

# Rangga Dwi Rendra

*by* Perpustakaan Usbypkp

---

**Submission date:** 28-Nov-2024 06:58AM (UTC+0430)

**Submission ID:** 2515573389

**File name:** TugasAkhir\_RanggaDwiRendra24.docx (2.01M)

**Word count:** 5092

**Character count:** 32893

**PENDAHULUAN**

Pada bab ini akan berisi pembahasn tentang latar belakang penelitian, permasalahan dan metode pengembangan aplikasi abit tracker dalam membantu menjadi salah satu dampak positif bagi para pengguna yang ingin merubah dan membentuk kebiasannya menjadi lebih baik.

31

**1.1 Latar Belakang Masalah**

Perkembangan teknologi digital yang pesat saat ini telah mengubah gaya hidup masyarakat modern menjadi serba digital, dimana banyak dari kegiatan dalam kehidupan masyarakat telah mengalami digitalisasi. Berkembangannya perangkat pintar seperti *smartphone*, laptop dan yang lainnya telah memberikan kemudahan untuk penggunanya dalam melakukan kegiatan sehari – hari, juga semakin mudahnya akses internet yang memudahkan dalam berbagi informasi menjadikan teknologi ini semakin banyak yang menggunakannya.

Sebuah survei menunjukkan bahwa lebih dari 19 persen remaja di Indonesia kecanduan internet. Ahli Adiksi Perilaku dr. Kristiana Siste mengatakan angka itu diperoleh berdasarkan survei kepada anak-anak dari 34 provinsi di Indonesia. Ia mengatakan sebagian besar waktu yang dihabiskan anak-anak dan remaja di internet adalah untuk bermain gim online serta media sosial.[1]

Terlihat bahwa penggunaan internet sudah mencapai titik dimana bagi sebagian orang menjadi kecaduan. Kecanduan ini tidak hanya mengganggu kesehatan mental, namun juga dapat mengurangi produktivitas dan menghambat perkembangan diri.

Diperlu solusi yang tidak hanya dapat membantu mengurangi kecanduan, tetapi juga meningkatkan produktif. Salah satu caranya adalah dengan membangun kebiasaan positif melalui teknologi yang sama dengan menjadi sumber kecanduan yaitu perangkat pintar dan internet. Oleh karena itu, aplikasi habit tracker dapat menjadi jawaban untuk membantu individu dalam membangun kebiasaan positif.

Aplikasi ini akan dibuat lintas platform sehingga semakin banyak yang dapat mengakses dan menggunakannya sehingga dapat membantu lebih banyak pengguna dalam membentuk kebiasasn positif dengan cara melacak

perkembangannya dari waktu ke waktu. Aplikasi ini memungkinkan penggunaannya untuk mencatat dan menganalisis perkembangan kebiasaan yang ingin mereka bentuk untuk perkembangan diri menjadi lebih baik kedepannya.

Dengan adanya perkembangan teknologi, dimana saat ini ada teknologi React Native yang memungkinkan untuk membuat aplikasi lintas platform menggunakan komponen yang sama, dengan penggunaan Node.js untuk disisi server membuat aplikasi ini dapat terintegrasi antar platformnya. Adanya internet membuat semua perubahan perkembangan terhadap kebiasaan pengguna dapat diakses dan diperbarui kapan saja dan dimana saja, selama pengguna memiliki perangkat pintar dan akses internet.

## 1.2 Perumusan Masalah

Berdasarkan latar belakang tersebut, rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana perancangan aplikasi *habit tracker* lintas platform yang efisien menggunakan React Native dan Node.js?
2. Apa saja fitur – fitur dalam aplikasi habit tracker yang dapat memenuhi kebutuhan dan meningkatkan efektifitas dalam membangun kebiasaan pengguna?
3. Apa saja cara pemantauan kebiasaan yang disediakan di dalam aplikasi *habit tracker* ini?

## 1.3 Ruang Lingkup Masalah

Ruang lingkup masalah terhadap rumusan masalah pada penelitian ini diantaranya adalah:

1. Aplikasi ini akan dibuat secara lintas platform, untuk saat ini hanya akan dibuat dua versi yaitu untuk mobile android dan web saja.
2. Fokus pada pengembangan fitur utama seperti menambah kebiasaan, mencatat perkembangan kebiasaan dan pemantauan perkembangan kebiasaan (analisis). selain itu, nantinya akan ada fitur – fitur tambahan yang akan diterapkan seperti adanya artikel.
3. Terdapat tiga tipe pemantauan kebiasaan yaitu, “Do or Not”, “Count”, dan “Time”.

4. Analisis akan dibuat dengan tampilan visual diagram sebagai alat bantu untuk melihat dan memantau perkembangan kebiasaan, juga untuk faktor keberhasilan dan kepuasan perkembangan kebiasaannya akan dikembalikan kepada pengguna aplikasi

Dengan adanya runag lingkup ini diharapkan dapat membatasi lingkup dan menunjukkan fokus utama terhadap permasalahan yang ada.

## 1.4 Maksud dan Tujuan Penelitian

### 1.4.1. Maksud

Pengembangan aplikasi habit tracker lintas platform ini dimaksudkan untuk memberikan solusi yang efektif dalam membantu pengguna mencatat, memantau atau melacak, dan mengelola kebiasaan yang ingin dibentuk.

### 1.4.2. Tujuan

Tujuan pengembangan aplikasi habit tracker lintas platform ini adalah sebagai berikut:

1. Menjadi alat bantu yang dapat diakses melalui perangkat android dan web untuk mengembangkan kebiasaan positif.
2. Mengimplemantasikan teknologi react native dan node.js dalam menciptakan aplikasi lintas platform.
3. Memenuhi kebutuhan pengguna dalam proses pengembangan kebiasaan.
4. Memberikan dampak positif pada pengguna dalam pemanfaatan teknologi dalam kegiatan sehari – hari

## 1.5 Metode Pengembangan Perangkat Lunak

Model Agile merupakan pendekatan pengembangan sistem yang memungkinkan perubahan setiap saat.[2] Metode pengembangan Agile dipilih dalam pengembangan aplikasi ini, dimana aspek fokus pada metode ini adalah kesederhanaan, fleksibel dan kecepatan yang memungkinkan memperpendek siklus hidup proyek. Dalam proses pengembangan aplikasi ini, pengembang hanya berkonsentrasi pada fungsi – fungsi atau fitur yang dibutuhkan langsung secara cepat yang nantinya akan bereaksi terhadap informasi umpan balik yang diterima. Dimana disetiap *sprint* sekitar satu sampai tiga minggu selama tiga bulan proyek ini

dibuat akan melakukan siklus *plan, design, develop, test, deploy, review*. Selama masa pengembangan aplikasi hingga adanya pembaruan aplikasi kedepannya.

1. Plan: Tahap penyusunan tujuan kecil yang dapat dicapai dalam jangka pendek. Dimana perencanaan bersifat cepat dan adaptif yang memungkinkan adanya perubahan saat proyek sedang berlangsung.
2. Design: Tahap mulainya perancangan untuk bagaimana solusi akan diwujudkan. *Design* atau perancangan meliputi visual dan bagaimana sistem atau fitur akan berfungsi.
3. Develop: Tahap pengerjaan pembuatan dimana kode mulai ditulis dan produk akan dibuat atau dibentuk.
4. Test: Tahap pengujian untuk memastikan kesesuaian jalannya produk sebagaimana mestinya. *Test* berlangsung paralel dengan *develop*, sehingga kesalahan dapat segera diperbaiki sebelum melangkah ke tahap berikutnya.
5. Deploy: Tahap dimana proyek diimplementasikan ke dalam lingkungan nyata, seperti server produksi atau aplikasi yang dapat diakses. Dimana berlangsung secara bertahap yang memungkinkan evaluasi terus menerus
6. Review: Tahap dimana *feedback* dari pengguna atau pemangku kepentingan dikumpulkan untuk memutuskan apa yang perlu ditingkatkan atau ditambahkan. Dengan siklus ini memastikan proyek tetap relevan dan terus berkembang.

#### 1.6 Sistematika Penulisan

Penulisan Tugas Akhir ini disusun dengan sistematika penulisan yang terdiri dari beberapa bab utama, yaitu:

1. BAB I Pendahuluan, mencakup latar belakang masalah, rumusan masalah, ruang lingkup masalah, maksud dan tujuan penelitian, metode pengembangan perangkat lunak, dan sistematika penulisan. Merupakan gambaran umum tentang mengapa penelitian/proyek ini dilakukan.
2. BAB II Dasar Teori, berisi teori - teori pendukung yang berkaitan pengembangan aplikasi yang dibuat.

3. **BAB III Analisa dan Pemodelan Sistem**, mencakup analisis sistem seperti konsep sistem, analisis target pengguna dan analisis perangkat dan pemodelan system aplikasi menggunakan prototype.
4. **BAB IV Implementasi dan Pengujian Sistem**, Penjabaran implementasi sistem dan pengujian sistem.
5. **BAB V Kesimpulan dan Saran**, mencangkup kesimpulan berupa penjabaran hasil yang didapatkan dari penulisan dan pengerjaan tugas akhir ini, saran berisi hal – hal yang dapat diterapkan nantinya untuk mengembangkan karya tulis selanjutnya.

**DASAR TEORI**

Pada bab ini berisi berbagai teori dan konsep yang menjadi dasar penelitian dan pengembangan proyek ini. Teori – teori ini memberikan landasan kuat dan kerangka pemikiran yang sistematis, serta memberikan pemahaman mendalam mengenai prinsip – prinsip yang diterapkan pada pengembangan proyek aplikasi habit tracker ini.

### 2.1 Aplikasi

Aplikasi merupakan perangkat lunak yang diciptakan dan digunakan untuk membantu menyelesaikan atau mengendalikan suatu pekerja yang spesifik. Aplikasi dapat berjalan melalui platform seperti mobile, web, atau desktop. Pengertian aplikasi secara umum adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya aplikasi merupakan suatu perangkat komputer yang siap pakai bagi user.[3]

Secara istilah pengertian aplikasi adalah suatu program yang siap untuk digunakan yang dibuat untuk melaksanakan suatu fungsi bagi pengguna jasa aplikasi serta penggunaan aplikasi lain yang dapat digunakan oleh suatu sasaran yang akan dituju.[4]

### 2.2 *Habit tracker*

#### 2.2.1. *Habit*

*Habit* adalah serangkaian tindakan yang diulang-ulang oleh seseorang untuk hal yang sama dan berlangsung tanpa proses berpikir.[5] *Habit* atau kebiasaan dapat terbentuk dai proses pembiasaan berulang kali sehingga menjadi bagian dari sebuah rutinitas seseorang.

#### 2.2.2. *Habit tracker*

*Habit tracker* atau pelacak kebiasaan adalah alat pembantu untuk seseorang mengembangkan kebiasaannya dimana *habit tracker* ini membantu penggunanya untuk melacak atau memantau perkembangan pembentukan kebiasaan.

## 2.3 Teknologi

<sup>25</sup> Teknologi merupakan suatu pengetahuan yang ditujukan untuk menciptakan suatu alat atau tindakan pengolahan dan konstruksi benda.[6] teknologi sendiri mencakup alat fisik seperti mesin dan perangkat keras, juga mencakup Teknik, metode, pengetahuan dan proses yang berguna untuk memecahkan masalah dan bahkan memenuhi kebutuhan makhluk hidup.

## 2.4 React Native

### 2.4.1. React native

React Native adalah kerangka kerja pengembangan perangkat lunak berbasis JavaScript yang memungkinkan pengembang <sup>19</sup> untuk membuat aplikasi mobile yang responsif dan dapat diakses di berbagai platform.[7] Dimana teknologi ini dikembangkan oleh Facebook dan pertama kali diluncurkan pada tahun 2015. React Native menggunakan komponen *User Interface (UI)* native dan menggabungkannya dengan *JavaScript*, yang menghasilkan performa yang mendekati aplikasi native untuk setiap platform.

### 2.4.2. React Native Web

React Native Web merupakan *library* yang memungkinkan React Native untuk bisa merender aplikasi web menggunakan sumber kode yang sama dengan aplikasi mobile. *Library* ini merupakan salah satu teknologi penting pengembangan aplikasi lintas platform yang nantinya aplikasi lintas platform itu dalam menjangkau dan menawarkan fleksibilitas bagi para penggunanya.

## 2.5 Expo

Expo merupakan kerangka kerja react native tingkat produksi untuk mengembangkan aplikasi untuk android, ios, dan web. Expo menyederhanakan proses pembangunan dikarenakan expo telah menyediakan lingkungan kerja yang telah dikonfigurasi dan juga memiliki perpustakaan standar modul asli.

## <sup>18</sup> 2.6 Node.js

Node.js adalah perangkat lunak yang didesain untuk mengembangkan aplikasi berbasis web dan ditulis dalam sintaks bahasa pemrograman JavaScript.[8]



dikembangkan oleh Ryan Dahl pada tahun 2009, dan telah menjadi salah satu platform paling populer saat ini untuk mengembangkan aplikasi di sisi server atau biasa dikenal dengan backend menggunakan JavaScript. Node.JS dapat berjalan di server karena dukungan dari V8 Engine buatan Google dan beberapa modul bawaan yang terintegrasi seperti modul HTTP, modul *file system*, modul *security* dan beberapa modul penting lainnya.[9]

Node.js mampu untuk menangani banyak koneksi secara simultan dengan tingkat efisiensi yang tinggi. Hal ini dapat Node.js lakukan karena memiliki arsitektur *Event Driven* dan *Non-Blockin I/O* didalamnya.

#### 2.5.1. *Event Driven*

Node.js dirancang untuk merespons *event* terhadap permintaan, dimana akan mengeksekusi callback terkait permintaan tersebut. Dengan demikian, Node.js dapat menangani permintaan yang bersanaab tanpa harus menunggu proses permintaan sebelumnya selesai terlebih dahulu,

#### 2.5.2. *Non-Blockin I/O*

Node js menungkinkan untuk melakukan operasi input output (I/O) secara *non-blocking* dalam Node.js akan melanjutkan eksekusi lain tanpa membuat eksekusi program lainnya berhenti atau mengganggu I/O tersebut selesai terlebih dahulu.

Arsitektur tersebut membuat Node.js cocok untuk aplikasi yang membutuhkan kinerja dan skalabilitas yang tinggi.

### 2.7 UML

UML adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek.[10] Sering juga didefinisikan sebagai salah satu standar visualisasi perancangan dan pendokumentasian pengembangan suatu sistem, juga dikenal sebagai standar *blueprint* sebuah perangkat lunak (software/aplikasi).

#### 2.6.1. *Use Case Diagram*

Use case diagram merupakan pemodelan untuk kelakuan sistem informasi yang akan dibuat. Use case bekerja dengan mendeskripsikan

tipikal interaksi antara pengguna sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sistem itu dipakai.[10]

22

### 1. Aktor

Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.[11] Digambarkan seperti gambar



Gambar 2.1 Aktor Use Case

7

### 2. *Include*

Sebuah use case dapat meng-include fungsionalitas use case lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa use case yang di-include akan dipanggil setiap kali use case yang meng-include dieksekusi secara normal.[11]

### 3. *Extend*

Sebuah use case juga dapat meng-extend use case lain dengan behaviour-nya sendiri. Sementara hubungan generalisasi antar use case menunjukkan bahwa use case yang satu merupakan spesialisasi dari yang lain.[11]

14

4

### 2.6.2. *Class Diagram*

*Class diagram* merupakan gambaran struktur sistem dari segi pendefinisian kelas – kelas yang akan dibuat untuk membangun sistem. *Class diagram* terdiri dari atribut dan operasi dengan tujuan pembuat program dapat membuat hubungan antara dokumentasi perancangan dan perangkat lunak sesuai.[10]

### 2.6.3. *Sequence Diagram*

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek.[10] Fokus diagram ini ada pada urutan kronologis dari interaksi antar objek didalam sistem yang menunjukkan alur kerjaseiring dengan berjalannya waktu.

### 2.6.4. *Activity Diagram*

Activity diagram merupakan diagram yang menggambarkan *workflow* atau aktivitas dari sebuah sistem yang ada pada perangkat lunak.[10] Diagram ini biasa digunakan untuk memodelkan alur kerja dari sistem, dimana menunjukan aktivitas dimulai, saling berhubungan, dan berakhir.

### 2.6.5. *Component diagram*

Diagram komponen adalah diagram yang dibuat untuk menunjukkan organisasi dan ketergantungan diantara kumpulan komponen dalam suatu sistem.[12]

### 2.6.6. *Deployment diagram*

Diagram Deployment adalah diagram yang menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi.[12]

## 2.8 *Black Box Testing*

Merupakan pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak, *tester* dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program.[13]

Pengujian *black box* merupakan salah satu jenis metode pengujian yang memperlakukan perangkat lunak yang tidak diketahui kinerja internalnya. Sehingga para tester memandang perangkat lunak seperti layaknya sebuah “kotak hitam” yang tidak penting dilihat isinya, tapi cukup dikenai proses testing dibagian luar.[14]

## 2.9 *MongoDB*

MongoDB adalah basis data NoSQL yang bersifat document based. MongoDB bersifat document based artinya MongoDB tidak memilki tabel, kolom

ataupun baris. MongoDB hanya memiliki koleksi dan dokumen. Data yang disimpan dalam basis data MongoDB berupa file JSON yang disebut dengan istilah BSON (Binary JSON). Sistem basis data MongoDB menggunakan *key-value*, artinya setiap dokumen dalam MongoDB dipastikan memiliki *key*. [15]

#### 2.10 *MongoDB Atlas*

MongoDB Atlas yang merupakan *cloud based Database Management System* (DBMS) yang menggunakan konsep NoSQL terfokus dengan data yang tidak terstruktur. [16]

#### 2.11 *Express*

*Express.js* adalah *framework* web app untuk Node.js yang ditulis dengan bahasa pemrograman JavaScript. Dimana *framework* ini digunakan untuk membangun aplikasi dari sisi back end secara efektif dan optimal. [17] *ExpressJS* merupakan sebuah Node.JS *framework* yang berguna untuk mempermudah pembuatan aplikasi berbasis Node.JS dengan menggunakan design pattern yang dapat disesuaikan dan sangat fleksibel. [9]

#### 2.12 *Web Server*

Web server merupakan software yang memberikan layanan data yang berfungsi menerima permintaan Hypertext Transfer Protocol (HTTP) dari client yang dikenal dengan browser web dan mengirimkan kembali hasilnya dalam bentuk halaman - halaman website yang umumnya menampilkan teks, gambar, animasi dan Video. Web server merupakan penyedia layanan akses kepada pengguna melalui protokol komunikasi HTTP atas berkas-berkas yang terdapat pada suatu situs web. [18]

#### 2.13 *Android*

Android adalah sistem operasi untuk telepon seluler yang berbasis linux. Android menyediakan platform terbuka bagi para pengembang untuk membuat aplikasi mereka sendiri. [19]

Pada saat perilis perdana Android, 5 November 2007, Android bersama *Open Handset Alliance* menyatakan mendukung pengembangan *open source* pada perangkat mobile. Di lain pihak, Google merilis kode-kode Android di bawah lisensi Apache, sebuah lisensi perangkat lunak dan open platform perangkat seluler. [4]

## 5 BAB III

### ANALISIS DAN PEMODELAN SISTEM

Pada bab tiga ini berisi pembahasan mengenai analisis dan pemodelan system yang dikembangkan pada proyek tugas akhir. Melalui analisis dan pemodelan ini diharapkan dapat diperoleh gambaran alur kerja juga alur kerja sistem yang memudahkan proses pengembangan juga implementasi pada tahap berikutnya.

#### 16 3.1 Analisis Sistem

Analisis sistem bertujuan untuk memahami kebutuhan aplikasi berdasarkan kebutuhan pengguna, mengevaluasi dan merancang solusi terhadap permasalahan kebutuhan sistem, proses ini mencakup beberapa tahapan penting, yaitu:

##### 3.1.1 Konsep Sistem

Aplikasi habit tracker merupakan aplikasi yang bertujuan untuk membantu para pengguna dalam memantau perkembangan yang akan kebiasaan pengguna. Pemantauan kebiasaan akan dibagi menjadi tiga tipe yaitu:

###### 1. *Do or Not*

Tipe dimana pengguna memantau perkembangan kebiasaan dengan mencatat kapan saja kebiasaan itu dilakukan. Tipe ini bertujuan untuk melihat peningkatan atau penurunan intensitas. Contohnya kebiasaan merapikan tempat tidur, dimana dalam tujuh hari pertama pengguna hanya melakukannya tiga kali saja dan seterusnya.

###### 2. *Count*

Tipe dimana pengguna memantau perkembangan kebiasaan berdasarkan jumlah objek atau kebiasaan harian terlaksana. Tipe ini bertujuan untuk melihat peningkatan atau penurunan kuantitas.

Contohnya kebiasaan minum air, dimana hari – hari pertama memulai hanya meminum empat gelas per hari, yang dikemudian hari meningkat menjadi lima dan seterusnya.

### 3. *Time*

Tipe dimana pengguna membandingkan waktu sebuah kebiasaan dilakukan.

Contohnya kebiasaan tidur malam, yang pada hari – hari awal mungkin tidur terlalu larut (misalnya: 23:00), namun dihari – hari berikutnya terpantau jam tidur sudah semakin dini (misalnya: 21:00).

Pengguna dapat menentukan sendiri kebiasaan seperti apa yang ingin di bentuk atau mengambil dari preset kebiasaan yang terdapat didalam aplikasi yang bisa di sesuaikan kembali nantinya.

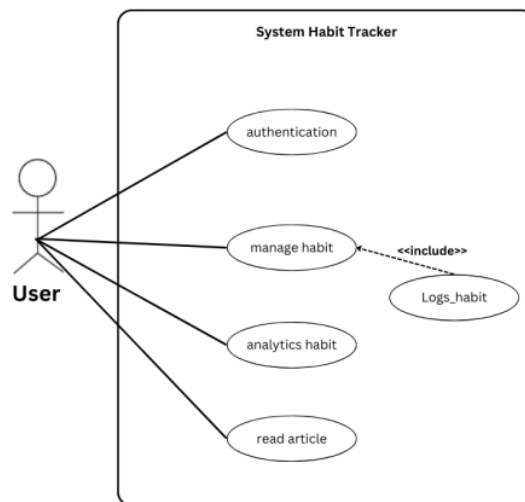
### 1. *UML (Unified Modeling Language)*

*UML* akan memudahkan penggambaran tentang konsep aplikasi.

Diagram yang akan ditampilkan diantaranya:

#### a. *Usecase diagram*

Dapat terlihat dari Gambar, menunjukan usecase diagram pada aplikasi habit tracker.



### 1) Aktor

Pada sistem ini hanya terdapat user, dimana user dapat mengakses keseluruhan fitur / sistem dalam aplikasi ini.

Use Case yang terdapat dalam sistem aplikasi *habit tracker* sebagai berikut:

#### a. Authentication

Proses registrasi dan login akun pengguna.

#### b. Manage habit

Mengatur *habit* atau kebiasaan meliputi *create habit*, *edit habit* dan *delete habit*.

#### c. Analytics habit

Menampilkan data catatan perkembangan kebiasaan.

#### d. Read article

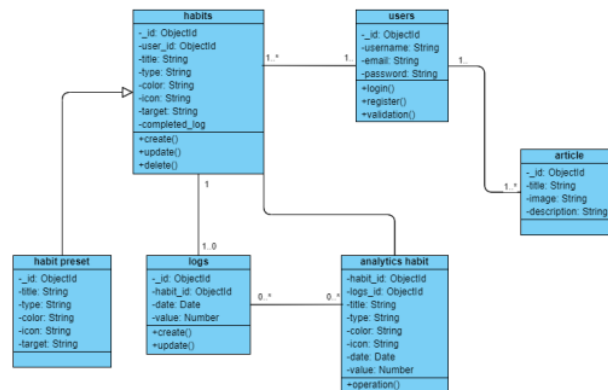
Menampilkan bacaan berupa artikel mengenai perkembangan kebiasaan.

### Include Usecase

*Logshabit*: proses pencatatan perubahan perkembangan kebiasaan.

36

### b. Class Diagram

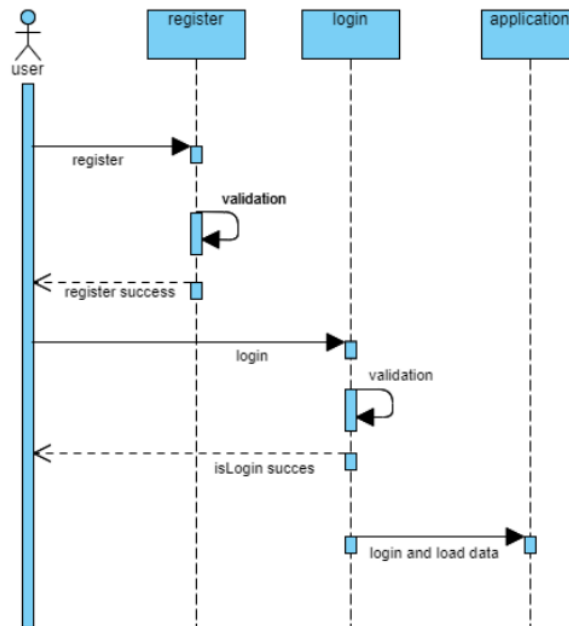


Gambar 3.2 class diagram aplikasi habit tracker

Class yang terdapat pada sistem aplikasi habit tracker, sebagai berikut:

1. *User* (Terdapat metode login dan register)
2. *Habit* (Terdapat metode create, update dan delete data)
3. *Habit\_Preset*
4. *Logs* (Terdapat metode create dan update data)
5. *Article*

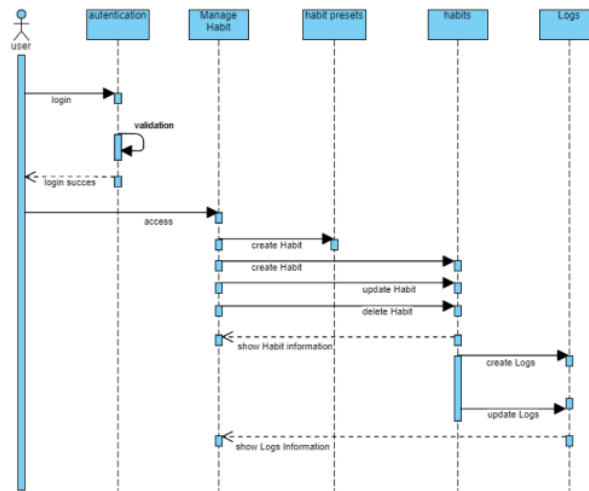
39  
c. *Sequence Diagram*



Gambar 3.3 *sequence diagram authentication*

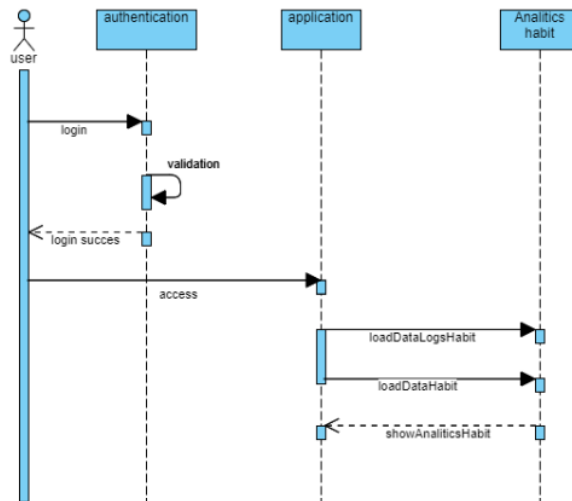
1. *user*: Pengguna melakukan register ke sistem (register)
2. *register*: Pengguna melakukan registrasi, dan sistem memverifikasi data register (validation).
3. *login*: jika pengguna sudah melakukan registrasi akun, maka pengguna bisa login, (validation) dilakukan untuk verifikasi data login user.
4. *application*: jika (islogin succes), maka akses diberikan (login and load data).





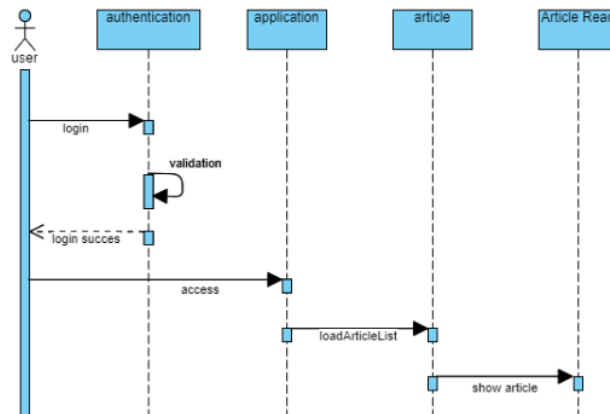
Gambar 3.4 sequence diagram manage habit

1. *user*: Pengguna melakukan login dan berinteraksi dengan aplikasi.
2. *authencation*: Pengguna melakukan (login), sistem melakukan (*validation*).
3. *Manage Habit*: (*login Succes*) maka (*access*) didapatkan, jika melakukan (*create habit, update habit, delete habit*), jika terdapat data *habit* maka (*ShowHabitInformation*) dan apabila terdapat logs terhadap *habit* juga (*ShowLogsInformation*).
4. *habit preset*: melanjutkan (*createHabit*) dan menentukan menggunakan preset atau tidak.
5. *Habit*: jika (*habitCreated*), (*showHabitsInformation*). Dapat melakukan (*createLogs* dan *updateLogs*).
6. *Logs*: jika terjadi (*createLogs* atau *updateLogs*), maka (*showLogsInformation*).



Gambar 3.5 sequence diagram analytics

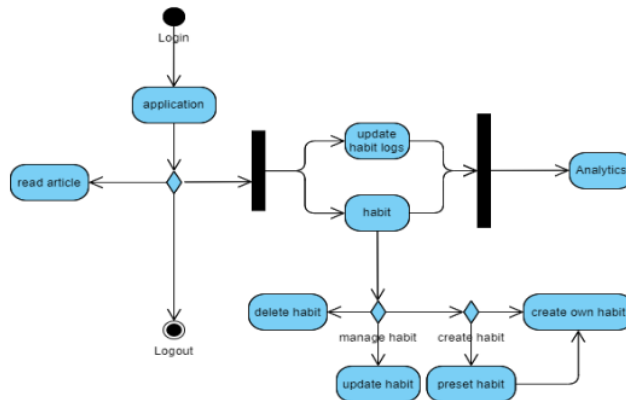
1. *user*: Pengguna melakukan login dan berinteraksi dengan aplikasi.
2. *authencation*: Pengguna melakukan (login), sistem melakukan (*validation*).
3. *application*: *application* disini merujuk pada halaman didalam aplikasi diluar login dan register. (*loginSucces*) maka (*access*) aplikasi, melakukan permintaan untuk *analytics habit*.
4. *Analytics*: (*loadDataLogsHabit*) dan (*loadDataHabit*) untuk nantinya ditampilkan (*showAnalyticsHabit*).



Gambar 3.6 sequence diagram article

1. *user*: Pengguna melakukan login dan berinteraksi dengan aplikasi.
2. *authencation*: Pengguna melakukan (login), sistem melakukan (validation).
3. *application*: (*loginSucces*) maka (*access*) aplikasi,
4. *article*: melakukan permintaan untuk (*loadArticleList*).
5. *article read*: artikel yang dipilih maka akan ditampilkan (*showArticle*).

23  
d. Activity Diagram



Gambar 3.5 activity diagram aplikasi habit tracker

1. Mulai (Start): Dimulai dengan aksi Login.
2. application: Setelah Login, akan diarahkan ke halaman utama aplikasi. Didalamnya terdapat pilihan aktifitas yaitu *read article*, *manage habit*, *update habit logs*, dan *analytics*.
3. *read article*: didalamnya pengguna dapat membaca berbagai artikel mengenai *habit* atau kebiasaan.
4. *habit manage*: melakukan pengelolaan seperti *create new habit*, *update habit*, dan *delete habit* terhadap habit yang ingin dibangun.
5. *create habit*: terdapat pilihan membuat habit baru dimana bisa menggunakan preset yang sudah disediakan atau membuat sendiri habit yang ingin dibangun.
6. *habit*: Kebiasaan yang sudah dibuat.
7. *Update habit logs*: Memperbarui catatan perkembangan kebiasaan.
8. *Analytics*: Pemantauan hasil perkembangan kebiasaan berdasarkan pencatatan yang dilakukan tiap harinya saat update logs.
9. *logout*: dimana pengguna keluar dari aplikasi.

### 3.1.2 Analisis Target Pengguna

Aplikasi ini memiliki target pengguna semua kalangan yang memiliki perangkat khususnya *smartphone android* dan pengguna aplikasi web yang dapat diakses di berbagai perangkat.

#### 1. Usia

Target pengguna aplikasi ini mulai dari anak – anak dengan bimbingan orang tua hingga orang dewasa, dengan target utama adalah remaja dikarenakan usia remaja mayoritas orang sudah dapat menggunakan perangkat komputer dan *smartphone* dalam berkehidupan sehari – hari, sehingga mereka dapat merasakan dampak positif penggunaan teknologi sedari kecil khususnya untuk memulai membangun kebiasaan positif diusianya.

## 2. Lokasi Geografis

Target pengguna berdasarkan Geografis adalah masyarakat yang tempat tinggalnya sudah memiliki akses internet, walau pada kenyataannya memang yang kebanyakan masih berada di daerah perkotaan karena selain akses internet juga kemampuan seseorang atau masyarakat suatu daerah untuk memiliki perangkat komputer dan smartphone.

## 3. Minat dan Gaya Hidup

Target pengguna aplikasi ini adalah masyarakat yang sudah terbiasa menggunakan dan memanfaatkan teknologi khususnya internet pada kehidupan dan kegiatan sehari – hari. Dimana mayoritas masyarakat modern sekarang sudah ketergantungan terhadap kebudayaan teknologi dan internet alat pembantu setiap harinya.

## 4. Pengalaman pengguna

### a. Harapan:

- 1) Aplikasi dapat yang responsif di berbagai perangkat
- 2) Tipe pengukuran dapat memenuhi kebutuhan pengguna
- 3) Dapat membantu pengembangan kebiasaan positif pada masyarakat dan individu manapun.

### b. Tantangan:

- 1) Memastikan aplikasi dapat berjalan di berbagai perangkat
- 2) Memastikan fitur – fitur yang sudah ada cukup untuk membantu memenuhi kebutuhan pengguna.
- 3) Membuat alur penggunaan aplikasi sesederhana mungkin.

### 3.1.3 Analisis Perangkat

Analisis perangkat akan berfokus pada spesifikasi perangkat seperti apa yang dapat digunakan pengguna untuk mengakses aplikasi dan spesifikasi teknis saat pengembangan.

#### 1. Perangkat pengguna

Aplikasi akan dibuat menjadi dua yaitu aplikasi berbasis android dan berbasis web.

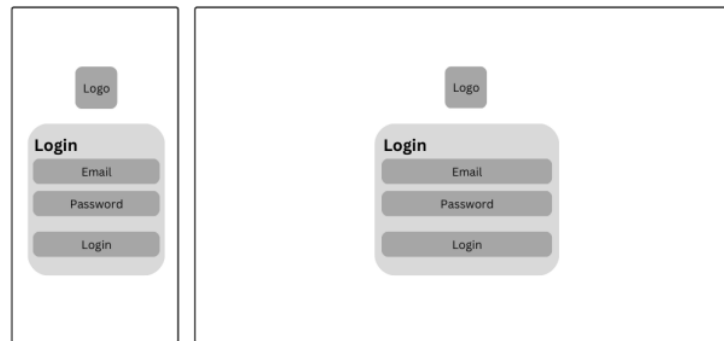
- a. Android
    - 1) Sistem Operasi: Android 8.0 (Oreo) ke atas untuk memastikan kompatibilitas dengan fitur – fitur yang ada dan aka nada nantinya.
    - 2) Resolusi Layar: mulai dari 4 inci hingga ukuran layer yang lebih besar.
    - 3) *Prosesor* : *Quad-core* atau *Octa-core* untuk performa yang lebih lancer.
    - 4) *Memori/Ram* : disarankan 3GB atau lebih untuk mendapatkan penggunaan yang lancer.
  - b. Web
    - 1) *Browser* : Chrome, Firefox, Edge.
    - 2) Resolusi Layar: Variatif, mulai dari layar Laptop hingga layar *smartphone*.
2. Pengembang
- Text Editor* : Visual Studio Code.
  - Bahasa Pemrograman* : JavaScript, JSX dan Kueri MongoDB
  - Framework* : React Native Expo.
  - Lainnya* : Node.js server dengan Express, Axios, dan MongoDB Atlas untuk basis datanya.

### 3.2 Pemodelan

Pemodelan sistem pada Aplikasi habit track berbasis web dan android, akan digambarkan melalui digital prototype. Digital prototype menggambarkan alur pengguna dalam penggunaan aplikasi habit tracker.

#### a. Halaman Login

Halaman login terbuka ketika pengguna belum melakukan login pada aplikasi. Pengguna tidak akan bisa mengakses halaman dan hal lain didalam aplikasi ketika tidak atau belum melakukan login.

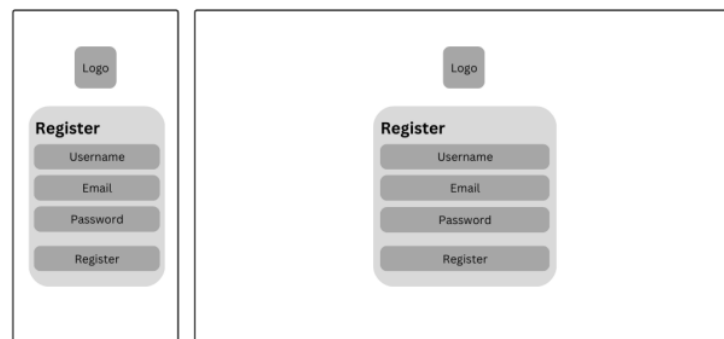


*Gambar 3.6 prototype login*

Terdapat dua inputan yaitu email dan password, dimana keduanya harus sesuai dengan akun yang pernah di registrasikan sebelumnya.

b. Halaman Register

Halaman register ditujukan untuk membuat akun bagi pengguna yang belum memiliki akun untuk dapat login.



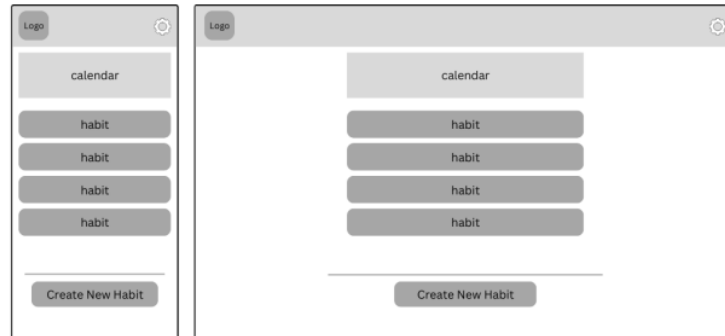
*Gambar 3.7 prototype register*

Untuk mendaftar, pengguna perlu memasukan inputan berupa username, email dan password. akan nada proses verifikasi terhadap alamat email yang sudah terdaftar sebelumnya. Jika register berhasil, halaman akan beralih kembali ke halaman login, pengguna sudah memiliki akses masuk / menggunakan aplikasi

c. Halaman Utama / WorkSpace

Halaman utama berisi list dari kebiasaan – kebiasaan yang sudah dibuat/tambahkan, pada halaman ini pembaruan perkembangan

kebiasaan dilakukan. Mayoritas navigasi ke halaman lain diakses melalui halaman utama ini.



*Gambar 3.8 prototype halaman workspace*

d. Halaman *Preset*

Halaman *Preset* diakses ketika tombol *Create New Habit* di halaman utama diklik. Berisi *preset* kebiasaan – kebiasaan yang dapat diambil atau sebagai acuan untuk mengambil habit seperti apa, *preset* dapat di atur atau disesuaikan kembali di halaman tambah kebiasaan baru.

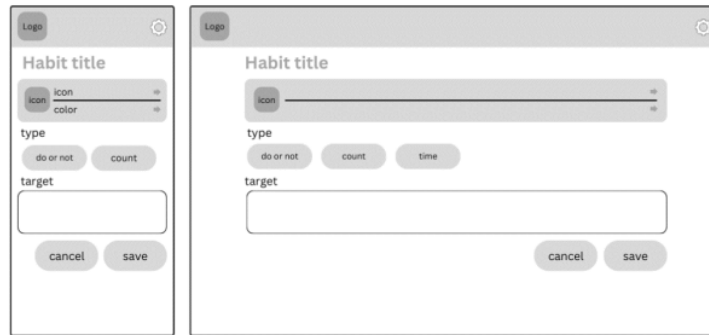


*Gambar 3.9 prototype preset habit*

e. Halaman *Create New Habit*

Halaman untuk mengatur habit seperti apa yang ingin ditambahkan dihalaman utama nantinya. Mencangkup pemberian nama atau judul kebiasaan, icon, warna, tipe pemantauan seperti apa yang ingin ambil dan target seperti apa yang ingin dicapai.

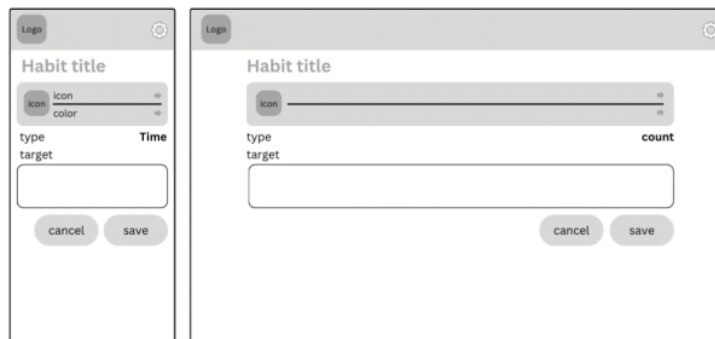




Gambar 3.10 prototype halaman create new habit

f. Halaman *Edit habit*

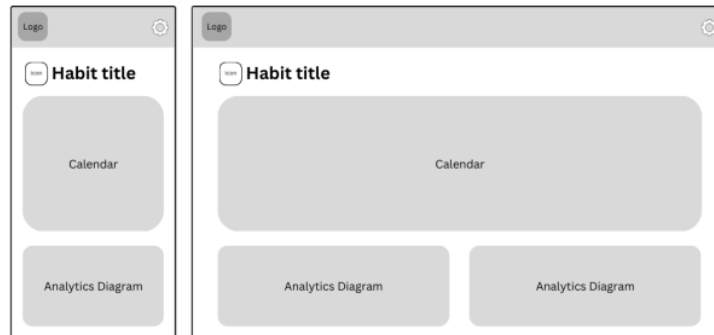
Halaman edit bertujuan untuk mengubah beberapa hal yang sudah pernah dibuat sebelumnya seperti judul habit, icon, warna dan target namun tidak dengan type.



Gambar 3.11 prototype edit habit

g. Halaman *Habit Analytics*

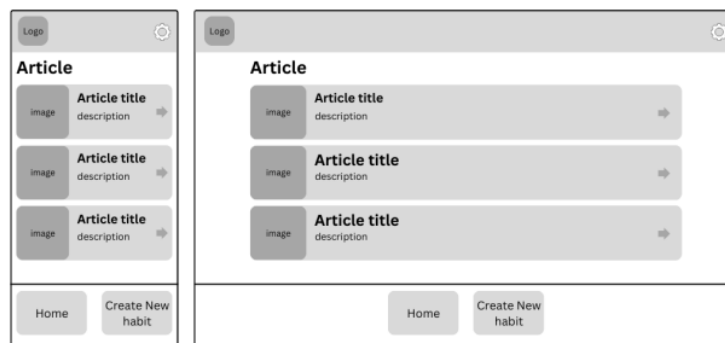
Habit analytics diakses melalui halaman utama dimana berisi tampilan dari kebiasaan yang dipilih di halaman utama sebelumnya.



*Gambar 3.12 prototype habit analytics*

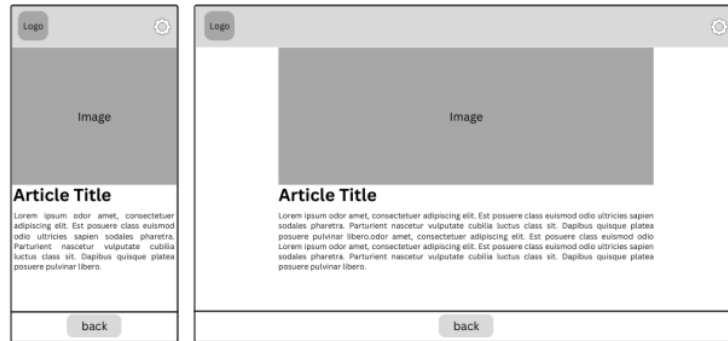
h. Halaman Artikel

Halaman ini menampilkan artiken – artiken informatif yang berhubungan dengan pemahaman dan meningkatkan pembentukan kebiasaan positif.



*Gambar 3.13 prototype pilihan artiken*

Halaman pertama yang akan tampil adalah halaman yang berisi list artiken, yang mana akan ditambahkan dan diperbarui kedepannya.



*Gambar 3.14 prototype artikel*

Saat salah satu artikel diklik, maka akan diarahkan pada halaman utama artikel yang berfungsi untuk membaca lebih lanjut mengenai artikel tersebut.

## 5 BAB IV

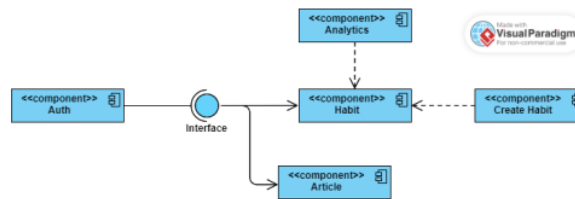
### IMPLEMENTASI & UJI COBA SISTEM

#### 4.1 Implementasi Sistem

Proses penerapan dan pengintegrasian terhadap solusi dari aplikasi atau sistem yang telah dirancang ke dalam lingkungan operasional.

##### 4.1.1 *Component & Deployment Diagram*

###### 1. *Component Diagram*



Gambar 4.1 component diagram

###### a. *User*

Komponen yang mewakili entitas pengguna dalam aplikasi. Pengguna dapat berinteraksi dengan aplikasi melalui modul atau komponen lainnya. 'User' terkait dengan komponen 'Auth' untuk otentikasi.

###### b. *Auth*

Komponen yang menangani proses otentifikasi dan otorisasi pengguna. Setelah itu pengguna dapat mengakses fitur lain dalam aplikasi. 'Auth' terhubung dengan 'User', 'Habit' dan 'Article'.

###### c. *Habit*

Komponen inti yang mengelola hampir seluruh data kebiasaan dalam aplikasi. 'Habit' berinteraksi dengan semua modul lain kecuali 'Article'.

###### d. *Create Habit*

Komponen dimana pengguna dapat membuat kebiasaan baru. 'Create Habit' berinteraksi dengan 'Habit' dimana menjadi pintu masuk untuk menambahkan data baru ke modul 'Habit'.

e. *Analytics*

Komponen yang berfungsi untuk menganalisis data kebiasaan pengguna untuk di visualisasikan.

'Analytics' berinteraksi dengan 'Habit' untuk mengambil data kebiasaan pengguna

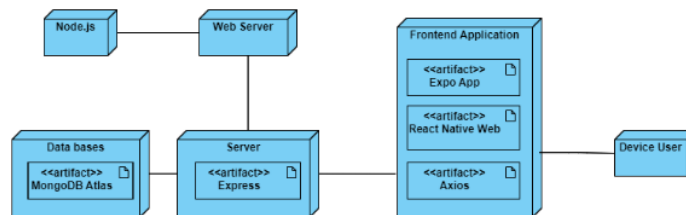
f. *Article*

Komponen yang menyediakan konten artikel terkait dengan pengembangan kebiasaan.

'Article' berinteraksi dengan 'User' dan 'Auth' untuk dapat mengaksesnya.

20

## 2. Deployment Diagram



Gambar 4.2 deployment diagram

Diagram ini menampilkan gambaran arsitektur sistem aplikasi yang terdiri dari beberapa entitas, berikut adalah penjelasannya:

a. *Node.js*

Digunakan sebagai runtime environment untuk menjalankan server aplikasi.

b. *Web Server*

Berfungsi sebagai penghubung antara server dan *frontend application*, dimana web server menerima *request* dari pengguna (device User) dan mengirimkan respons yang sesuai.

c. *Server*

Bertanggung jawab untuk menangani logika bisnis aplikasi. Dimana aplikasi ini menggunakan Express yang merupakan framework untuk Node.js. Express bertindak sebagai perantara antara frontend application dan backend, mengelola request HTTP dari aplikasi frontend.

d. *Frontend Application*

Terdiri dari beberapa artifak, yaitu:

1) *Expo App*

Framework dan platform utama pembangunan aplikasi khusus dibagian *frontend*.

2) *React Native Web*

Library untuk membuat component react native bisa di build menjadi web.

3) *Axios*

Digunakan untuk menangani komunikasi seperti mengirim request HTTP dari aplikasi frontend ke server.

e. *Device User*

Pengguna aplikasi (*Device User*) berinteraksi dengan aplikasi melalui perangkat pengguna, mengirimkan request melalui frontend application yang nantinya akan menerima respons yang telah diolah server.

f. *Databases*

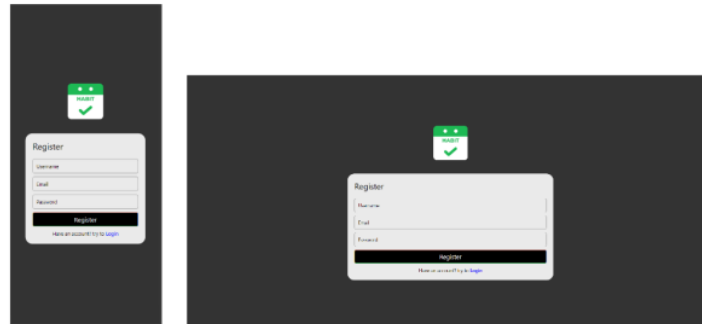
Menggunakan data bases cloud MongoDB Atlas.

4.1.2 Antarmuka Pengguna

Antar muka pengguna atau *User Interface (UI)* merupakan bagian dari sebuah aplikasi dimana bertugas untuk berinteraksi langsung dengan pengguna melalui tampilan visual. Pada aplikasi habit tracker lintas platform web dan mobile android ini tampilan *UI* dibuat semirip mungkin untuk yang bertujuan untuk memudahkan pengguna.

### 1. Register page

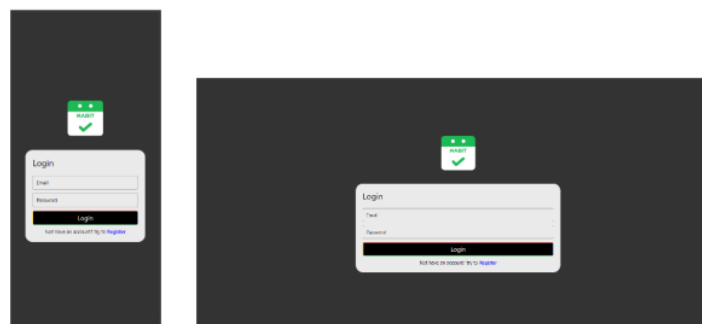
Halaman yang berfungsi untuk daftar atau membuat akun untuk login pada aplikasi bagi pengguna yang belum memiliki akun.



Gambar 4.3 UI register

### 2. Login page

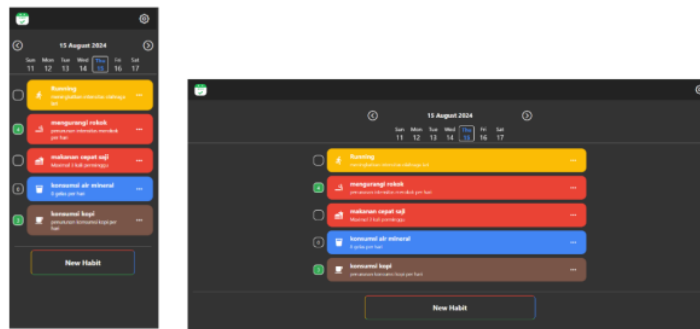
Halaman yang berfungsi sebagai pintu masuk untuk pengguna sebelum menggunakan aplikasi secara keseluruhan, yang nantinya jika login berhasil pengguna akan mendapat akses masuk dan menggunakan aplikasi *habit tracker*, login juga berfungsi untuk menentukan data pengguna mana yang di akan diambil dan ditampilkan dalam aplikasi nantinya.



Gambar 4.4 UI login

### 3. Main page

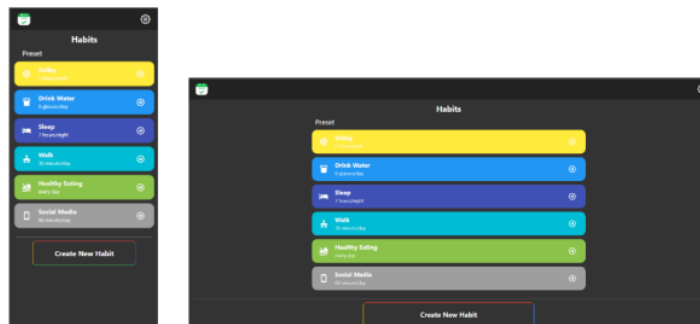
Halaman utama yang berisi hampir semua akses navigasi ke halaman lain, juga terdapat proses memberikan action perubahan atau perkembangan kebiasaan yang sudah dibuat atau ambil.



Gambar 4.5 UI halaman utama

### 4. Preset page

Halaman yang menampilkan preset kebiasaan – kebiasaan yang mungkin bisa menjadi inspirasi pengguna dimana preset sudah disediakan didalam aplikasi yang nantinya tetap bisa disesuaikan oleh pengguna sebelum mengambilnya.

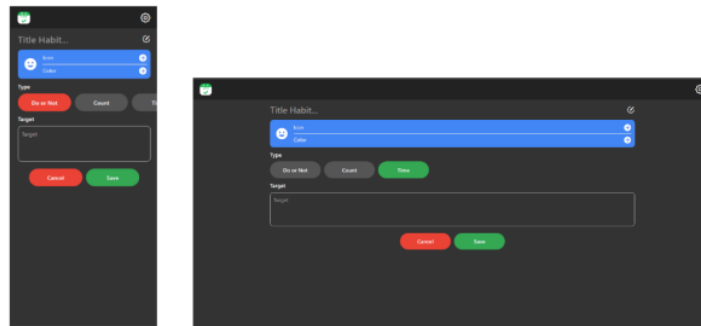


Gambar 4.6 UI preset



### 5. Create Habit page

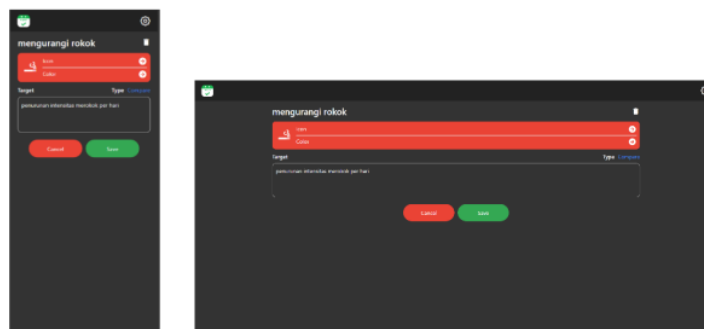
Halaman untuk menyesuaikan kebiasaan yang ingin dibuat sendiri maupun mengambil dari preset. Dimana pengguna dapat menentukan nama atau judul, icon, warna, tipe dan target kebiasaan yang nantinya akan disimpan dan ditampilkan dihalaman utama.



Gambar 4.7 UI halaman create habit

### 6. Edit Habit page

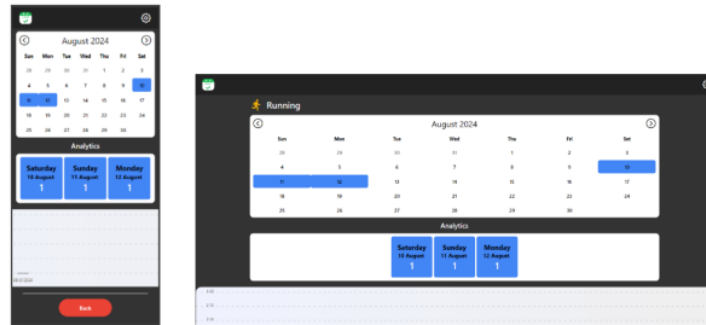
Halaman yang berisi fitur mengubah beberapa hal dari kebiasaan yang sudah dibuat sebelumnya, dihalaman ini juga terdapat fitur hapus kebiasaan.



Gambar 4.8 UI halaman edit

### 7. Analytics page

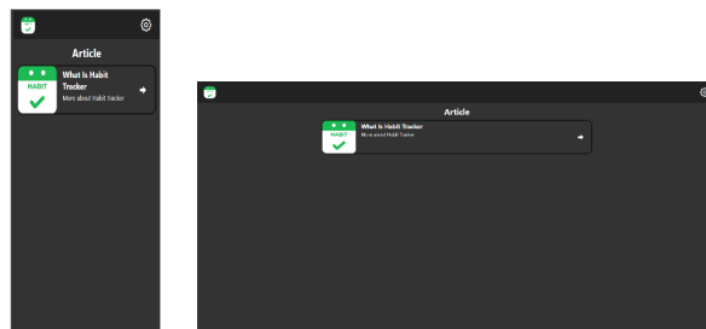
Halaman yang berisi fitur visualisasi data perkembangan kebiasaan pengguna.



Gambar 4.9 UI halaman analytics

### 8. Article page

Halaman berisi artikel – artikel informatif seputar pengembangan kebiasaan untuk memberi pemahaman dan informasi yang mungkin dapat membantu pengguna meningkatkan perkembangan kebiasaan.



Gambar 4.10 UI halaman article list

## 4.2 Pengujian Sistem

Pengujian sistem akan menggunakan metode *black box testing*, pengujian dibagi menjadi dua tahap, yaitu untuk aplikasi mobile android dan aplikasi web.

### 1. Aplikasi android

Tabel 4.1 Hasil Pengujian Android

No	Uji	Harapan	Validitas Y/T	Skor
1	Auth	Username, Email dan Password sesuai dan dapat tervalidasi	Y	100
2	API Data	Data yang di ambil dan dikirim sesuai.	Y	100
3	Navigasi halaman	Semua halaman ternavigasi dan dapat diakses dengan baik	Y	100
4	Tombol	Semua tombol berjalan dengan baik	Y	100
5	Analisis habit	Fitur dan komponen analisis berjalan dengan baik	Y	80
6	Artikel	Tampilan dan konten artikel sesuai	Y	100
7	Aplikasi terinstall dengan baik	Aplikasi dapat terinstal pada perangkat android	Y	100

## 2. Aplikasi web

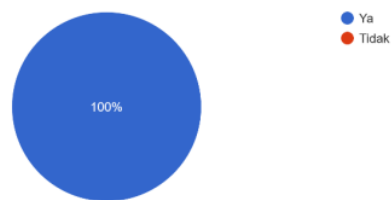
Tabel 4.2 Hasil Pengujian Web

No	Nama	Harapan	Validitas Y/T	Skor
1	Auth	Username, Email dan Password sesuai dengan validasi	Y	100
2	API Data	Data yang di ambil dan dikirim sesuai.	Y	100
3	Navigasi halaman	Semua halaman ternavigasi dan dapat diakses dengan baik	Y	100
4	Tombol	Semua tombol berjalan dengan baik	Y	100
5	Analisis habit	Fitur dan komponen analisis berjalan dengan baik	Y	80
6	Artikel	Tampilan dan konten artikel sesuai	Y	100
7	Aplikasi terinstall dengan baik	Aplikasi dapat terinstal pada perangkat android	Y	100

## 4.3 Survei Pengguna

Survei terhadap aplikasi dilakukan kepada empat belas responden, sebagai berikut:

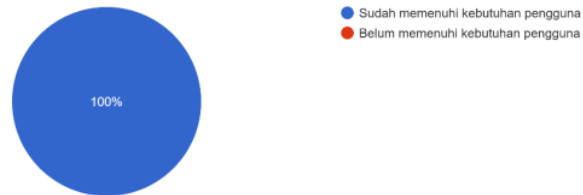
Apakah Anda memiliki keinginan untuk mengembangkan kebiasaan positif dalam diri Anda  
14 jawaban



Gambar 4.11 hasil survei 1

Survei pertama menanyakan tentang keinginan untuk mengembangkan kebiasaan positif. Hasil survei menunjukkan bahwa seluruh responden memiliki keinginan dalam mengembangkan kebiasaan positif.

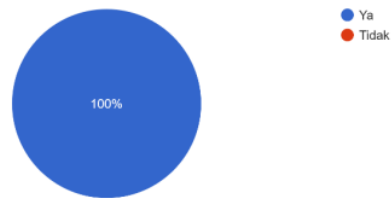
berikan tanggapan anda!  
14 jawaban



*Gambar 4.12 hasil survei 2*

Survei kedua menanyakan apa aplikasi habit tracker beserta fitur – fitur yang tersedia sudah memenuhi kebutuhan pengguna. Hasil survei menunjukkan bahwa seluruh responden menjawab sudah memenuhi kebutuhan.

Apakah dengan adanya aplikasi habit tracker ini, pengguna merasakan dampak positif penggunaan teknologi dalam kehidupan sehari  
14 jawaban



*Gambar 4.13 hasil survei 3*

Survei ketiga menanyakan apa pengguna merasakan dampak positif dalam penggunaan teknologi dari adanya aplikasi *habit tracker* ini. Hasil survei menunjukkan bahwa aplikasi ini memberikan dampak positif.

## KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan dari hasil penulisan berdasarkan isi laporan yang menjawab rumusan masalah yang ada, juga terdapat saran sebagai bahan pembelajaran kedepan

### 5.1 Kesimpulan

Berdasarkan hasil penelitian dan pengembangan yang telah dilakukan pada proyek aplikasi *habit tracker* lintas platform, dapat ditarik kesimpulan sebagai berikut:

1. Aplikasi *habit tracker* lintas platform ini dirancang menjadi dua versi yaitu web dan mobile android dimana aplikasi ini bertujuan untuk membantu penggunanya dalam mengembangkan kebiasaan positif.
2. Aplikasi *habit tracker* ini menggunakan teknologi react native dan node.js dalam perancangannya dimana hal ini memungkinkan aplikasi dibangun secara lintas platform.
3. Berdasarkan hasil survei yang dilakukan menunjukkan bahwa aplikasi ini telah memenuhi kebutuhan pengguna dalam menembangkan kebiasaan yang ingin mereka bentuk.
4. Hasil survei juga menunjukkan bahwa aplikasi ini telah memberikan dampak positif dalam pemanfaatan teknologi dalam.

### 5.2 Saran

terdapat beberapa saran terhadap hasil dari penelitian ini penelitian, diantaranya adalah:

1. Lebih bisa meningkatkan kualitas tampilan aplikasi yang masih belum maksimal dimana masih terlihat membosankan dan kurang menarik bagi sebagian pengguna.
2. Menambahkan aplikasi berbeda untuk akses admin dalam mengelola aplikasi.
3. Fitur yang masih bisa dimaksimalkan lagi, terutama pada analisis *habit* dan pengukuran *habit* bertipe time. Membuat komponen diagram dan input

waktu yang lebih baik lagi mungkin akan memaksimalkan penggunaan aplikasi ini.

4. Konten artikel yang masih tergolong sangat sedikit mungkin dapat diperbanyak lagi.
5. Untuk fitur notifikasi pada aplikasi ini juga mungkin harus dirancang ulang agar pengguna lebih maksimal dalam menggunakan aplikasi habit tracker.

Semoga dengan adanya penelitian ini dapat menjadi bahan acuan dan sumber informasi juga bahan pembelajaran bagi banyak orang.

# Rangga Dwi Rendra

---

## ORIGINALITY REPORT

---

22%

SIMILARITY INDEX

21%

INTERNET SOURCES

13%

PUBLICATIONS

15%

STUDENT PAPERS

---

## PRIMARY SOURCES

---

1	123dok.com Internet Source	2%
2	media.neliti.com Internet Source	2%
3	text-id.123dok.com Internet Source	1%
4	jurnal.unprimdn.ac.id Internet Source	1%
5	docplayer.info Internet Source	1%
6	repository.uisu.ac.id Internet Source	1%
7	misliapoltesa.blogspot.com Internet Source	1%
8	Submitted to stidalhadid Student Paper	1%
9	id.scribd.com Internet Source	1%

---



10	<a href="http://repository.unsub.ac.id">repository.unsub.ac.id</a> Internet Source	1 %
11	<a href="http://repository.bsi.ac.id">repository.bsi.ac.id</a> Internet Source	1 %
12	<a href="http://elib.pnc.ac.id">elib.pnc.ac.id</a> Internet Source	1 %
13	<a href="http://jurnal.usbypkp.ac.id">jurnal.usbypkp.ac.id</a> Internet Source	1 %
14	<a href="http://begawe.unram.ac.id">begawe.unram.ac.id</a> Internet Source	1 %
15	<a href="http://ejournal.raharja.ac.id">ejournal.raharja.ac.id</a> Internet Source	1 %
16	<a href="http://id.123dok.com">id.123dok.com</a> Internet Source	<1 %
17	Submitted to Universitas Gadjah Mada Student Paper	<1 %
18	<a href="http://repository.untag-sby.ac.id">repository.untag-sby.ac.id</a> Internet Source	<1 %
19	<a href="http://widuri.raharja.info">widuri.raharja.info</a> Internet Source	<1 %
20	<a href="http://adoc.pub">adoc.pub</a> Internet Source	<1 %
21	<a href="http://jurnal.atmaluhur.ac.id">jurnal.atmaluhur.ac.id</a> Internet Source	<1 %

22	<a href="https://repository.its.ac.id">repository.its.ac.id</a> Internet Source	<1 %
23	<a href="https://www.slideshare.net">www.slideshare.net</a> Internet Source	<1 %
24	<a href="https://digilib.iain-palangkaraya.ac.id">digilib.iain-palangkaraya.ac.id</a> Internet Source	<1 %
25	Submitted to Singapore American High School Student Paper	<1 %
26	Submitted to Universitas Bina Darma Student Paper	<1 %
27	Submitted to Universitas Brawijaya Student Paper	<1 %
28	Submitted to Universitas Pamulang Student Paper	<1 %
29	Zulfian Syahril Shiddiq, Mochamad Alfian Rosid, Arif Senja Fitriani, Hindarto Hindarto. "Information System For New Residents Registration Based On Web and Android", Procedia of Engineering and Life Science, 2023 Publication	<1 %
30	<a href="https://repo.darmajaya.ac.id">repo.darmajaya.ac.id</a> Internet Source	<1 %
31	<a href="https://repository.uniks.ac.id">repository.uniks.ac.id</a> Internet Source	<1 %

<1 %

32

[repository.syekhnurjati.ac.id](https://repository.syekhnurjati.ac.id)

Internet Source

<1 %

33

[jurnal.pancabudi.ac.id](https://jurnal.pancabudi.ac.id)

Internet Source

<1 %

34

[repository.ittelkom-pwt.ac.id](https://repository.ittelkom-pwt.ac.id)

Internet Source

<1 %

35

Romi Nur asfi Akbar, Fahmi Indiarto, Arfani Aristiantoro, Yudo Utomo. "Aplikasi Online Berbasis Android "SI TekO" (Sistem Informasi Teknisi Online) Sebagai Solusi Mempermudah Masyarakat Dalam Mendapatkan Jasa Service", Generation Journal, 2021

Publication

<1 %

36

[journals.usm.ac.id](https://journals.usm.ac.id)

Internet Source

<1 %

37

[sinta.unud.ac.id](https://sinta.unud.ac.id)

Internet Source

<1 %

38

[eprintslib.ummgl.ac.id](https://eprintslib.ummgl.ac.id)

Internet Source

<1 %

39

[informatika.uin-suka.ac.id](https://informatika.uin-suka.ac.id)

Internet Source

<1 %

40

[kleido.com](https://kleido.com)

Internet Source

<1 %

41 lib.unnes.ac.id <1 %  
Internet Source

---

42 repository.polman-babel.ac.id <1 %  
Internet Source

---

43 repository.uinjkt.ac.id <1 %  
Internet Source

---

44 smart.stmikplk.ac.id <1 %  
Internet Source

---

Exclude quotes On

Exclude matches Off

Exclude bibliography On