

## BAB 3 METODOLOGI PENELITIAN

*Waterfall* merupakan metodologi yang digunakan dalam mengembangkan sistem rekomendasi acara pada website pencarian acara ini. Model *waterfall* merupakan salah satu pendekatan tradisional yang digunakan dalam pengembangan perangkat lunak secara berurutan atau sekuensial [19]. Gambar 3.1 merupakan tahapan yang digunakan dalam model *waterfall*.



Gambar 3.1. Waterfall methodology

Sesuai dengan tahapan dari metodologi *waterfall*, maka metodologi penelitian akan mencakup tahapan sebagai berikut:

### 3.1 Analisa Kebutuhan

Tahap ini dilakukan untuk menganalisis kebutuhan pengguna melalui survey mengenai kebutuhan dan pentingnya implementasi sistem rekomendasi dalam website pencarian acara. Pada tahap ini, telah ditemukan bahwa penerapan sistem rekomendasi untuk pencarian acara sangat dibutuhkan pengguna karena dapat memudahkan pengguna dalam mencari suatu acara yang sesuai dengan preferensi mereka di tengah banyaknya informasi acara. Oleh karena itu, pada penelitian ini akan dibangun sistem rekomendasi acara pada website pencarian acara yang dapat

memberikan rekomendasi sesuai dengan preferensi pengguna berdasarkan interaksi yang telah dilakukan pengguna terhadap item acara sebelumnya.

### **3.2 Studi Literatur dan Observasi**

Tahap studi literatur dilakukan untuk memperoleh pemahaman yang mendalam tentang konsep, metode, dan teknologi yang terkait dengan pengembangan sistem rekomendasi. Pada tahap ini, peneliti akan mencari dan meninjau berbagai sumber literatur seperti jurnal ilmiah, buku, artikel, dan riset terkait untuk memahami pendekatan serta algoritma yang sesuai dalam membangun sistem rekomendasi acara. Selain studi literatur, observasi terhadap platform penyedia acara lain juga dilakukan untuk perbandingan sebagai dasar untuk pembangunan situs web pencarian acara beserta dengan sistem rekomendasinya.

### **3.3 Pembangunan Dataset Acara untuk Rekomendasi**

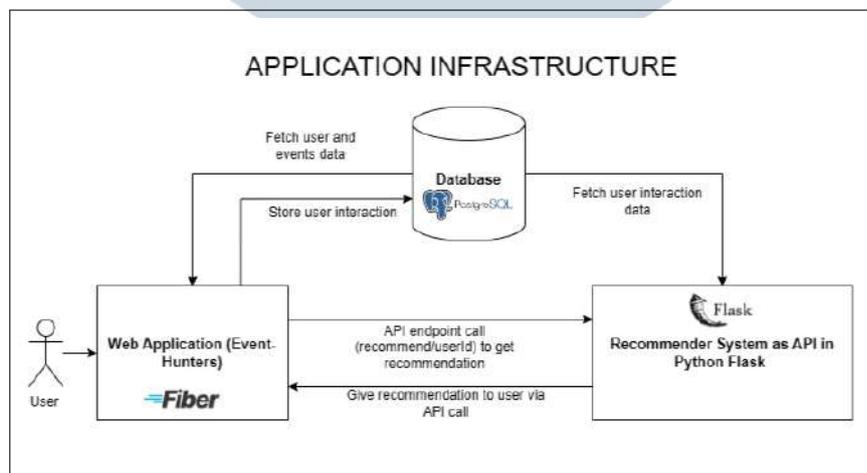
Data acara yang akan digunakan oleh aplikasi dan sistem rekomendasi akan dikumpulkan dan dibuat secara manual dengan menyalin informasi dari platform pencarian acara lain seperti Goers, Instagram, dan EventKampus. Fitur seperti tanggal pendaftaran dan tanggal mulai acara pada data acara akan diubah untuk keperluan simulasi rekomendasi. Hal ini dilakukan karena sebagian besar data acara yang tersedia sudah kadaluarsa dan terbatasnya ketersediaan data acara baru.

### **3.4 Perancangan Aplikasi**

Setelah pembangunan dataset acara selesai, tahap selanjutnya adalah perancangan aplikasi. Tahap ini mencakup perancangan infrastruktur aplikasi secara menyeluruh, termasuk rancangan *website* pencarian acara dan sistem rekomendasi yang diterapkan pada *website* pencarian acara tersebut. *Website* pencarian acara dengan nama Event Hunters yang dibangun dengan menggunakan bahasa pemrograman Golang dengan *framework fiber*. Golang dikembangkan oleh Google bersama Ken Thompson, Robert Griesemer, dan Rob Pike pada tahun 2009, merupakan salah satu bahasa pemrograman yang mengutamakan kecepatan, keandalan, skalabilitas, dan kesederhanaan. Golang didasarkan pada bahasa C dan ideal untuk mengembangkan aplikasi, *website*, dan perangkat lunak lainnya [20]. *Golang Fiber* merupakan salah satu *framework website* yang ringan dan cepat untuk bahasa pemrograman Go. *Framework* ini menyediakan fitur *routing*,

*middleware*, dan validasi yang kuat untuk mengembangkan aplikasi *website* yang responsif. *Golang Fiber* memungkinkan pengembang untuk membuat API dan aplikasi *website* berkinerja tinggi dengan cepat [21].

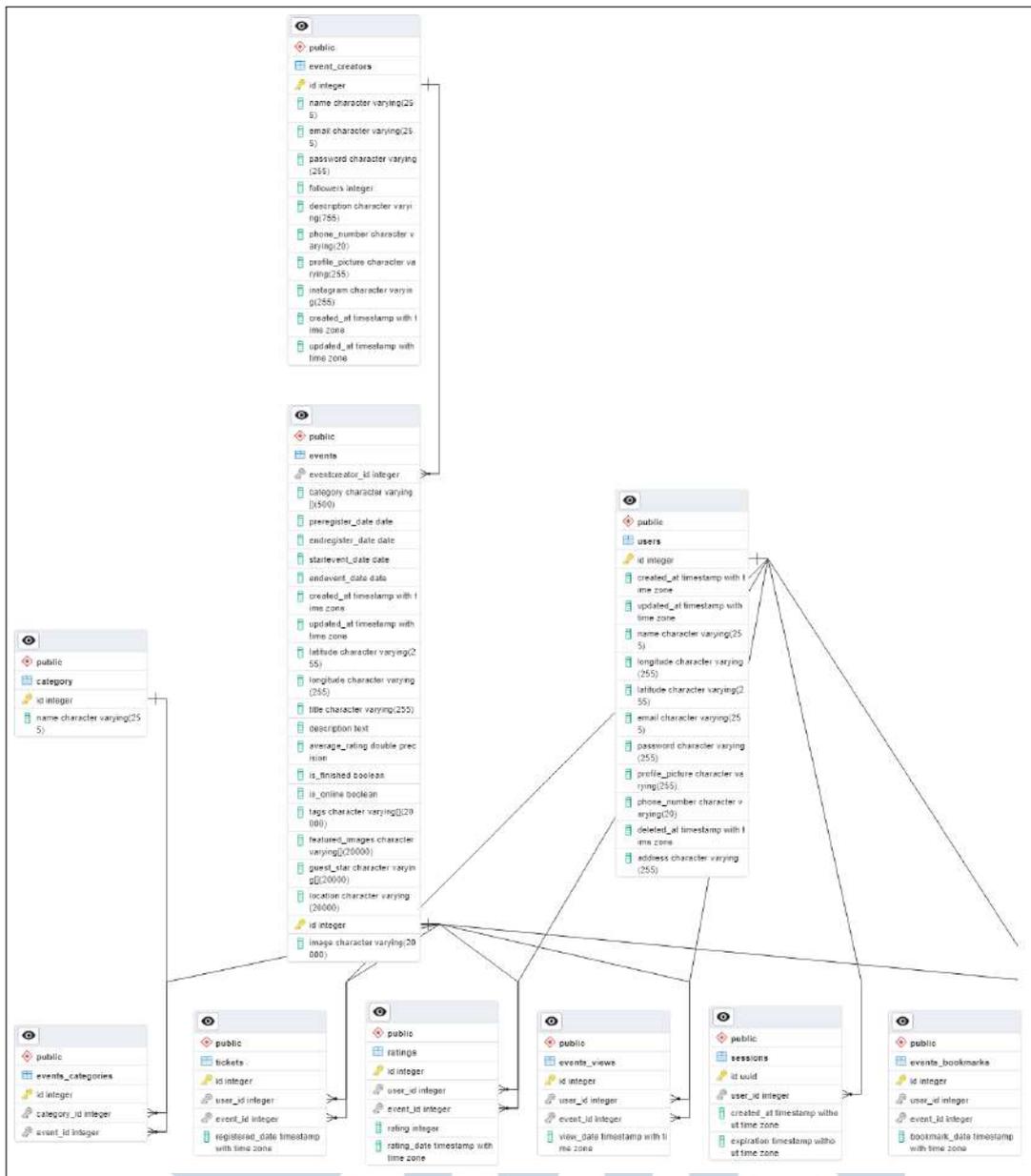
Sistem rekomendasi acara diintegrasikan ke website pencarian acara dalam bentuk *machine learning* yang dibangun menggunakan *Flask python* sebagai API. Python adalah bahasa pemrograman tingkat tinggi yang diinterpretasikan dan dikenal luas dalam penggunaan komputasi ilmiah, ilmu data, dan pembelajaran mesin dengan berbagai macam *library* yang sangat membantu [22]. Flask merupakan salah satu *micro-framework* dari *python* yang dirancang untuk membuat aplikasi *website* dalam waktu singkat. Flask hanya mengimplementasikan fungsi inti, memberikan fleksibilitas kepada pengembang untuk menambahkan fitur sesuai kebutuhan selama proses implementasi [23]. Database aplikasi *website* dan sistem rekomendasi menggunakan PostgreSQL. PostgreSQL merupakan salah satu database relasional *open-source* yang populer dan sering digunakan baik untuk aplikasi berskala menengah maupun besar [24]. Gambar 3.2 menggambarkan infrastruktur aplikasi secara keseluruhan



Gambar 3.2. Infrastruktur Aplikasi

### 3.4.1 Perancangan Skema Database Aplikasi

*Website* pencarian acara dan sistem rekomendasi yang diterapkan akan menggunakan *relational database* PostgreSQL. Skema *database* adalah deskripsi formal tentang struktur fisik dari basis data yang mencakup tabel, kolom, tipe data, constraints, indeks, dan hubungan antar tabel [25]. Gambar 3.3 adalah skema dari *database* yang dibangun:



Gambar 3.3. Event-Hunters Database Schema

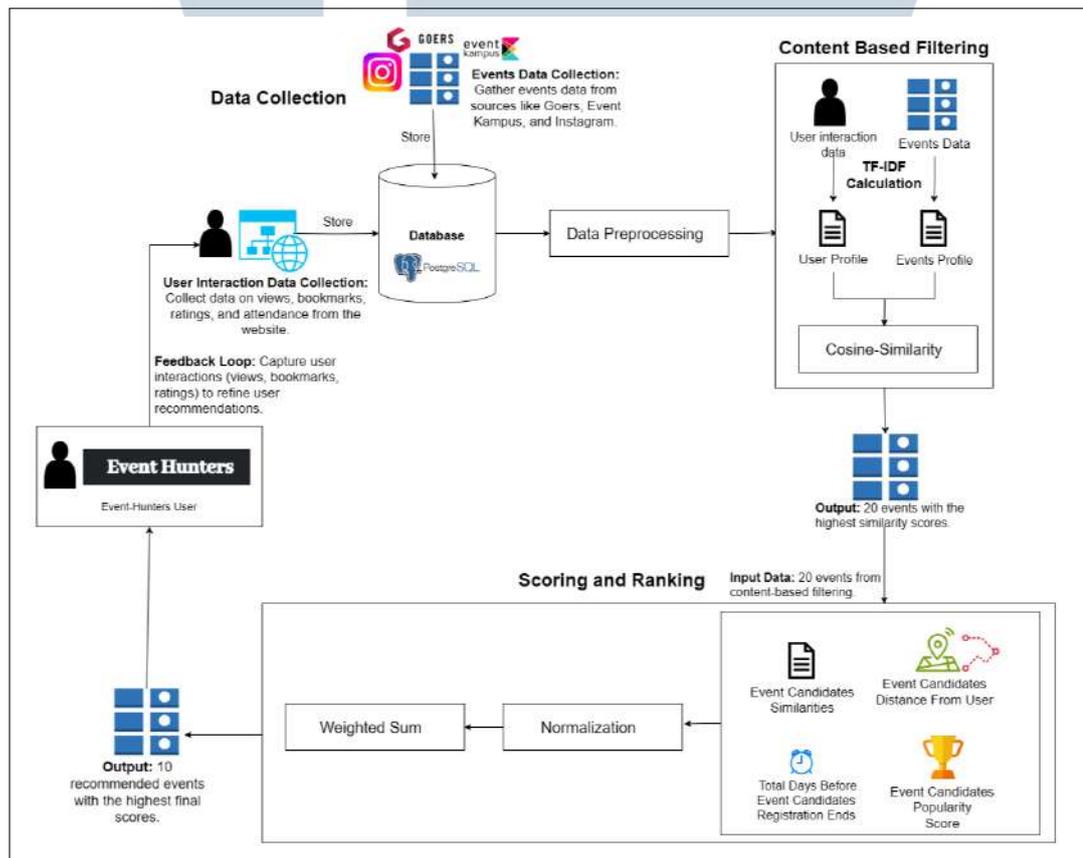
Terdapat total 10 tabel yang akan digunakan untuk aplikasi ini. Berikut adalah penjelasan untuk masing-masing tabel yang terpapar pada skema database:

Tabel 3.1. Daftar penjelasan masing-masing tabel *database*

Nama Tabel	Deskripsi
events	Tabel yang digunakan untuk menyimpan seluruh data acara pada aplikasi
users	Tabel yang digunakan untuk menyimpan seluruh data pengguna pada aplikasi
event creators	Tabel yang digunakan untuk menyimpan data pembuat acara pada aplikasi.
category	Tabel yang digunakan untuk menyimpan daftar kategori yang tersedia untuk acara.
events categories	Tabel yang digunakan untuk menyimpan relasi <i>many-to-many</i> antara acara dengan kategori
tickets	Tabel interaksi pengguna yang digunakan untuk menyimpan data kehadiran pengguna terhadap suatu acara, relasi ini bersifat <i>many-to-many</i> dan merupakan data yang esensial untuk kepentingan rekomendasi
events views	Tabel interaksi pengguna yang digunakan untuk menyimpan data views pengguna terhadap suatu acara, relasi ini bersifat <i>many-to-many</i> dan merupakan data yang esensial untuk kepentingan rekomendasi.
events bookmarks	Tabel interaksi pengguna yang digunakan untuk menyimpan data bookmark pengguna terhadap suatu acara, relasi ini bersifat <i>many-to-many</i> dan merupakan data yang esensial untuk kepentingan rekomendasi.
ratings	Tabel interaksi pengguna yang digunakan untuk menyimpan data rating dari pengguna terhadap suatu acara, relasi ini bersifat <i>many-to-many</i> dan merupakan data yang esensial untuk kepentingan rekomendasi.
sessions	Tabel ini menyimpan <i>session</i> untuk pengguna yang telah <i>login</i> , <i>session</i> akan dihancurkan setelah pengguna melakukan <i>logout</i>

### 3.4.2 Perancangan sistem rekomendasi acara

Sistem rekomendasi acara pada penelitian ini akan dibangun dengan salah satu *framework web Python* yaitu *flask* sebagai API (*Application Programming Interface*) yang dapat diakses melalui aplikasi *website* pencarian acara untuk mendapatkan daftar rekomendasi. API ini hanya memiliki 1 *endpoint* dan dibangun dengan pendekatan *pipelining hybridization*. *Pipelining hybridization* merupakan salah satu desain dari pendekatan rekomendasi *hybrid* dimana sistem rekomendasi awal akan memproses masukan untuk sistem rekomendasi setelahnya menyerupai sebuah pipa. Desain ini tentunya digunakan dengan tujuan meningkatkan kualitas rekomendasi yang diberikan [26]. Gambar 3.4 merupakan *framework* dari sistem rekomendasi yang digunakan pada penelitian ini.



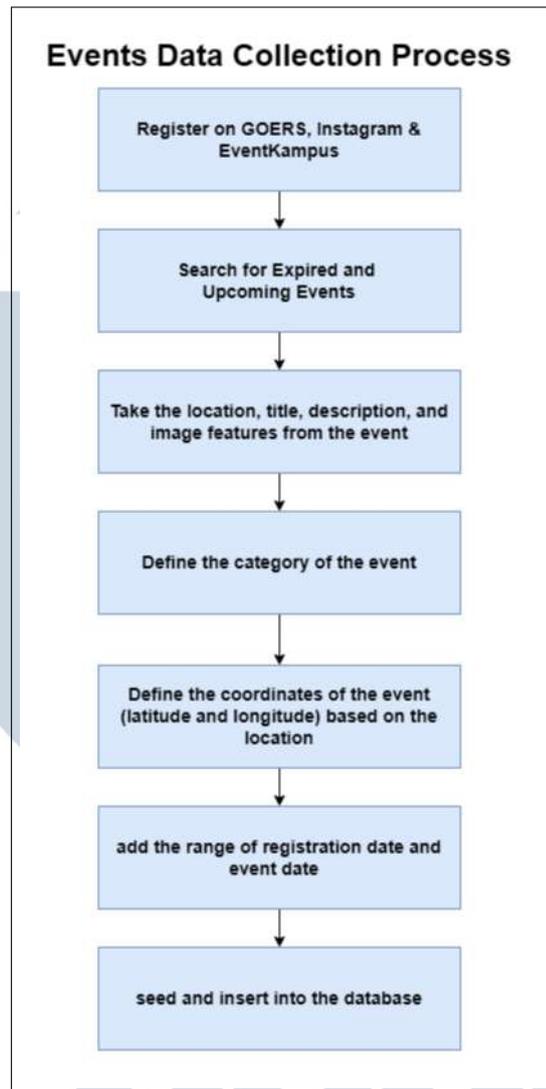
Gambar 3.4. *Framework* Sistem Rekomendasi yang dibangun

Berikut adalah penjelasan untuk masing-masing komponen yang terdapat pada infrastruktur sistem rekomendasi yang dirancang:

#### 1. Data Collection (Pengumpulan Data)

- **Events Data Collection:** Proses pengumpulan data acara dimulai dengan mendaftar atau registrasi pada platform acara seperti GOERS, Instagram, dan EventKampus untuk mengakses serta mendapatkan informasi mengenai acara. Setelah itu, melalui platform tersebut, dilakukan pencarian acara yang telah berakhir atau yang akan datang. Acara yang telah ditemukan akan diambil fitur pentingnya seperti lokasi, judul, deskripsi, dan gambar acara. Setiap acara kemudian dikategorikan, dan koordinat geografisnya ditentukan berdasarkan deskripsi lokasi. Informasi tentang rentang tanggal registrasi dan tanggal acara juga ditambahkan sesuai dengan kepentingan rekomendasi. Semua data yang terkumpul kemudian akan dimasukkan dalam basis data dan siap untuk digunakan dalam sistem rekomendasi acara. Proses ini menghasilkan 110 data acara dengan kategori serta tipe yang beraneka ragam sehingga dapat mensimulasikan rekomendasi yang lebih personal dan memahami preferensi pengguna. Gambar 3.5 merupakan proses untuk pengumpulan data acara.

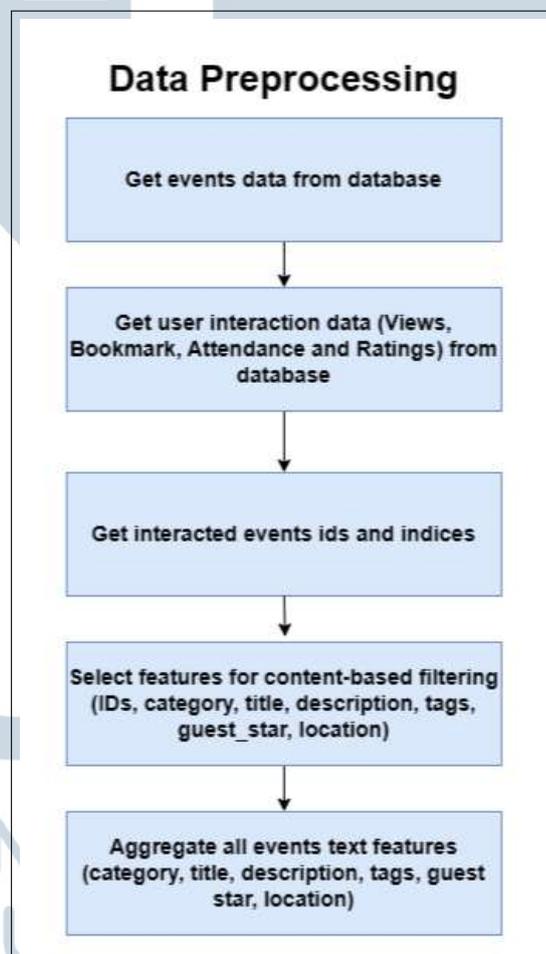




Gambar 3.5. Proses *data collection* untuk *events*

- **User Interaction Data Collection:** Mengumpulkan data interaksi pengguna dari website pencarian acara Event Hunters yang dibuat, data interaksi yang diambil antara lain: views, bookmarks, ratings, dan kehadiran pengguna terhadap acara. Data ini penting untuk memahami preferensi pengguna.
2. **Database:** Data yang dikumpulkan dari pengguna dan acara disimpan dalam *database PostgreSQL*. *Database* ini berfungsi sebagai penyimpanan sentral untuk semua informasi yang diperlukan oleh sistem rekomendasi.
  3. **Data Preprocessing (Pra-Pemrosesan Data):**

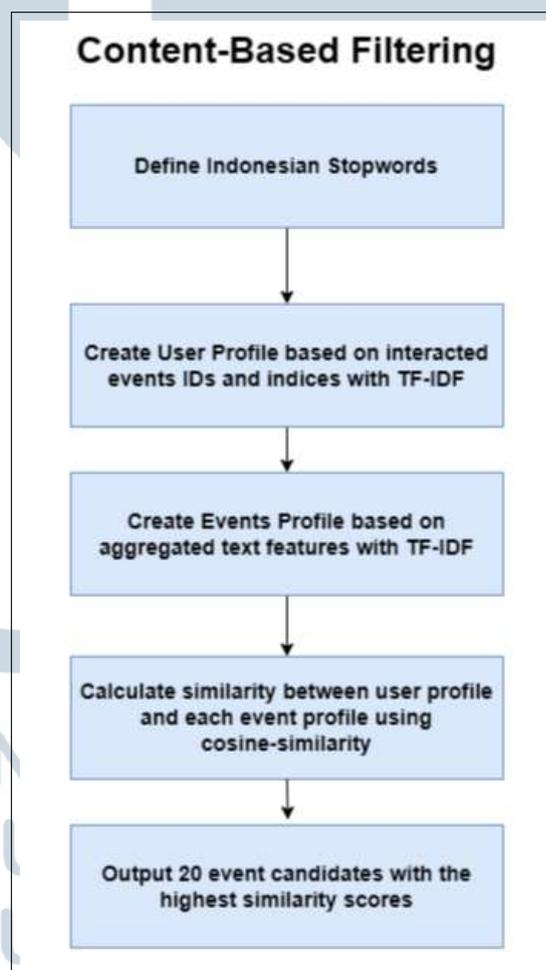
Tahap *preprocessing data* dimulai dengan pengambilan data interaksi pengguna dan data acara dari *database*. Data interaksi pengguna ini kemudian diidentifikasi berdasarkan ID dan indeks acara yang telah diinteraksi oleh pengguna tersebut. Selanjutnya, dilakukan seleksi fitur teks untuk metode *content-based filtering*, mencakup ID, kategori, judul, deskripsi, tag, bintang tamu, dan lokasi acara. Semua fitur teks dari acara kemudian diagregasikan untuk persiapan lebih lanjut dalam sistem rekomendasi, memastikan bahwa data yang digunakan seluruhnya berupa teks agar dapat dijadikan sebagai masukan untuk *content-based filtering*. Gambar 3.6 merupakan proses untuk *data preprocessing*.



Gambar 3.6. Proses *Data Preprocessing*

4. **Content-Based Filtering** Tahap selanjutnya adalah penerapan metode content-based filtering. Proses ini dimulai dengan mendefinisikan *stopwords* dalam Bahasa Indonesia untuk membersihkan teks dari kata-kata yang

tidak memiliki makna signifikan dalam analisis. Kemudian, dibuat profil pengguna berdasarkan ID acara dan indeks acara yang telah diinteraksi oleh pengguna menggunakan metode TF-IDF. Selanjutnya, dibuat profil acara berdasarkan agregasi fitur teks acara yang telah dilakukan pada *preprocessing* dengan metode yang sama, TF-IDF. Setelah itu, dilakukan perhitungan kesamaan antara profil pengguna dengan setiap profil acara menggunakan *cosine similarity*. Hasil dari perhitungan ini adalah 20 acara dengan skor kesamaan tertinggi yang kemudian akan diambil sebagai kandidat acara untuk dijadikan sebagai masukan untuk tahap rekomendasi selanjutnya. Proses ini memastikan bahwa rekomendasi yang dihasilkan sesuai dengan preferensi pengguna berdasarkan interaksi sebelumnya. Gambar 3.7 merupakan proses untuk *content-based filtering*

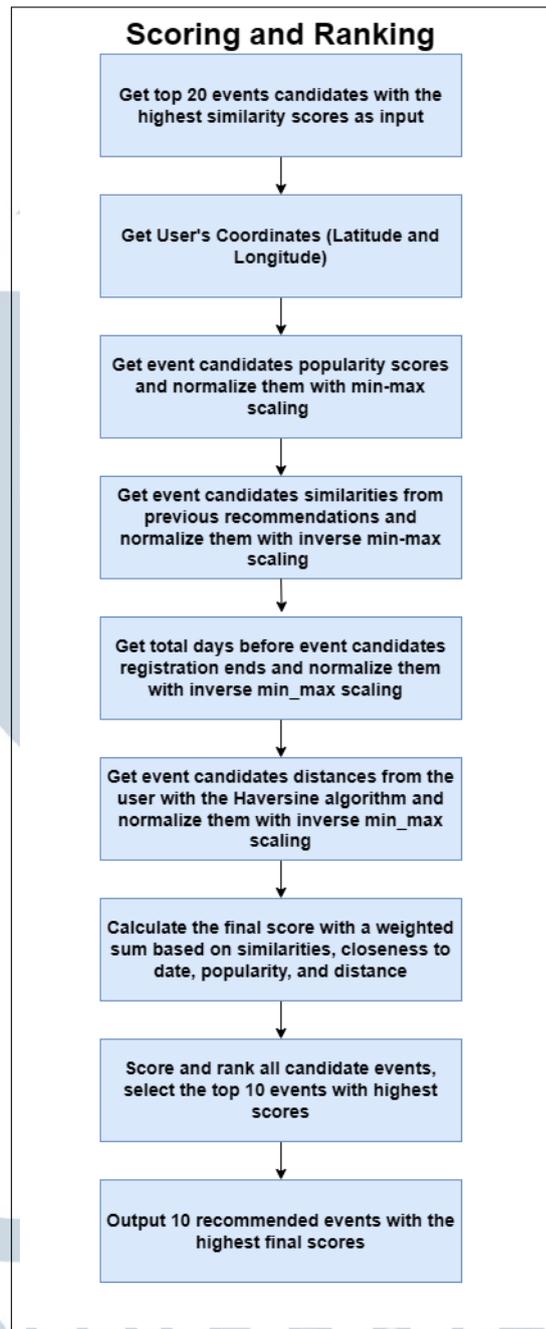


Gambar 3.7. Proses *Content-Based Filtering*

5. **Scoring and Ranking:** Tahap berikutnya adalah proses *scoring and*

*ranking* yang dimulai dengan mengambil 20 kandidat acara dengan skor kesamaan tertinggi dari tahap *content-based filtering*. Selanjutnya, diperoleh koordinat pengguna (*latitude* dan *longitude*) untuk digunakan dalam analisis lokasi. Skor popularitas setiap kandidat acara diambil dan dinormalisasi menggunakan skala *min-max*. Kesamaan kandidat acara dari rekomendasi sebelumnya juga dinormalisasi dengan metode yang sama. Kemudian, dihitung jumlah hari sebelum pendaftaran acara berakhir dan dinormalisasi menggunakan skala *min-max* invers karena semakin kecil jumlah hari yang tersisa maka nilainya semakin besar. Jarak antara pengguna dan lokasi acara dihitung menggunakan algoritma *Haversine* dan dinormalisasi juga dengan skala *min-max* invers. Gambar 3.8 merupakan proses untuk *scoring and ranking*.





Gambar 3.8. Proses *Scoring and Ranking*

Setelah empat kriteria tersebut dihitung dan dinormalisasi, skor akhir akan dihitung dengan metode *weighted sum* berdasarkan kesamaan, kedekatan tanggal, popularitas, dan jarak. Semua kandidat acara kemudian diberi skor dan diurutkan, dengan sepuluh acara teratas dengan skor tertinggi dipilih sebagai rekomendasi akhir. Hasil akhirnya adalah 10 acara yang

direkomendasikan kepada pengguna dengan skor tertinggi.

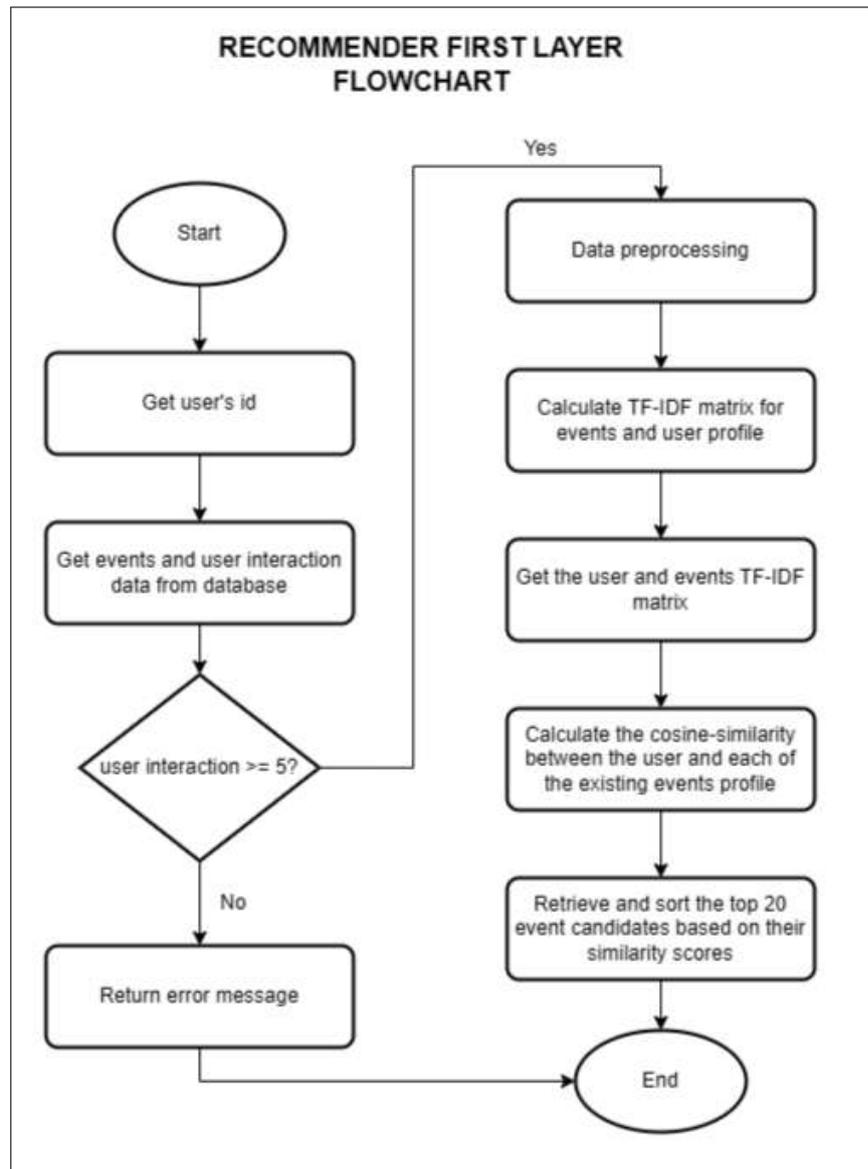
## 6. Output (Penyajian Rekomendasi)

- **10 Recommended Events:** Dari 20 kandidat acara, hanya 10 acara dengan skor tertinggi dari tahap *Scoring and Ranking* direkomendasikan kepada pengguna.
  - **Event-Hunters User:** Pengguna website akan menerima rekomendasi acara berdasarkan preferensi dan interaksi mereka yang telah dianalisis oleh sistem rekomendasi.
7. **Feedback Loop:** Menangkap interaksi pengguna yang berkelanjutan pada masa mendatang untuk menyempurnakan hasil rekomendasi selanjutnya. Sistem ini memungkinkan pembaruan profil pengguna secara dinamis berdasarkan perilaku terbaru.

### A Flowchart perhitungan sistem rekomendasi acara tahap pertama

Pada bagian ini akan dipaparkan alur cara kerja sistem rekomendasi acara pada tahap pertama. Gambar 3.9 merupakan *flowchart* untuk bagian pertama dari sistem rekomendasi dengan melibatkan metode *content-based filtering* dengan hasil 20 kandidat acara dengan nilai similaritas tertinggi.





Gambar 3.9. Diagram alur rekomendasi tahap pertama

Sistem rekomendasi tahap pertama ini akan menerima input id pengguna dari parameter URL API *endpoint* yaitu *"recommend/:userId"*. Id pengguna ini nantinya akan digunakan untuk mendapatkan interaksi yang sudah dilakukan oleh pengguna. Apabila interaksi dari pengguna yang bersangkutan masih kurang dari lima, maka API sistem rekomendasi akan mengembalikan pesan *error*, sedangkan apabila interaksi user sama dengan atau melebihi lima, proses perhitungan rekomendasi akan dilanjutkan. Pada sebagian besar kasus dunia nyata, *raw data* (data mentah) tidak dapat langsung digunakan untuk analisis dalam pemodelan pembelajaran mesin, data mentah cenderung memiliki fitur yang tidak diperlukan

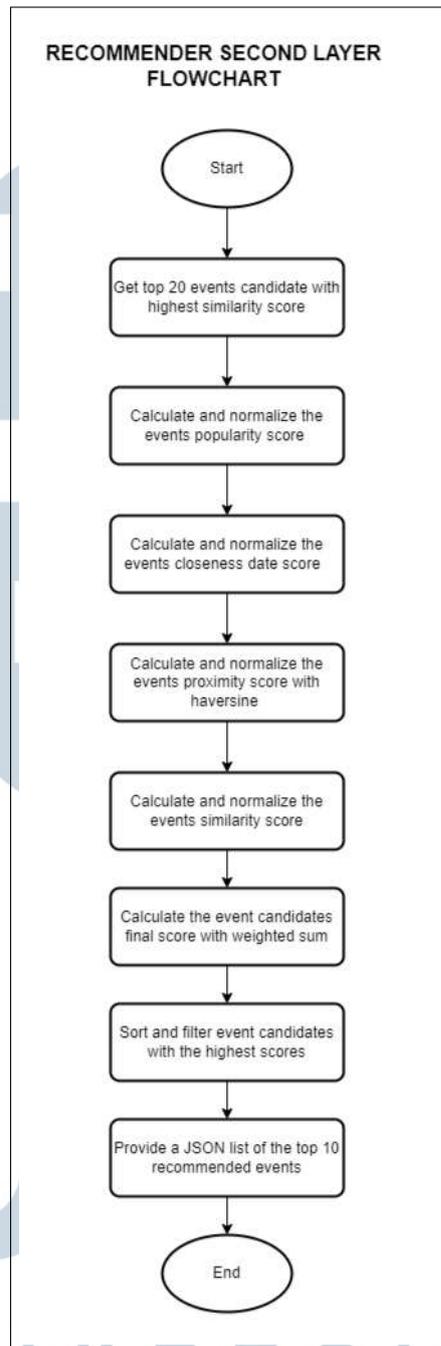
serta *noise* yang berpotensi menghambat kalkulasi di tahap analisis atau pemodelan [27]. Oleh karena itu, terdapat tahap prapemrosesan data yang melibatkan *feature engineering* dan *data cleaning* untuk mempersiapkan dan membersihkan data sebelum diproses lebih lanjut pada model yang telah disiapkan.

Hasil akhir dari prapemrosesan data ini adalah data teks acara yang telah diagregasi dari berbagai macam fitur, seperti tag, deskripsi, judul, kategori, dan bintang tamu dari acara. Dengan adanya data teks acara ini, kalkulasi TF-IDF dapat diterapkan untuk menentukan tingkat kepentingan dari masing-masing kata kunci yang terdapat pada suatu acara. Dengan TF-IDF, profil pengguna dan acara dapat dibangun dalam bentuk matriks. Setelah didapatkan profil pengguna dan seluruh acara, perhitungan *cosine similarity* dapat diterapkan untuk mendapatkan nilai similaritas antara masing-masing acara dengan pengguna. Semakin tinggi nilai similaritas, maka semakin relevan konten acara tersebut dengan preferensi pengguna. Hasil akhir dari rekomendasi tahap pertama ini adalah 20 kandidat acara dengan nilai *cosine similarity* tertinggi. Kandidat-kandidat ini akan digunakan sebagai data masukan untuk tahap rekomendasi selanjutnya.

## **B Flowchart perhitungan sistem rekomendasi acara tahap kedua**

Pada bagian ini akan dipaparkan alur cara kerja sistem rekomendasi acara tahap kedua. Gambar 3.10 merupakan *flowchart* untuk bagian kedua dari sistem rekomendasi dengan melibatkan metode *weighted sum* untuk menghasilkan 10 acara dengan skor tertinggi yang akan direkomendasikan ke pengguna.

U M N  
U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A



Gambar 3.10. Diagram alur rekomendasi tahap kedua

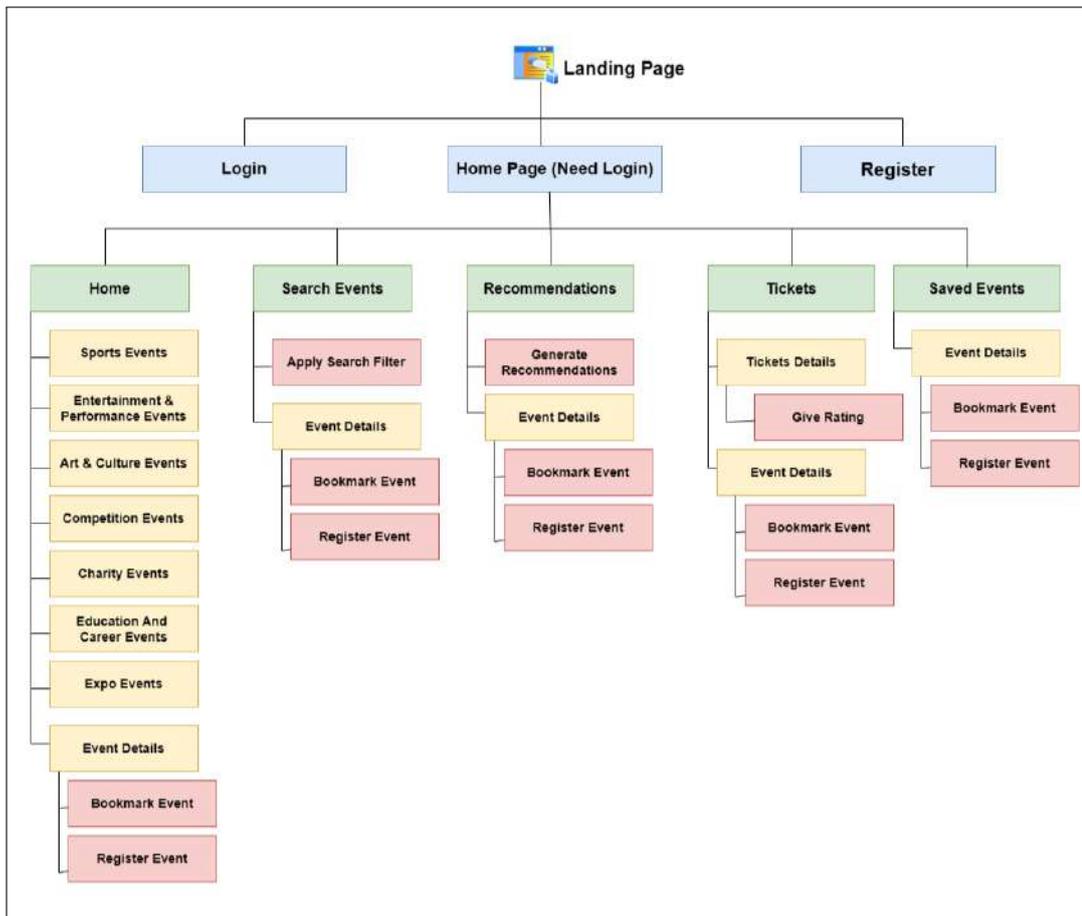
Sistem rekomendasi tahap kedua ini akan menerima *input* 20 kandidat acara dengan skor similaritas tertinggi dari hasil rekomendasi tahap pertama. Setiap kandidat acara akan dinilai popularitasnya berdasarkan jumlah interaksi oleh seluruh pengguna terhadap acara tersebut. Selanjutnya, akan dihitung nilai kedekatan waktu berakhirnya registrasi acara dengan waktu saat ini. Semakin

sedikit hari yang tersisa sebelum tanggal registrasi berakhir, semakin tinggi skornya, dan sebaliknya. Selain itu, nilai kedekatan jarak antara acara dan lokasi pengguna akan dihitung menggunakan algoritma *haversine* berdasarkan *latitude* dan *longitude* masing-masing acara dan pengguna. Selain ketiga kriteria tersebut, nilai similaritas yang diperoleh dari tahap rekomendasi sebelumnya juga akan dimasukkan dalam perhitungan untuk memberikan kontribusi pada nilai skor akhir.

Skor dari keempat kriteria tersebut akan dinormalisasi agar nilainya tetap konsisten dan berada dalam rentang yang sama. Setelah dinormalisasi, langkah selanjutnya adalah menerapkan metode *weighted sum* dengan bobot yang telah disesuaikan untuk setiap kriteria. Berdasarkan hasil survei, mayoritas pengguna menempatkan kedekatan lokasi acara sebagai prioritas utama, diikuti oleh konten acara, kedekatan tanggal berakhirnya registrasi acara, dan popularitas acara. Oleh karena itu, bobot yang digunakan dalam perhitungan *weighted sum* akan disesuaikan dengan tingkat prioritas yang diberikan oleh pengguna. Dengan menggunakan perhitungan *weighted sum*, akan dihasilkan nilai atau skor akhir untuk setiap kandidat acara. Setelah itu, 20 kandidat acara akan diurutkan kembali berdasarkan skor terbesar dan diseleksi kembali sehingga hanya terdapat 10 acara untuk direkomendasikan kepada pengguna yang dikirimkan dalam bentuk *JSON response* ke *website* pencarian acara.

### 3.4.3 Perancangan website pencarian acara

Aplikasi website akan dikembangkan sebagai *monolithic application* menggunakan *framework Golang* yang disebut *Fiber*. *Frontend* dan *backend* pada website ini tidak dipisahkan, melainkan disatukan menjadi satu kesatuan. Perancangan website ini memiliki tujuan untuk menjelaskan alur sistem website pencarian acara yang bernama Event Hunters secara menyeluruh, dimulai dari perancangan *flowchart*, *Data Flow Diagram (DFD)* dan *Wireframe*. Gambar 3.11 adalah perancangan *sitemap* untuk *website* Event Hunters:

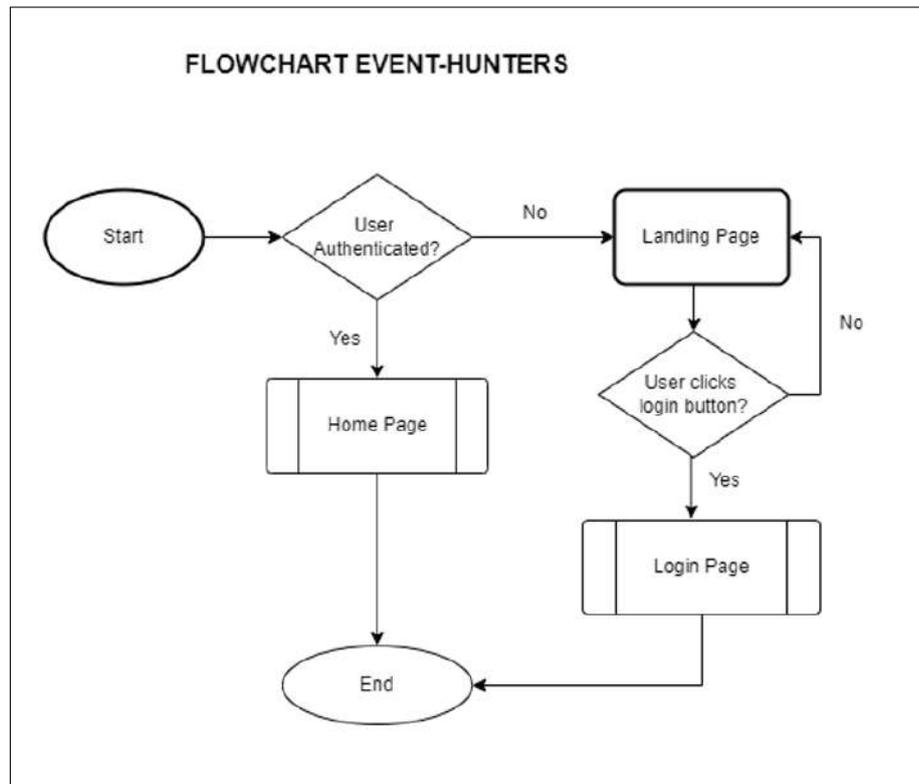


Gambar 3.11. Sitemap website Event Hunters

## A Flowchart

Desain website menggunakan flowchart bertujuan untuk memberikan gambaran yang jelas mengenai bagaimana website terkait akan beroperasi. Gambar 3.12 menampilkan flowchart utama untuk website Event Hunters:

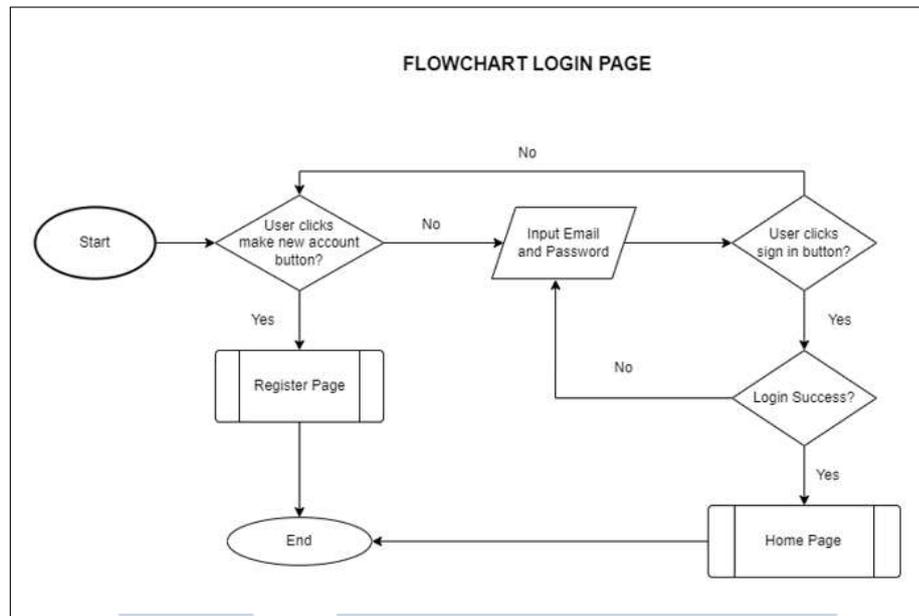
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 3.12. Flowchart website Event Hunters

Ketika mengakses website Event Hunters, sistem akan melakukan pengecekan terhadap sesi pengguna yang sedang aktif. Jika sesi pengguna sudah aktif, maka pengguna akan dialihkan ke halaman utama (home). Sebaliknya, jika tidak ada sesi yang aktif, pengguna akan diarahkan ke halaman awal (*landing page*) yang dilengkapi dengan tombol untuk melakukan *login*. Saat tombol *login* ditekan, pengguna akan diarahkan ke halaman *login*. Gambar 3.13 menampilkan *flowchart* untuk halaman login pada website Event Hunters:

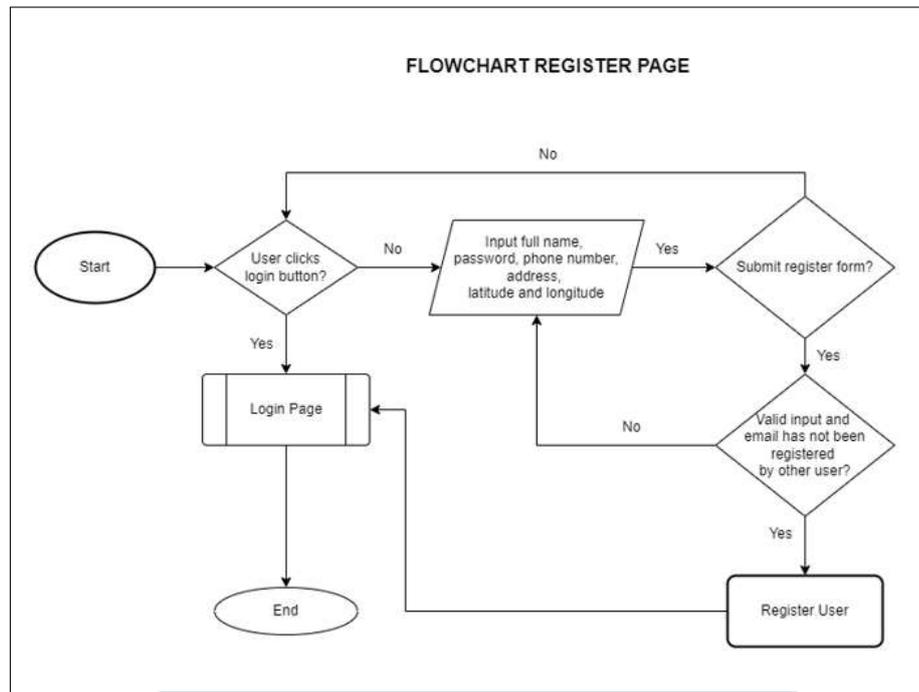
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 3.13. Flowchart *Login Page*

Pada halaman *login*, pengguna dapat memasukkan *email* dan kata sandi yang telah terdaftar, kemudian sistem akan memvalidasi data tersebut dengan informasi yang tersimpan di *database*. Jika proses validasi berhasil dan tidak ada kesalahan, pengguna akan berhasil masuk dan langsung diarahkan ke halaman utama (*home*). Namun, jika pengguna belum memiliki akun, mereka dapat mengklik tombol untuk membuat akun baru, yang akan mengarahkan mereka ke halaman pendaftaran (*register*). Gambar 3.14 menampilkan *flowchart* untuk halaman *register* pada *website* Event Hunters:

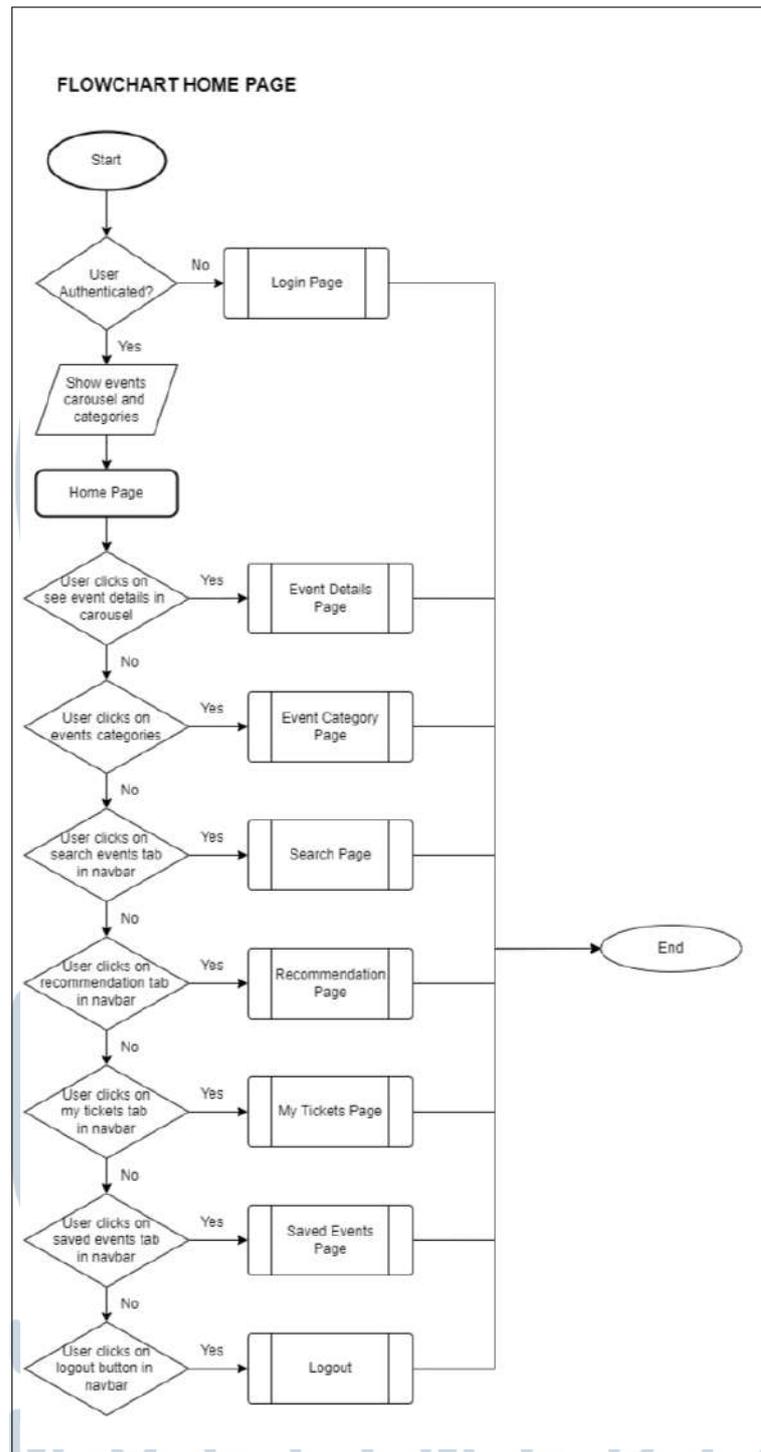
U M N  
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 3.14. Flowchart *Register Page*

Pada halaman *register*, pengguna dapat mengisi nama lengkap, kata sandi, nomor telepon, alamat, serta koordinat *latitude* dan *longitude* untuk membuat akun. Setelah semua kolom terisi, data tersebut akan diproses dan diverifikasi terlebih dahulu. Verifikasi dilakukan untuk memastikan bahwa semua informasi yang dimasukkan oleh pengguna benar dan valid, terutama *email*, *latitude*, dan *longitude*. Verifikasi email bertujuan memastikan bahwa tidak ada pengguna yang menggunakan alamat *email* yang sudah terdaftar dalam *database*. Setelah verifikasi berhasil, data pengguna akan didaftarkan dalam *database* dan pengguna akan langsung diarahkan ke halaman *login*. Jika pengguna sudah memiliki akun yang terdaftar, mereka dapat langsung mengklik tombol *login* untuk diarahkan ke halaman *login*. Gambar 3.15 merupakan *flowchart* untuk halaman *home* pada Event Hunters:

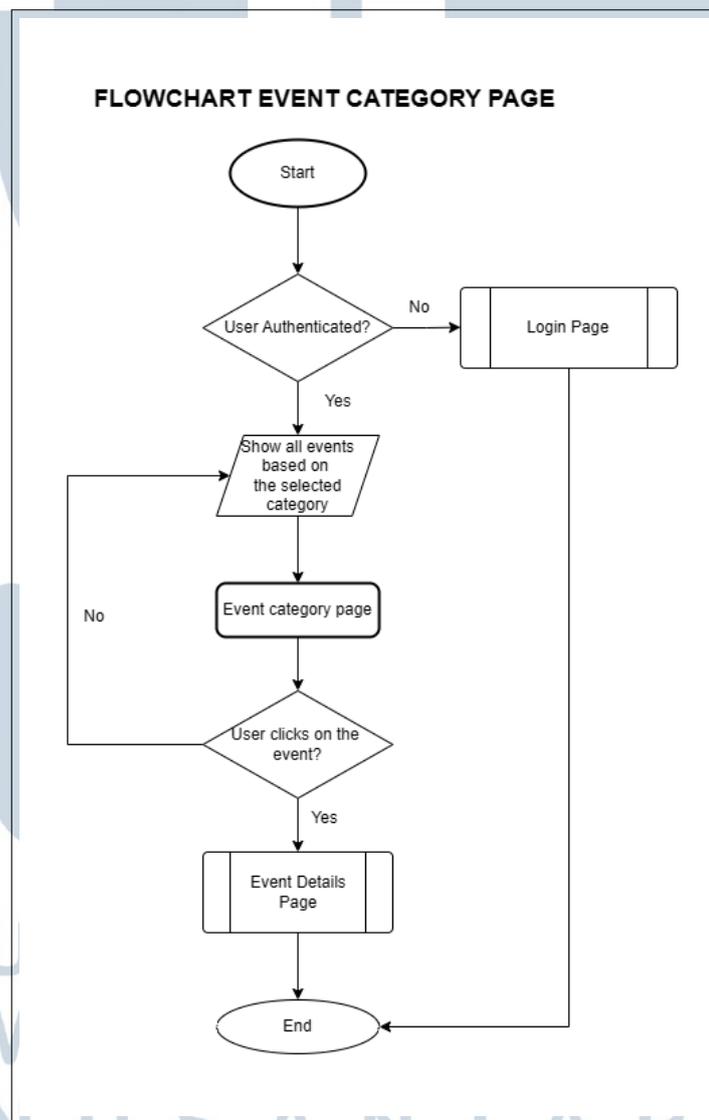
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 3.15. Flowchart *Home Page*

Pada halaman utama, terjadi pengecekan terhadap sesi pengguna dalam *database* dan *cookies* untuk memverifikasi otentikasi pengguna. Jika pengguna belum terotentikasi, mereka akan dialihkan ke halaman *login*. Setelah otentikasi

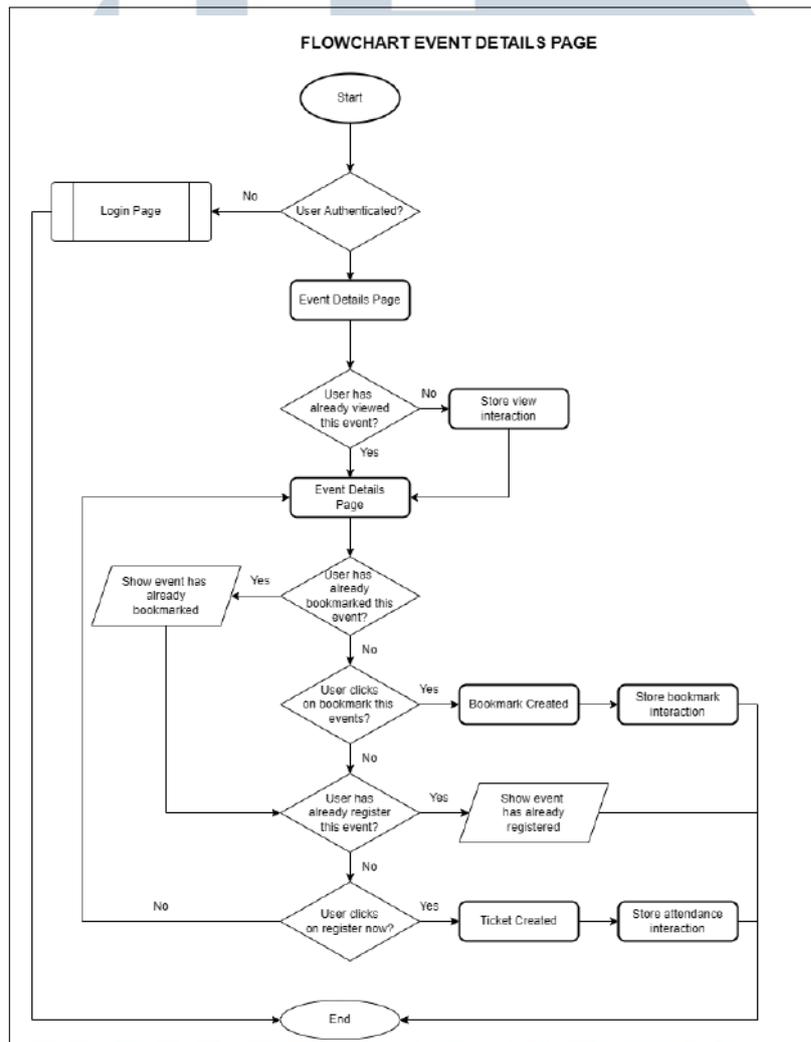
berhasil, pengguna akan diarahkan ke halaman utama yang menampilkan *carousel* acara dalam waktu terdekat serta berbagai kategori acara. Pengguna dapat mengklik salah satu *carousel* acara untuk melihat detail acara tersebut, atau mereka dapat memilih halaman kategori acara berdasarkan kategori yang diminati. Selain itu, pada halaman utama juga tersedia navigasi untuk mengakses halaman pencarian, rekomendasi, tiket pengguna, daftar acara yang disimpan, serta tombol *logout*. Gambar 3.16 merupakan *flowchart* untuk halaman kategori acara pada Event Hunters:



Gambar 3.16. Flowchart Category Page

Seperti halaman utama, pada halaman kategori acara juga dilakukan verifikasi otentikasi pengguna. Halaman ini menampilkan daftar acara sesuai

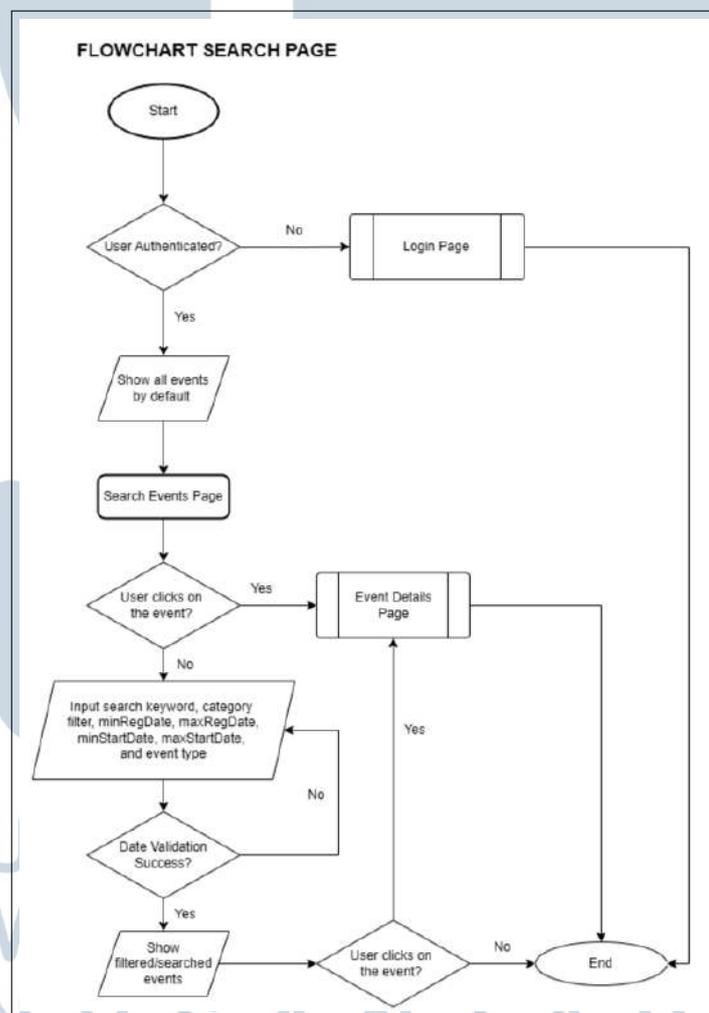
dengan kategori yang diakses oleh pengguna. Misalnya, jika pengguna membuka halaman kategori acara dengan tema "Olahraga", maka semua acara yang terkategori "Olahraga" akan ditampilkan di halaman tersebut. Pengguna juga dapat mengklik acara yang terpampang untuk menuju ke halaman detail acara. Gambar 3.17 merupakan *flowchart* untuk halaman detail acara pada Event Hunters:



Gambar 3.17. Flowchart *Event Details Page*

Pada halaman detail acara, akan dilakukan verifikasi otentikasi pengguna terlebih dahulu. Halaman ini akan menampilkan detail dari acara sesuai dengan ID acara yang dipilih oleh pengguna. Jika pengguna belum pernah melihat detail acara ini sebelumnya, maka interaksi *view* akan dicatat ke dalam *database*. Di halaman ini, pengguna dapat melihat informasi detail acara seperti judul, tanggal, gambar, lokasi, dan deskripsi acara tersebut. Jika pengguna belum melakukan

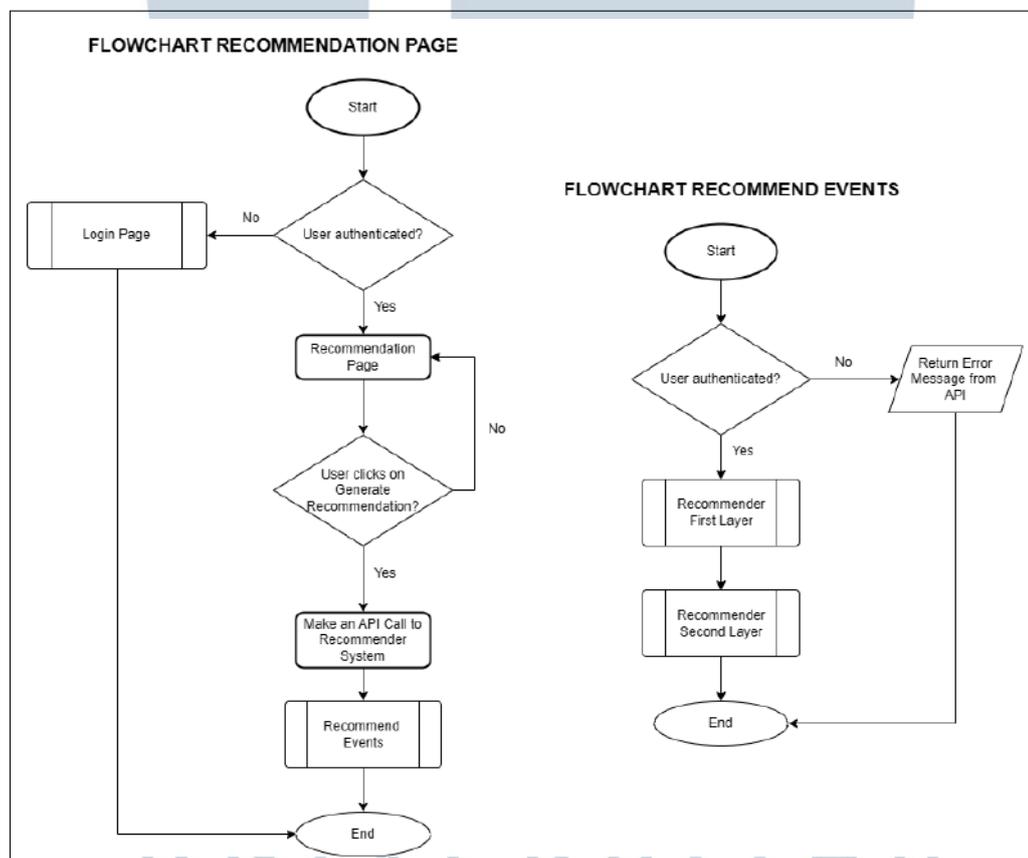
tindakan *bookmark* pada acara tersebut, maka dapat dilakukan dengan mengklik tombol *bookmark* yang akan menyimpan interaksi tersebut ke dalam *database*. Namun, jika pengguna sudah melakukan *bookmark*, maka pesan "event has already bookmarked" akan ditampilkan dan tombol *bookmark* akan dinonaktifkan. Sama halnya dengan proses registrasi acara, jika pengguna sudah melakukan registrasi, maka tombol registrasi akan dinonaktifkan dan pesan bahwa acara tersebut telah diregistrasi akan ditampilkan. Namun, jika belum, pengguna dapat mengklik tombol *register* yang akan mencatat pembuatan tiket dan menyimpan interaksi kehadiran pengguna ke dalam *database*. Gambar 3.18 merupakan *flowchart* untuk halaman pencarian pada Event Hunters:



Gambar 3.18. Flowchart Search Page

Pada halaman pencarian juga terdapat verifikasi otentikasi pengguna terlebih dahulu. Secara default, saat pengguna mengakses halaman pencarian, semua

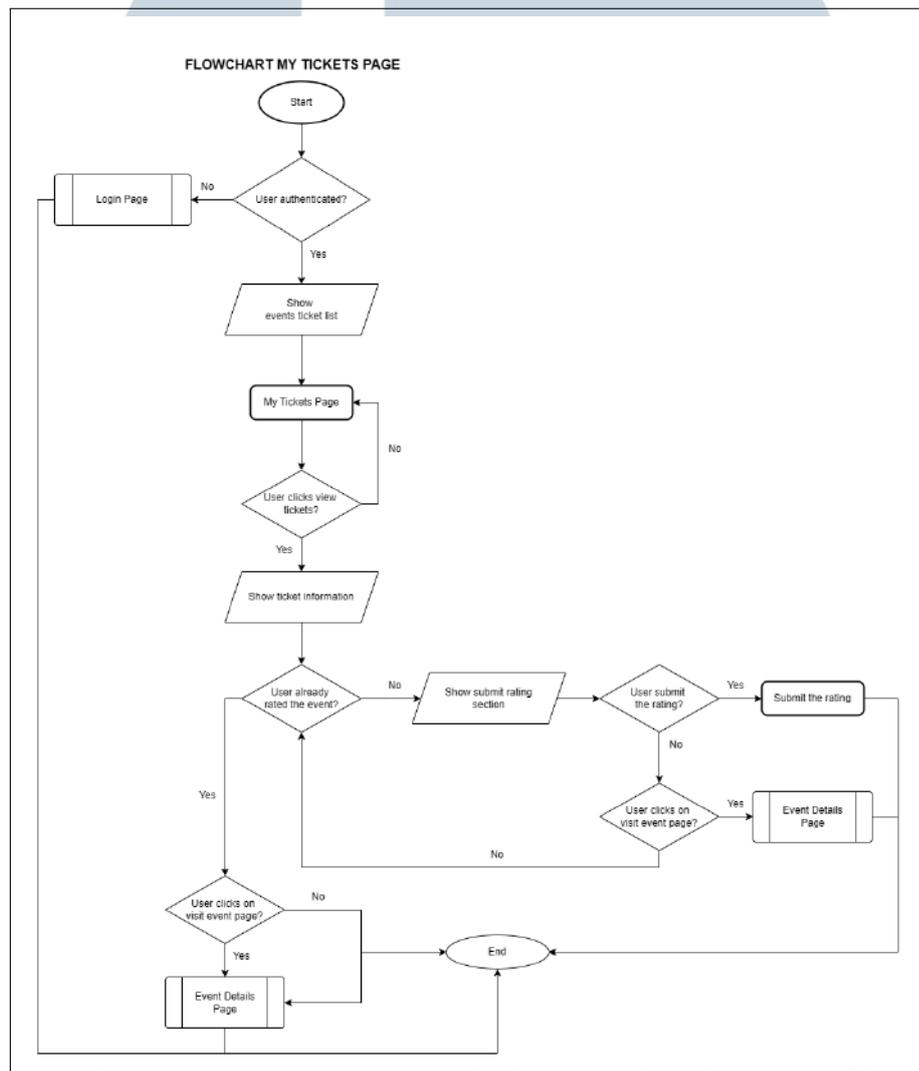
acara akan ditampilkan. Pengguna dapat mengklik acara tersebut untuk diarahkan ke halaman detail acara. Selain itu, pengguna juga dapat melakukan *filter* atau pencarian berdasarkan kata kunci, kategori acara, rentang tanggal registrasi acara (minimum dan maksimum), rentang tanggal mulai acara (minimum dan maksimum), serta tipe acara (*online* atau *offline*). Proses validasi juga dilakukan terhadap *filter* tanggal yang dimasukkan oleh pengguna, misalnya, tanggal registrasi minimum tidak dapat melebihi tanggal registrasi maksimum. Setelah validasi berhasil, halaman ini akan menampilkan daftar acara sesuai dengan kriteria yang ditentukan oleh pengguna. Gambar 3.19 merupakan *flowchart* untuk halaman rekomendasi pada Event Hunters:



Gambar 3.19. Flowchart *Recommendation Page*

Pada halaman rekomendasi juga diperlukan verifikasi otentikasi pengguna. Halaman ini bertujuan untuk menampilkan daftar 10 rekomendasi acara yang disediakan oleh model rekomendasi melalui API. Untuk mendapatkan rekomendasi, pengguna harus mengklik tombol *Generate Recommendation*. Tombol ini akan memanggil API endpoint *recommend/UserId* dimana *parameter* "UserId"

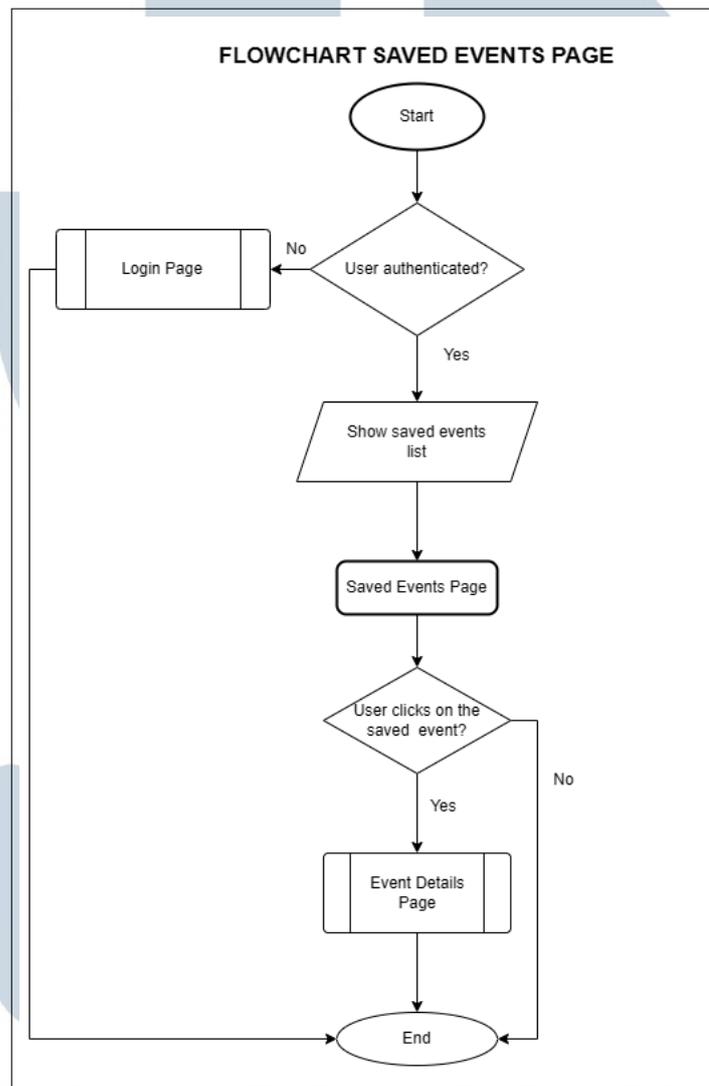
digunakan untuk menghitung rekomendasi berdasarkan id pengguna dari sisi API. Proses perhitungan ini terdiri dari 2 tahap rekomendasi yang telah dijelaskan pada gambar 3.9 dan 3.10 sebelumnya dalam *flowchart* sistem rekomendasi yang diajukan. Sebelum melakukan pemanggilan ke API endpoint, akan dilakukan pengecekan ketersediaan *session* pengguna yang melakukan *request* di *database*. Gambar 3.20 merupakan *flowchart* untuk halaman *my tickets* pada Event Hunters:



Gambar 3.20. Flowchart My Tickets Page

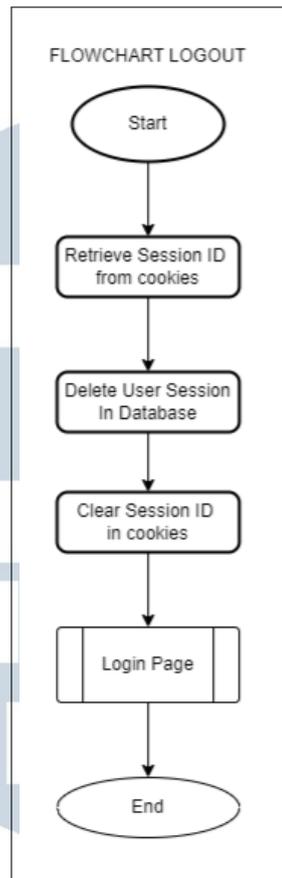
Pada halaman tiket, juga ada verifikasi otentikasi pengguna. Halaman ini berfungsi menampilkan daftar tiket yang telah dimiliki pengguna melalui registrasi acara. Pengguna dapat mengklik tiket yang ditampilkan untuk diarahkan ke halaman informasi tiket. Pada halaman informasi tiket, terdapat informasi yang menampilkan data pengguna dan acara, lalu bagian untuk memberikan *rating*

terhadap acara terkait dan tombol untuk menuju ke halaman detail acara. Jika pengguna telah memberikan *rating*, maka bagian untuk memberikan *rating* pada halaman informasi tiket tidak akan ditampilkan. Gambar 3.21 merupakan *flowchart* untuk halaman *saved events* pada Event Hunters:



Gambar 3.21. Flowchart *Saved Events Page*

Pada halaman *saved events*, akan dilakukan verifikasi otentikasi pengguna terlebih dahulu. Halaman ini berperan dalam menampilkan daftar acara yang telah disimpan (*bookmark*) oleh pengguna. Pengguna dapat mengklik acara yang ditampilkan untuk diarahkan ke halaman detail acara. Gambar 3.22 merupakan *flowchart* untuk proses *logout*:

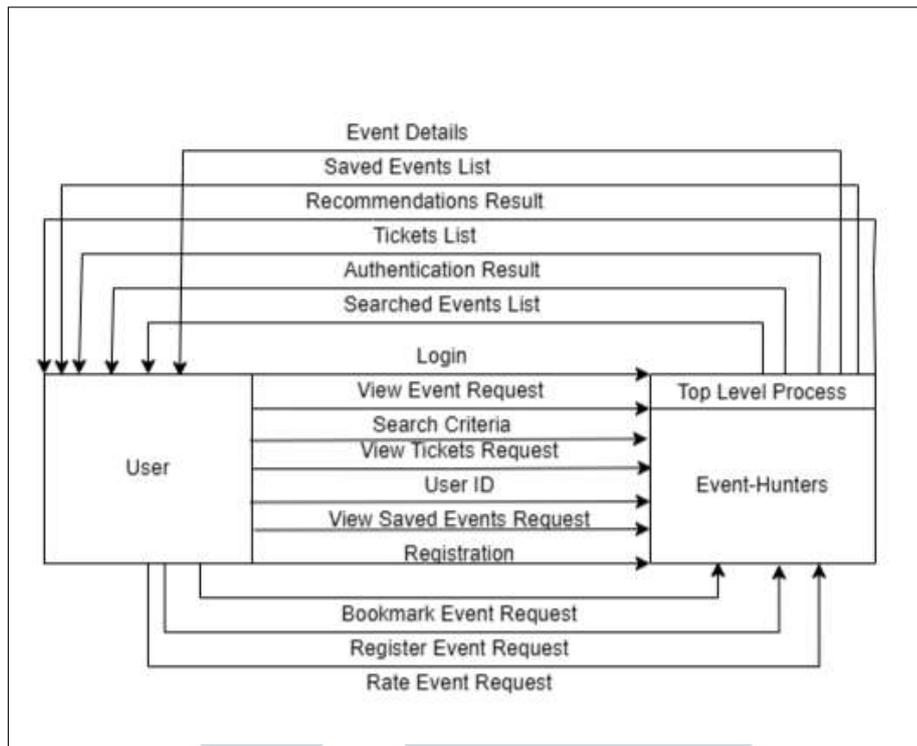


Gambar 3.22. Flowchart Logout

Proses *logout* dimulai dengan mengambil *session* id yang tersimpan di *cookies*. Setelah mendapatkan *session* id, langkah selanjutnya adalah menghapus *session* tersebut dari *database* berdasarkan *session* id yang telah diperoleh. Jika penghapusan *session* di *database* berhasil, *session* di *cookies* juga akan dihapus, dan pengguna akan diarahkan kembali ke halaman *login*.

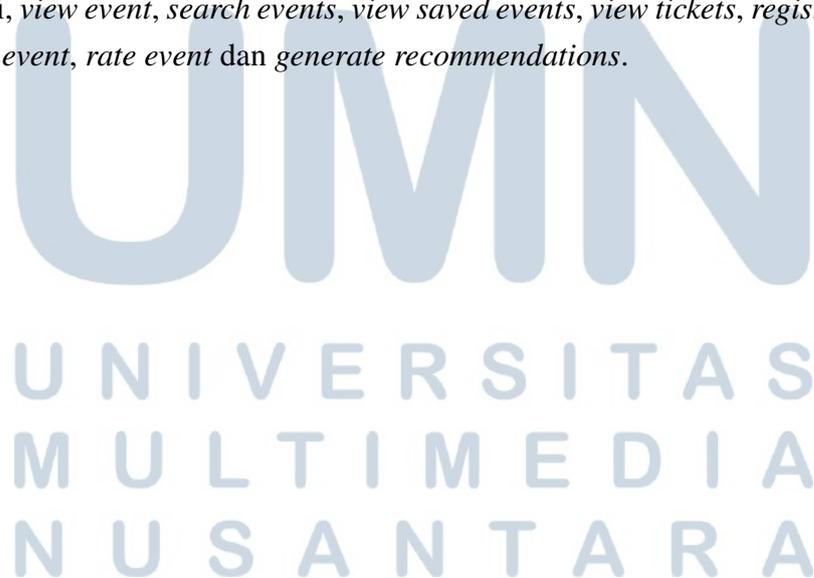
## B Data Flow Diagram (DFD)

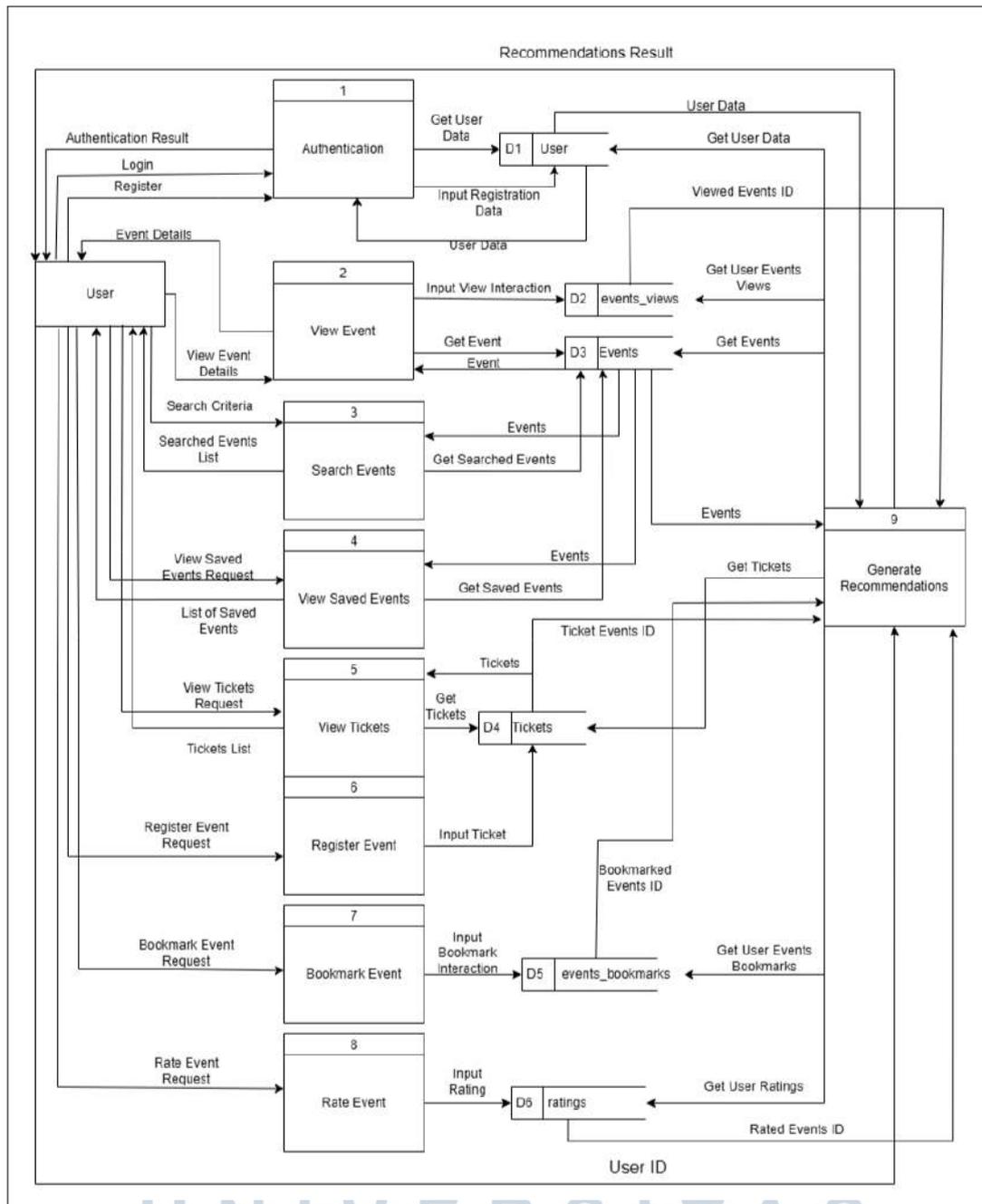
Diagram alur data (DFD) adalah sebuah diagram yang menggunakan notasi-notasi tertentu untuk menggambarkan alur data dalam sistem. DFD ini sangat berguna dalam memahami proses kerja sistem secara logis, terstruktur, dan sistematis [28]. Gambar 3.23 merupakan keseluruhan alur yang dilakukan antara pengguna dengan aplikasi Event-Hunters:



Gambar 3.23. *Data Flow Diagram* level nol

Gambar 3.24 menunjukkan DFD level satu yang menjabarkan proses yang terdapat pada DFD level nol secara lebih rinci. Terdapat 9 subproses utama yaitu, autentikasi, *view event*, *search events*, *view saved events*, *view tickets*, *register event*, *bookmark event*, *rate event* dan *generate recommendations*.



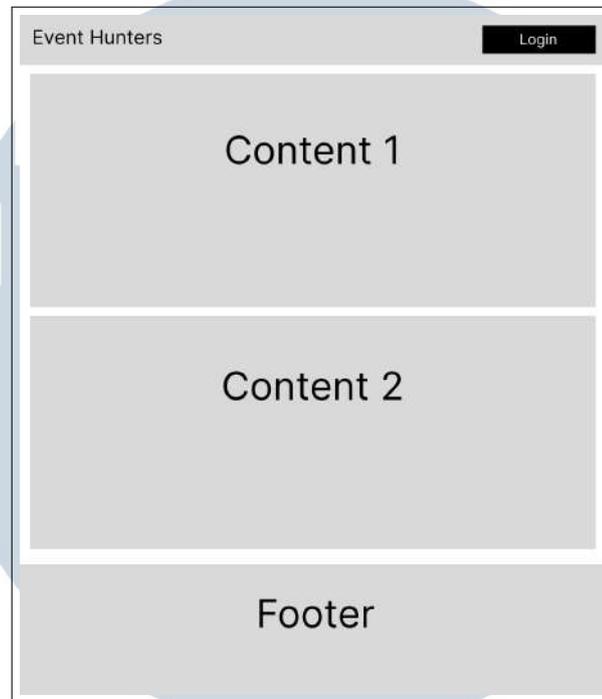


Gambar 3.24. Data Flow Diagram level satu

### C Wireframe

*Wireframe* merujuk pada representasi visual sederhana dari antarmuka pengguna suatu aplikasi atau situs web. *Wireframe* biasanya digunakan pada tahap awal desain untuk menggambarkan struktur dan tata letak elemen-elemen utama tanpa detail grafis. Komponen utama dari *wireframe* mencakup *banner*, *header*,

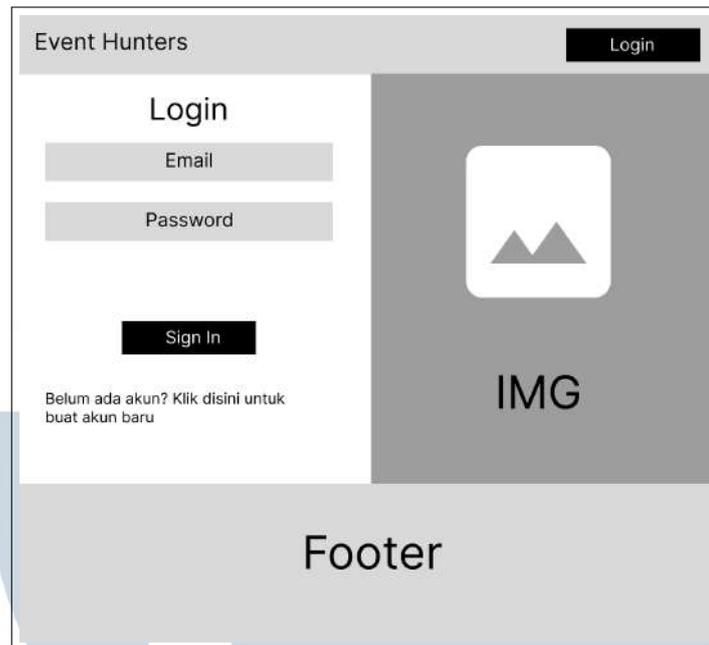
*content, footer, link, serta form* [29]. Gambar 3.25 merupakan *wireframe* halaman *landing page* Event Hunters:



Gambar 3.25. *Wireframe landing page*

*Wireframe landing page* terdiri dari beberapa bagian. Pada bagian atas terdapat *navbar* yang mencakup nama *website* dan tombol menuju halaman *login*. Bagian berikutnya adalah konten pertama dari halaman awal yang memberikan pengantar tentang aplikasi Event Hunters. Selanjutnya, dibawah konten pertama terdapat *carousel* yang menampilkan tanggapan dari pengguna tiruan terhadap manfaat yang diberikan oleh aplikasi Event Hunters. Bagian terakhir adalah *footer*. Gambar 3.26 menampilkan *wireframe* halaman *login*.

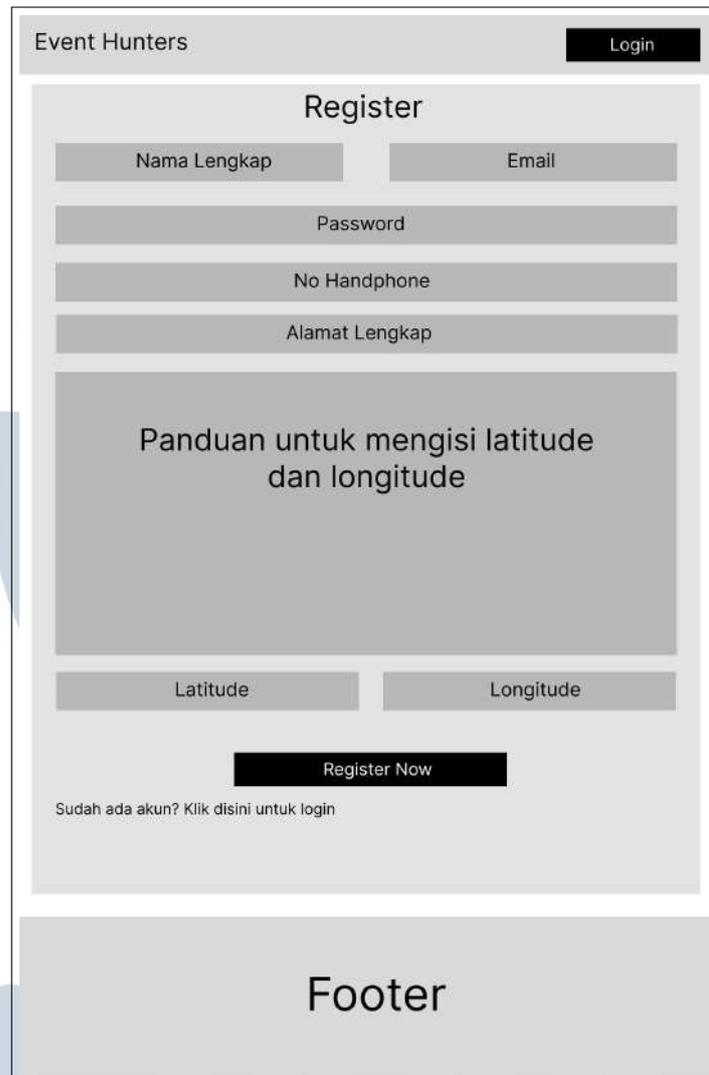
U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A



Gambar 3.26. Wireframe Login Page

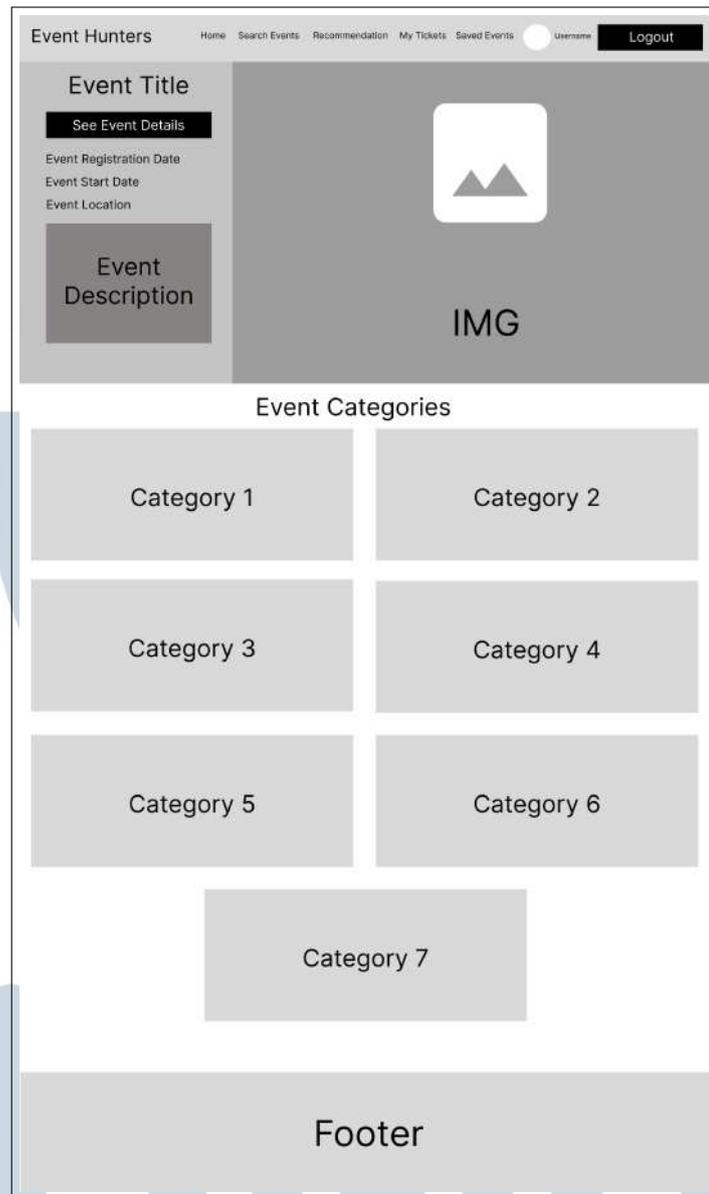
Wireframe halaman *login* memiliki beberapa komponen yang serupa dengan *wireframe* sebelumnya, seperti *navbar* dan *footer*. Pada halaman *login*, terdapat *form login* yang mencakup *input form* untuk *email* dan *password*, serta tombol *sign in* untuk masuk ke halaman utama jika *email* dan *password* telah sesuai. Selain itu, terdapat juga *link* di bawah tombol *sign in* yang mengarahkan pengguna ke halaman registrasi jika belum memiliki akun. Gambar 3.27 menampilkan *wireframe* halaman registrasi.

U M I N  
U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A



Gambar 3.27. Wireframe Register Page

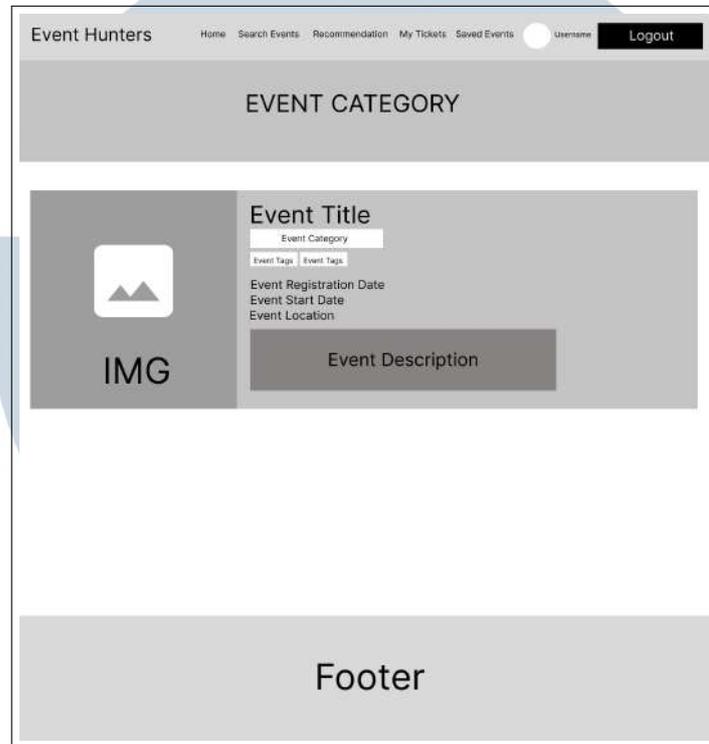
Wireframe halaman registrasi juga memiliki beberapa komponen yang serupa dengan wireframe sebelumnya. Halaman registrasi menampilkan form registrasi yang mencakup beberapa input form, seperti nama lengkap, email, password, nomor handphone, alamat lengkap, serta titik latitude dan longitude. Pada bagian pengisian titik latitude dan longitude, terdapat panduan untuk mendapatkan dan mengisi kedua titik tersebut dari Google Maps. Selanjutnya, terdapat tombol Register Now untuk melakukan registrasi berdasarkan data yang telah diisi. Di bawah tombol tersebut, terdapat link yang mengarahkan pengguna ke halaman login jika sudah memiliki akun dan tidak perlu melakukan registrasi. Gambar 3.28 menunjukkan wireframe untuk halaman utama (home page).



Gambar 3.28. Wireframe Home Page

Pada *wireframe* halaman utama (*home page*), terdapat *navbar* yang berbeda secara keseluruhan dari *wireframe* sebelumnya. *Navbar* ini adalah struktur *navbar* yang muncul setelah pengguna berhasil *login* ke halaman utama, dimana terdapat navigasi tambahan untuk menuju halaman utama, pencarian, rekomendasi, tiket pengguna, dan acara yang disimpan. Selain itu, terdapat juga foto profil pengguna, nama pengguna, dan tombol untuk *logout*. Bagian pertama dari *wireframe* ini berisi *carousel* acara yang menampilkan gambar dan deskripsi acara yang diputar, diikuti dengan bagian kategori acara yang menampilkan seluruh kategori acara yang

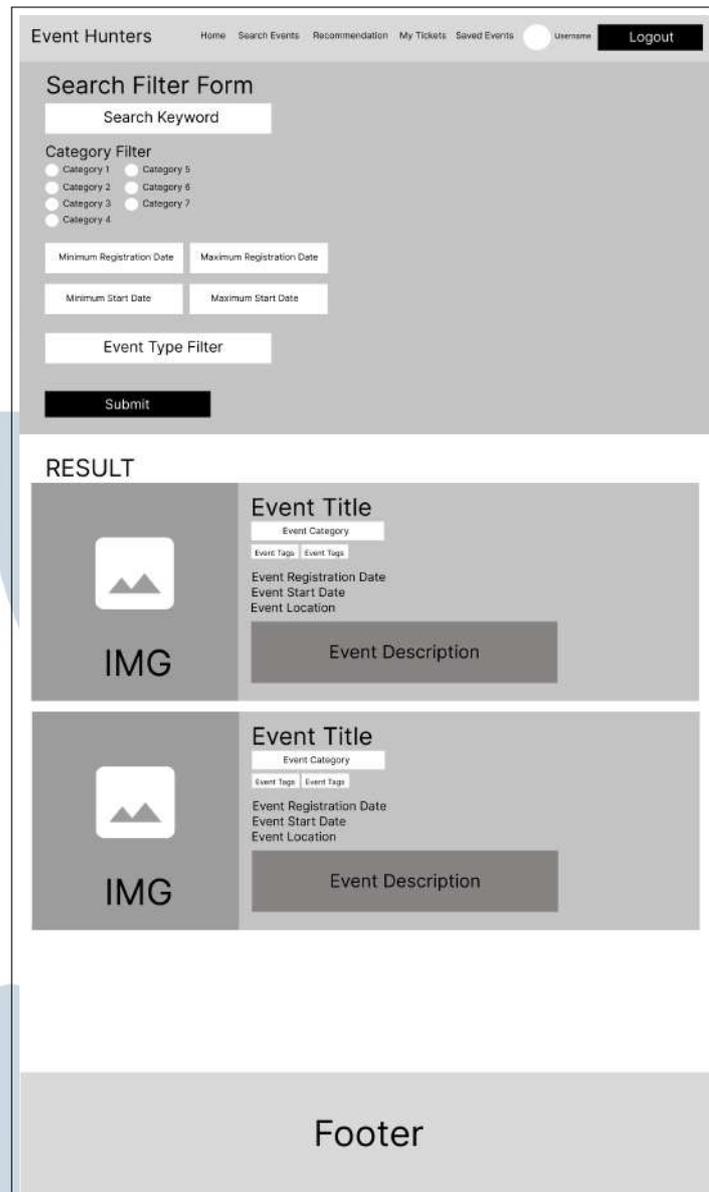
tersedia, dengan total 7 kategori. Bagian kategori ini memungkinkan pengguna untuk mengakses halaman daftar acara berdasarkan kategori yang dipilih. Gambar 3.29 menunjukkan *wireframe* untuk halaman kategori acara yang dipilih.



Gambar 3.29. *Wireframe Event Category Page*

*Wireframe* halaman kategori acara memiliki komponen *navbar* yang identik dengan halaman utama. Di bawah *navbar*, terdapat *header* yang menampilkan jenis kategori acara, diikuti oleh bagian yang menampilkan daftar acara sesuai dengan jenis kategori yang dipilih oleh pengguna. Gambar 3.30 menampilkan *wireframe* untuk halaman pencarian acara.

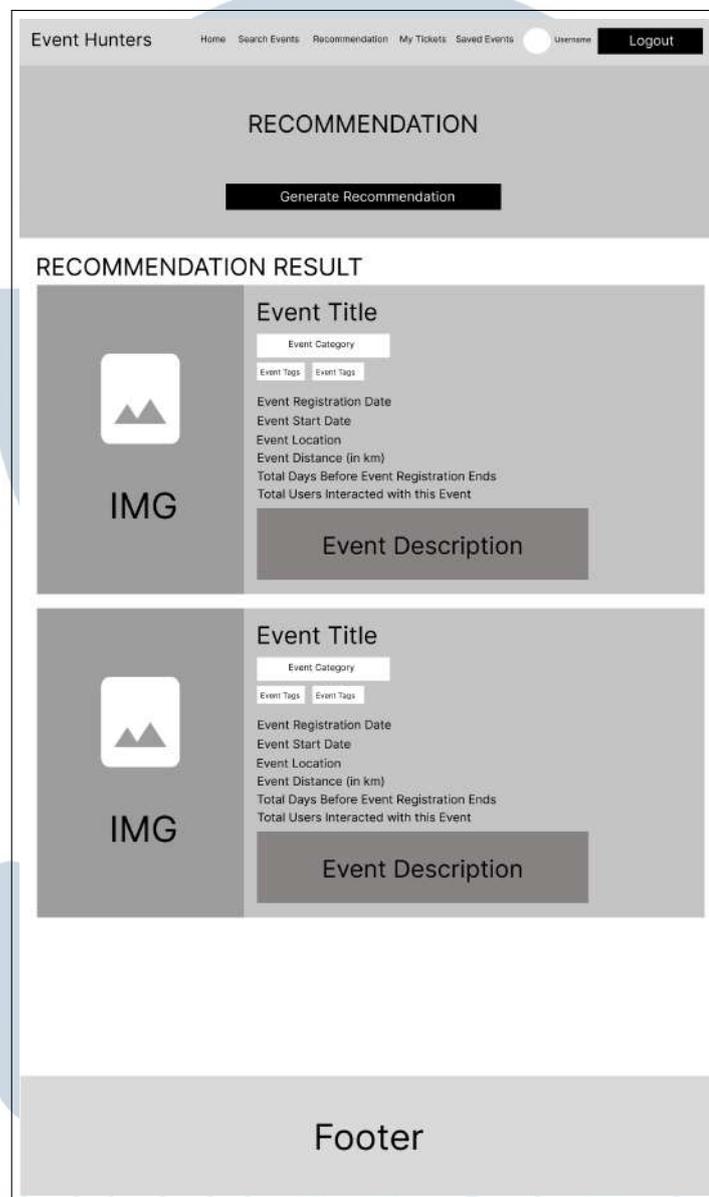
U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A



Gambar 3.30. Wireframe Search Page

Wireframe halaman pencarian acara juga memiliki komponen *navbar* yang sama dengan halaman utama. Di bawah *navbar*, terdapat *form* pencarian acara yang memuat beberapa *input*, seperti kata kunci pencarian, kategori yang diinginkan, rentang tanggal registrasi acara, rentang tanggal mulai acara, dan tipe acara yang diinginkan. Di bawah *input* tipe acara, terdapat tombol untuk mengirimkan permintaan pencarian berdasarkan input pengguna. Setelahnya, terdapat bagian yang menampilkan hasil pencarian acara dengan format yang telah ditentukan. Format tampilannya mencakup gambar, judul acara, kategori acara, *tag* acara,

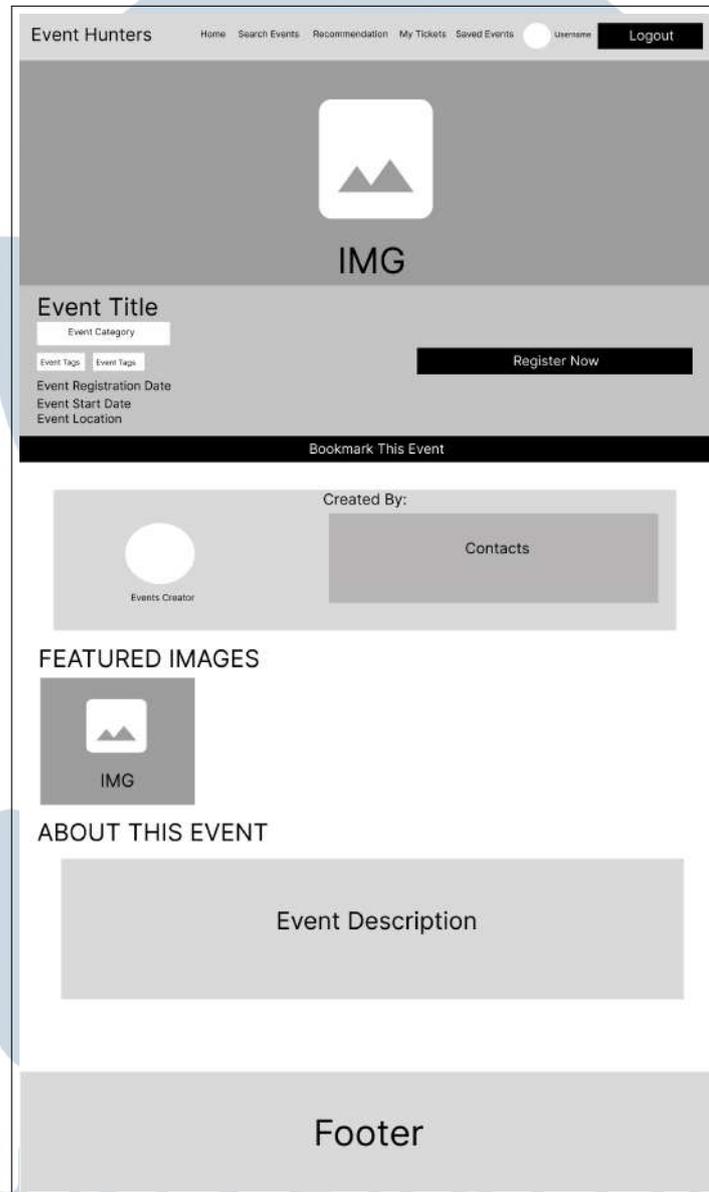
tanggal registrasi acara, tanggal mulai acara, lokasi acara, dan deskripsi acara. Gambar 3.31 menunjukkan *wireframe* untuk halaman rekomendasi acara.



Gambar 3.31. *Wireframe Recommendation Page*

*Wireframe* halaman rekomendasi acara juga memiliki komponen *navbar* yang sama dengan halaman utama. Di bawah *navbar*, terdapat *header* yang memuat tombol untuk *generate* rekomendasi, lalu di bawah *header* terdapat bagian untuk menampilkan hasil rekomendasi acara. Untuk format tampilan hasil rekomendasi acara, kurang lebih mirip dengan format tampilan acara di halaman pencarian acara. Namun, terdapat beberapa *field* tambahan yang ditampilkan dalam format

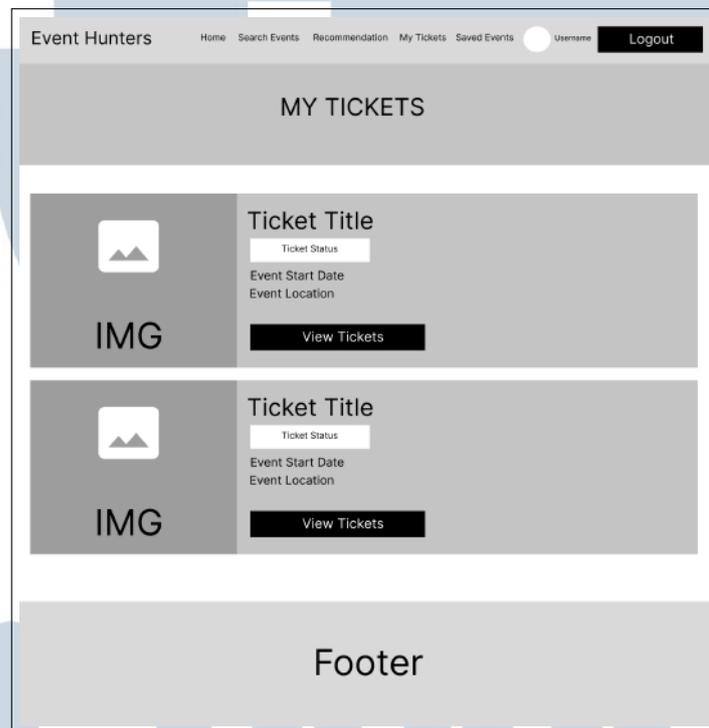
rekomendasi acara, yaitu jarak acara dalam km, jumlah hari sebelum registrasi acara berakhir, dan total pengguna yang berinteraksi dengan acara tersebut. Gambar 3.32 menunjukkan *wireframe* untuk halaman detail acara.



Gambar 3.32. *Event Details Page Wireframe*

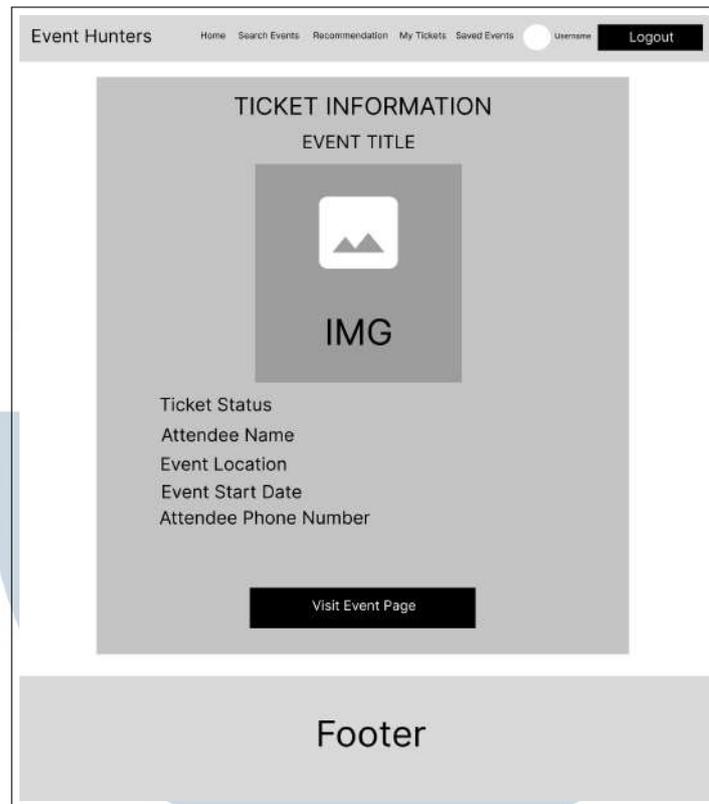
*Wireframe* halaman detail acara juga memiliki komponen *navbar* yang sama dengan halaman utama. Di bawah *navbar*, terdapat gambar dari acara yang ditampilkan, lalu di bawah gambar tersebut terdapat deskripsi dari acara tersebut yang mencakup judul acara, kategori acara, *tags* acara, tanggal registrasi acara, tanggal mulai acara, dan lokasi acara. Pada bagian kanan deskripsi acara, terdapat

tombol untuk melakukan registrasi acara terkait. Di bawah deskripsi acara, terdapat tombol untuk menyimpan (*bookmark*) acara. Dilanjut dengan bagian deskripsi dari pembuat acara terkait. Di bawah deskripsi pembuat acara, terdapat bagian *featured images* yang memuat gambar dari acara tersebut, pengguna dapat mengklik gambar tersebut untuk melihat gambar acara secara keseluruhan. Lalu di bawah *featured images*, terdapat deskripsi acara terkait secara keseluruhan. Gambar 3.33 menunjukkan *wireframe* untuk halaman *my tickets*.



Gambar 3.33. *My Tickets Page Wireframe*

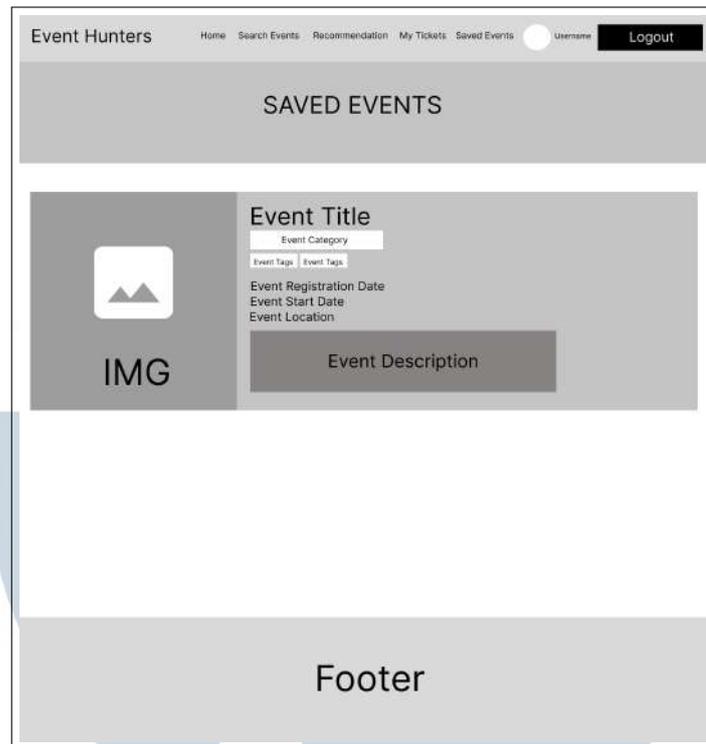
*Wireframe* halaman *my tickets* memiliki komponen *navbar* yang sama dengan halaman utama. Di bawah *navbar*, terdapat *header* yang dilanjutkan dengan daftar tiket yang dimiliki pengguna. Pengguna dapat mengklik tiket tersebut untuk melihat informasi lengkap mengenai tiket serta acara yang bersangkutan. Gambar 3.34 menunjukkan *wireframe* untuk halaman informasi tiket.



Gambar 3.34. *Ticket Information Page Wireframe*

*Wireframe* halaman informasi tiket memiliki komponen *navbar* yang sama dengan halaman utama. Pada *wireframe* ini, terdapat informasi tiket mengenai acara yang telah diregistrasi oleh pengguna. Bagian ini mencakup judul acara, status tiket, nama pengguna, lokasi acara, tanggal mulai acara, dan nomor telepon pengguna. Selain itu, terdapat tombol untuk mengunjungi halaman detail acara sesuai dengan yang tercantum pada tiket tersebut. Gambar 3.35 menampilkan *wireframe* untuk halaman acara yang disimpan.

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 3.35. *Saved Events Page Wireframe*

*Wireframe* halaman *saved events* memiliki komponen *navbar* yang sama dengan halaman utama. Struktur *wireframe* ini mirip dengan struktur halaman *My Ticket*, di mana terdapat *header* yang diikuti oleh daftar acara yang telah disimpan oleh pengguna. Pengguna dapat mengklik acara tersebut untuk melihat informasi lengkap mengenai acara terkait.

### 3.5 Pembangunan aplikasi Event Hunters dan model sistem rekomendasi

Setelah selesai merancang aplikasi website, database, dan sistem rekomendasi, langkah berikutnya adalah membangun aplikasi dan sistem rekomendasi tersebut sesuai dengan rancangan yang telah didefinisikan sebelumnya. Database akan diimplementasikan menggunakan PostgreSQL. Aplikasi website akan dikembangkan sebagai *monolithic application* menggunakan *framework Golang* yang disebut *Fiber*, di mana *frontend* dan *backend* tidak dipisahkan, melainkan disatukan menjadi satu kesatuan. Sementara itu, sistem rekomendasi akan dibangun menggunakan *Flask*, sebuah *framework Python*, yang akan dipanggil sebagai API dari aplikasi.

### **3.6 Integrasi model sistem rekomendasi dengan aplikasi website event-hunters**

Setelah pembangunan aplikasi Event Hunters dan sistem rekomendasi selesai, langkah selanjutnya adalah mengintegrasikan sistem rekomendasi tersebut dengan aplikasi website Event Hunters. Integrasi dilakukan dengan memanggil API *endpoint* ke `"/recommend/userId"` untuk mendapatkan hasil rekomendasi yang telah dihitung. Dalam proses integrasi ini, hasil perhitungan seperti jarak, jumlah hari sebelum registrasi berakhir, dan jumlah interaksi pengguna terhadap suatu acara yang didapatkan dari perhitungan rekomendasi juga turut digunakan dalam menampilkan hasil rekomendasi kepada pengguna *website*.

### **3.7 Testing serta Debugging**

Tahap ini bertujuan untuk memastikan bahwa aplikasi yang telah dibangun dapat berjalan dengan lancar tanpa ada *bug* ketika diakses dan diuji oleh pengguna secara langsung.

### **3.8 Deployment**

Setelah memastikan bahwa aplikasi bebas dari *bug* atau masalah, langkah selanjutnya adalah melakukan *deployment* aplikasi sehingga dapat diuji dan dievaluasi oleh pengguna secara langsung. *Deployment* aplikasi akan dilakukan menggunakan *platform cloud* yang bernama Railway.

### **3.9 Pengujian dan Evaluasi Sistem**

Setelah *deployment* berhasil dan aplikasi telah berjalan di lingkungan produksi, langkah selanjutnya adalah melakukan pengujian dan evaluasi sistem. Pengujian akan menggunakan metode EUCS (*End User Computing Satisfaction*) untuk mengukur kepuasan pengguna terhadap website pencarian acara beserta sistem rekomendasinya. Selain pengujian oleh pengguna, juga akan dilakukan pengujian dengan metode *blackbox testing* untuk memastikan bahwa seluruh komponen aplikasi berjalan sesuai dengan harapan.

### 3.10 Penulisan Laporan

Setelah pengujian dan evaluasi selesai, langkah selanjutnya adalah menulis laporan sebagai dokumentasi atas proses pembuatan dan hasil akhir dari sistem tersebut.

