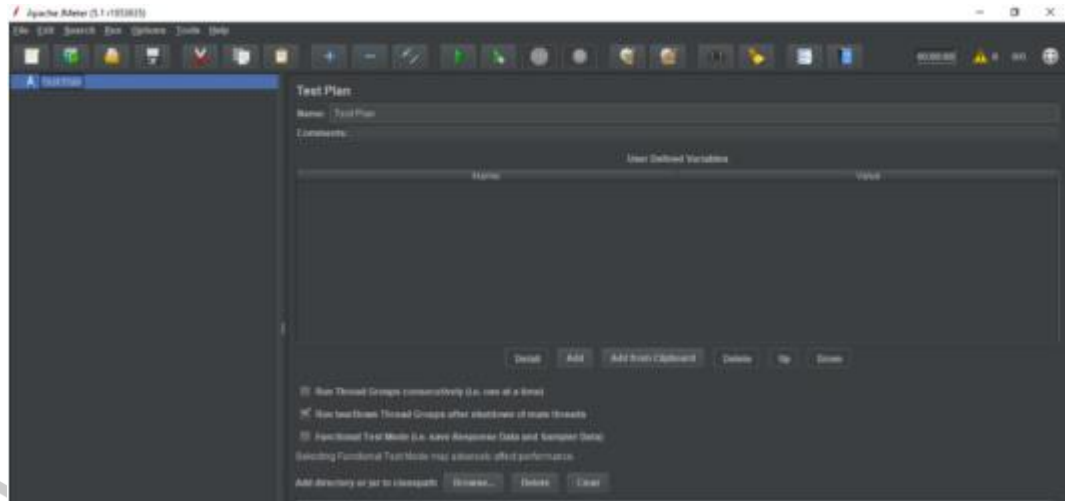
### 3. Install Jmeter

Setelah mendownload jmeter, maka kita perlu mengekstrak file yang sudah kita download, setelah itu kita sudah bisa menggunakan jmeter mode *Graphical User Interface* (GUI) dengan mengeksekusi file `/bin/jmeter.bat`,



Gambar 3. 45 Tampilan File Execute JMeter

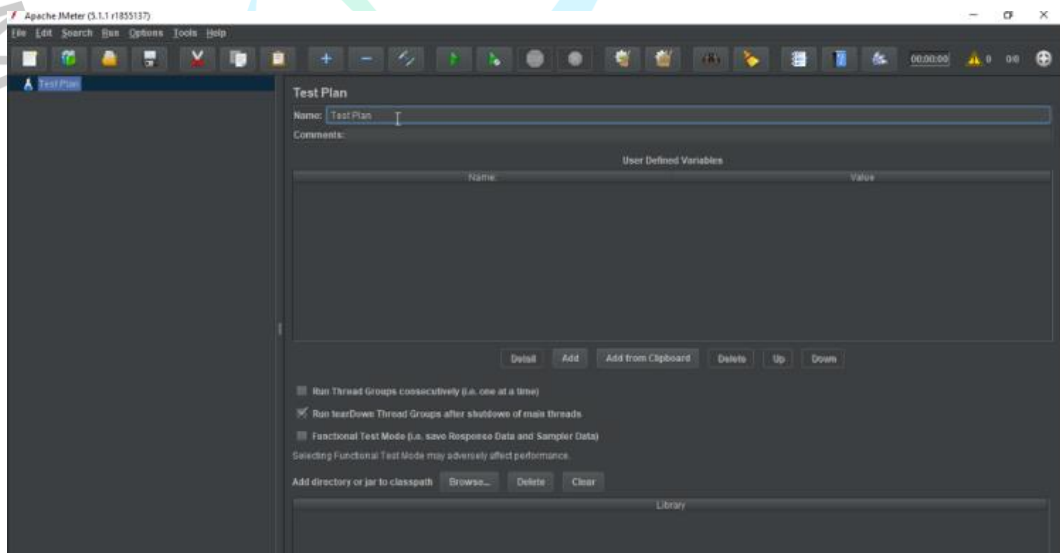
Gambar 3.46 adalah tampilan Jmeter



Gambar 3. 46 Tampilan Awal JMeter

Dan untuk melakukan performa test dengan jmeter, berikut ini adalah langkah – langkahnya

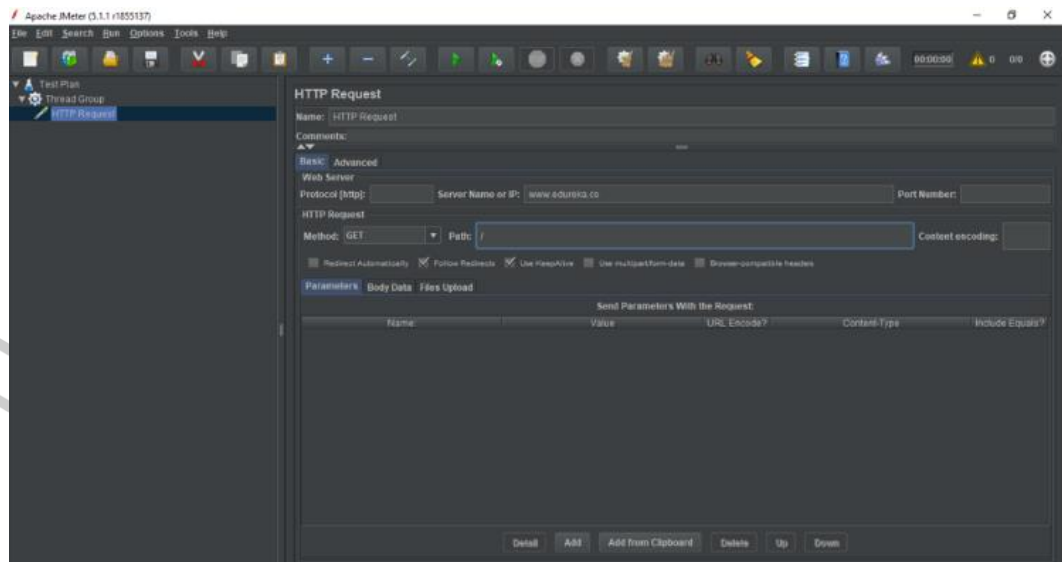
1. Membuat *Test Plan*



Gambar 3. 47 Tampilan Test Plan JMeter

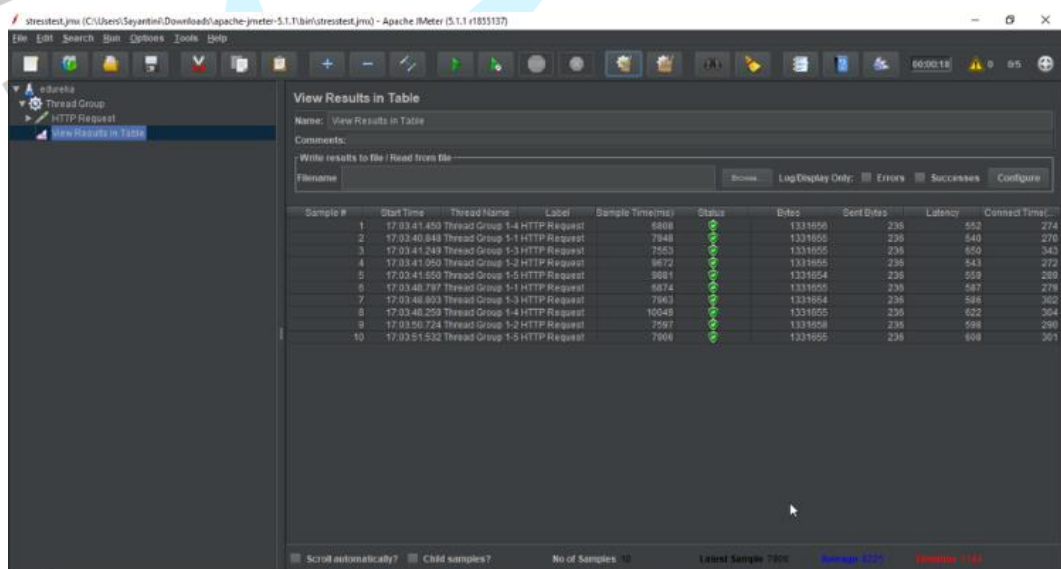
Test Plan adalah tempat kita menaruh project yang akan kita test dalam performance test di JMeter.

2. Memasukkan Thread Group dan menambahkan sebuah HTTP Request dengan nama server dari website yang ingin dilakukan pengujian.



Gambar 3. 48 Tampilan HTTP Request JMeter

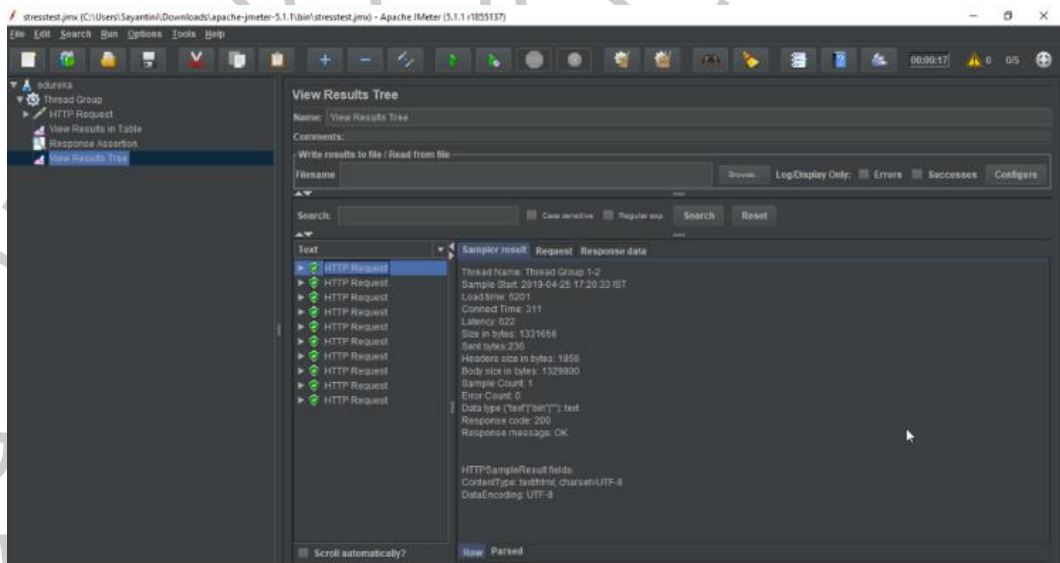
3. Memasukkan listener kedalam thread group dan view test results. Hal itu untuk menunjukkan status test sudah sejauh mana.



Gambar 3. 49 Tampilan View Result in Table JMeter

4. Lalu tambahkan *response assertion* ke dalam thread group. Ini adalah salah satu elemen terpenting karena hal ini dapat membantu untuk menegaskan response dari request di dalam aplikasi.

Kita akan menegaskan untuk response code 200 menggunakan response assertion. Response code 200 adalah response yang merepresentasikan sukses dalam HTTP Request.



Gambar 3. 50 Tampilan View Results Tree JMeter

Lalu berikutnya kita bisa menambahkan banyaknya user atau banyaknya thread dan menambahkan iterasi banyaknya pengujian. Sehingga hal ini bisa menambahkan beban ke server dan mengecek apakah server cukup kuat melakukan tugasnya dengan beban yang telah kita tentukan.

### 3.3 Kendala Yang Dihadapi

Dalam menjalankan Kerja Profesi di PT. Code Development Indonesia, praktikan mengalami beberapa kendala yang cukup menghambat praktikan selama Kerja Profesi, terlebih lagi dengan kondisi pandemi yang mengharuskan praktikan bekerja dari rumah atau *Work From Home* (WFH). Berikut beberapa kendala yang praktikan alami :

- a. Praktikan mengerjakan lebih dari satu proyek.

- b. Jaringan yang terkadang mengalami gangguan ketika melakukan pertemuan secara daring.
- c. Koordinasi yang perlu dilakukan secara intens.

### 3.4 Cara Mengatasi Kendala

Untuk mengatasi kendala yang dihadapi, solusi praktikan antara lain adalah :

- a. Menentukan skala prioritas dari masing – masing proyek, jika ada kendala yang *critical*, maka hal tersebutlah yang menjadi prioritas.
- b. Menyiapkan dua perangkat jaringan agar jika satu jaringan tidak berfungsi, bisa memakai perangkat jaringan cadangan.
- c. Melakukan koordinasi setiap hari melalui pertemuan secara daring, untuk membahas sudah sejauh mana dan masalah apa yang dihadapi.

### 3.5 Pembelajaran Yang Diperoleh dari Kerja Profesi

Selama masa Kerja Profesi, banyak pembelajaran serta manfaat yang praktikan dapatkan, diantaranya :

- a. Belajar bertanggung jawab dalam pekerjaan, terutama dalam membangun sebuah aplikasi yang akan dipakai oleh masyarakat banyak, dengan cara memastikan tidak adanya *bug*, serta melakukan pengetesan
- b. Belajar teknologi baru, dalam pembuatan aplikasi ini, praktikan dihadapi dengan sebuah teknologi baru, sehingga praktikan harus belajar dengan cepat agar aplikasi bisa dipakai
- c. Memberikan pengalaman bekerja bagi praktikan serta memberikan suatu gambaran bagaimana dunia industri berjalan
- d. Belajar berkomunikasi, dalam dunia industri komunikasi adalah suatu hal yang penting, karena tidak jarang akibat

kurangnya atau salahnya informasi, sebuah pekerjaan menjadi berantakan, dan tidak selarasnya kebutuhan antar tim

- e. Melatih kemampuan *problem solving*, pada saat bekerja praktikan diharapkan mampu memberikan sebuah solusi atas permasalahan yang terjadi dalam bidang kerja praktikan
- f. Mendapatkan wawasan baru dari PT Code Development Indonesia mengenai cara membangun dan memelihara aplikasi dengan baik terhadap *client*.
- g. Bertemu dan berdiskusi mengenai pengembangan aplikasi dengan orang – orang yang ahli di bidangnya.

