

## BAB III

### METODOLOGI PENELITIAN DAN PERANCANGAN SISTEM

#### 3.1 Metodologi Penelitian

Beberapa metode yang digunakan dalam merancang dan membangun sebuah sistem pendukung keputusan yaitu:

1. Studi Literatur

Studi literatur penelitian ini melakukan pembelajaran terhadap teori-teori yang berhubungan dengan *data mining* khususnya pada teknik klasifikasi yaitu algoritma *Naive Bayes Classifier*, teknik untuk mempersiapkan dataset yang baik dan benar, metode validasi model klasifikasi yaitu *confusion matrix*.

2. Pengumpulan data

Pengumpulan data yang akan dilakukan dengan cara pengambilan data nilai akhir mata kuliah dengan kode IF semester satu sampai dengan empat di prodi informatika dan nilai akhir penjurusan setiap individu angkatan 2014 dari BAAK Universitas Multimedia Nusantara.

3. Perancangan dan pembangunan sistem

Tahapan perancangan dan pembangunan sistem adalah tahapan dimana sistem yang digunakan untuk penelitian akan dibuat. Perancangan sistem pendukung keputusan peminatan akan dirancang dengan diagram *flowehart*, *data flow diagram* dan *mock-up* untuk antarmuka. Pembangunan sistem pada penelitian ini akan membuat sistem klasifikasi yang berbasis web dibuat dengan bahasa pemrograman PHP menggunakan *framework Codeigniter* dimana *data training* akan disimpan didalam *database* MySQL. Pembangunan *front-end* akan menggunakan *framework bootstrap* dimana *front-end* sistem akan *responsive*.

#### 4. Uji coba dan evaluasi

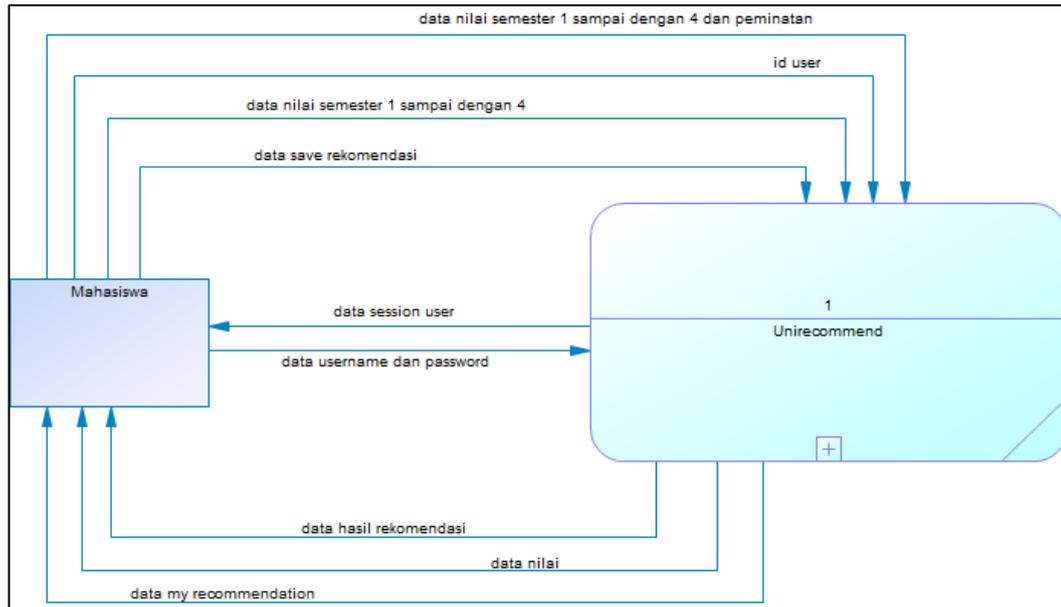
Tahapan ini melakukan uji coba (*testing*) sistem dengan *data test* yang ada sebagai parameter sistem. Metode evaluasi yang digunakan adalah *confusion matrix* *dataset* akan dibagi 2 yaitu *data testing* dan *data training* lalu hasil evaluasi akan diisi menggunakan *confusion matrix*. Uji coba algoritma *Naive Bayes Classifier* akan dilakukan dengan cara melakukan perhitungan manual sesuai dengan skenario yang ada dan akan mencocokkan hasil klasifikasi antara perhitungan manual dengan di sistem.

### 3.2 Perancangan Sistem

Beberapa penjelasan mengenai perancangan sistem yang akan digunakan pada penelitian ini diantaranya adalah *Data Flow Diagram*, *Flowchart* Sistem, Struktur Tabel *Database* dan Perancangan Antarmuka Sistem. Sistem pendukung keputusan peminatan yang telah dibuat ini diberi nama sistem *unirecommend*.

#### 3.2.1 Data Flow Diagram

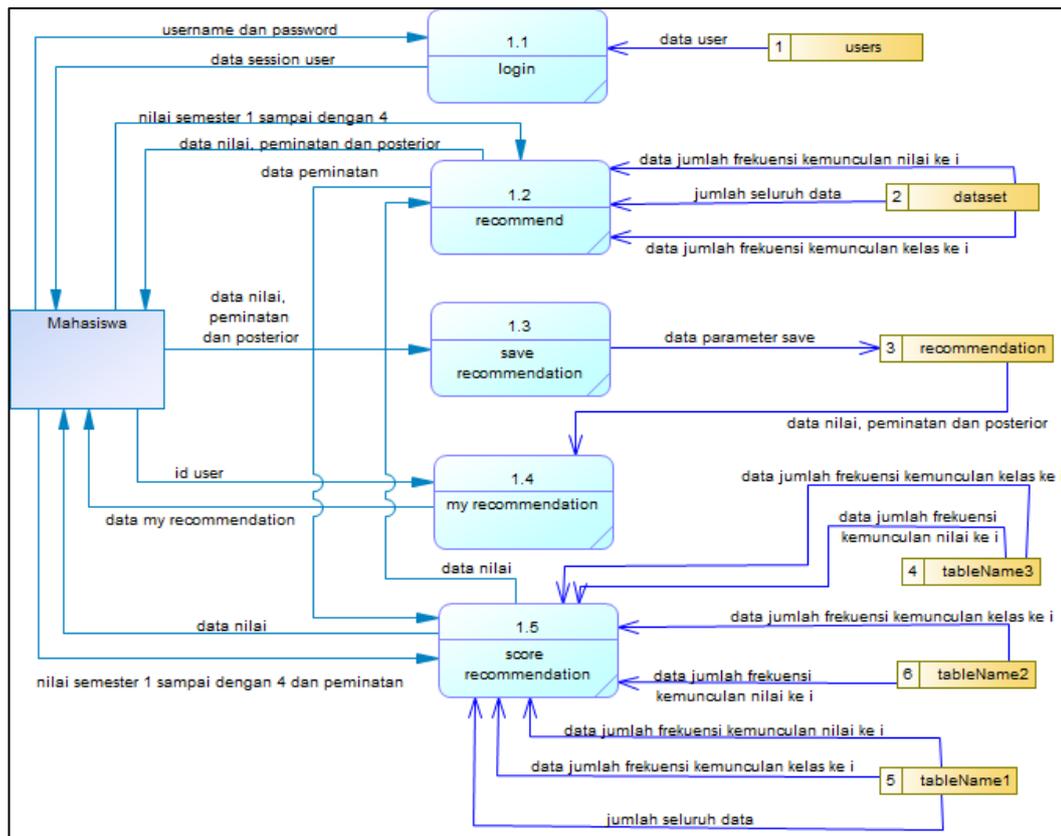
Pembuatan *data flow* diagram digunakan untuk menggambarkan pertukaran data antara sistem dan pengguna sistem. Gambar 3.1 menunjukkan diagram konteks dari Sistem pendukung keputusan pemilihan peminatan.



Gambar 3.1 Diagram Konteks Unirecommend

Diagram konteks *unirecommend* hanya mempunyai satu entitas yaitu entitas mahasiswa. Entitas mahasiswa adalah entitas yang terdaftar dalam *database* tabel *user* yang dapat melakukan input nilai-nilai mata kuliah yang berkode IF dari semester satu sampai dengan empat, melakukan input peminatan, melakukan input data *username* dan *password*. Entitas mahasiswa dapat memperoleh hasil rekomendasi peminatan, mendapatkan hasil nilai akhir peminatan tersebut, mendapatkan data *login session* dan mendapatkan nilai probabilitas seluruh posterior kelas.

UMMN  
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 3.2 Diagram Level 0 Unirecommend

Gambar 3.2 menunjukkan diagram *level* satu dari *unirecommend* dimana terdapat 5 proses yaitu proses *login*, proses *recommend*, proses *save recommendation*, proses *my recommendation* dan proses *score recommend*. Proses pertama dari diagram level 1 sistem pendukung keputusan pemilihan peminatan yaitu proses *login*. Proses *login* adalah proses dimana validasi input data *username* dan *password* yang dikirim oleh user (mahasiswa) dilakukan. Data yang dikirim *user* diverifikasi ke database untuk mengonfirmasi status validasi *username* dan *password* di dalam *database*. Apabila *username* dan *password* terdaftar, maka proses *login* akan mengirim data *session login* yang didapatkan dari data *user* yang berasal dari *database users*.

Proses kedua dari diagram level 1 *unirecommend* adalah proses *recommend*. Proses *recommend* adalah proses rekomendasi peminatan dilakukan dimana proses

*recommend* menerima data parameter yang dikirim entitas mahasiswa, yaitu nilai semester 1 sampai dengan 4. Proses *recommend* kemudian akan melakukan perhitungan jumlah frekuensi kemunculan setiap nilai (*attribute*), melakukan perhitungan jumlah frekuensi kemunculan setiap kelas, dan melakukan perhitungan jumlah data di *database* pada tabel *dataset*. Proses *recommend* selanjutnya mengirimkan data hasil peminatan ke proses *score recommendation* dan menerima data nilai dari proses *score recommendation*. Proses rekomendasi selanjutnya akan mengirimkan data nilai, data peminatan dan data probabilitas posterior seluruh kelas ke entitas mahasiswa.

Proses ketiga dari diagram level 1 *unirecommend* adalah proses *save recommendation*. Proses *save recommendation* adalah proses penyimpanan hasil rekomendasi peminatan, nilai akhir peminatan dan nilai probabilitas posterior seluruh kelas ke dalam *database* pada tabel *recommendation*. Data *save* rekomendasi berupa tiga parameter tersebut dikirim oleh entitas mahasiswa.

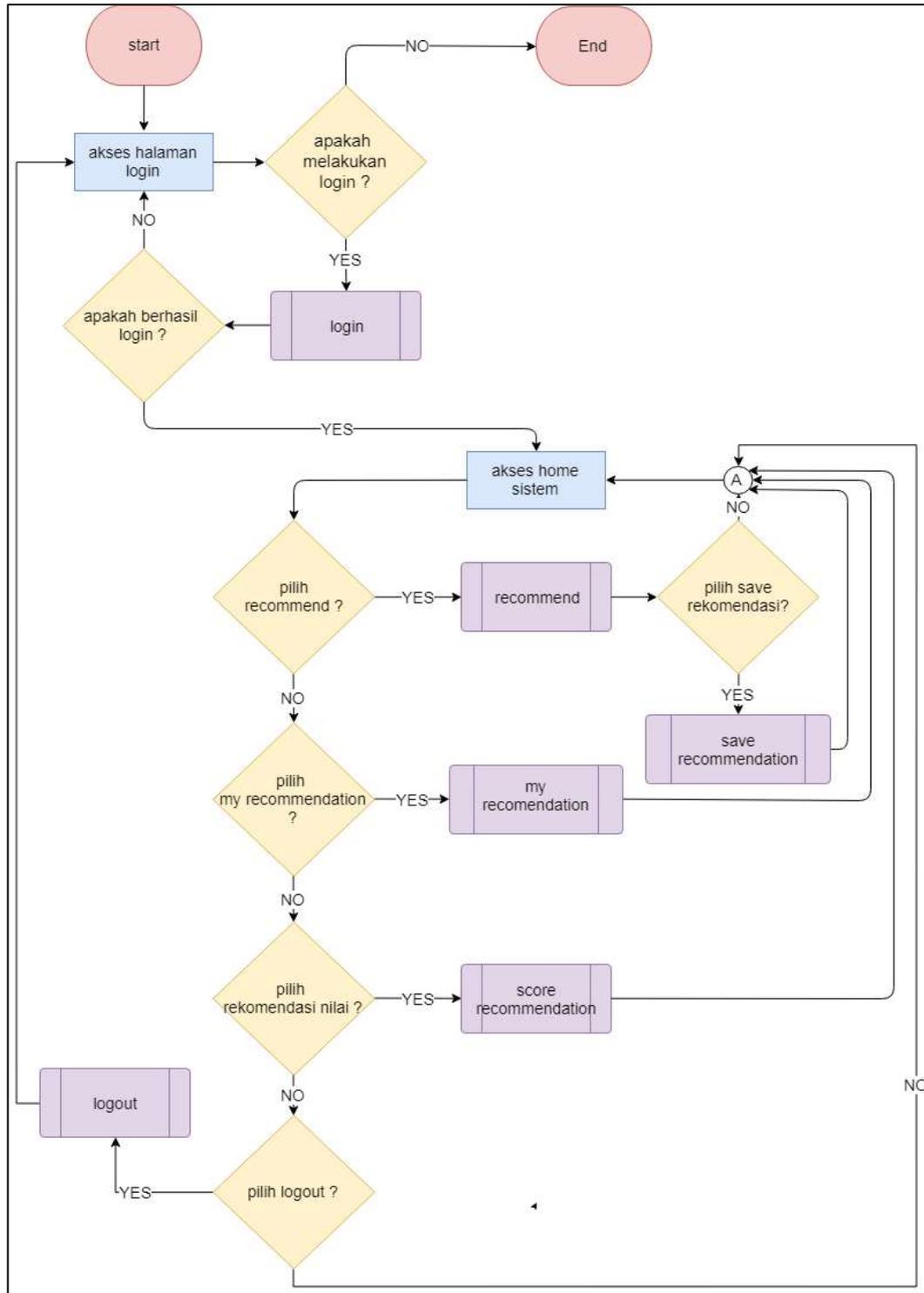
Proses keempat dari diagram level 1 *unirecommend* adalah *my recommendation*. Proses *my recommendation* adalah proses pengambilan data rekomendasi user dari *database* pada tabel *recommendation*. Proses *my recommendation* menerima data parameter *user id* dari entitas mahasiswa dan mengirimkan data *my recommendation* (data peminatan, nilai akhir peminatan dan nilai posterior seluruh kelas) sesuai *user id* entitas mahasiswa yang diambil dari tabel *recommendation* kepada entitas mahasiswa tersebut.

Proses kelima dari diagram level 1 *unirecommend* adalah *score recommendation*. Proses *score recommendation* menerima data parameter dari entitas mahasiswa, yaitu nilai semester satu sampai dengan empat serta peminatan,

dan mengirimkan data nilai. Proses *score recommendation* mengambil data jumlah frekuensi kemunculan nilai kesekian dan data jumlah frekuensi kemunculan kelas kesekian dari tabel peminatan pertama kedua dan ketiga. Proses *score recommendation* mengambil data mengenai jumlah banyaknya seluruh data hanya pada tabel peminatan pertama (*tableName1*) karena jumlah banyaknya seluruh data pada tabel peminatan kedua (*tableName2*) dan ketiga (*tableName3*) selalu sama dengan peminatan pertama. Proses *score recommendation* juga menerima data hasil peminatan dari proses *recommend* dan mengirim data nilai ke proses *recommend*.

### 3.2.2 Flowchart

Pada bagian ini, ditunjukkan perancangan *flowchart*. Perancangan *flowchart* digunakan untuk menyederhanakan rangkaian proses dalam menyelesaikan masalah. Perancangan *flowchart* pada sistem ini dibagi menjadi dua yaitu *flowchart* fungsi dan *flowchart* utama sistem. *Flowchart* fungsi adalah *flowchart* yang menjelaskan proses kerja fungsi dari pengisian *input* sampai menghasilkan *output* yang diinginkan. *Flowchart* utama sistem adalah *flowchart* yang menjelaskan urutan langkah-langkah dari proses kerja keseluruhan sistem. *Flowchart* fungsi terdiri dari *flowchart login*, *recommend*, *save recommendation*, *my recommendation* dan *score recommendation*.



Gambar 3.3 Flowchart Utama Sistem Unirecommend

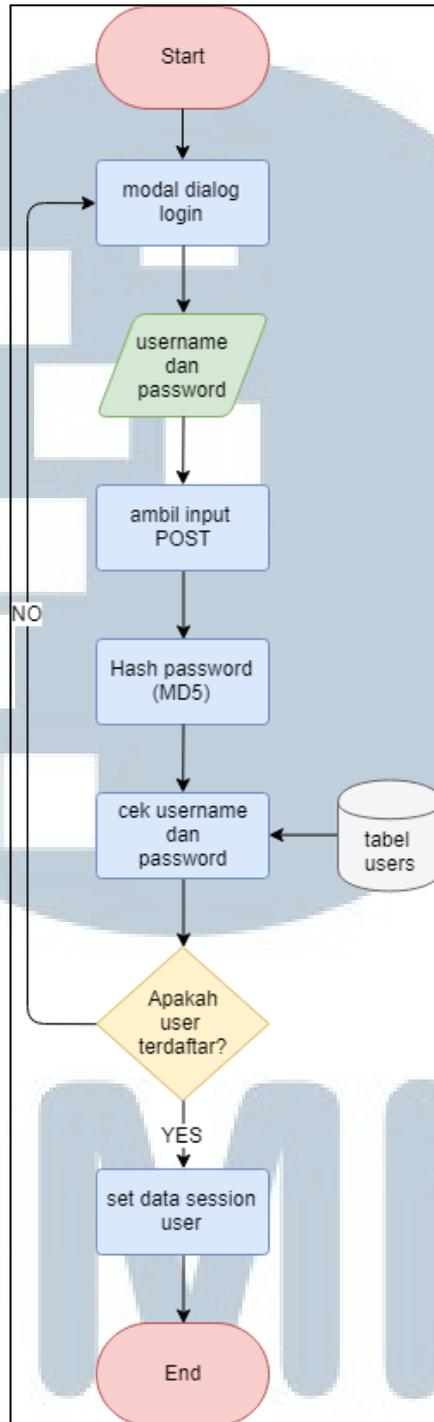
Gambar 3.3 menunjukkan *flowchart* utama *unirecommend* yang dapat diakses oleh mahasiswa. Sistem pertama akan menampilkan halaman *login*, jika mahasiswa melakukan *login* maka sistem akan mengarahkan ke subproses *login*

kemudian jika *login* berhasil maka sistem akan mengarahkan *user* (mahasiswa) ke *home* sistem.

Mahasiswa dapat melakukan memilih menu-menu yang sudah tersedia di dalam *home* sistem, berikut hal-hal yang dapat dilakukan oleh mahasiswa terhadap *home* sistem:

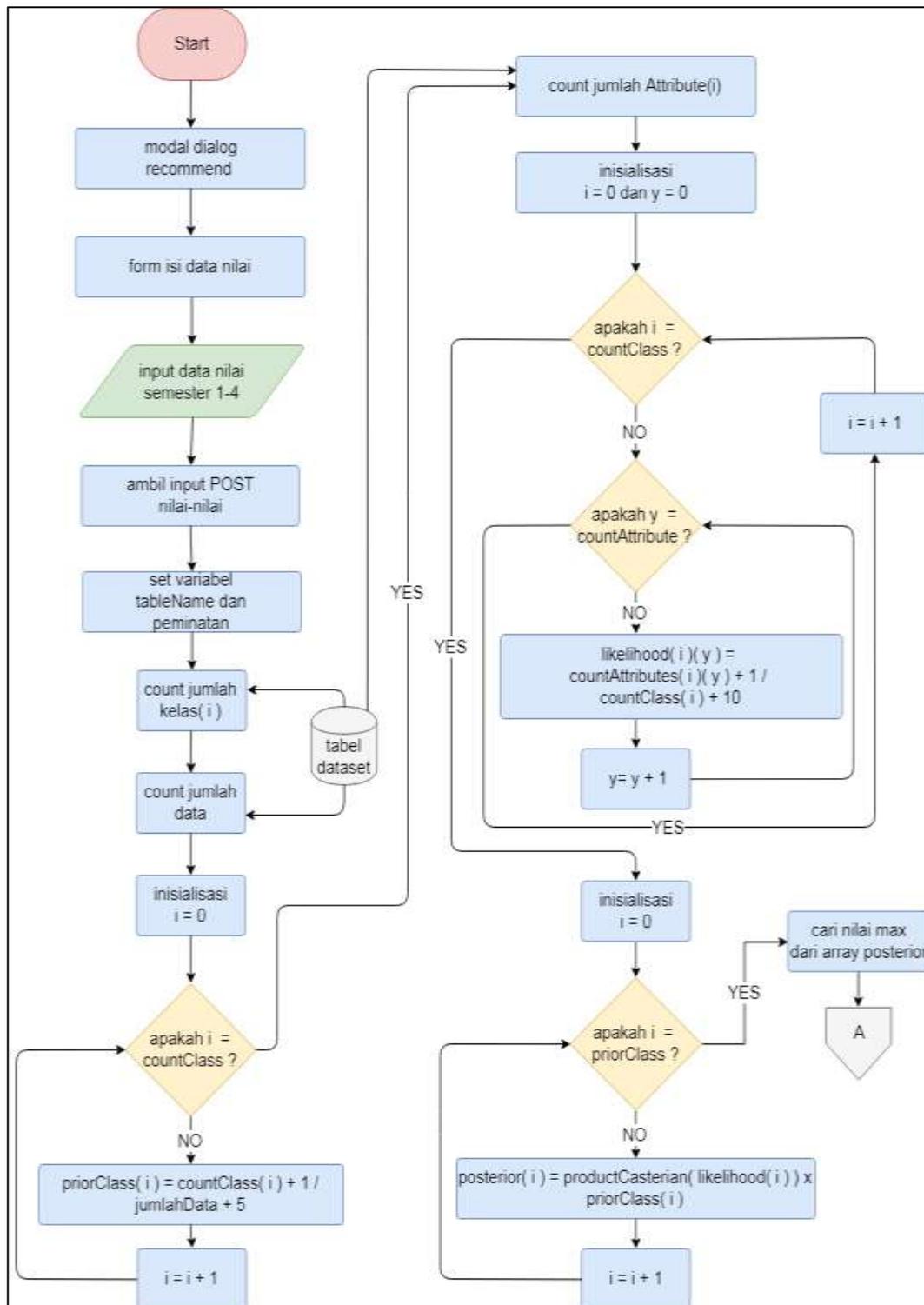
1. Memilih menu *recommend*
2. Memilih menu *my recommendation*
3. Memilih menu *score recommenndation*
4. Memilih menu *logout*
5. Memilih menu *save recommendation*

Gambar 3.4 di halaman berikutnya menunjukkan *flowchart* proses *login* untuk mahasiswa. Pertama-tama proses *login* menampilkan *modal dialog login*. Kemudian proses *login* akan meminta *user* memasukkan *username* dan *password*, dan selanjutnya proses *login* akan mengambil input *POST user*. Selanjutnya proses *login* melakukan *hashing* (MD5) *password* yang dikirim dari *user*. Kemudian proses *login* akan melakukan pengecekan apakah *username* dan *password* telah terdaftar di dalam tabel *users*. *User* yang terdaftar dalam proses *login* akan membuat data *session* yang berisikan *id user*, *username* dan *password* setelah proses *login* membuat data *session* maka proses *login* akan berhasil, namun jika *username* dan *password* tidak terdaftar pada tabel *login* maka proses *login* akan diulang.



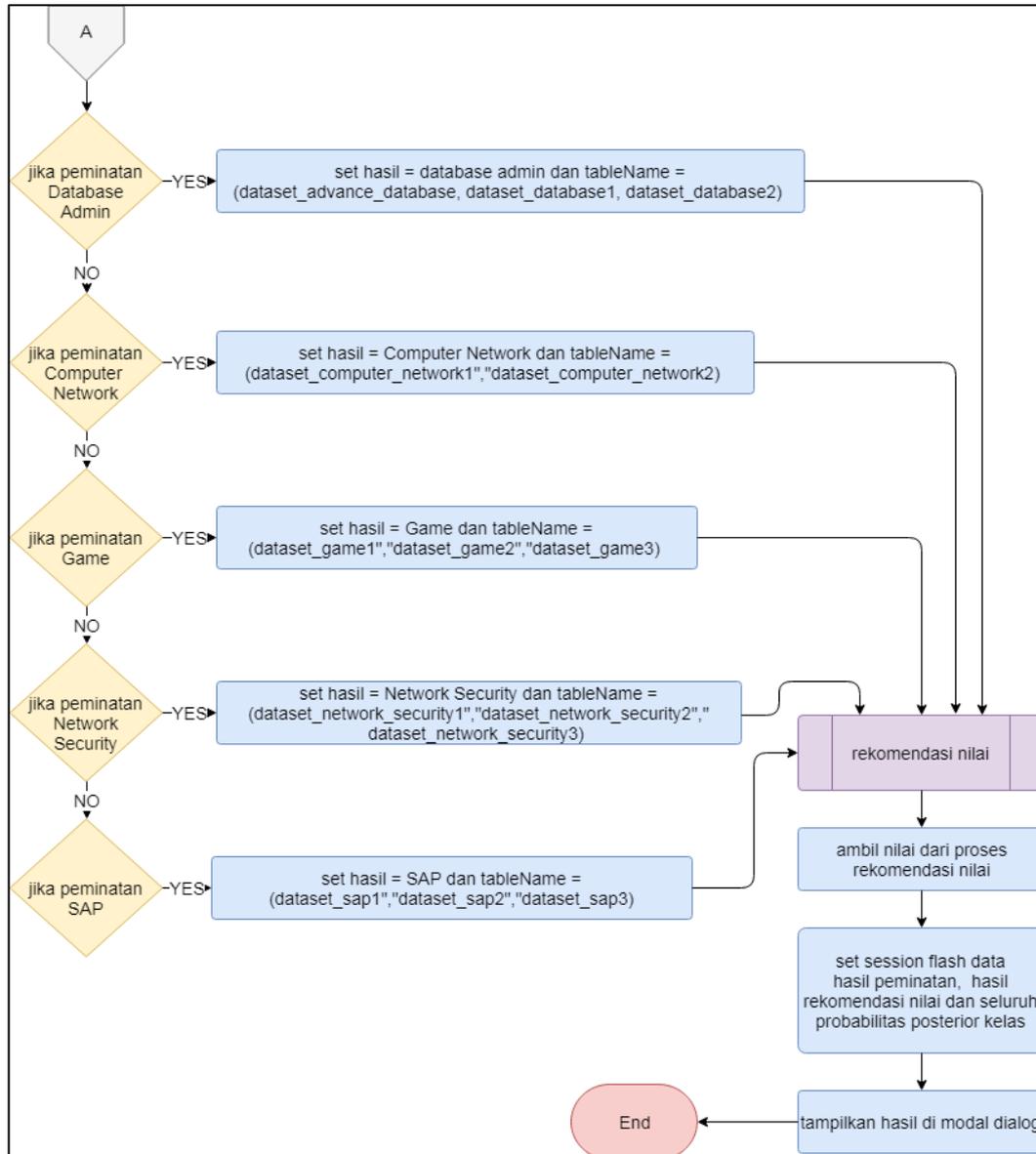
Gambar 3.4 Flowchart Proses Login Unirecommend

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 3.5 Flowchart Proses Recommend Unirecommend

MULTIMEDIA  
NUSANTARA



Gambar 3.6 Flowchart Proses Recommend Unirecommend (Lanjutan)

Gambar 3.5 dan gambar 3.6 menunjukkan *flowchart* proses *recommend* digambarkan, dimana pada proses ini algoritma *naïve bayes classifier* digunakan. Pertama-tama proses *recommend* akan menampilkan *modal dialog recommend form* isi dari data nilai semester satu sampai dengan empat. Selanjutnya *user* akan diminta untuk melakukan input data nilai akhir mata kuliah kode IF semester satu sampai dengan empat secara manual. Proses *recommend* akan mengambil input *POST* nilai-nilai yang dikirim oleh *user*. Kemudian proses *recommend* akan

melakukan *set* variabel *tableName* dan peminatan dimana *tableName* diisi dengan data *string* yaitu *dataset* dan *peminatan* diisi dengan *array* data peminatan Universitas Multimedia Nusantara yang berisi *database admin, computer network, game, network security* dan *SAP*.

Algoritma *naïve bayes classifier* mulai berjalan pada saat melakukan perhitungan jumlah frekuensi kemunculan setiap kelas (peminatan) pada tabel *dataset*. Kemudian proses *recommend* melakukan perhitungan banyaknya seluruh data pada tabel *dataset*. Setelah itu proses *recommend* akan menginisialisasi nilai *integer* *i* sama dengan nol untuk melakukan iterasi perhitungan nilai probabilitas *prior* setiap kelas (peminatan) dengan cara *countClass* dijumlahkan dengan satu lalu dibagi dengan jumlah data ditambah dengan lima, dimana angka lima adalah *denominator* (banyaknya kelas yang ada). Proses *recommend* akan melakukan perhitungan frekuensi kemunculan atribut nilai setiap mata kuliah semester satu sampai dengan empat pada tabel *dataset* dan melakukan inisialisasi *integer* *i* dan *y* dengan nilai awal nol untuk melakukan iterasi *array* dua dimensi. Perhitungan nilai probabilitas setiap atribut (mata kuliah) terhadap setiap kelas (peminatan) dengan cara *countAttributes* dijumlahkan dengan satu lalu dibagi dengan *countClass* ditambah dengan 10, dimana angka 10 adalah nilai *denominator* (banyaknya karakteristik atribut). Setelah itu proses *recommend* melakukan inisialisasi *integer* *i* dengan nilai awal 0 untuk melakukan iterasi perhitungan nilai probabilitas *posterior* tiap kelas atau label (peminatan) dengan cara melakukan operasi *casterian product*.

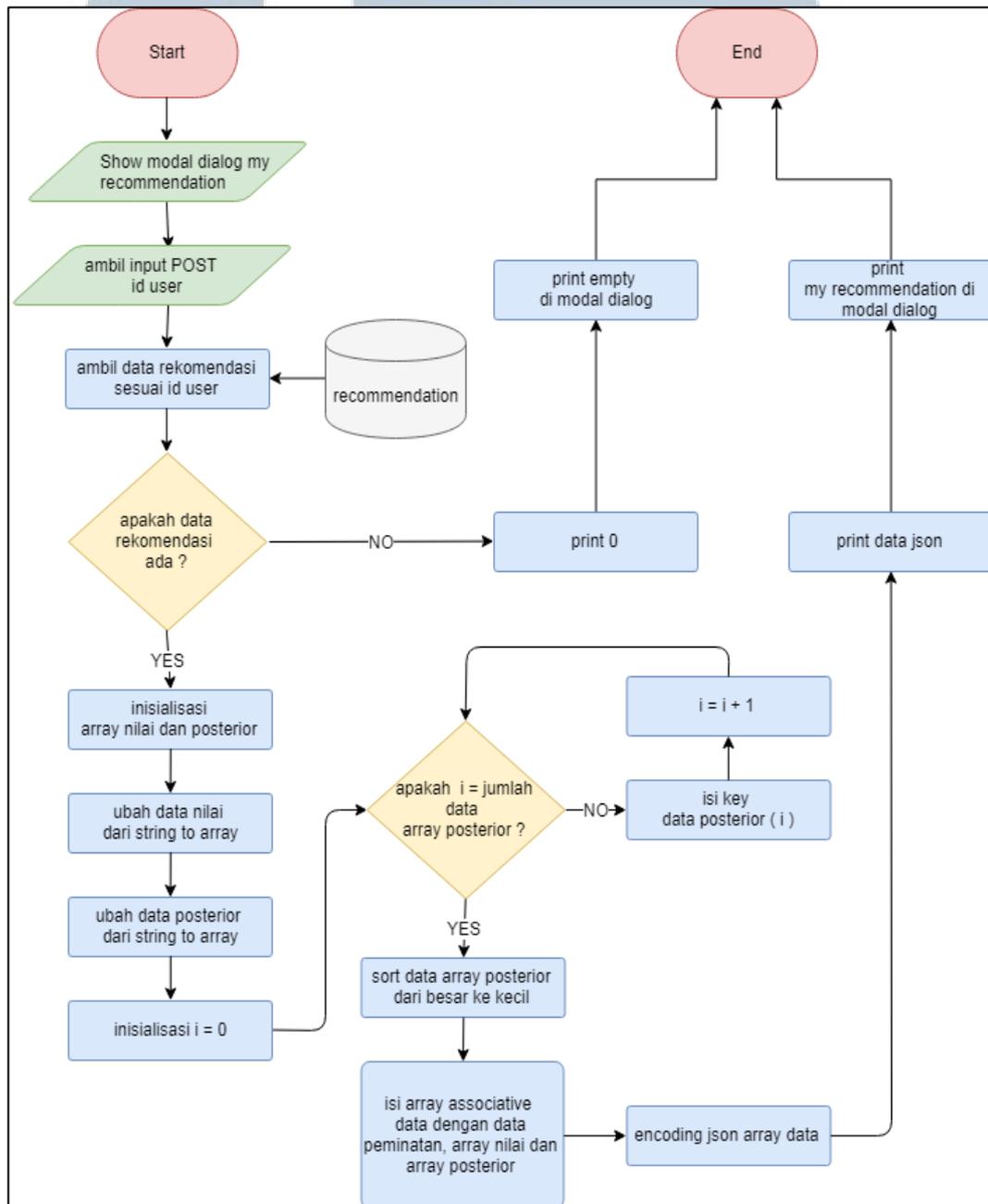
Seluruh nilai probabilitas setiap atribut dikalikan dengan nilai probabilitas *prior* setiap kelas. Setelah mendapatkan nilai probabilitas *posterior* kelas proses

*recommend* mencari nilai tertinggi di *array posterior*, saat nilai probabilitas *posterior* kelas tertinggi sudah didapatkan maka algoritma *naïve bayes classifier* selesai.

Proses *recommend* akan melakukan pengecekan terhadap hasil klasifikasi, jika hasil klasifikasi adalah peminatan *database admin*, maka variabel hasil akan di masukan dengan data *string*, yaitu *database admin* dan variabel *tableName* akan dimasukan dengan data *array* nama tabel, yaitu *dataset\_advance\_database*, *dataset\_database1* dan *dataset\_database2*. Namun jika hasil klasifikasinya adalah peminatan *computer network*, maka variabel hasil akan dimasukan dengan data *string* yaitu *computer network* dan variabel *tableName* akan dimasukan dengan data *array* nama tabel yaitu *dataset\_computer\_network1* dan *dataset\_computer\_network2*. Hasil klasifikasi berupa *game*, akan menghasilkan variabel hasil yang akan dimasukan dengan data *string* yaitu *game* dan variabel *tableName* akan dimasukan dengan data *array* nama tabel yaitu *dataset\_network\_security1*, *dataset\_network\_security2* dan *dataset\_network\_security3*. Hasil klasifikasi berupa *SAP*, membuat variabel hasil akan dimasukan dengan data *string* *SAP* dan variabel *tableName* akan dimasukan dengan data *array* nama tabel yaitu *dataset\_SAP1*, *dataset\_SAP2* dan *dataset\_SAP3*.

Proses *recommend* selanjutnya akan melakukan *passing* parameter *tableName* dan input nilai semester satu sampai dengan empat kepada proses *score recommendation*. Kemudian proses *recommend* akan menerima *return* data dari proses *score recommendation*, yaitu data *score recommendation* akhir sesuai hasil klasifikasi peminatan yang didapatkan. Selanjutnya proses *recommend* akan

melakukan set *session flashdata* yang isinya adalah hasil peminatan, hasil *score recommendation* dan seluruh nilai probabilitas *posterior* kelas untuk ditampilkan kepada *user* dengan *modal dialog*. Setelah proses *recommend* menampilkan di *modal dialog* maka proses *recommend* selesai.



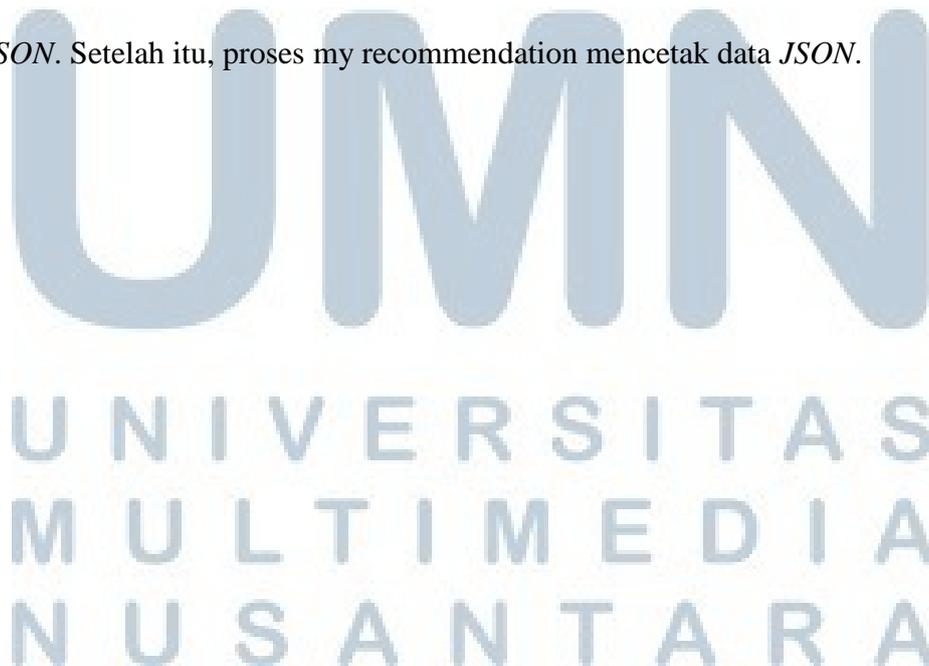
Gambar 3.7 Flowchart My Recommendation Unirecommend

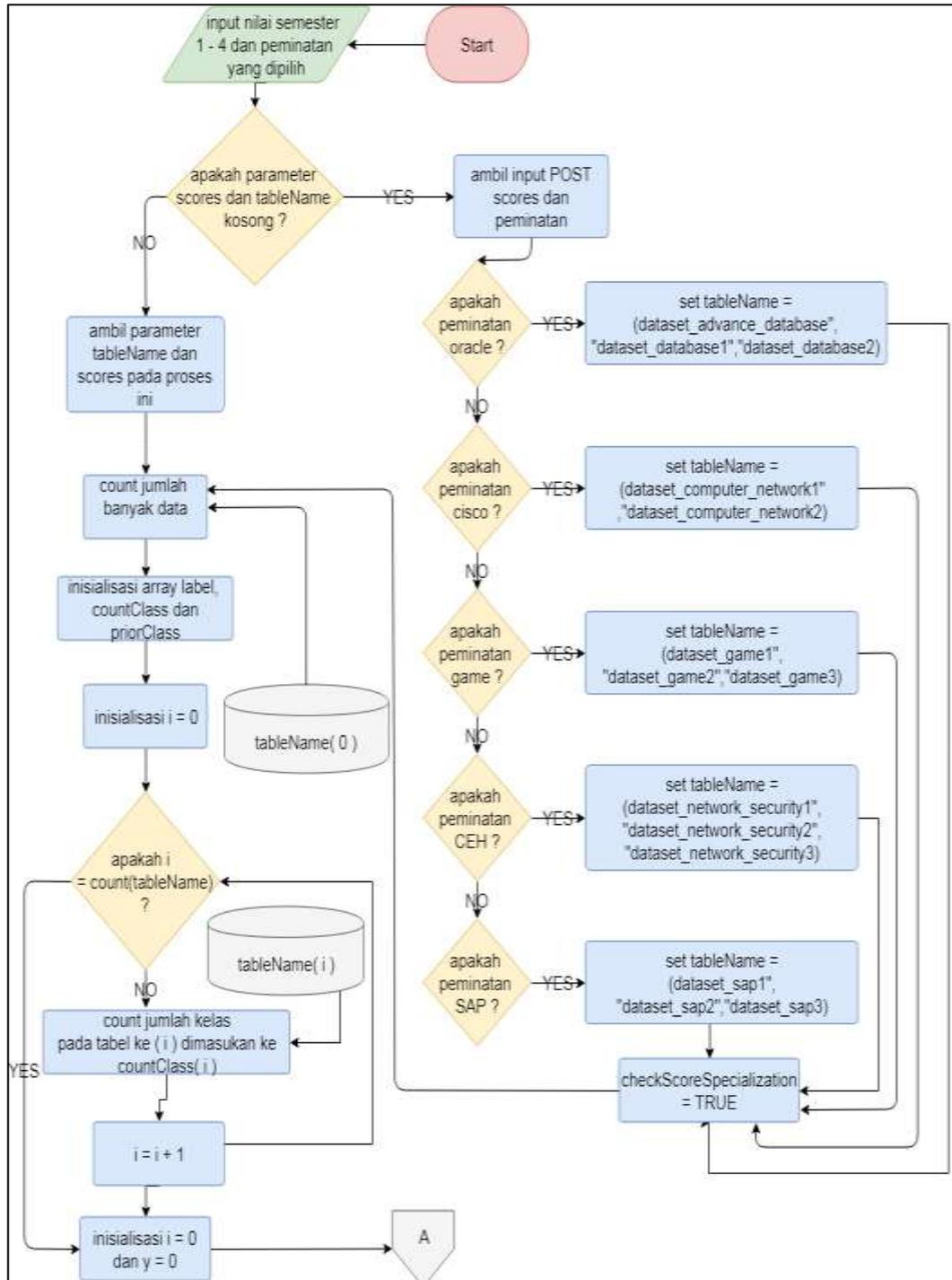
Gambar 3.7 menunjukkan proses *my recommendation*. Proses *my Recommendation* pertama-tama akan menampilkan *modal dialog my*

*recommendation*. Kemudian proses *my recommendation* menerima input *POST* id user dari fungsi *AJAX POST* yang akan digunakan untuk mengambil data rekomendasi sesuai dengan parameter id *user* pada tabel *recommendation*.

Proses *my recommendation* selanjutnya akan melakukan pengecekan apakah data rekomendasinya kosong atau tidak. Data yang kosong akan membuat proses yg *recommendation* melakukan cetak angka 0 lalu menampilkan *empty di modal dialog*. Namun, jika tidak kosong maka proses *my recommendation* akan melakukan inialisasi *array* nilai dan *posterior*. Data nilai dan *posterior* yang diterima dari tabel *recommendation* akan diubah dari tipe data *string* ke *array of string*. Proses *my recommendation* selanjutnya melakukan iterasi untuk mengubah *array posterior* menjadi *array associative posterior*.

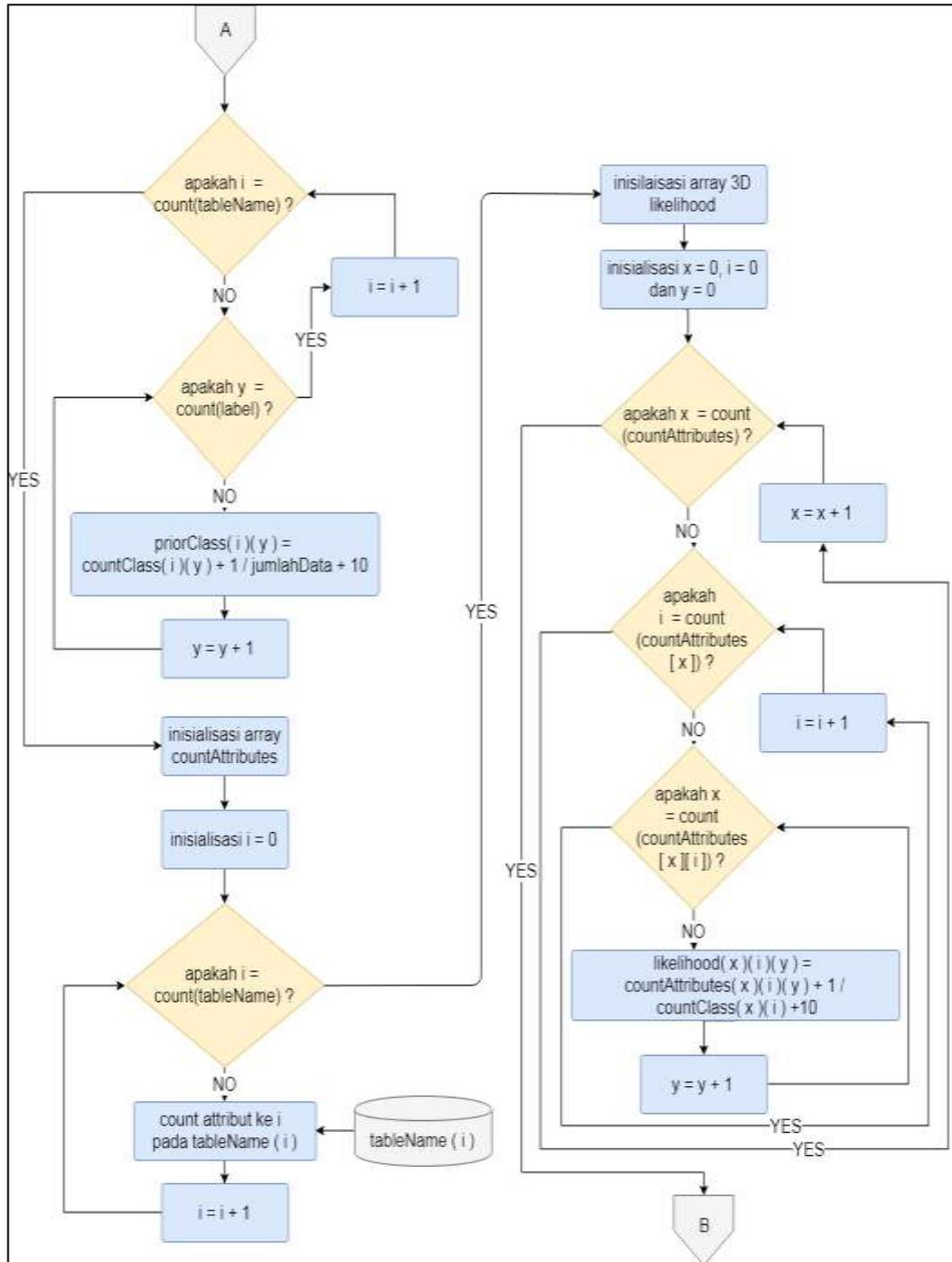
Proses *my recommendation* kemudian melakukan *sorting array associative posterior* dari paling besar ke paling kecil. Setelah itu, proses *my recommendation* mengisi *array associative* data dengan data peminatan, data *array* nilai dan *array associative posterior*. *Array associative* data kemudian diubah ke dalam format *JSON*. Setelah itu, proses *my recommendation* mencetak data *JSON*.





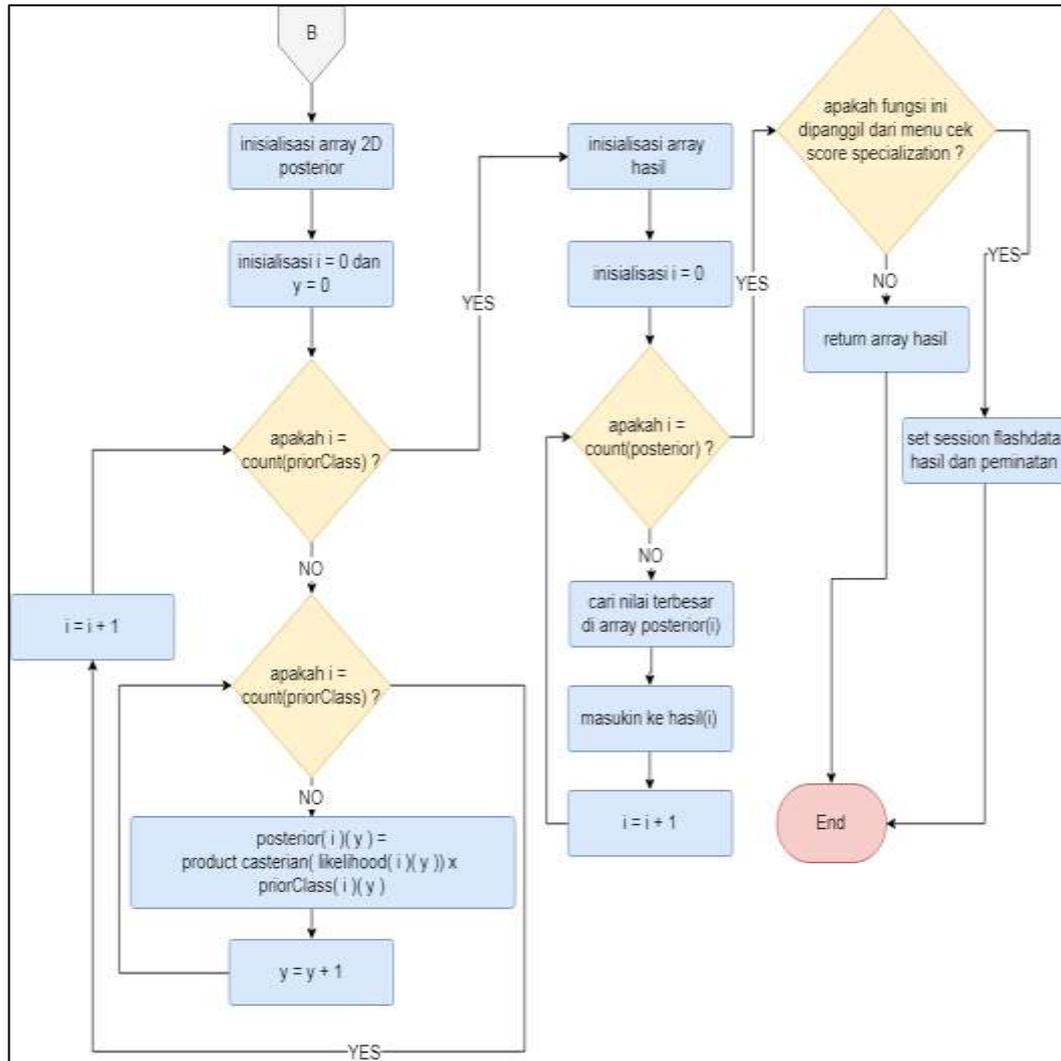
Gambar 3.8 Flowchart Score Recommendation Unirecommend

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 3.9 Flowchart Score Recommendation Unirecommend (Lanjutan)

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 3.10 Flowchart Score Recommendation Unirecommend (Lanjutan)

Gambar 3.8, gambar 3.9 dan gambar 3.10 menunjukkan *flowchart* proses *score recommendation* digambarkan. Proses *score recommendation* menggunakan algoritma *naïve bayes classifier* seperti pada proses *recommend* memiliki perbedaan, yaitu pada proses ini *algoritma naïve bayes classifier* dijalankan sebanyak sekian kali sesuai dengan banyaknya tabel peminatan ke sekian.

Proses *score recommendation* pertama-tama akan meminta *user* untuk melakukan input manual nilai akhir mata kuliah kode IF semester satu sampai dengan empat dan peminatan yang dipilih. Kemudian proses *score recommendation* melakukan pengecekan apakah parameter nilai-nilai dan *tableName* yang di-

*passing* dari proses *recommend* kosong. Proses *score recommendation* akan melakukan pengambilan data input *POST* nilai-nilai dan peminatan (data inputan user). Parameter nilai-nilai dan *tableName* yang kosong akan melakukan set variabel *tableName* dengan data *array* nama-nama tabel peminatan yang sesuai inputan user, dan melakukan set variabel *checkScoreSpecialization* dengan data *boolean* berisi *true* yang bertujuan untuk memberi tanda bahwa proses *score recommendation* dipanggil oleh *user* bukan dari proses *recommend*. Namun, jika tidak kosong maka proses *score recommendation* akan mengambil parameter *tableName* dan nilai-nilai yang di *passing* dari proses *recommend*.

Algoritma *naïve bayes classifier* pada proses ini mulai berjalan pada saat perhitungan banyaknya data pada tabel peminatan pertama (*tableName*). Proses *score recommendation* melakukan inisialisasi *array* label (kelas) dengan isi variasi nilai huruf yang ada di Universitas Multimedia Nusantara, *countClass* dan *priorClass*. Setelah itu proses *score recommendation* akan melakukan iterasi untuk perhitungan jumlah frekuensi kemunculan setiap kelas (*countClass*) pada setiap tabel peminatan. Proses *score recommendation* melakukan iterasi *array* dua dimensi untuk perhitungan nilai probabilitas *prior* setiap kelas pada setiap tabel peminatan, lalu melakukan iterasi *array* tiga dimensi untuk perhitungan nilai probabilitas setiap atribut terhadap setiap kelas pada setiap tabel peminatan. Selanjutnya proses *score recommendation* melakukan iterasi *array* dua dimensi untuk perhitungan nilai probabilitas *posterior* dengan melakukan operasi *casterian product* nilai probabilitas atribut kemudian dikalikan dengan nilai probabilitas *prior* kelas. Proses *score recommendation* selanjutnya melakukan iterasi untuk mencari nilai probabilitas *posterior* terbesar dari seluruh kelas pada setiap tabel

peminatan(*tableName*) dan melakukan pengecekan apakah proses ini dipanggil oleh user atau dari *proses recommend*. Hasilnya, jika dipanggil oleh user maka proses *score recommendation* akan melakukan set variabel *session flashdata*.

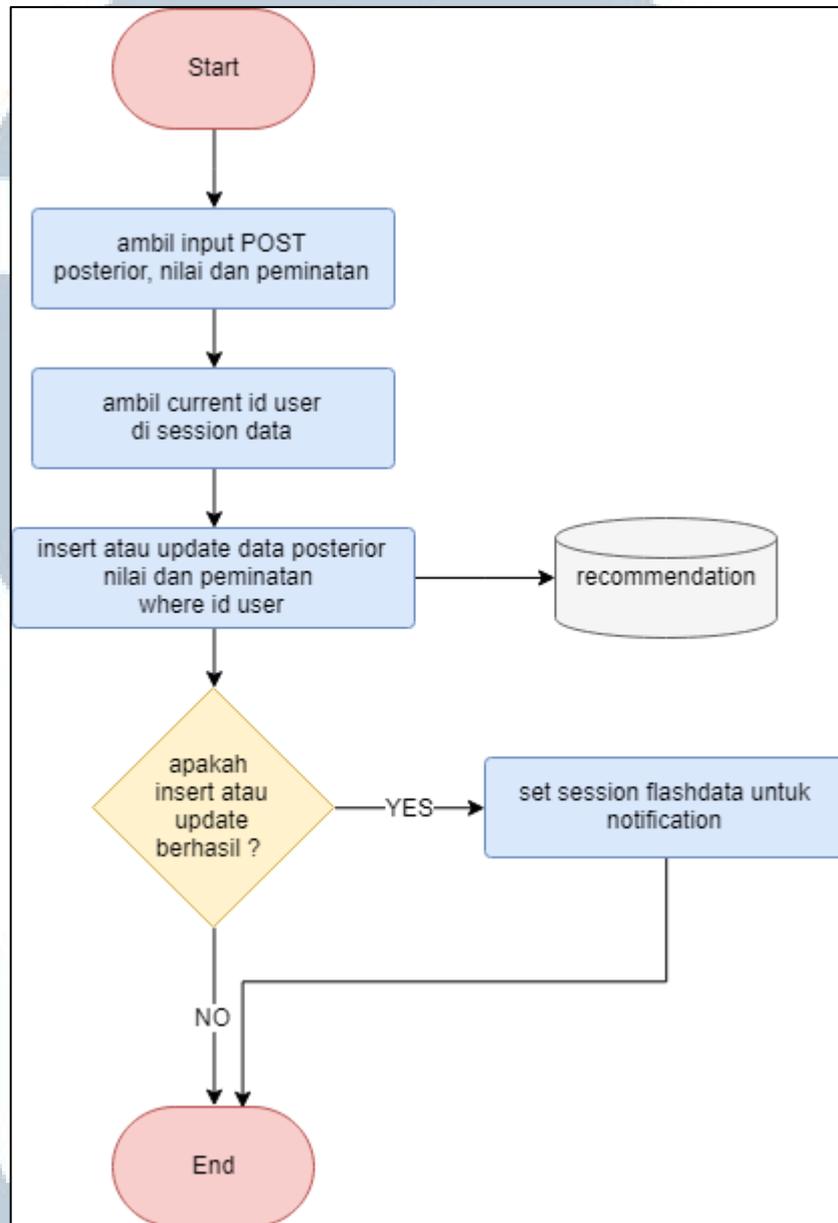
Hasil klasifikasinya berupa nilai-nilai akhir peminatan dan peminatan yang dipilih oleh *user* saat awal proses. Namun jika dipanggil oleh proses *recommend*, maka proses *score recommendation* akan mengirim data *return*, yaitu hasil klasifikasi nilai-nilai akhir peminatan. Tahap ini merupakan tahap terakhir dari proses *score recommendation*.



Gambar 3.11 Flowchart logout Unirecommend

Gambar 3.11 menunjukkan *flowchart logout*. Proses *logout* pertama-tama akan menghapus seluruh data *user* yang ada di dalam *session data login*, setelah itu

proses *logout* akan melakukan *destroy session* dan mengeluarkan *modal dialog login*.



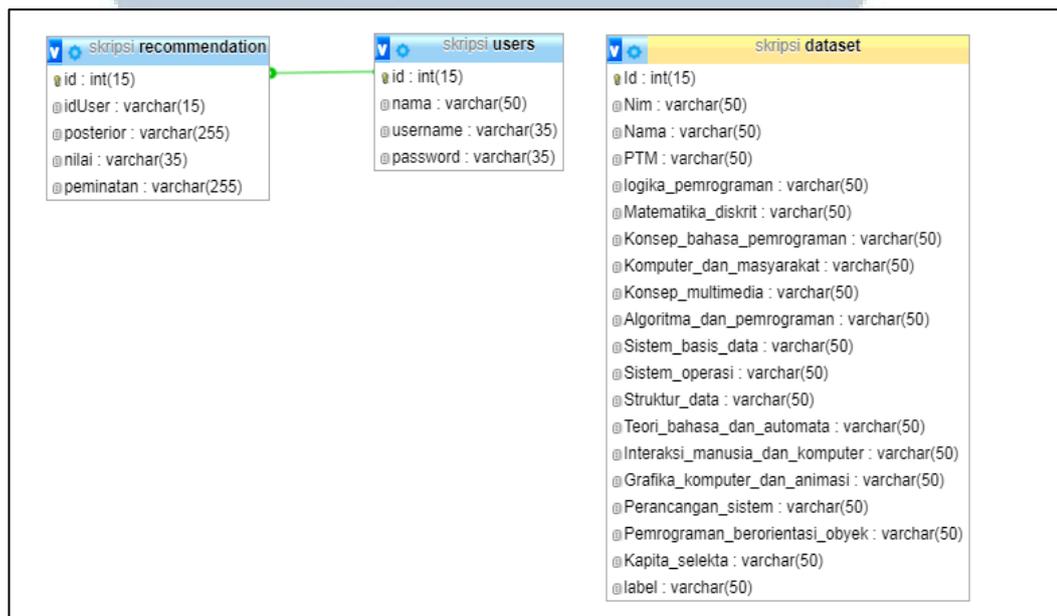
Gambar 3.12 Flowchart Save Recommendation Unirecommend

Gambar 3.12 menunjukkan *flowchart* proses *save recommendation* digambarkan. Proses *save recommendation* pertama-tama mengambil input *POST posterior*, nilai dan peminatan. Kemudian proses *save recommendation* mengambil data *id user* di dalam data *session* yang akan digunakan untuk melakukan *insert* data *posterior*, data nilai dan peminatan jika *user* belum mempunyai data

rekomendasi, melakukan *update* data *posterior*, data nilai dan peminatan jika *user* telah mempunyai data rekomendasi ke tabel *recommendation*. Setelah itu, proses *save recommendation* melakukan pengecekan terhadap keberhasilan operasi *insert* atau *update*. Proses *save recommendation* akan melakukan *set sessionflashdata* untuk notifikasi jika operasi *insert* atau *update* berhasil. Operasi *insert* atau *update* yang gagal akan membuat proses *save recommendation* selesai.

### 3.2.3 Database Schema

Pada bagian ini akan memperlihatkan *database schema* yang berfungsi untuk menjelaskan hubungan antar data dalam basis data.

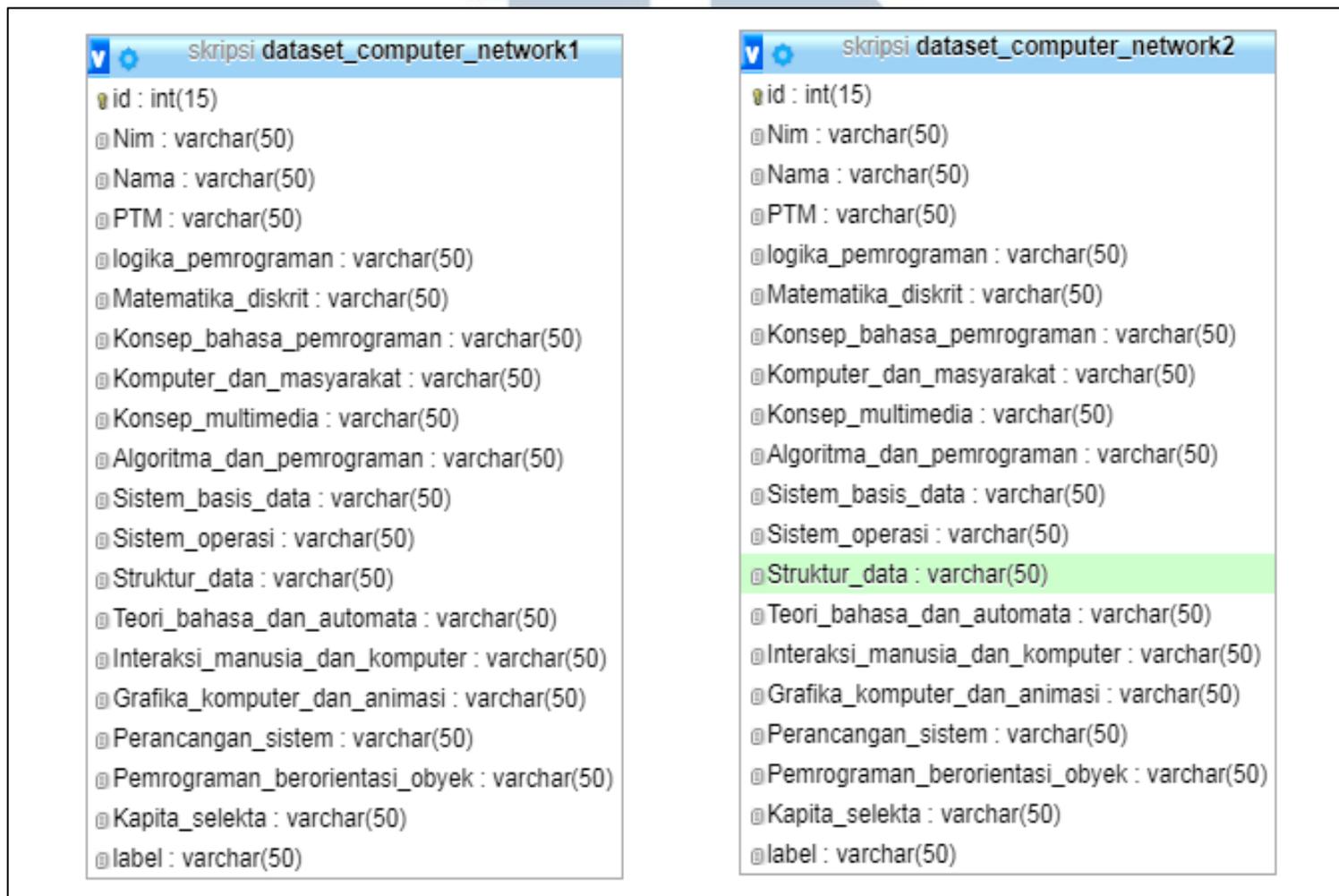


Gambar 3.13 Database Schema Unirecommend

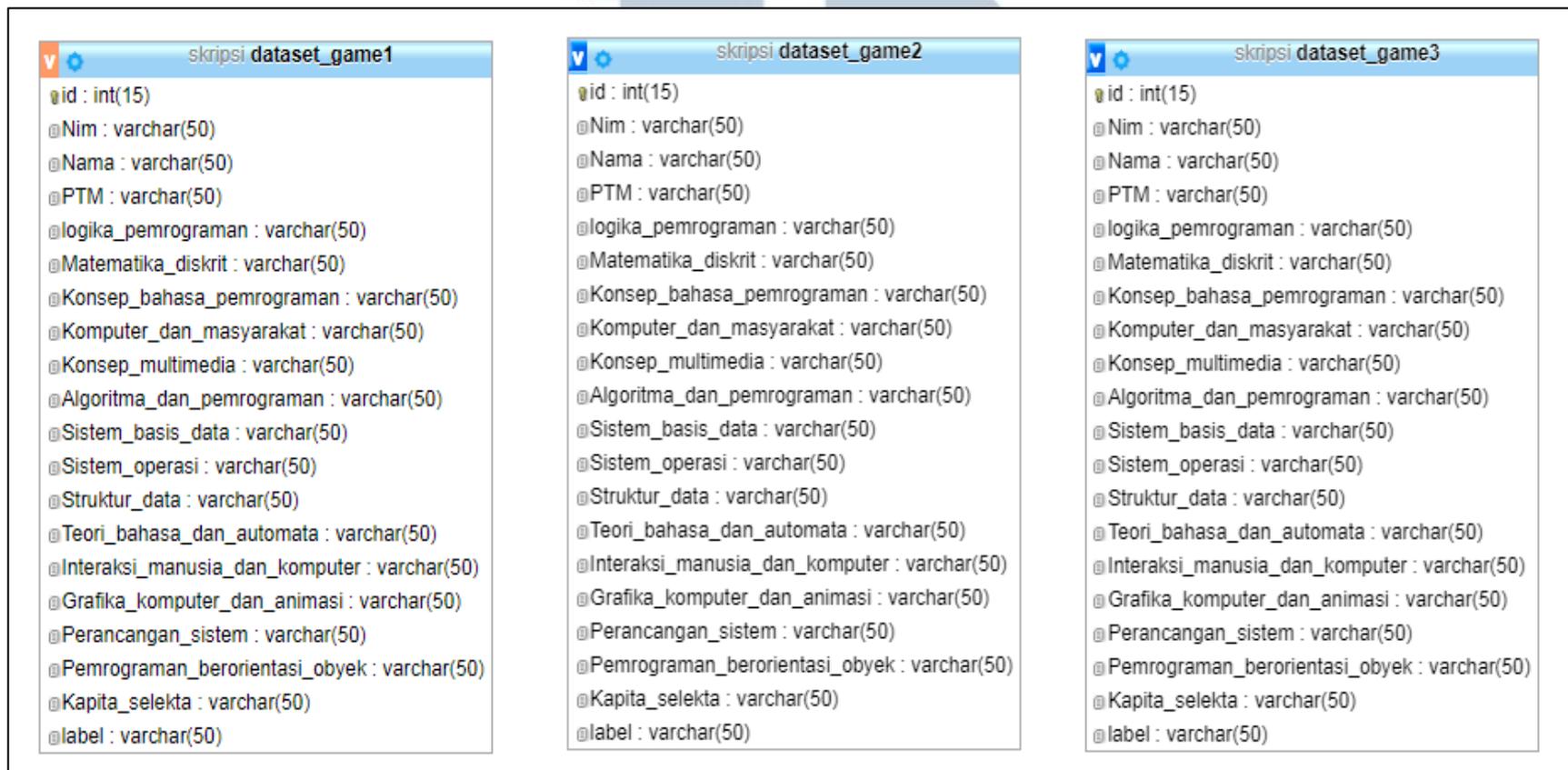
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

skripsi dataset_advance_database	skripsi dataset_database2	skripsi dataset_database1
id : int(15)	id : int(15)	id : int(15)
Nim : varchar(50)	Nim : varchar(50)	Nim : varchar(50)
Nama : varchar(50)	Nama : varchar(50)	Nama : varchar(50)
PTM : varchar(50)	PTM : varchar(50)	PTM : varchar(50)
logika_pemrograman : varchar(50)	logika_pemrograman : varchar(50)	logika_pemrograman : varchar(50)
Matematika_diskrit : varchar(50)	Matematika_diskrit : varchar(50)	Matematika_diskrit : varchar(50)
Konsep_bahasa_pemrograman : varchar(50)	Konsep_bahasa_pemrograman : varchar(50)	Konsep_bahasa_pemrograman : varchar(50)
Komputer_dan_masyarakat : varchar(50)	Komputer_dan_masyarakat : varchar(50)	Komputer_dan_masyarakat : varchar(50)
Konsep_multimedia : varchar(50)	Konsep_multimedia : varchar(50)	Konsep_multimedia : varchar(50)
Algoritma_dan_pemrograman : varchar(50)	Algoritma_dan_pemrograman : varchar(50)	Algoritma_dan_pemrograman : varchar(50)
Sistem_basis_data : varchar(50)	Sistem_basis_data : varchar(50)	Sistem_basis_data : varchar(50)
Sistem_operasi : varchar(50)	Sistem_operasi : varchar(50)	Sistem_operasi : varchar(50)
Struktur_data : varchar(50)	Struktur_data : varchar(50)	Struktur_data : varchar(50)
Teori_bahasa_dan_automata : varchar(50)	Teori_bahasa_dan_automata : varchar(50)	Teori_bahasa_dan_automata : varchar(50)
Interaksi_manusia_dan_komputer : varchar(50)	Interaksi_manusia_dan_komputer : varchar(50)	Interaksi_manusia_dan_komputer : varchar(50)
Grafika_komputer_dan_animasi : varchar(50)	Grafika_komputer_dan_animasi : varchar(50)	Grafika_komputer_dan_animasi : varchar(50)
Perancangan_sistem : varchar(50)	Perancangan_sistem : varchar(50)	Perancangan_sistem : varchar(50)
Pemrograman_berorientasi_obyek : varchar(50)	Pemrograman_berorientasi_obyek : varchar(50)	Pemrograman_berorientasi_obyek : varchar(50)
Kapita_selekta : varchar(50)	Kapita_selekta : varchar(50)	Kapita_selekta : varchar(50)
label : varchar(50)	label : varchar(50)	label : varchar(50)

Gambar 3.14 Database Schema Unirecommend (Lanjutan)



Gambar 3.15 Database Schema Unirecommend (Lanjutan)



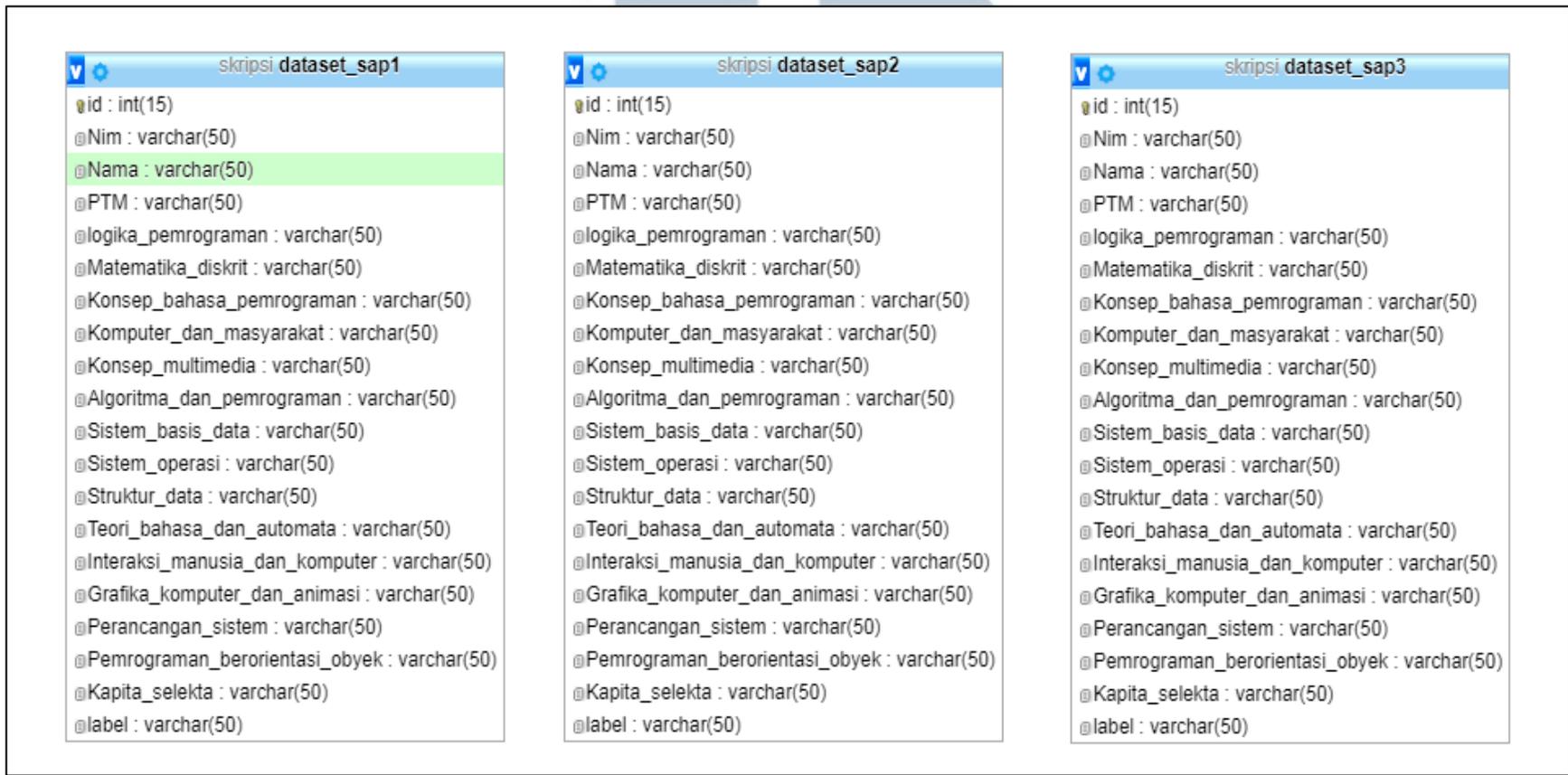
Gambar 3.16 Database Schema Unirecommend (Lanjutan)

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 3.17 Database Schema Unirecommend (Lanjutan)

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 3.18 Database Schema Unirecommend (Lanjutan)



Gambar 3.13, gambar 3.14, gambar 3.15, gambar 3.16, gambar 3.17 dan gambar 3.18 menunjukkan *database schema* pada sistem *unirecommend* terdapat 2 tabel yang saling berelasi yaitu tabel *users* dan tabel *recommendation* dimana tabel *recommendation* mempunyai *foreign key* yang *referenced key*nya ada pada tabel *users*. *Database* sistem pendukung keputusan pemilihan peminatan memiliki 17 tabel dimana 15 tabel berguna sebagai *knowledge base* yaitu tabel *dataset*, tabel *dataset\_advance\_database*, tabel *dataset\_database1*, tabel *dataset\_database2*, tabel *dataset\_computer\_network1*, tabel *dataset\_computer\_network2*, tabel *dataset\_game\_1*, tabel *dataset\_game2*, tabel *dataset\_game3*, tabel *dataset\_network\_security1*, tabel *dataset\_network\_security2*, tabel *dataset\_network\_security3*, tabel *dataset\_SAP1*, tabel *dataset\_SAP2* dan tabel *dataset\_SAP3* dan dua tabel sisa berguna sebagai informasi pribadi *user* yaitu tabel *users* dan tabel *recommendation*.

### 3.2.4 Struktur Tabel

Perancangan struktur tabel digunakan untuk menjelaskan tabel yang digunakan dalam proses membangun sebuah sistem. Struktur tabel yang ada pada sistem *unirecommend* yaitu tabel *users*, tabel *recommendation* dan tabel *dataset*. Struktur tabel sistem *unirecommend* hanya ada tiga dikarenakan struktur 14 tabel yang digunakan sebagai *knowledge base* yaitu tabel *dataset\_advance\_database*, tabel *dataset\_database1*, tabel *dataset\_database2*, tabel *dataset\_computer\_network1*, tabel *dataset\_computer\_network2*, tabel *dataset\_game\_1*, tabel *dataset\_game2*, tabel *dataset\_game3*, tabel *dataset\_network\_security1*, tabel *dataset\_network\_security2*, tabel

*dataset\_network\_security3*, tabel *dataset\_SAP1*, tabel *dataset\_SAP2* dan tabel *dataset\_SAP3* mempunyai struktur tabel yang sama dengan tabel *dataset*.

#### A. Tabel Users

Tabel 3.1 menunjukkan struktur tabel *users*. Tabel ini menyimpan data informasi *user*. Kolom *id* digunakan untuk sebagai *primary key* setiap baris data informasi *user*. Kolom *nama* digunakan untuk menyimpan nama *user* mahasiswa. Kolom *username* digunakan untuk menyimpan *username* yang unik untuk melakukan *login*. Kolom *password* digunakan untuk menyimpan kata sandi *user* dimana kolom *password* menggunakan fungsi *hashing*(MD5).

Tabel 3.1 Struktur Tabel Users

Column	Type	Keterangan
id	int(15)	id (auto increment)
nama	varchar(50)	berisikan nama user
username	varchar(35)	username user
password	varchar(35)	kata sandi user

Tabel 3.2 menunjukkan struktur tabel *recommend*. Tabel *recommend* menyimpan data informasi hasil rekomendasi *user*. Kolom *id* digunakan sebagai *primary key* setiap baris data informasi hasil rekomendasi. Kolom *id user* digunakan sebagai *foreign key* dimana *referenced key* dari tabel *user* pada kolom *id*. Kolom *posterior* digunakan untuk menyimpan nilai seluruh probabilitas posterior kelas dari hasil klasifikasi. Kolom *nilai* digunakan untuk menyimpan data nilai-nilai akhir peminatan yang direkomendasikan ke *user*. Kolom *peminatan* digunakan untuk menyimpan hasil rekomendasi peminatan oleh sistem ke *user*.

Tabel 3.2 Struktur Tabel recommendation

Column	Type	Keterangan
id	int(15)	id (auto increment)
idUser	varchar(15)	foreign key pada tabel users
posterior	varchar(255)	nilai probabilitas posterior

Tabel 3.3 Struktur Tabel recommendation (Lanjutan)

Column	Type	Keterangan
nilai	varchar(35)	hasil prediksi nilai user
peminatan	varchar(255)	hasil klasifikasi peminatan user

Tabel 3.3 menunjukkan struktur tabel *dataset*. Tabel *dataset* digunakan untuk menyimpan data nilai-nilai akhir mata kuliah mahasiswa 2014 yang berkode IF dari semester satu sampai dengan empat. Tabel *dataset* digunakan untuk menyimpan *knowledge base* pada sistem. Kolom id digunakan untuk sebagai *primary key* setiap baris data. Kolom nim digunakan untuk menyimpan nomor induk mahasiswa. Kolom nama digunakan untuk menyimpan nama mahasiswa. Kolom PTM, logika\_pemrograman, matematika\_diskrit, konsep\_bahasa\_pemrograman, komputer\_dan\_masyarakat, konsep\_multimedia, algoritma\_dan\_pemrograman, sistem\_basis\_data, sistem\_operasi, struktur\_data, teori\_bahasa\_dan\_automata, interaksi\_manusia\_dan\_komputer, grafika\_komputer\_dan\_animasi, perancangan\_sistem, pemrograman\_berorientasi\_obyek dan kapita\_selekta digunakan untuk menyimpan nilai-nilai akhir dimana nilai-nilai akhir tersebut berguna sebagai *knowledge base*(atribut). Kolom label digunakan untuk menyimpan data kelas atau label dari kolom yang berisikan nilai-nilai akhir.

Tabel 3 4 Struktur Tabel Dataset

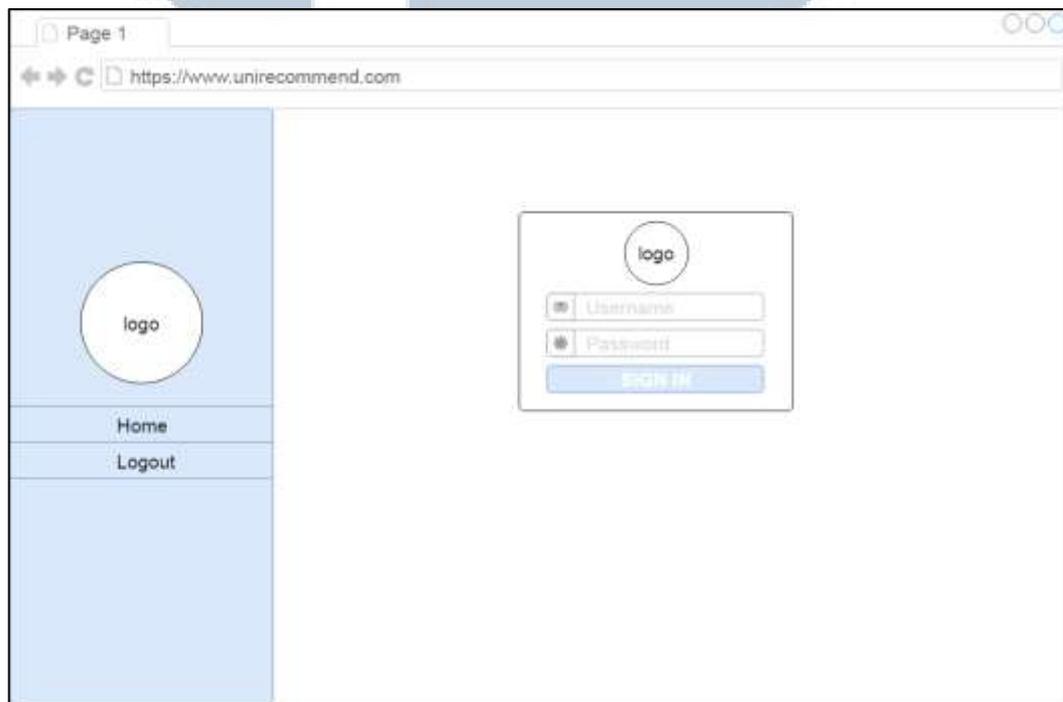
Column	Type	keterangan
id	int(15)	id (auto increment)
nim	varchar(50)	berisi no induk mahasiswa
nama	varchar(50)	berisi nama mahasiswa
PTM	varchar(50)	berisi nilai akhir huruf
logika_pemrograman	varchar(50)	berisi nilai akhir huruf
matematika_diskrit	varchar(50)	berisi nilai akhir huruf
konsep_bahasa_pemrograman	varchar(50)	berisi nilai akhir huruf
komputer_dan_masyarakat	varchar(50)	berisi nilai akhir huruf
konsep_multimedia	varchar(50)	berisi nilai akhir huruf
algoritma_dan_pemrograman	varchar(50)	berisi nilai akhir huruf
sistem_basis_data	varchar(50)	berisi nilai akhir huruf

Tabel 3.5 Struktur Tabel Dataset (Lanjutan)

Column	Type	Link to
sistem_operasi	varchar(50)	berisi nilai akhir huruf
struktur_data	varchar(50)	berisi nilai akhir huruf
teori_bahasa_dan_automata	varchar(50)	berisi nilai akhir huruf
interaksi_manusia_dan_komputer	varchar(50)	berisi nilai akhir huruf
grafika_komputer_dan_animasi	varchar(50)	berisi nilai akhir huruf
perancangan_sistem	varchar(50)	berisi nilai akhir huruf
pemrograman_berorientasi_obyek	varchar(50)	berisi nilai akhir huruf
kapital_selekt	varchar(50)	berisi nilai akhir huruf
label	varchar(50)	berisi label / kelas tujuan

### 3.2.5 Perancangan Antarmuka

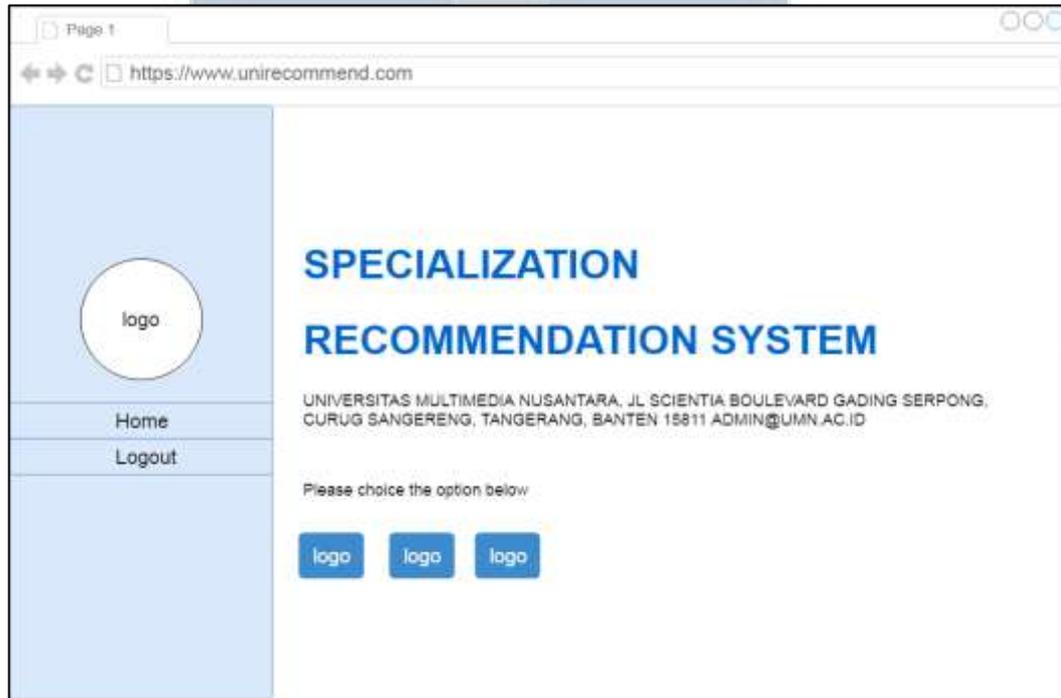
Perancangan antarmuka dapat membantu dalam proses pembangunan sistem. Perancangan antarmuka dapat memberikan gambaran mengenai antarmuka atau *interface* dari sistem yang dibangun.



Gambar 3.19 Rancangan Antarmuka Login

Gambar 3.19 menunjukkan rancangan antarmuka login untuk *user*. Antarmuka *login* terdiri dari *modal dialog login* dan *sidebar*. *Modal dialog login* terdapat *form* yang terdiri dari logo, input tipe *text* untuk melakukan input data

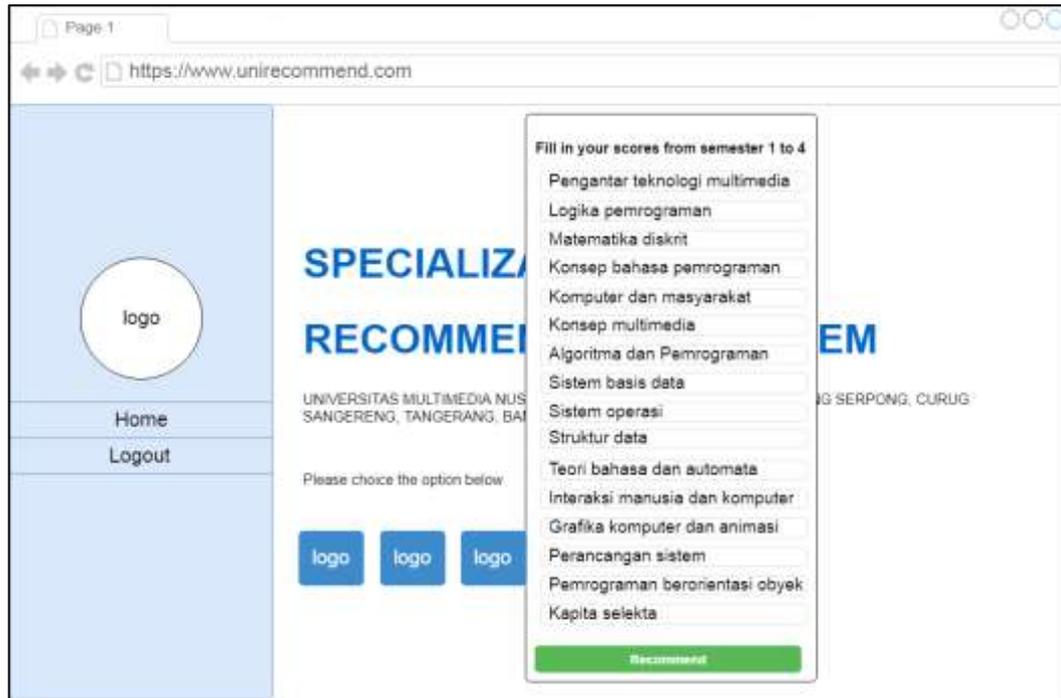
*username*, input tipe *password* untuk melakukan input data *password* dan tombol *sign in*. *Sidebar* terdiri dari logo sistem, menu *home* dan menu *logout*.



Gambar 3.20 Rancangan Antarmuka Home

Gambar 3.20 menunjukkan rancangan antarmuka pada *home* sistem. Antarmuka *home* terdiri dari *sidebar*, *container* dan *header*. *Sidebar* terdiri dari menu *home* dan *logout*. *Header* berisi tulisan tentang informasi sistem. *Container* berisi tiga tombol yaitu tombol *recommend*, tombol *my recommendation* dan tombol *score recommendation*.

UMMN  
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



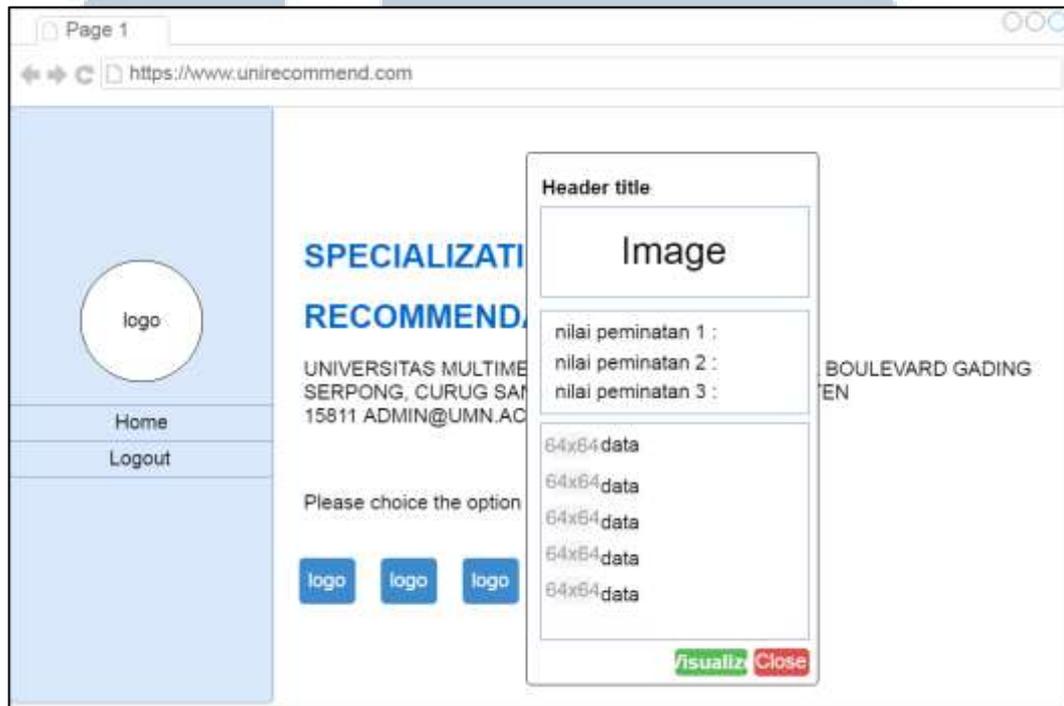
Gambar 3.21 Rancangan Antarmuka Modal dialog Recommend

Gambar 3.21 menunjukkan rancangan antarmuka *modal dialog recommend* yang terletak pada *Home* sistem. Antarmuka *modal dialog recommend* terdiri dari *form* yang berisikan 16 input *textbox* tipe *text* dan tombol *recommend*.



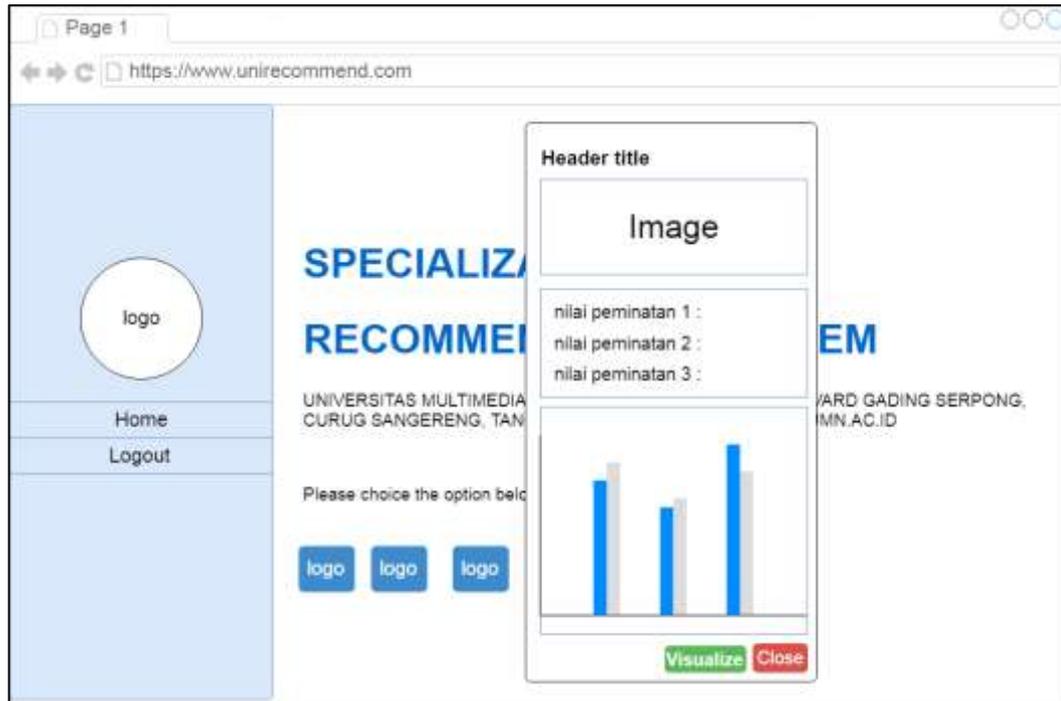
Gambar 3.22 Rancangan Antarmuka Modal dialog Save Recommendation

Gambar 3.22 menunjukkan rancangan antarmuka *modal dialog save recommendation* yang terletak pada *Home* sistem. Antarmuka *modal dialog save recommendation* terdiri dari *header title*, *image*, *list* nilai-nilai akhir peminatan, tombol *save* dan tombol *close modal dialog*.



Gambar 3.23 Rancangan Antarmuka Modal Dialog My Recommendation

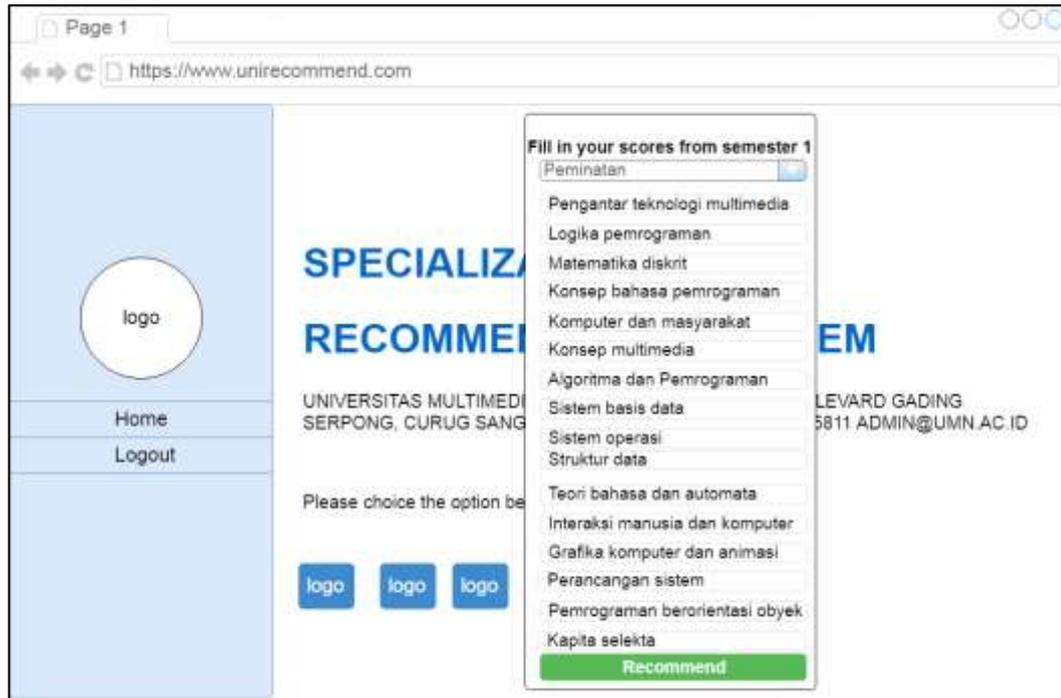
UMMN  
 UNIVERSITAS  
 MULTIMEDIA  
 NUSANTARA



Gambar 3.24 Rancangan Antarmuka Modal Dialog My Recommendation (Lanjutan)

Gambar 3.23 dan gambar 3.24 menunjukkan rancangan antarmuka *modal dialog my recommendation* yang terletak pada *home* sistem. Antarmuka *modal dialog my recommendation* terdiri dari *header title*, *image*, *list* nilai akhir peminatan, *list* urutan nilai presentase kecocokan mahasiswa di setiap peminatan, grafik nilai presentase kecocokan mahasiswa di setiap peminatan, tombol *visualize* dan tombol *close modal dialog*. Tombol *Visualize* berfungsi sebagai *toggle* antara pergantian tampilan dari *list* urutan nilai peluang posterior ke grafik nilai peluang posterior atau sebaliknya.

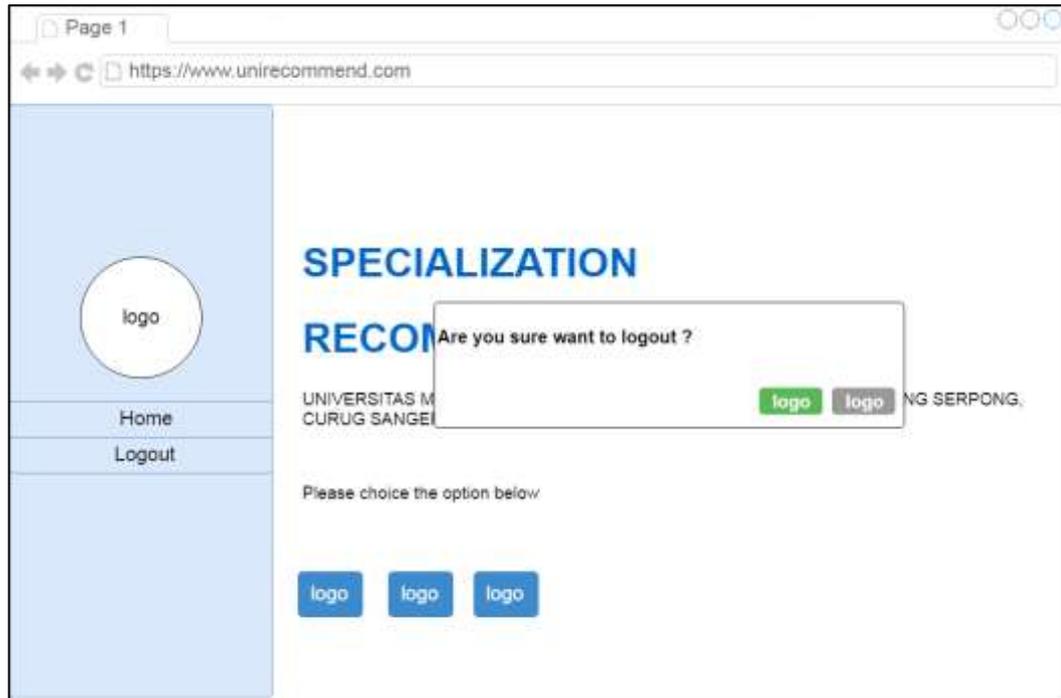
U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A



Gambar 3 25 Rancangan Antarmuka Modal Dialog Score Recommendation

Gambar 3.25 menunjukkan rancangan antarmuka *modal dialog score recommendation* yang terletak pada *home* sistem. Antarmuka *modal dialog score recommendation* terdiri dari *form* yang berisikan 16 input *textbox* tipe *text*, input *select dropdown* dan tombol *recommend*.

UMN  
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 3.26 Rancangan Antarmuka Modal Dialog Logout

Gambar 3.26 menunjukkan rancangan antarmuka *modal dialog logout* yang terletak pada *home* sistem. Antarmuka *modal dialog logout* terdiri dari *header title*, tombol *logout* dan tombol *close modal dialog*.

UMN  
 UNIVERSITAS  
 MULTIMEDIA  
 NUSANTARA