

PROJECT MOBILE APPLICATION DEVELOPMENT

**Pengembangan Aplikasi Seluler Bimbingan Belajar “BIMSI”
untuk Program Studi Sistem Informasi
di Universitas Multimedia Nusantara**



Disusun Oleh :

Amelia Rahmanita	(00000044296)
Bill Kiki	(00000043854)
Meilani Yaputri	(00000045878)
Mettadewi	(00000045973)
Reynardthan Handoko	(00000044437)

**Program Studi Sistem Informasi
Fakultas Teknik dan Informatika
UNIVERSITAS MULTIMEDIA NUSANTARA
TANGERANG
2021**

KATA PENGANTAR

Puji syukur kami ucapkan kepada Tuhan Yang Maha Esa karena atas berkat dan karunianya kami dapat menyelesaikan laporan yang berjudul “Pengembangan Aplikasi Seluler Bimbingan Belajar “BIMSI” untuk Program Studi Sistem Informasi di Universitas Multimedia Nusantara” ini tepat pada waktunya. Adapun tujuan dari penulisan laporan ini adalah untuk memenuhi tugas akhir pada mata kuliah “Mobile Application Development”. Selain itu, laporan ini juga bertujuan untuk menambah wawasan dalam merancang sebuah aplikasi seluler untuk bimbingan belajar khususnya secara teori maupun teknis. Dalam proses perancangan dan pengembangan aplikasi seluler ini, alat atau *software* yang kami gunakan adalah Android Studio.

Laporan ini tidak dapat terselesaikan tanpa dukungan dari berbagai pihak. Oleh karena itu, kami ingin menyampaikan ucapan terima kasih kepada semua pihak yang telah membantu dalam penyusunan laporan ini terutama kepada:

1. Segenap keluarga kami, terutama orang tua, yang senantiasa memberikan kasih sayang dan dukungannya kepada kami,
2. Bapak Agus Sulaiman, S.Kom., M.M selaku dosen kami yang telah membimbing dan memberikan arahan untuk membantu kami dalam penyusunan laporan ini,
3. Rekan-rekan mahasiswa jurusan Sistem Informasi yang senantiasa berjuang bersama dan bertukar semangat,
4. Serta kepada semua pihak yang telah memberikan motivasi dan bantuan yang tidak dapat kami sebutkan satu per satu.

Adapun, kami menerima masukan dari para pembaca mengenai kritik dan saran yang dapat membangun laporan ini. Semoga laporan ini dapat bermanfaat bagi para pembaca dan semua pihak khususnya dalam hal perancangan dan pengembangan aplikasi seluler untuk bimbingan belajar.

Tangerang, 25 November 2021

Tim Penulis

DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI	ii
DAFTAR GAMBAR	v
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Gambaran Umum Organisasi	2
1.3 Rumusan Masalah	3
1.4 Tujuan Penelitian	3
BAB II ISI	4
2.1 Proses Bisnis	4
2.2 Entity Relationship Diagram	6
2.3 <i>Layout</i> Aplikasi bagi Pengguna	9
2.3.1 Layer <i>Splash Screen</i>	9
2.3.2 Layer <i>Login</i>	10
2.3.3 Layer <i>Sign Up</i>	11
2.3.4 Layer <i>Home</i>	12
2.3.5 Layer <i>Student Notes</i>	15
2.3.6 Layer <i>Forum</i>	17
2.3.7 Layer <i>Information</i>	20
2.3.8 Layer <i>BIMSI Reservation</i>	21
2.3.9 Layer <i>Tutor BIMSI</i>	24
2.3.10 Layer <i>Reservation Page</i>	26
2.4 <i>Layout</i> Aplikasi bagi Admin	27
2.4.1 Layer <i>Tutor BIMSI</i>	27
2.4.2 Layer <i>Update Tutor</i>	29
2.4.3 Layer <i>Invoice</i>	30
2.5 <i>Coding</i> Aplikasi	31
2.5.1 Android Studio	31
1. Splash screen	31
2. Login	32
3. Sign Up	34

4. Main Activity	35
5. Button menu utama : Home, Forum, Information, Reservation	37
6. View Pager Elearning	38
7. View Pager MyUMN	39
8. Mata Kuliah Data Analysis	39
9. Mata Kuliah Accounting	40
10. Mata Kuliah Database	41
11. Mata Kuliah Algorithm	41
12. Mata Kuliah Mobile App	42
13. dbHandlerNote	43
14. Edit Notes	45
15. StudentNote	46
16. StudentNotes	47
17. Tambah Student Note	49
18. Forum	50
19. Tambah Forum	52
20. Edit Forum	54
21. Informasi	57
22. CustomMaps	59
23. Reservation	60
24. Detail	63
25. Form Pembelian	64
26. Read Admin	66
27. Update Admin	69
28. Admin Order	72
29. Order	73
30. Read	75
31. Tutor	76
32. Order Adapter	77
33. Read Adapter	78
34. Tutor Adapter	80
2.5.6 File PHP	82
1. conn.php	82

2. db_connect.php	82
3. DataBase.php	83
4. LogIn.php	84
5. SignUp.php	85
6. getForum.php	85
7. getForumById.php	86
8. addForum.php	87
9. updateForum.php	88
10. deleteForum.php	88
11. tutor.php	89
12. pemesanan.php	90
13. bacaAdmin.php	91
14. updateTutorStok.php	91
15. lihatOrder.php	92
16. deleteTutor.php	93
BAB III PENUTUP	94
3.1 Kesimpulan	94
3.2 Peran Anggota	95
3.3 Link Video Presentasi Project	96

DAFTAR GAMBAR

Gambar 2.1 Proses Bisnis BIMSI	4
Gambar 2.2 Entity Relationship Diagram BIMSI.....	6
Gambar 2.3 Layer Splash Screen.....	9
Gambar 2.4 Layer Log In.....	10
Gambar 2.5 Layer Sign Up	11
Gambar 2.6 Layer Home.....	13
Gambar 2.7 Layer View Pager.....	14
Gambar 2.8 Layer Student Notes.....	16
Gambar 2.9 Layer Forum.....	18
Gambar 2.10 Layer Information	20
Gambar 2.11 Layer BIMSI Reservation	22
Gambar 2.12 Layer Tutor BIMSI	24
Gambar 2.13 Layer Reservation Page	26
Gambar 2.14 Layer Tutor BIMSI Untuk Admin	27
Gambar 2.15 Layer Update Tutor Untuk Admin.....	29
Gambar 2.16 Layer Invoice Untuk Admin	30
Gambar 2.17 Code Splash Screen.....	31
Gambar 2.18 Code Log In	33
Gambar 2.19 Code Sign Up.....	34
Gambar 2.20 Code Main Activity.....	36
Gambar 2.21 Code Button Menu Utama	37
Gambar 2.22 Code View Pager E-learning.....	38
Gambar 2.23 Code View Pager My UMN.....	39
Gambar 2.24 Code Mata Kuliah Data Analysis.....	39
Gambar 2.25 Code Mata Kuliah Accounting	40
Gambar 2.26 Code Mata Kuliah Database	41
Gambar 2.27 Code Mata Kuliah Algorithm	41
Gambar 2.28 Code Mata Kuliah Mobile App.....	42
Gambar 2.29 dbHandlerNote	44
Gambar 2.30 Code Edit Notes	45
Gambar 2.31 Code Student Note	46
Gambar 2.32 Code Student Notes.....	48
Gambar 2.33 Code Tambah Student Note	49
Gambar 2.34 Code Forum.....	51
Gambar 2.35 Code Tambah Forum	53
Gambar 2.36 Code Edit Forum.....	56
Gambar 2.37 Code Informasi.....	58
Gambar 2.38 Code Custom Maps.....	59
Gambar 2.39 Code Reservation	62
Gambar 2.40 Code Detail	63
Gambar 2.41 Code Form Pembelian.....	65

Gambar 2.42 Code Read Admin.....	68
Gambar 2. 43 Code Update Admin.....	71
Gambar 2.44 <i>Code Admin Order</i>	73
Gambar 2.45 Code Order	74
Gambar 2.46 Code Read	75
Gambar 2.47 Code Tutor	76
Gambar 2.48 Code Order Adapter	77
Gambar 2.49 Code Read Adapter	79
Gambar 2.50 Code Tutor Adapter	80
Gambar 2.51 Conn.php	82
Gambar 2.52 dbConnect.php	82
Gambar 2.53 Database.php	83
Gambar 2.54 LogIn.php	84
Gambar 2.55 SignUp.php	85
Gambar 2.56 getForum.php	85
Gambar 2. 57 getForumById.php	86
Gambar 2.58 addForum.php	87
Gambar 2.59 updateForum.php	88
Gambar 2.60 deleteForum.php	88
Gambar 2.61 Tutor.php	89
Gambar 2.62 Pemesanan.php	90
Gambar 2.63 bacaAdmin.php	91
Gambar 2.64 updateTutorStok.php	91
Gambar 2.65 lihatOrder.php	92
Gambar 2.66 deleteTutor.php	93

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dengan adanya perkembangan teknologi yang signifikan, perkembangan dunia dan peradaban manusia juga semakin maju dari waktu ke waktu. Khususnya dalam beberapa tahun terakhir, informasi dan teknologi dibutuhkan dalam hampir semua bidang dan industri, termasuk sosial, budaya, kesehatan, komunikasi, bisnis, dan pendidikan. Alhasil, banyak aspek dari kehidupan manusia yang mengalami perubahan. Salah satunya adalah kemudahan dalam memperoleh informasi secara cepat dan akurat sesuai kebutuhan dari manapun dan kapanpun.

Kebutuhan terhadap teknologi dan informasi harus didukung dengan adanya sistem yang baik dan terintegrasi. Ini dapat diperoleh dengan adanya sistem informasi yang menggabungkan prosedur kerja, informasi, orang, dan teknologi informasi yang terorganisir. Hal lainnya yang tidak kalah penting adalah adanya aplikasi seluler. Aplikasi seluler yang berjalan pada perangkat seluler genggam ini sangatlah umum dan dibutuhkan oleh banyak orang karena bentuknya yang kecil, *portable*, mudah digunakan, dan dapat diakses dari mana saja. Terutama, di zaman sekarang, hampir semua orang mempunyai aplikasi dan perangkat seluler. Tidak hanya untuk penggunaan pribadi, aplikasi seluler ini juga sangat berguna untuk bisnis dalam proses pekerjaan dan berlangsungnya kegiatan dalam perusahaan, instansi, ataupun lembaga. Oleh karena itu pengembangan dan peningkatan kualitas perangkat seluler, seperti resolusi layar dan penyimpanan memori yang lebih besar, dan konektivitas yang lebih baik, sangatlah penting dalam menciptakan lingkungan yang lebih baik untuk aplikasi seluler modern.

Untuk itu perancangan dan pengembangan sistem informasi dan aplikasi seluler juga dipelajari dalam jenjang pendidikan kuliah baik secara formal maupun non-formal. Dalam hal ini, perancangan dan pengembangan sistem informasi dan aplikasi seluler akan diterapkan dalam pembuatan Bimbingan Belajar “BIMSI”. BIMSI sendiri nantinya akan dimasukkan ke dalam salah satu program kerja dari himpunan mahasiswa program studi Sistem Informasi. Program studi sistem informasi adalah bidang studi yang mengkombinasikan ilmu komputer dengan bisnis serta manajemen berbagai hal. Sedangkan, bimbingan belajar adalah bantuan yang dilakukan seseorang yang terlatih dalam suatu hal untuk memberikan pembelajaran

tambahan kepada individu, dengan tujuan mengatasi permasalahan pembelajaran yang dihadapi. Bimbingan belajar biasanya dilakukan kepada siswa agar mendapatkan penyesuaian pada pembelajarannya secara maksimal. Dengan membuat aplikasi seluler, pembelajaran dalam BIMSI juga akan menjadi lebih mudah dan efektif.

1.2 Gambaran Umum Organisasi

HIMSI merupakan singkatan dari Himpunan Mahasiswa Sistem Informasi yang berada dibawah naungan Universitas Multimedia Nusantara. HIMSI didirikan pada tahun 2007 dengan nama DISC (D'Information System Community) dan berubah nama secara resmi pada tahun 2011 menjadi HIMSI. Perubahan nama ini juga diiringi dengan perubahan sebutan dan juga logo. HIMSI dibentuk dengan tujuan untuk menjadi wadah bagi mahasiswa-mahasiswa sistem informasi dalam menampung dan menyalurkan ide, aspirasi dan kreativitas mereka sebagai generasi penerus bangsa yang inovatif dan inspiratif.

Sama seperti organisasi lain HIMSI juga memiliki visi dan misi. Visi mereka adalah untuk menjadikan HIMSI sebagai wadah penampung aspirasi dan pengembang aspek akademik maupun non-akademik KBM SI dengan berlandaskan atas profesionalitas dan kekeluargaan. HIMSI memiliki beberapa misi, yakni:

1. Menampung dan menyalurkan aspirasi KBM SI melalui program kerja yang membangun
2. Melaksanakan dan mengembangkan program kerja HIMSI agar membawa dampak positif bagi KBM SI
3. Meningkatkan profesionalitas sekaligus mempererat hubungan HIMSI dengan pihak eksternal
4. Mempererat hubungan pengurus HIMSI yang solid dan kompak

HIMSI memiliki beberapa program kerja yang setiap generasinya selalu berinovasi. Program kerja tersebut meliputi *Siberian Week*, *Social Is Me*, ISFEST, Pengenalan Prodi, *Company Visit*, *DISCO (D'Information System Community and Outbound)*, Inspirasi, Studi Banding, *HIMSI Anniversary*, dan Bimbel HIMSI. Pada *project* ini, kami akan menggunakan Bimbel HIMSI sebagai pedoman dan acuan dalam membuat aplikasi seluler Bimbingan Belajar “BIMSI”.

BIMSI merupakan sebuah sistem yang ditujukan untuk memberikan layanan berupa informasi terkait bimbingan belajar sehingga memudahkan Himpunan Mahasiswa Sistem

Informasi (HIMSI) dalam melakukan pendataan mahasiswa yang tertarik untuk mengikuti bimbingan belajar. Serta, membantu mahasiswa untuk mendaftar dan mengikuti kegiatan bimbingan belajar. BIMSI merupakan singkatan dari Bimbingan Belajar Mahasiswa Sistem Informasi. Dengan begitu, diharapkan mahasiswa dapat meningkatkan pemahaman, kemampuan penyelesaian masalah dalam *project* yang sedang mereka kerjakan, dan nilai akademik mereka. Selain itu, para tutor juga dapat menambah pengalaman mengajar mereka.

1.3 Rumusan Masalah

Berdasarkan latar belakang tersebut kami sebagai peneliti merumuskan beberapa masalah, yakni:

1. Bagaimana rancangan aplikasi seluler Bimbingan Belajar “BIMSI” dapat mempermudah dan mempercepat pengaksesan dan proses pendaftaran bimbingan belajar?
2. Fitur apa saja yang ada dalam aplikasi seluler Bimbingan Belajar “BIMSI”?

1.4 Tujuan Penelitian

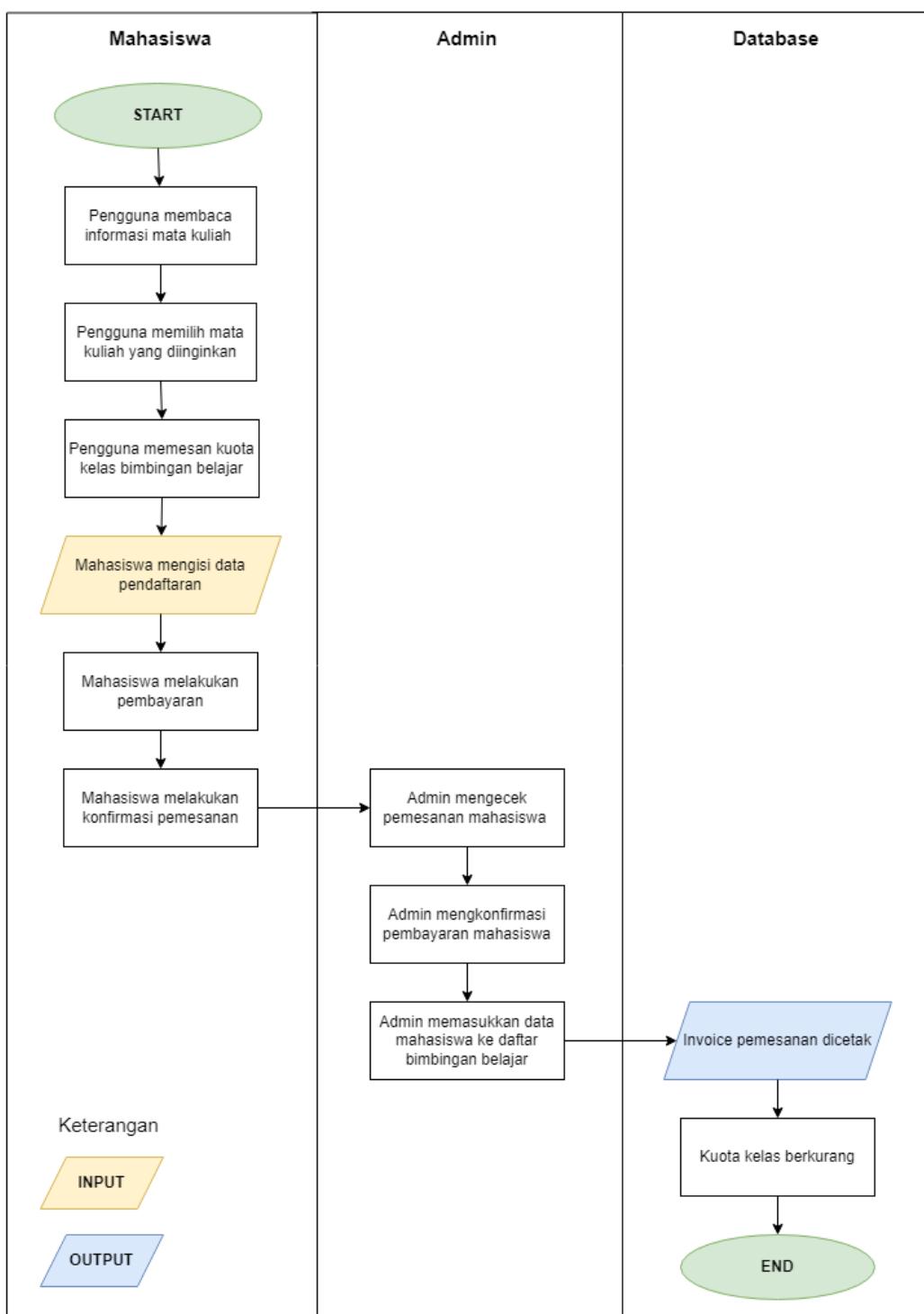
Proyek ini memiliki tujuan untuk mengetahui proses pelaksanaan dari Bimbingan Belajar “BIMSI”. Selain itu ada juga beberapa tujuan lain dari pembuatan proyek ini, diantaranya:

1. Merancang dan membuat aplikasi seluler dalam lingkup bimbingan belajar program studi Sistem Informasi yang efektif, efisien, dan mudah digunakan.
2. Menyediakan aplikasi seluler bagi bimbingan belajar program studi Sistem Informasi dengan fitur-fitur yang mampu menampilkan informasi secara *realtime* dan mudah dipahami.
3. Membantu bimbingan belajar program studi Sistem Informasi dalam mendata semua informasi dan data yang dibutuhkan untuk melakukan segala bentuk administrasi secara terstruktur.

BAB II

ISI

2.1 Proses Bisnis



Gambar 2.1 Proses Bisnis BIMSI

Proses bisnis merupakan sekumpulan tahapan, prosedur, atau aktivitas yang saling terkait satu dengan lainnya dalam suatu perusahaan atau organisasi. Proses bisnis mencakup strategi bisnis yang diembankan kepada para pemangku kepentingan atas suatu pekerjaan spesifik untuk mencapai tujuan atau objektif yang diinginkan, seperti memberikan produk atau layanan kepada pelanggan. Tahapan ini dilakukan berulang kali oleh beberapa penguji dengan spesialisasinya masing-masing demi meraih tujuan yang diinginkan dengan cara atau standar optimal dengan memaksimalkan produktivitas dan kinerja dari proses yang ada.

Diagram di atas merupakan proses bisnis yang terdapat dalam bimbingan belajar BIMSI. Untuk menggunakan fungsi aplikasi BIMSI, mahasiswa sebagai pengguna harus mengunduh aplikasi BIMSI pada Google Play Store ataupun App Store terlebih dahulu. Setelah melakukan pengunduhan, pengguna dapat membuka aplikasi BIMSI yang akan ditujukan pada laman *Log In*. Bagi pengguna yang telah memiliki akun, pengguna dapat langsung memasukkan alamat *e-mail* dan *password* sesuai akun yang telah terdaftar pada database aplikasi BIMSI lalu menekan tombol *Log In*. Namun, bagi pengguna yang belum memiliki akun, pengguna harus menekan tulisan *Sign Up* yang berada pada bagian bawah halaman *Log In* kemudian menginput data diri mereka untuk mendaftar sebagai pengguna aplikasi BIMSI.

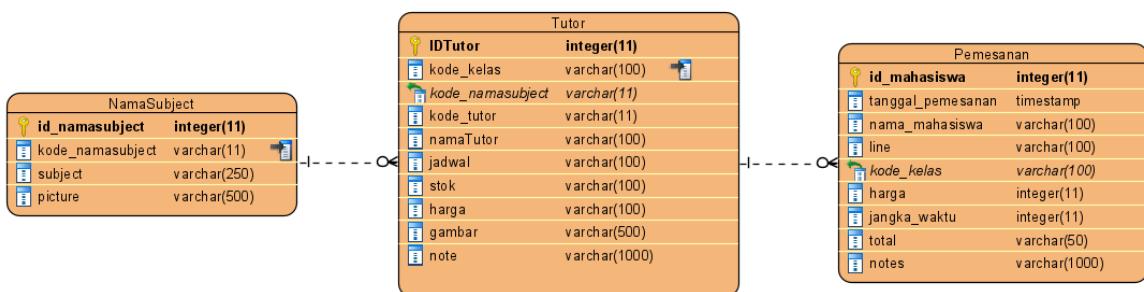
Setelah berhasil masuk, pengguna akan dihadapkan pada halaman *Home* aplikasi BIMSI yang berisi ikon dari setiap mata kuliah yang disediakan oleh bimbingan belajar BIMSI. Jika pengguna menekan salah satu ikon mata kuliah, pengguna dapat membaca informasi singkat mengenai mata kuliah tersebut dan jadwal yang tersedia. Selanjutnya, pada *layer* yang sama, pengguna dapat menekan tombol *Reserve* untuk melakukan reservasi atau pemesanan terhadap jadwal bimbingan belajar mata kuliah tersebut.

Setelah pengguna menekan tombol *Reserve*, pengguna akan diarahkan ke halaman *BIMSI Reservation* yang terdiri dari dua bagian utama. Pada bagian atas, pengguna dapat melihat kembali ikon setiap mata kuliah yang disediakan oleh bimbingan belajar BIMSI. Sedangkan, pada bagian bawah, pengguna dapat melihat seluruh daftar tutor berupa foto dan nama dari setiap tutor yang aktif mengajar di bimbingan belajar BIMSI. Ikon mata kuliah pada bagian atas layer *BIMSI Reservation* berfungsi untuk memfilter foto dan nama tutor yang ditampilkan sesuai dengan ikon mata kuliah yang dipilih. Saat pengguna menekan masing-masing foto tutor, pengguna akan diarahkan ke halaman baru yakni *Tutor BIMSI* yang menampilkan data atau informasi seperti nama tutor, kode, kuota, jadwal, dan harga setiap kelas, serta deskripsi tambahan mengenai tutor tersebut. Pengguna dapat membaca informasi

tersebut lalu menekan tombol *Reserve Now* yang terletak di paling bawah untuk memesan kuota kelas bimbingan belajar bersama tutor tersebut.

Terakhir, dalam penyelesaian proses pemesanan, pengguna akan diarahkan pada halaman *Reservation Page* untuk mengisi nama mahasiswa, ID Line, dan jumlah pemesanan kelas; dimana jumlah kelas yang dipesan harus empat. Hal ini dikarenakan BIMSI hanya memfasilitasi pemesanan kuota kelas sebanyak satu bulan pertemuan bimbingan belajar. Setelah itu, pengguna harus menekan tombol *Calculate* untuk dapat melihat hasil kalkulasi total biaya dari bimbingan belajar sesuai jumlah kelas yang telah dipesan. Di bawahnya juga terdapat kolom catatan atau *note* pembayaran yang wajib diisi pengguna untuk menginformasikan opsi atau pilihan pembayaran yang dipilih. Setelah selesai mengisi seluruh kolom pendataan yang dibutuhkan dan mengecek kesesuaianya, pengguna harus menekan tombol *Confirm* yang terletak di bagian paling bawah halaman untuk mengkonfirmasi pemesanan kuota bimbingan belajar BIMSI.

2.2 Entity Relationship Diagram



Gambar 2.2 Entity Relationship Diagram BIMSI

ERD atau *Entity Relationship Diagram* merupakan suatu diagram yang berfungsi untuk menggambarkan hubungan atau relasi antar objek yang ada pada suatu sistem dalam suatu rangka yang terstruktur. ERD banyak dimanfaatkan dalam berbagai proyek sistem informasi untuk memudahkan dalam pembuatan model dan penjelasan struktur suatu basis data dari proyek yang sedang dikembangkan. ERD aplikasi BIMSI terdiri dari 3 tabel yakni NamaSubject, Tutor, dan Pemesanan. Tabel NamaSubject memiliki atribut id subject, kode

subject, subject, dan picture. Tabel Tutor memiliki atribut yang terdiri dari id tutor, kode kelas, kode_namasubject, kode_tutor, nama_tutor, jadwal, stok, harga, gambar, dan note. Tabel Pemesanan memiliki atribut yakni id mahasiswa, tanggal pemesanan, nama mahasiswa, line, kode kelas, harga, jangka waktu, total, dan notes.

Tabel NamaSubject dengan kode_namasubject sebagai *index* dihubungkan ke Tabel Tutor dengan hubungan *one to many* karena setiap kode_namasubject dapat muncul atau dipilih berulang kali oleh seorang tutor tertentu. Tabel Tutor dengan kode_kelas sebagai *index* dihubungkan ke Tabel Pemesanan dengan hubungan *one to many* karena setiap kode_kelas dapat muncul atau dipilih berulang kali dalam pemesanan kelas bimbingan belajar mahasiswa terkait. Berikut merupakan rincian dari masing-masing atribut yang terkandung dalam tabel database BIMSI.

1. **Tabel NamaSubject**

Atribut : id_namasubject (Primary Key)

Atribut : kode_namasubject (Index)

Atribut : subject

Atribut : picture

2. **Tabel Tutor**

Atribut : IDTutor (Primary Key)

Atribut : kode_kelas (Index)

Atribut : kode_namasubject (Foreign Key)

Atribut : kode_tutor

Atribut : namaTutor

Atribut : jadwal

Atribut : stok

Atribut : harga

Atribut : gambar

Atribut : note

3. Tabel Pemesanan

Atribut : id_mahasiswa (Primary Key)

Atribut : tanggal_mahasiswa

Atribut : nama_mahasiswa

Atribut : line

Atribut : kode_kelas (Foreign Key)

Atribut : harga

Atribut : jangka_waktu

Atribut : total

Atribut : notes

2.3 Layout Aplikasi bagi Pengguna

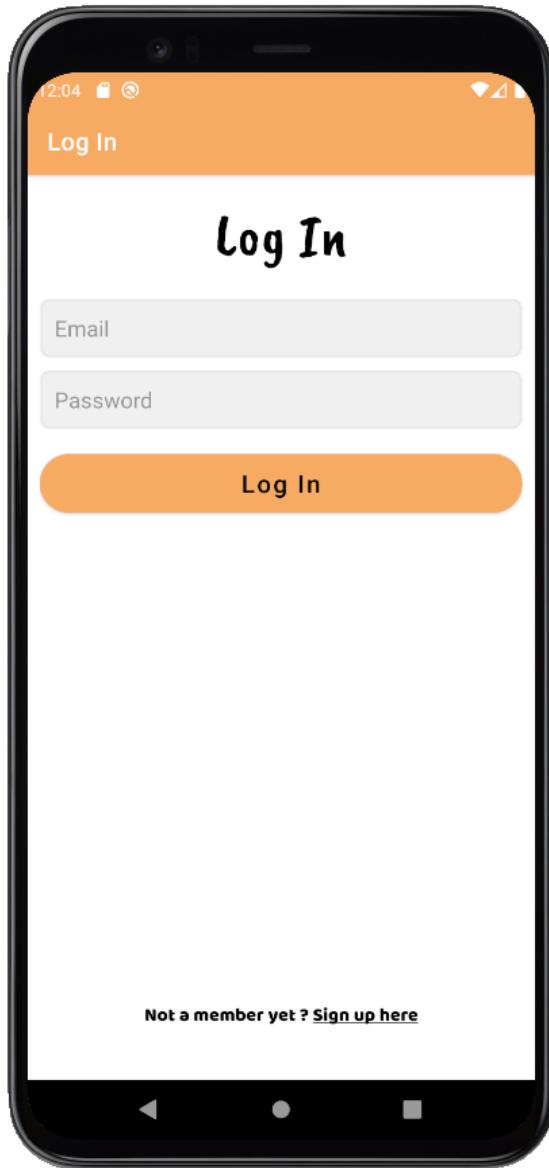
2.3.1 Layer *Splash Screen*



Gambar 2.3 Layer *Splash Screen*

Ini merupakan tampilan *Splash Screen* dari aplikasi bimbingan belajar BIMSI, dimana terdapat logo BIMSI dengan warna yang terinspirasi dari logo HIMSI, yakni biru tua dan oranye.

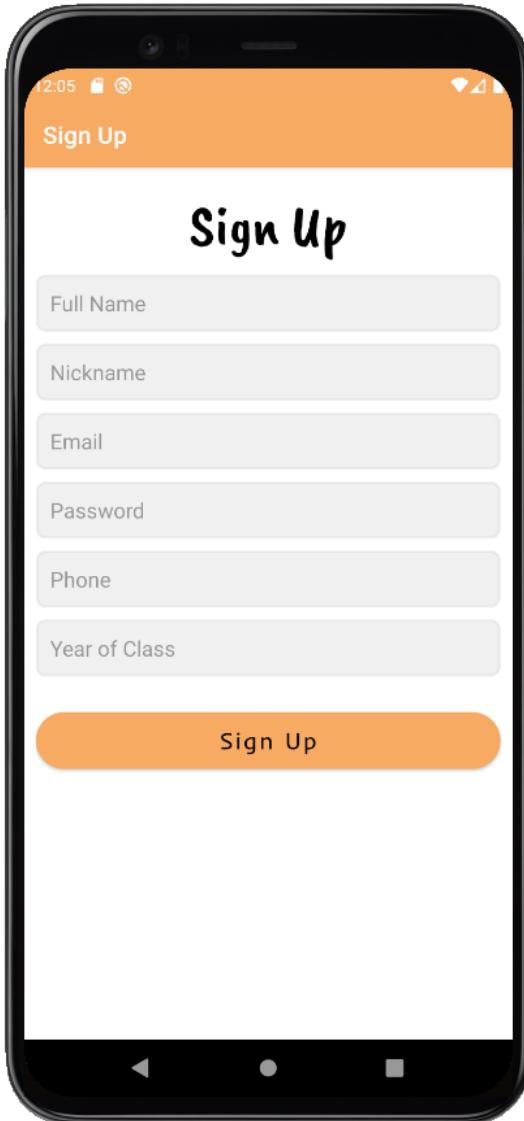
2.3.2 Layer Login



Gambar 2.4 Layer Log In

Pada tampilan *Log In*, terdapat dua bagian pada tampilan yaitu tombol *Log In* dan tulisan “Sign up” yang dapat ditekan. Jika pengguna telah mendaftar dalam aplikasi, pengguna dapat langsung *log in* dengan email dan password yang telah didaftarkan. *Email* dan *password* yang dimasukkan harus benar atau valid, karena ketika data yang dimasukkan tidak sesuai akan muncul *toast* “*Wrong Email/Password*”. Namun, jika pengguna belum mempunyai atau mendaftarkan akunnya, pengguna dapat menekan bagian tulisan “Sign up” yang berada di bagian bawah tampilan untuk melakukan pendaftaran akun terlebih dahulu. Pada saat pengguna menekan bagian tulisan “Sign up”, pengguna akan langsung diarahkan ke halaman *Sign up* untuk mengisi data diri.

2.3.3 Layer Sign Up



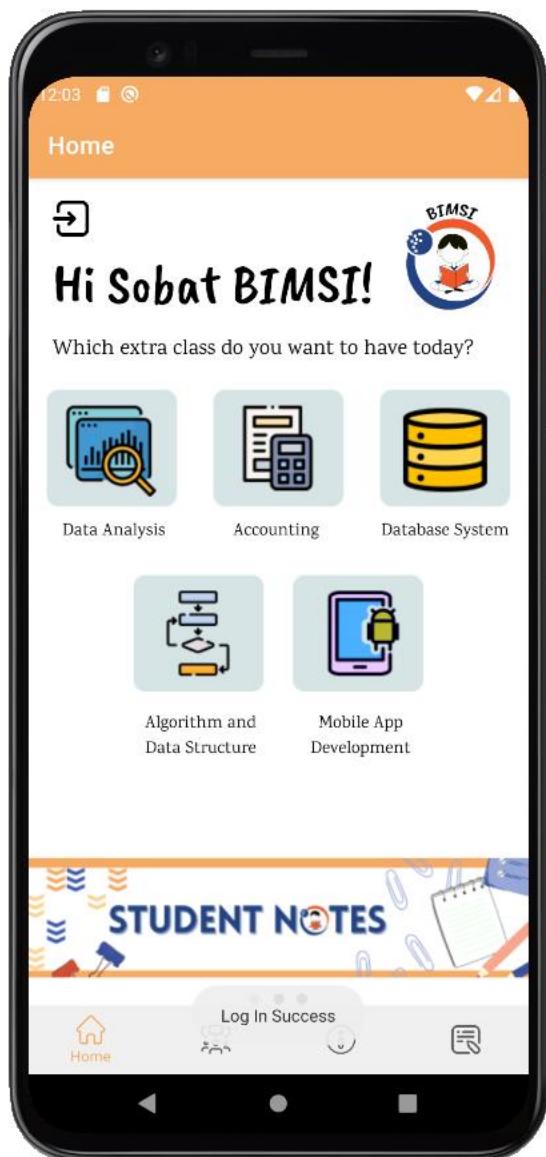
Gambar 2.5 Layer Sign Up

Pada tampilan *Sign Up*, apabila user belum memiliki akun dan ingin membuat akun, maka pengguna dapat mengisi data yang diperlukan untuk membuat akun. Pengguna akan diminta melengkapi data-data yang dibutuhkan seperti *full name*, *nickname*, *email*, *password*, nomor telepon, dan tahun angkatan. Setelah data-data tersebut telah terisi secara lengkap maka pengguna dapat menekan button *sign up*.

Jika *email* yang dimasukkan sudah terdaftar sebelumnya atau data-data yang diisi tidak benar, akan muncul *toast* berupa “*User already registered or email invalid, Sign up failed*”. Sedangkan, saat ada salah satu data yang tidak terisi atau masih kosong akan muncul *toast*

berupa “*All fields must be filled*”. Ketika data yang diisi sudah sesuai dan valid maka akan muncul *toast* “*Sign up successful*” dan pengguna akan diarahkan kembali ke halaman *Log In* untuk menginput *email* dan *password*.

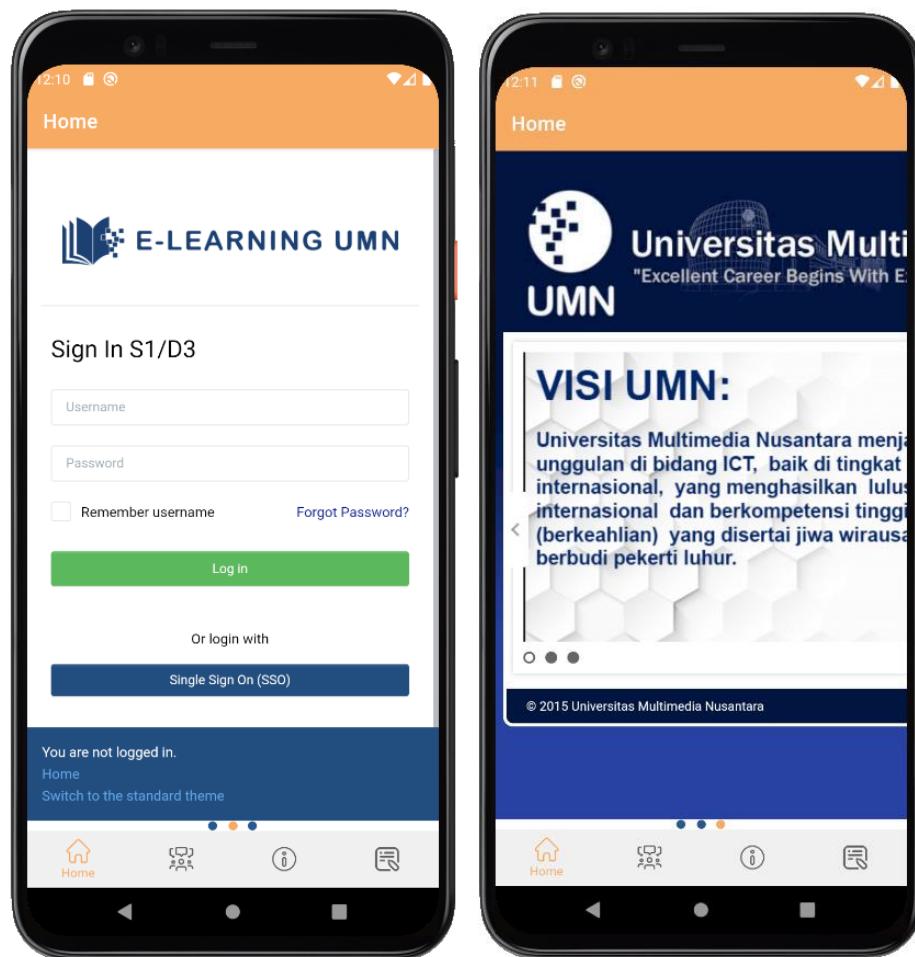
2.3.4 Layer Home





Gambar 2.6 Layer Home

Pada tampilan *Home*, pengguna disambut dengan tulisan “Hi, Sobat BIMSI” dan pertanyaan “*Which extra class do you want to have today?*”. Pengguna juga dapat melihat beberapa pilihan mata kuliah yang tersedia pada bimbingan belajar BIMSI. Saat pengguna menekan masing-masing logo mata kuliah, pengguna akan diarahkan ke halaman baru, dimana akan muncul *toast* yang bergantung pada mata kuliah terkait seperti “*Move to Algorithm and Data Structure*”. Halaman ini berisi deskripsi dan jadwal yang tersedia untuk mata kuliah tersebut. Setelah menentukan pilihan mata kuliah, pengguna dapat langsung menekan tombol *Reserve* dan akan diarahkan ke halaman *BIMSI Reservation* untuk melihat dan mengisi detail reservasi. Pada bagian bawah *layout* terdapat *student notes* yang akan mengarahkan pengguna ke halaman *student notes*. Terakhir, terdapat tombol *Log Out* pada bagian kiri atas yang berfungsi untuk keluar dari aplikasi.

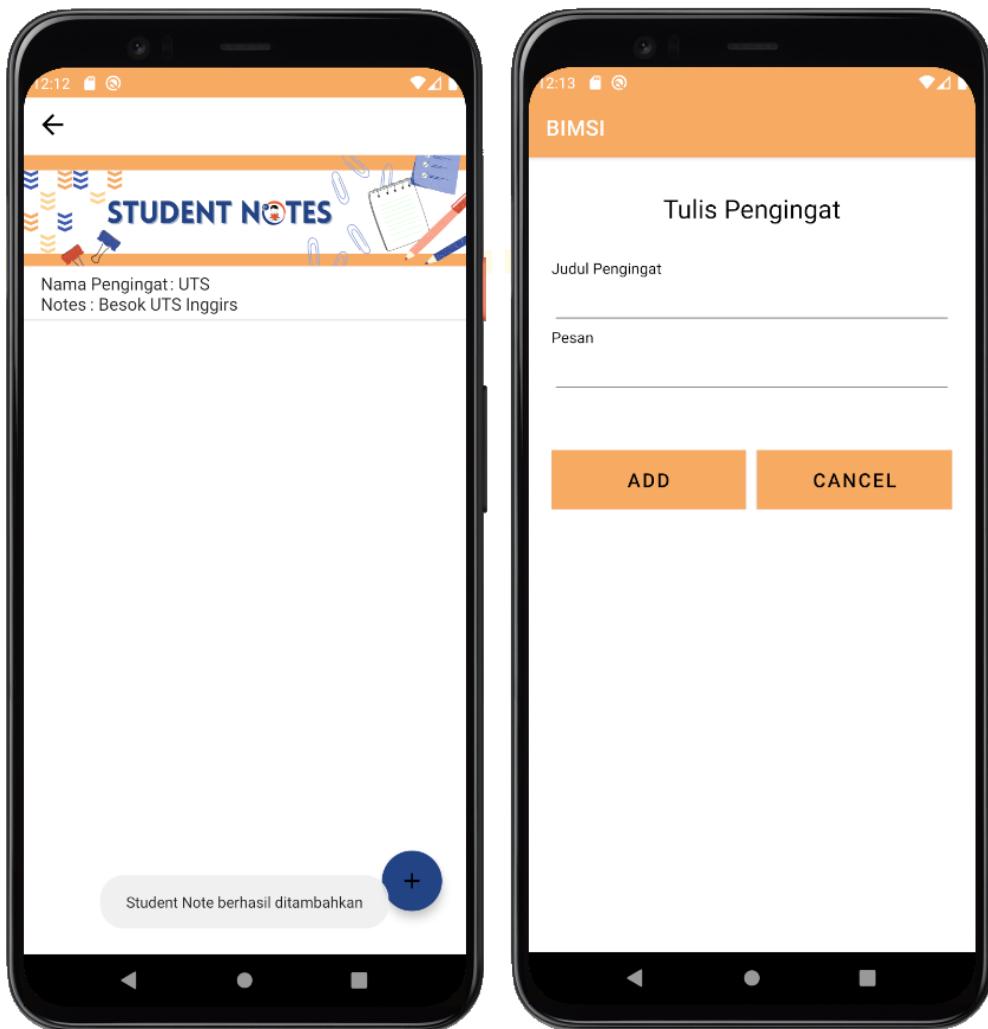


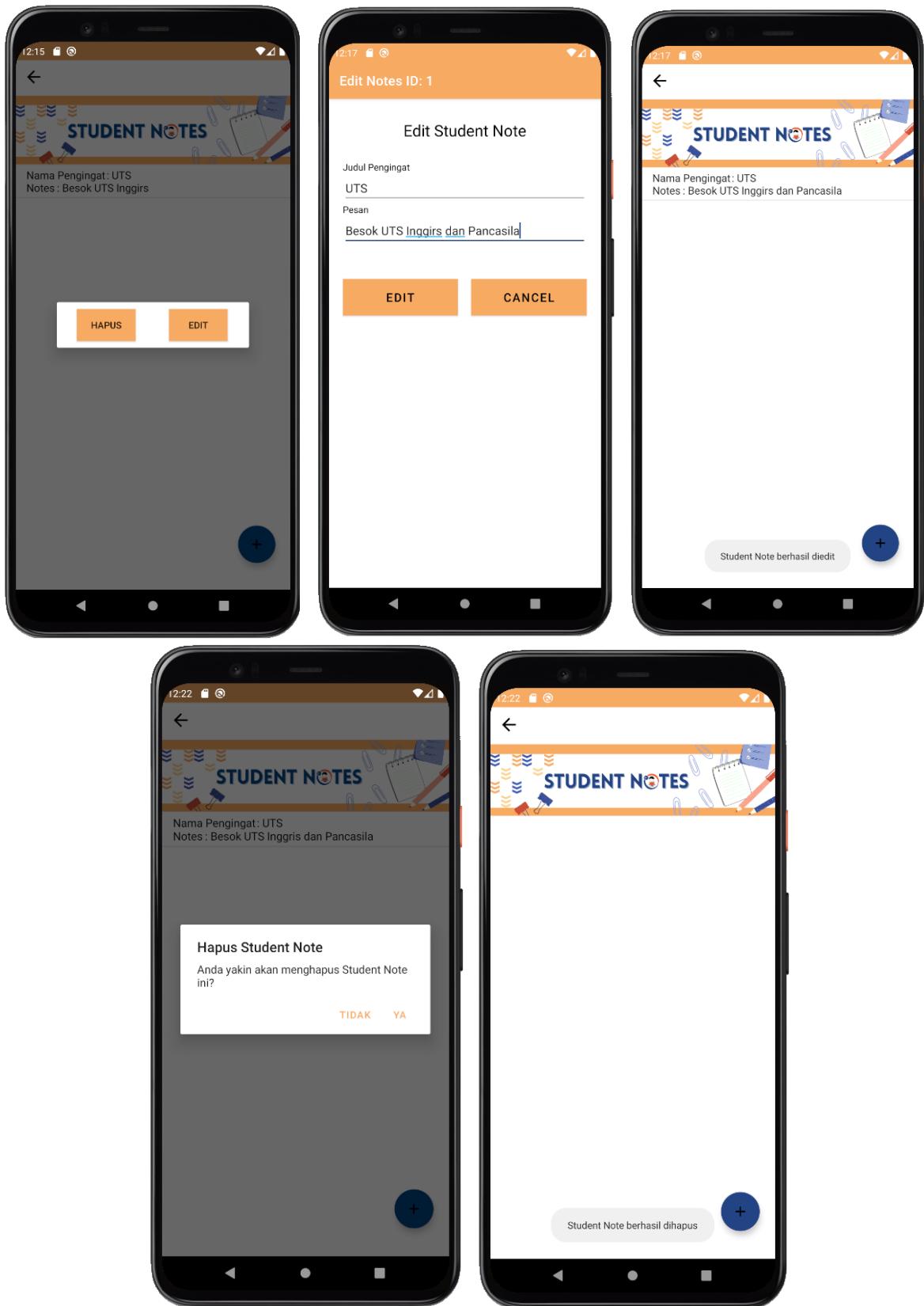
Gambar 2.7 Layer View Pager

Halaman *Home* terdiri dari tiga *view pager*, dimana pengguna dapat menggunakan *view pager* hanya dengan menggeser halaman ke kiri maupun ke kanan. Pada halaman *view pager*

yang kedua terdapat *E-Learning* dan MYUMN pada *view pager* ketiga, dimana pada kedua *view pager* tersebut pengguna dapat mengakses *E-Learning* dan MYUMN melalui aplikasi ini, tanpa harus membuka *web browser*.

2.3.5 Layer Student Notes

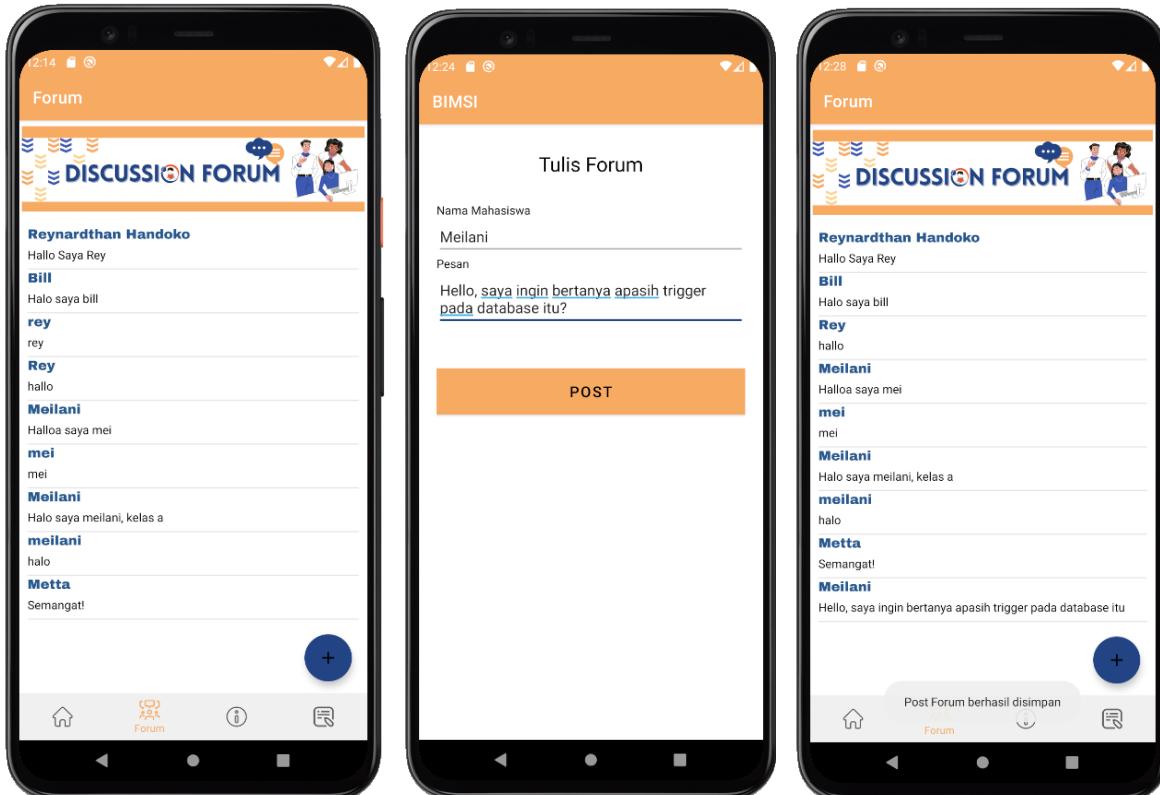


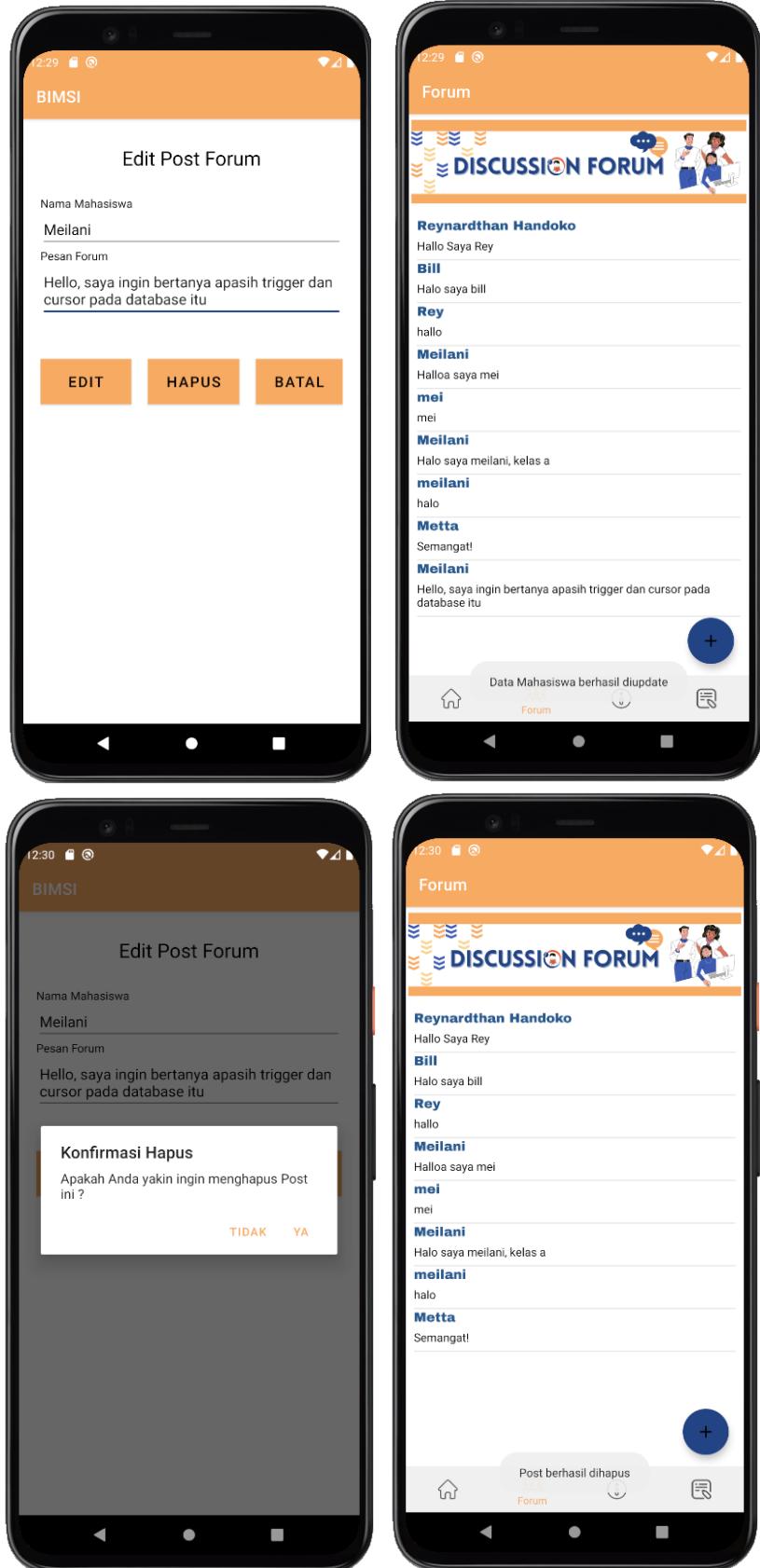


Gambar 2.8 Layer Student Notes

Ketika pengguna menekan *image button Student Notes*, pengguna akan diarahkan ke halaman *Student Notes* dan akan muncul *toast* “Move to Student Notes”. Database yang digunakan pada halaman ini adalah database lokal atau SQLite, dimana catatan hanya disimpan pada perangkat dan tidak akan muncul jika diakses menggunakan perangkat lain. Untuk menambahkan catatan, pengguna perlu menekan *floating button* yang berada di kanan bawah dan akan diarahkan ke halaman baru dimana pengguna dapat memasukkan catatan yang ingin ditulis. Ketika catatan sudah yakin untuk ditambahkan pengguna perlu menekan tombol *add*, tetapi jika pengguna tidak yakin dan ingin membatalkan catatan pengguna dapat menekan tombol *cancel*. Setelah catatan berhasil ditambahkan akan muncul *toast* “*Student note berhasil ditambahkan*”. Saat pengguna ingin mengubah ataupun menghapus isi catatan, pengguna hanya perlu menekan catatan yang telah dibuat, dimana kemudian akan muncul *pop up* dua tombol yakni *delete* dan *edit*. Jika sudah yakin dengan perubahannya, pengguna dapat menekan tombol *edit*. Namun saat pengguna ingin menghapus catatan yang telah ditulis, pengguna dapat menekan tombol *delete*. Sebelum benar-benar terhapus akan muncul *alert dialog* yang berisikan peringatan “Anda yakin akan menghapus student note ini?”, jika pengguna menekan iya maka catatan akan terhapus dan muncul *toast* “*Student Note berhasil dihapus*”.

2.3.6 Layer Forum

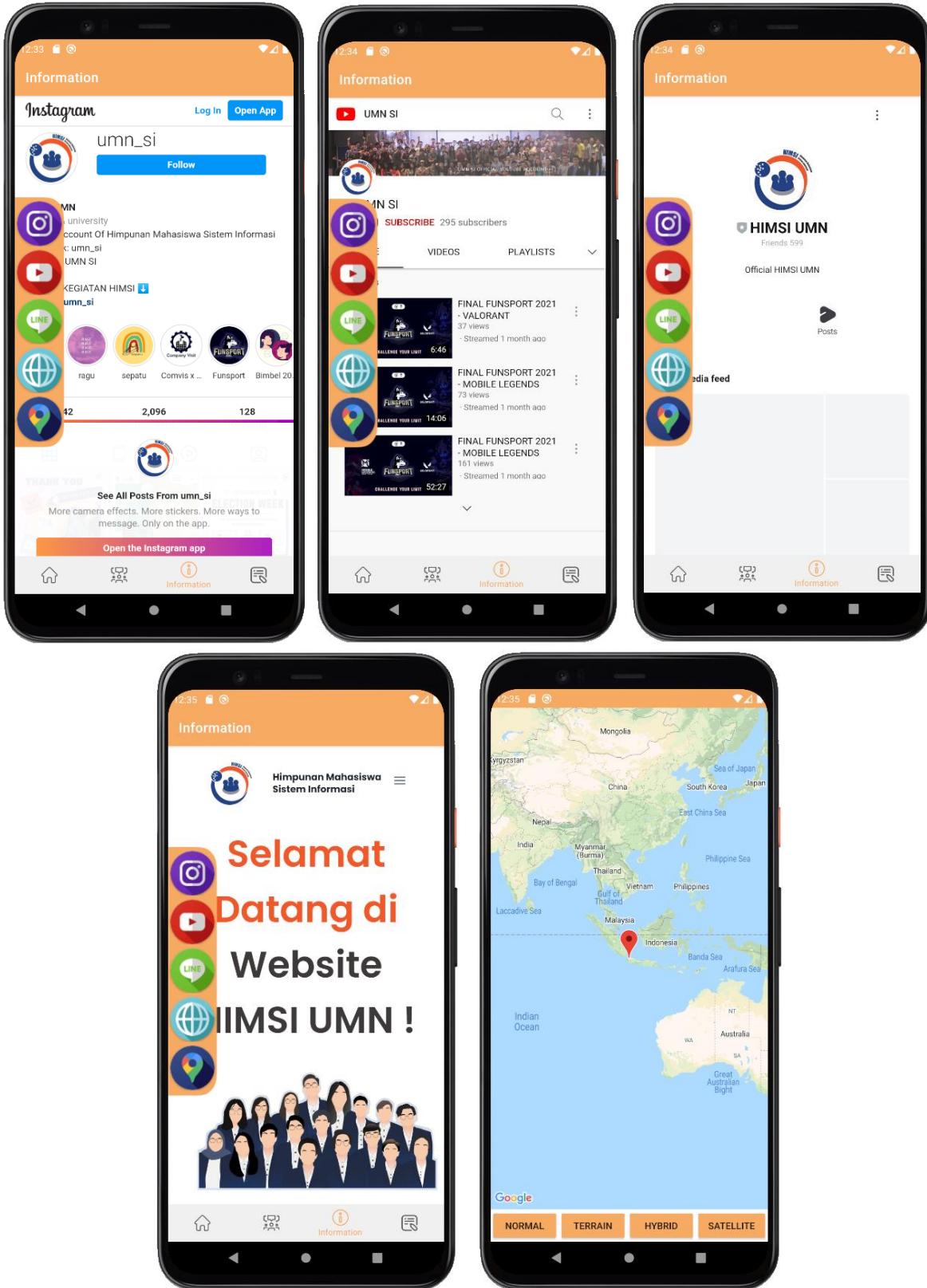




Gambar 2. 9 Layer Forum

Pada tampilan *Forum*, pengguna dapat melihat pesan-pesan yang ditulis oleh pengguna, dimana pesan bisa berupa pertanyaan, jawaban, maupun informasi. Database yang digunakan pada halaman ini adalah database MySQL, dimana catatan disimpan secara *online* atau *cloud* sehingga dapat dilihat dan diakses oleh banyak pengguna. Untuk menambahkan pesan forum, pengguna perlu menekan *floating button* yang terletak di kanan bawah dan akan diarahkan ke halaman baru dimana pengguna dapat memasukkan pesan forum yang ingin ditulis. Ketika pesan sudah yakin untuk ditambahkan pengguna perlu menekan tombol *post*, tetapi kalau pengguna tidak yakin dan ingin membatalkan pesan pengguna dapat menekan tombol *cancel*. Setelah catatan berhasil ditambahkan akan muncul *toast* “*Post Forum berhasil disimpan*”. Saat pengguna ingin mengubah ataupun menghapus isi pesan pada forum, pengguna dapat menekan pesan dan akan diarahkan ke halaman baru yakni Edit Post Forum. Pada halaman ini, pengguna dapat mengubah Nama Mahasiswa maupun Pesan Forum. Jika sudah yakin dengan perubahannya, pengguna dapat menekan tombol *edit* dan akan muncul *toast* “*Data mahasiswa berhasil diupdate*”. Namun saat pengguna tidak jadi mengubah isi pesan, pengguna dapat menekan tombol batal. Terakhir saat pengguna ingin menghapus forum yang ditulis, pengguna dapat menekan tombol hapus. Sebelum benar-benar terhapus akan muncul *alert dialog* yang berisikan peringatan “*Anda yakin akan menghapus post ini?*”, jika pengguna menekan iya maka catatan akan terhapus dan muncul *toast* “*Post berhasil dihapus*”.

2.3.7 Layer Information



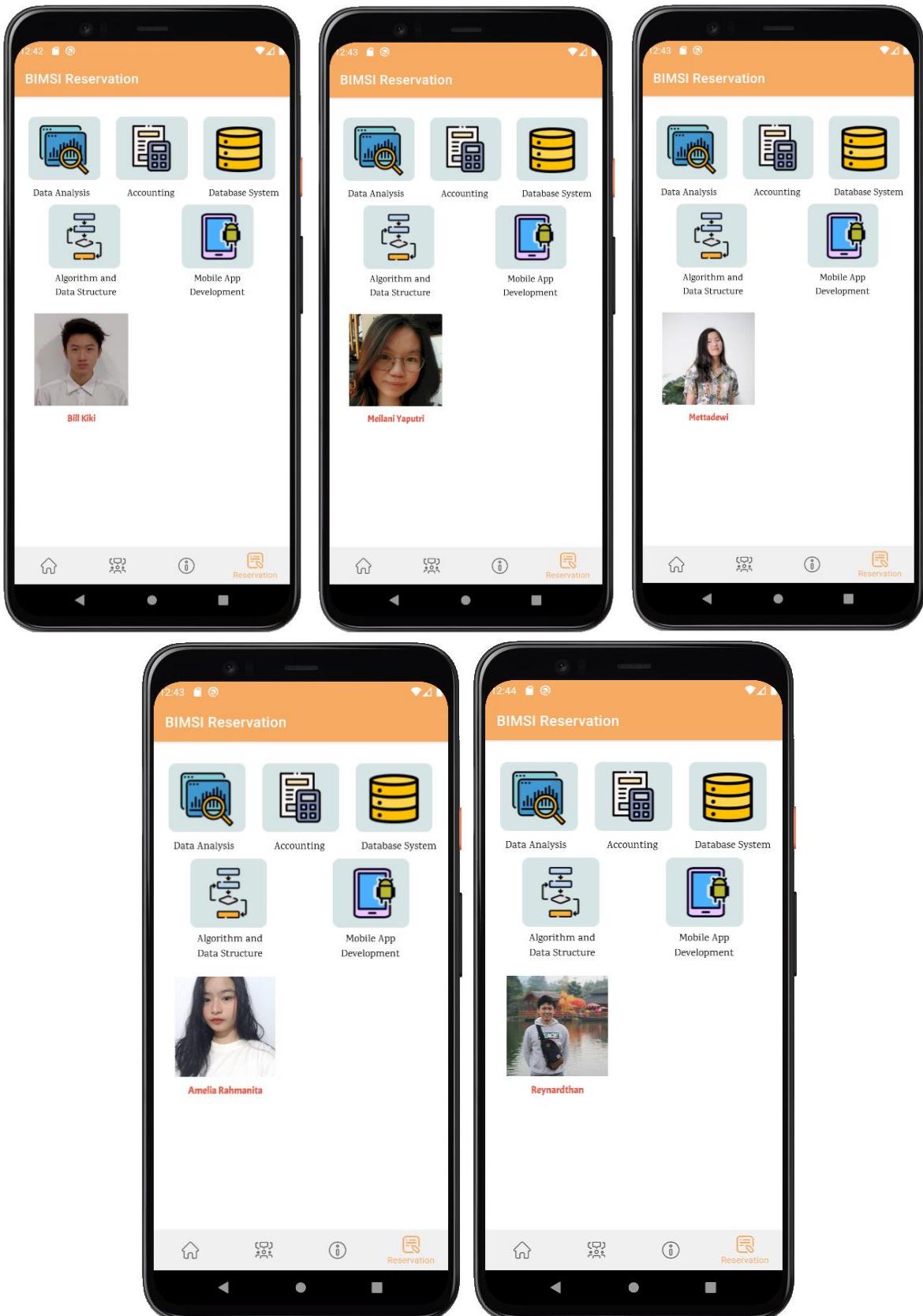
Gambar 2.10 Layer Information

Pada tampilan Information, pengguna dapat melihat segala macam media sosial dan platform terkait jurusan Sistem Informasi khususnya HIMSI. Terdapat empat tombol yang masing-masingnya terhubung ke web view, dimana logo-logo tersebut merepresentasikan media sosial dari HIMSI. Tombol pertama merupakan official instagram dari HIMSI yaitu umn_si. Tombol kedua merupakan channel YouTube dari jurusan Sistem Informasi yakni UMN SI. Tombol ketiga merupakan line dari HIMSI UMN. Tombol keempat merupakan official website dari HIMSI. Terakhir, terdapat tombol maps atau peta yang akan mengarahkan pengguna ke layer maps.

Pada *layer maps*, pengguna dapat melihat dan mengetahui lokasi gedung Universitas Multimedia Nusantara. Saat pengguna menekan penanda lokasi, akan muncul tulisan “*Welcome to UMN*”. Terdapat 4 jenis *view* yang memiliki gambaran yang berbeda, yaitu *normal, terrain, hybrid* dan *satellite*.

2.3.8 Layer *BIMSI Reservation*

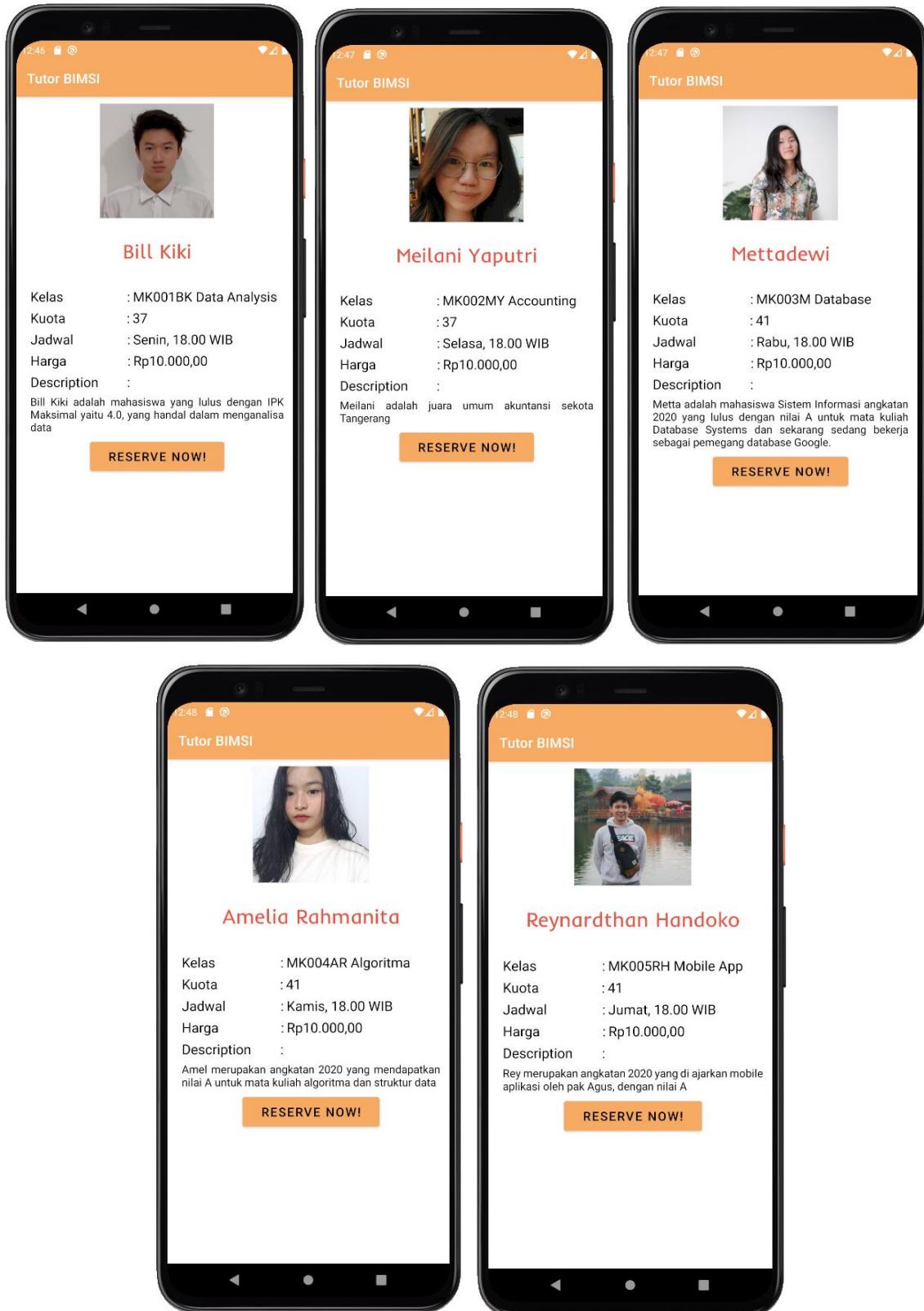




Gambar 2.11 Layer BIMSI Reservation

BIMSI Reservation merupakan halaman yang akan muncul setelah pengguna menekan tombol *Reserve*, dimana pengguna akan diarahkan ke halaman *BIMSI Reservation* yang terdiri dari dua bagian utama. Pada bagian atas, pengguna dapat melihat kembali ikon setiap mata kuliah yang disediakan oleh bimbingan belajar BIMSI. Sedangkan, pada bagian bawah, pengguna dapat melihat seluruh daftar tutor berupa foto dan nama dari setiap tutor yang aktif mengajar di bimbingan belajar BIMSI. Ikon mata kuliah pada bagian atas layer *BIMSI Reservation* berfungsi untuk memfilter foto dan nama tutor yang ditampilkan sesuai dengan ikon mata kuliah yang dipilih.

2.3.9 Layer Tutor BIMSI

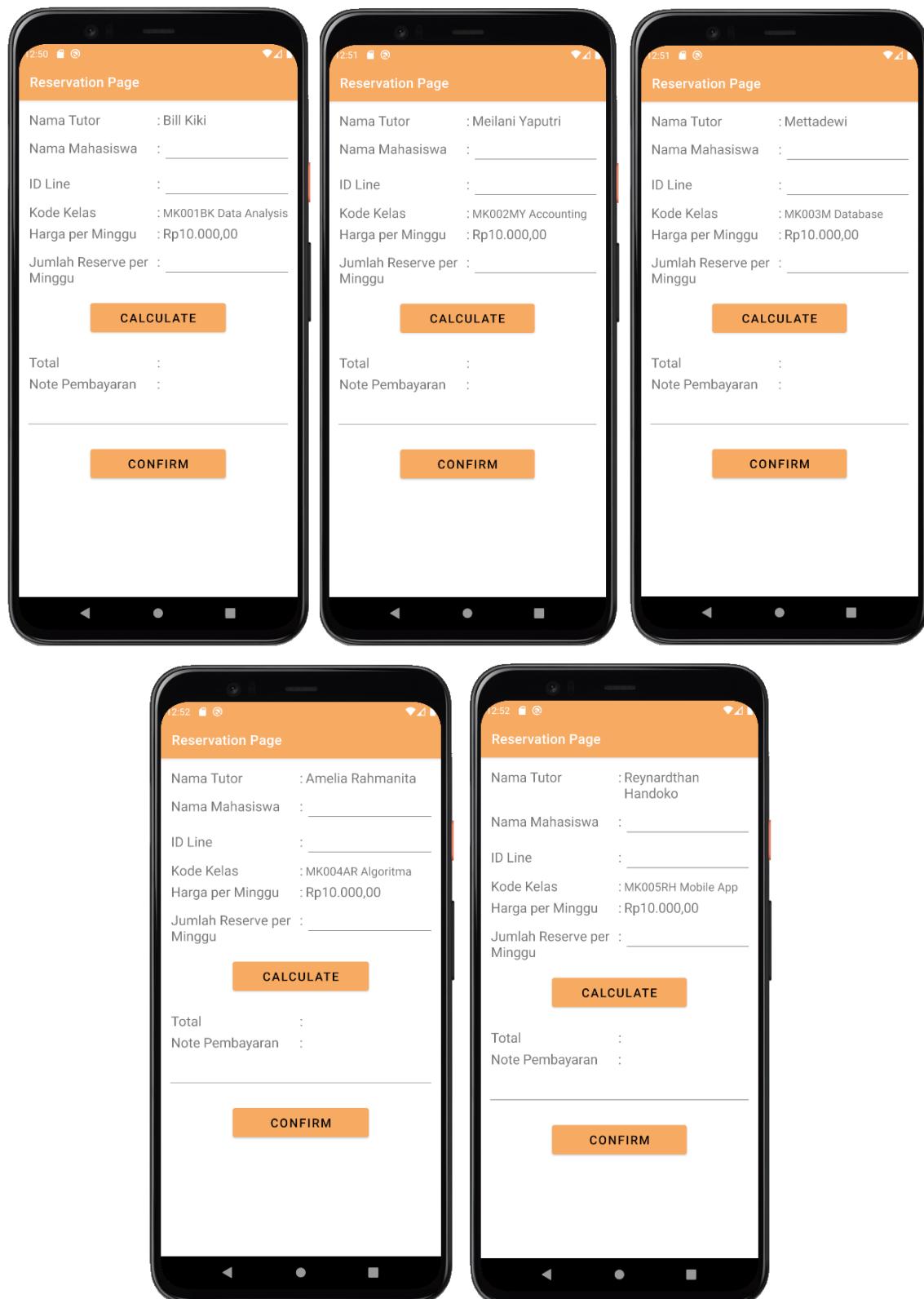


Gambar 2.12 Layer Tutor BIMSI

Saat pengguna menekan masing-masing foto tutor, pengguna akan diarahkan ke halaman baru yakni *Tutor BIMSI* yang menampilkan data atau informasi seperti nama tutor,

kode, kuota, jadwal, dan harga setiap kelas, serta deskripsi tambahan mengenai tutor tersebut. Pengguna dapat membaca informasi tersebut lalu menekan tombol *Reserve Now* yang terletak di paling bawah untuk memesan kuota kelas bimbingan belajar bersama tutor tersebut.

2.3.10 Layer Reservation Page

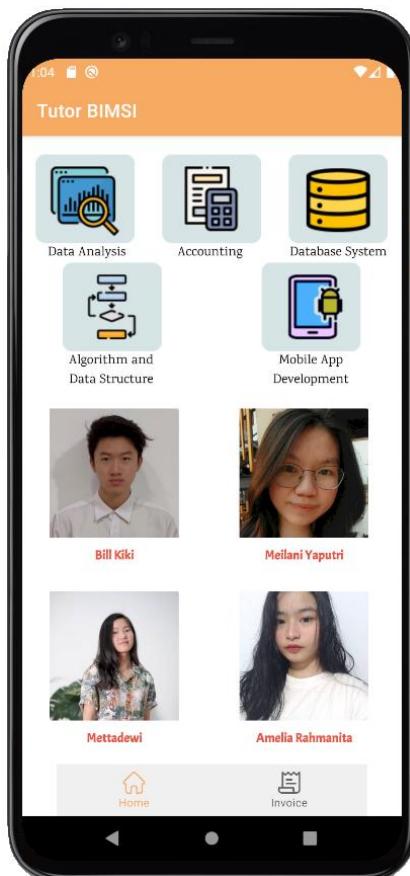


Gambar 2.13 Layer Reservation Page

Halaman Reservation Page akan muncul saat pengguna menekan tombol *Reserve Now* yang berada pada *layer* tutor. Pada halaman ini pengguna perlu mengisi nama mahasiswa, ID Line, dan jumlah pemesanan kelas; dimana jumlah kelas yang dipesan harus empat. Hal ini dikarenakan BIMSI hanya memfasilitasi pemesanan kuota kelas sebanyak satu bulan pertemuan bimbingan belajar. Setelah itu, pengguna harus menekan tombol *Calculate* untuk dapat melihat hasil kalkulasi total biaya dari bimbingan belajar sesuai jumlah kelas yang telah dipesan. Di bawahnya juga terdapat kolom catatan atau *note* pembayaran yang wajib diisi pengguna untuk menginformasikan opsi atau pilihan pembayaran yang dipilih. Setelah selesai mengisi seluruh kolom pendaftaran yang dibutuhkan dan mengecek kesesuaianya, pengguna harus menekan tombol *Confirm* yang terletak di bagian paling bawah halaman untuk mengkonfirmasi pemesanan kuota bimbingan belajar BIMSI.

2.4 Layout Aplikasi bagi Admin

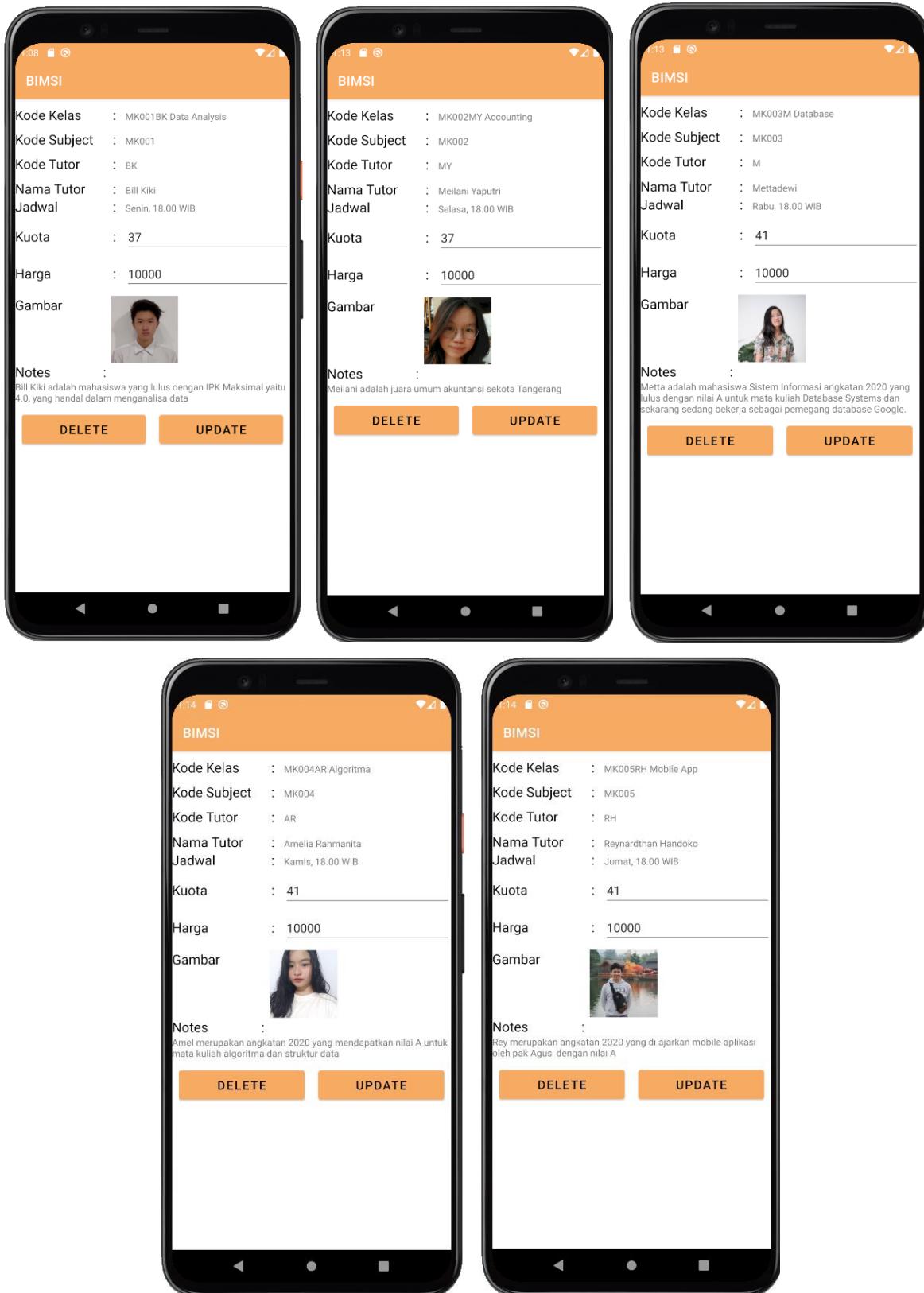
2.4.1 Layer Tutor BIMSI



Gambar 2.14 *Layer Tutor BIMSI Untuk Admin*

Tampilan dari halaman Tutor BIMSI merupakan layer yang akan pertama kali dihadapkan setelah admin berhasil masuk ke dalam aplikasi. Tutor *BIMSI* terdiri dari dua bagian utama, pada bagian atas, admin dapat melihat ikon setiap mata kuliah yang diajarkan oleh masing-masing tutor. Sedangkan, pada bagian bawah, admin dapat melihat seluruh daftar tutor berupa foto dan nama dari setiap tutor yang aktif mengajar di bimbingan belajar BIMSI. Ikon mata kuliah pada bagian atas layer Tutor *BIMSI* berfungsi untuk memfilter foto dan nama tutor yang ditampilkan sesuai dengan ikon mata kuliah yang dipilih. Saat foto tutor ditekan, admin akan diarahkan ke halaman Update Tutor.

2.4.2 Layer Update Tutor



Gambar 2.15 Layer Update Tutor Untuk Admin

Pada tampilan *Update Tutor*, admin dapat mengubah data-data seperti jumlah kuota dan harga untuk masing-masing mata kuliah atau kelas bimbingan belajar. Jika sudah yakin, admin dapat menekan tombol *update*. Selain itu, admin juga dapat menghapus data tutor dengan menekan tombol *delete*.

2.4.3 Layer *Invoice*



Gambar 2.16 *Layer Invoice Untuk Admin*

Pada tampilan *Invoice*, admin dapat melihat kumpulan data mahasiswa yang melakukan pendaftaran bimbingan belajar. Data tersebut antara lain, tanggal pemesanan, kode kelas, nama mahasiswa, ID Line, jangka waktu bimbingan belajar, total dan *notes* pembayaran. *Invoice* digunakan admin untuk melakukan verifikasi pembayaran yang telah dilakukan oleh mahasiswa.

2.5 Coding Aplikasi

2.5.1 Android Studio

1. Splash screen



```
1 package project1.com.Main;
2
3 import ...
4
5 public class Splash extends Activity {
6
7     @Override
8     protected void onCreate(Bundle savedInstanceState) {
9         super.onCreate(savedInstanceState);
10        setContentView(R.layout.activity_splash);
11
12        Thread thread = run() -> {
13            try {
14                sleep( millis: 1500 );
15            } catch ( InterruptedException e ) {
16                e.printStackTrace();
17            } finally {
18                startActivity(new Intent( packageContext: Splash.this, LogIn.class));
19                finish();
20            }
21        };
22        thread.start();
23    }
24
25 }
26
27 }
28
29 }
30
31 }
32 }
```

Gambar 2.17 Code Splash Screen

Code diatas digunakan untuk menampilkan *splash screen* ketika pengguna masuk ke dalam aplikasi. Interval waktu yang digunakan untuk menampilkan *splash screen* adalah 1,5 detik, setelah itu pengguna akan langsung diarahkan ke halaman *Log In*. *Code Finish()* digunakan dengan tujuan pengguna tidak akan kembali ke halaman *splash screen* saat pengguna telah masuk ke halaman utama. *Code sleep()* berfungsi untuk mengatur lamanya waktu *splash screen* atau membekukannya sesuai waktu yang ditentukan.

2. Login

```
public class LogIn extends AppCompatActivity {

    private EditText edtEmail, edtPassword;
    private View btnSignin;
    final String URL_Login = "https://bimbelhimsi.000webhostapp.com/BIMBEL/SignIn.php";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_log_in);
        setTitle("Log In");

        edtEmail = findViewById(R.id.edtEmail);
        edtPassword = findViewById(R.id.edtPassword);
        btnSignin = findViewById(R.id.btnSignin);

        btnSignin.setOnClickListener(new View.OnClickListener(){
            @Override
            public void onClick(View v) {
                String iEmail = edtEmail.getText().toString().trim();
                String iPassword = edtPassword.getText().toString().trim();

                if(!iEmail.isEmpty() || !iPassword.isEmpty()){
                    SignIn(iEmail,iPassword);
                } else {
                    edtEmail.setError("Please insert valid email");
                    edtPassword.setError("Please Insert valid password");
                }
            }
        });
        TextView textView = findViewById(R.id.notmember);
        String text = "Not a member yet ? Sign up here";
        SpannableString ss = new SpannableString(text);

        ClickableSpan clickableSpan1 = new ClickableSpan() {
            @Override
            public void onClick(View v) {
                Intent i = new Intent(getApplicationContext(), SignUp.class);
                startActivity(i);
            }

            @Override
            public void updateDrawState(TextPaint ds) {
                super.updateDrawState(ds);
                ds.setColor(Color.BLACK);
                ds.setUnderlineText(true);
            }
        };
        ss.setSpan(clickableSpan1, start: 19, end: 31, Spanned.SPAN_EXCLUSIVE_EXCLUSIVE);

        textView.setText(ss);
        textView.setMovementMethod(LinkMovementMethod.getInstance());
    }

    private void SignIn(String email, String password){
        StringRequest stringRequest = new StringRequest(Request.Method.POST, URL_Login, new Response.Listener<String>() {
            @Override
            public void onResponse(String response) {
                try {
                    JSONObject jsonObject = new JSONObject(response);
                    String success = jsonObject.getString( name: "success");
                    JSONArray jsonArray = jsonObject.getJSONArray( name: "SignIn");
                }
            }
        });
    }
}
```

```

        if (success.equals("1")) {
            for (int i = 0; i < jsonArray.length(); i++) {
                Toast.makeText( context: LogIn.this, text: "Log In Success", Toast.LENGTH_SHORT).show();
                if (edtEmail.getText().toString().equals("adminbimsi@gmail.com") && edtPassword.getText().toString().equals("akuadminbimsi")) {
                    Intent intent = new Intent(getApplicationContext(), ReadAdmin.class);
                    startActivity(intent);
                } else {
                    Intent intent = new Intent(getApplicationContext(), Button.class);
                    startActivity(intent);
                }
            }
        } else if (success.equals("0")) {
            Toast.makeText( context: LogIn.this, text: "Wrong Email/Password", Toast.LENGTH_SHORT).show();
        }
    } catch (JSONException e) {
        e.printStackTrace();
        Toast.makeText( context: LogIn.this, text: "error", Toast.LENGTH_SHORT).show();
    }
}

}, new Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError error) {
        Toast.makeText( context: LogIn.this, text: "Error"+error.toString(), Toast.LENGTH_SHORT).show();
    }
});

{
    @Override
    protected Map<String, String> getParams() throws AuthFailureError {
        Map<String, String> params = new HashMap<>();
        params.put( k: "email",email);
        params.put( k: "password",password);
        return params;
    }
};

RequestQueue requestQueue = Volley.newRequestQueue( context: LogIn.this);
requestQueue.getCache().clear();
requestQueue.add(stringRequest);

}
}

```

Gambar 2.18 *Code Log In*

Code diatas digunakan untuk saat pengguna melakukan *log in*. *Code final String URL_Login = "<https://bimbelhimsi.000webhostapp.com/BIMBEL/SignIn.php>";* menandakan bahwa data-data yang akan dimasukkan pengguna pada halaman *log in* merupakan data yang telah tersimpan di *web hosting* secara *online* atau *cloud*. Menu *Log In* dapat digunakan oleh pengguna dan admin. Pengguna akan memasukkan data-data yang diminta, seperti *email* dan *password*. Untuk admin, email dan password telah ditentukan sehingga admin hanya perlu melakukan *log in* tanpa perlu *sign up*. Sedangkan pengguna harus memasukkan email dan password yang benar, sehingga saat pengguna menekan tombol *log in* akan muncul *toast* “*Log In Successful*”. Namun, saat pengguna memasukkan email dan password yang tidak valid maka muncul *toast* “*Wrong Email/Password*”.

3. Sign Up

The screenshot shows the Android Studio code editor with the Java file `Main.java` open. The code implements a `SignUp` activity that handles user input and sends data via a POST request to a PHP endpoint. The code includes imports for `AppCompatActivity`, `EditText`, `View`, and `Handler`. It defines fields for `edtNamaFull`, `edtNickname`, `edtEmail`, `edtPassword`, `edtPhone`, and `edtYoc`, and a button `btnSignup`. The `onCreate` method sets up the view and initializes these fields. The `btnSignup` button's `onClick` listener checks if all fields are filled and non-empty. If so, it creates a `Handler` to post a `Runnable` that constructs a `String[] field` and a `String[] data`. It then creates a `PutData` object with the URL `"https://bimbelhimsi.000webhostapp.com/BIMBEL/SignUp.php"` and method `"POST"`. The `putData.startPut()` method is called, followed by `putData.onComplete()`. Inside `onComplete`, the result is checked for success. If successful, a `Toast` is shown, and the user is directed to the `LogIn` activity. If unsuccessful, a `Toast` indicates that all fields must be filled.

```
1 package projectc1.com.Main;
2
3 import ...
4
5
6 public class SignUp extends AppCompatActivity {
7     EditText edtNamaFull, edtNickname, edtEmail, edtPassword, edtPhone, edtYoc;
8     View btnSignup;
9
10
11     @Override
12     protected void onCreate(Bundle savedInstanceState) {
13         super.onCreate(savedInstanceState);
14         setContentView(R.layout.activity_sign_up);
15         setTitle("Sign Up");
16
17         edtNamaFull = findViewById(R.id.edtNamaFull);
18         edtNickname = findViewById(R.id.edtNickname);
19         edtEmail = findViewById(R.id.edtEmail);
20         edtPassword = findViewById(R.id.edtPassword);
21         edtPhone = findViewById(R.id.edtPhone);
22         edtYoc = findViewById(R.id.edtYoc);
23
24         btnSignup = findViewById(R.id.btnSignup);
25
26         btnSignup.setOnClickListener(new View.OnClickListener() {
27             @Override
28             public void onClick(View v) {
29                 String NamaFull, nickname, email, password, phone, yoc;
30                 NamaFull = String.valueOf(edtNamaFull.getText());
31                 nickname = String.valueOf(edtNickname.getText());
32                 email = String.valueOf(edtEmail.getText());
33                 password = String.valueOf(edtPassword.getText());
34                 phone = String.valueOf(edtPhone.getText());
35                 yoc = String.valueOf(edtYoc.getText());
36                 if(!NamaFull.equals("") && !nickname.equals("") && !email.equals("") && !password.equals("") && !phone.equals("") && !yoc.equals("") ) {
37
38                     Handler handler = new Handler(Looper.getMainLooper());
39                     handler.post(new Runnable() {
40                         @Override
41                         public void run() {
42                             String [] field = new String[6];
43                             field[0] = "NamaFull";
44                             field[1] = "nickname";
45                             field[2] = "email";
46                             field[3] = "password";
47                             field[4] = "phone";
48                             field[5] = "yoc";
49
50                             String[] data = new String[6];
51                             data[0] = NamaFull;
52                             data[1] = nickname;
53                             data[2] = email;
54                             data[3] = password;
55                             data[4] = phone;
56                             data[5] = yoc;
57
58                             PutData putData = new PutData ( url: "https://bimbelhimsi.000webhostapp.com/BIMBEL/SignUp.php", method: "POST",field, data);
59                             if(putData.startPut()) {
60                                 if(putData.onComplete()) {
61                                     String result = putData.getResult();
62                                     if (result.equals("SignUp Success")) {
63                                         Toast.makeText(getApplicationContext(),result,Toast.LENGTH_SHORT).show();
64                                     }else {
65                                         Toast.makeText(getApplicationContext(), result, Toast.LENGTH_SHORT).show();
66                                         Intent intent = new Intent(getApplicationContext(), LogIn.class);
67                                         startActivity(intent);
68                                     }
69                                 }
70                             }
71                         }
72                     });
73                 } else {
74                     Toast.makeText(getApplicationContext(), text: "All fields must be filled", Toast.LENGTH_SHORT).show();
75                 }
76             });
77         });
78     }
79
80
81
82
83     });
84 } else {
85     Toast.makeText(getApplicationContext(), text: "All fields must be filled", Toast.LENGTH_SHORT).show();
86 }
87
88 });
89 }
90 }
```

Gambar 2.19 Code Sign Up

Code diatas digunakan untuk saat pengguna melakukan *sign in*. *Code PutData putData = new PutData ("https://bimbelhimsi.000webhostapp.com/BIMBEL/SignUp.php", "POST", field, data);* menandakan bahwa data-data yang akan dimasukkan pengguna pada halaman *sign up* merupakan data yang akan disimpan pada *web hosting* secara *online* atau *cloud*. Menu *Sign Up* hanya dapat digunakan oleh pengguna, dimana pengguna akan memasukkan data-data yang diminta, seperti *full name*, *nickname*, *email*, *password*, nomor telepon, dan tahun angkatan. Pengguna harus mendaftar menggunakan email yang valid, jika tidak akan muncul *toast* “*User already registered or email invalid, Sign up failed*”. Selain itu pengguna juga perlu untuk mengisi seluruh data-data, ketika pengguna mengosongkan salah satu data akan muncul *toast* “*All field must be filled*”. Saat pengguna telah memasukkan seluruh data dengan benar dan menekan tombol *sign up* akan muncul *toast* “*Sign Up Successful*”.

4. Main Activity

```
1 package projectc1.com.Main;
2
3 import ...
4
5
6 public class MainActivity extends Fragment {
7
8
9     @Override
10    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
11        View viewMain = inflater.inflate(R.layout.fragment_main, container, attachToRoot: false);
12
13        ImageButton btnDataanal = (ImageButton) viewMain.findViewById(R.id.imageDataanal);
14        btnDataanal.setOnClickListener(new View.OnClickListener() {
15            @Override
16            public void onClick(View v) {
17                Toast.makeText(getApplicationContext(), text: "Move to Data Analysis!", Toast.LENGTH_LONG).show();
18                Intent i = new Intent(getApplicationContext(), dataanalysis.class);
19                startActivity(i);
20            }
21        });
22
23        ImageButton btnAccounting = (ImageButton) viewMain.findViewById(R.id.imageAccounting);
24        btnAccounting.setOnClickListener(new View.OnClickListener() {
25            @Override
26            public void onClick(View v) {
27                Toast.makeText(getApplicationContext(), text: "Move to Accounting!", Toast.LENGTH_LONG).show();
28                Intent i = new Intent(getApplicationContext(), accounting.class);
29                startActivity(i);
30            }
31        });
32    }
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
```

```

49
50
51
52 ImageButton btnDatabase = (ImageButton) viewMain.findViewById(R.id.imageDatabase);
53 btnDatabase.setOnClickListener(new View.OnClickListener() {
54     @Override
55     public void onClick(View v) {
56         Toast.makeText(getApplicationContext(), text "Move to Database System!", Toast.LENGTH_LONG).show();
57         Intent i = new Intent(getApplicationContext(), database.class);
58         startActivity(i);
59     }
60 });
61
62 ImageButton btnAlgoritm = (ImageButton) viewMain.findViewById(R.id.imageAlgorithm);
63 btnAlgoritm.setOnClickListener(new View.OnClickListener() {
64     @Override
65     public void onClick(View v) {
66         Toast.makeText(getApplicationContext(), text "Move to Algorithm and Data Structure!", Toast.LENGTH_LONG).show();
67         Intent i = new Intent(getApplicationContext(), algorithm.class);
68         startActivity(i);
69     }
70 });
71
72 ImageButton btnMobile = (ImageButton) viewMain.findViewById(R.id.imageMobileapp);
73 btnMobile.setOnClickListener(new View.OnClickListener() {
74     @Override
75     public void onClick(View v) {
76         Toast.makeText(getApplicationContext(), text "Move to Mobile App Development!", Toast.LENGTH_LONG).show();
77         Intent i = new Intent(getApplicationContext(), mobileapp.class);
78         startActivity(i);
79     }
80 });
81
82 ImageButton btnStudentNote = (ImageButton) viewMain.findViewById(R.id.imgStudentNote);
83 btnStudentNote.setOnClickListener(new View.OnClickListener() {
84     @Override
85     public void onClick(View v) {
86         Toast.makeText(getApplicationContext(), text "Move to Student Note!", Toast.LENGTH_LONG).show();
87         Intent i = new Intent(getApplicationContext(), StudentNotes.class);
88         startActivity(i);
89     }
90 });
91
92 ImageView btnlogout = (ImageView) viewMain.findViewById(R.id.btnlogout);
93 btnlogout.setOnClickListener(new View.OnClickListener() {
94     @Override
95     public void onClick(View v) {
96         Toast.makeText(getApplicationContext(), text Anda Berhasil Keluar Dari Aplikasi!, Toast.LENGTH_LONG).show();
97         Intent i = new Intent(getApplicationContext(), LogIn.class);
98         startActivity(i);
99     }
100 });
101
102 return viewMain;
103 }
104 }
105 }

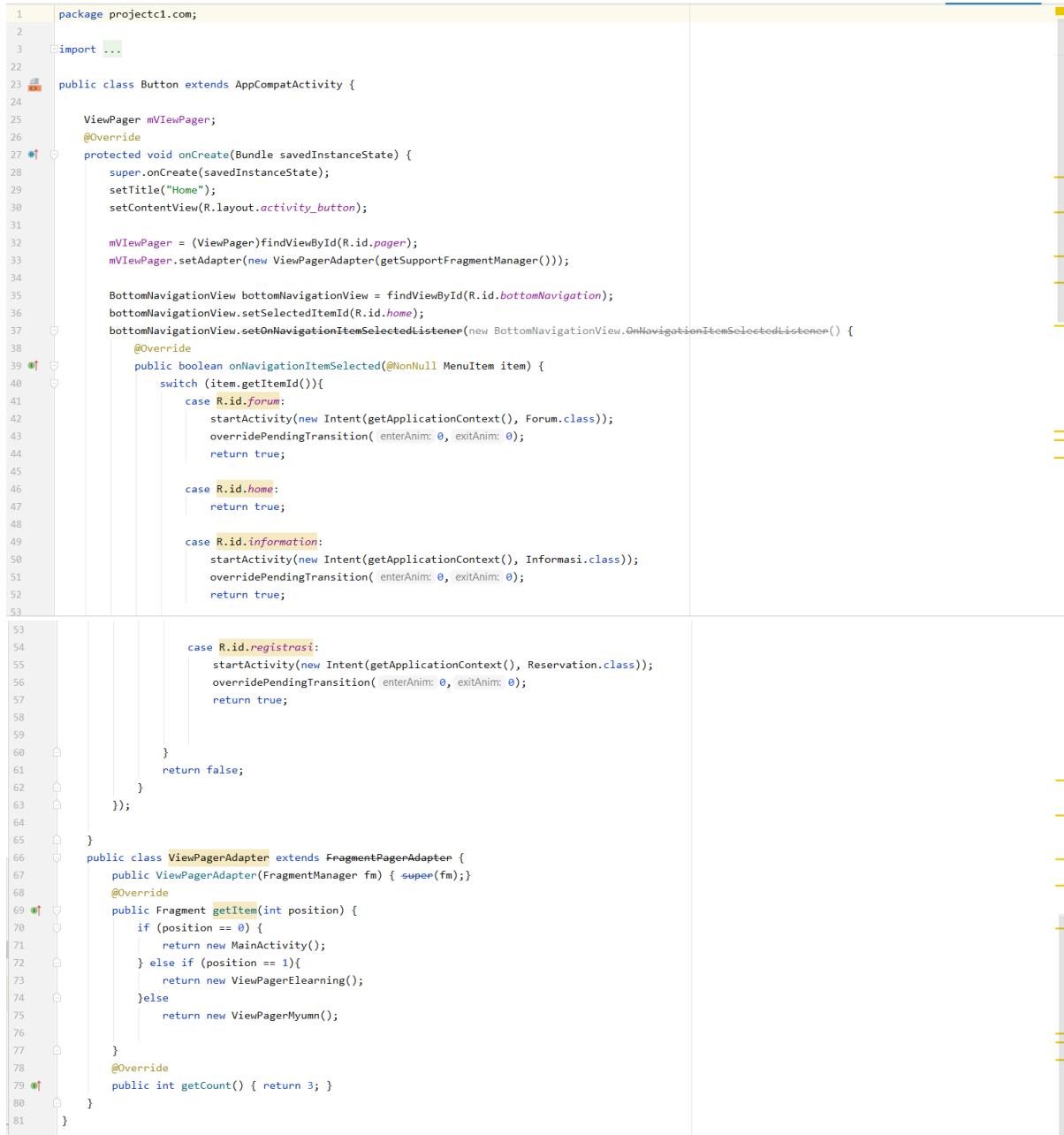
```

Gambar 2.20 Code Main Activity

Fragment MainActivity project aplikasi BIMSI dimuat atau ditampilkan pada *class Button*. Fragment ini memuat *ImageButton* dari ikon mata kuliah yang disediakan oleh BIMSI, seperti *Data Analysis*, *Accounting*, *Database System*, *Algorithm and Structure Data*, dan *Mobile Application Development*, dimana ketika pengguna menekan *ImageButton*, akan berpindah ke tiap kelas yang menampilkan penjelasan-penjelasan mengenai mata kuliah tersebut dan muncul *toast*, seperti “Move to Algorithm and Data Structure!”. Pada *fragment mainactivity* juga dimuat ikon untuk *log out* akun yang dapat digunakan ketika pengguna ingin keluar dari aplikasi. Pengguna hanya perlu menekan *ImageView* yang berada pada kiri atas *layout*, dimana akan muncul *toast* “Anda berhasil keluar dari aplikasi”. Selain itu terdapat

ImageButton Student Note yang ketika ditekan akan menampilkan kelas *Student Note* dan memunculkan *toast* “Move to Student Note!”.

5. Button menu utama : Home, Forum, Information, Reservation



```
1 package projectc1.com;
2
3 import ...
4
5 public class Button extends AppCompatActivity {
6
7     ViewPager mViewPager;
8     @Override
9     protected void onCreate(Bundle savedInstanceState) {
10         super.onCreate(savedInstanceState);
11         setTitle("Home");
12         setContentView(R.layout.activity_button);
13
14         mViewPager = (ViewPager)findViewById(R.id.pager);
15         mViewPager.setAdapter(new ViewPagerAdapter(getSupportFragmentManager()));
16
17         BottomNavigationView bottomNavigationView = findViewById(R.id.bottomNavigation);
18         bottomNavigationView.setSelectedItemId(R.id.home);
19         bottomNavigationView.setOnNavigationItemSelected(new BottomNavigationView.OnNavigationItemSelectedListener() {
20             @Override
21             public boolean onNavigationItemSelected(@NonNull MenuItem item) {
22                 switch (item.getItemId()){
23                     case R.id.forum:
24                         startActivity(new Intent(getApplicationContext(), Forum.class));
25                         overridePendingTransition(enterAnim: 0, exitAnim: 0);
26                         return true;
27
28                     case R.id.home:
29                         return true;
30
31                     case R.id.information:
32                         startActivity(new Intent(getApplicationContext(), Informasi.class));
33                         overridePendingTransition(enterAnim: 0, exitAnim: 0);
34                         return true;
35
36                     case R.id.registrasi:
37                         startActivity(new Intent(getApplicationContext(), Reservation.class));
38                         overridePendingTransition(enterAnim: 0, exitAnim: 0);
39                         return true;
40
41                 }
42                 return false;
43             }
44         });
45
46     }
47
48     public class ViewPagerAdapter extends FragmentPagerAdapter {
49         public ViewPagerAdapter(FragmentManager fm) { super(fm); }
50         @Override
51         public Fragment getItem(int position) {
52             if (position == 0) {
53                 return new MainActivity();
54             } else if (position == 1){
55                 return new ViewPagerElearning();
56             }else
57                 return new ViewPagerMyumn();
58
59         }
60         @Override
61         public int getCount() { return 3; }
62     }
63 }
64
65 }
```

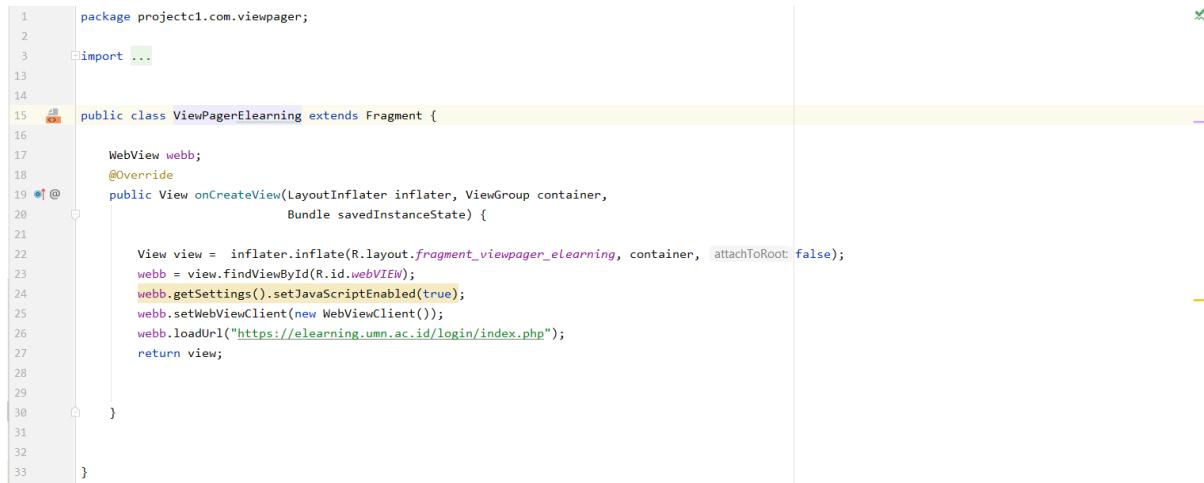
Gambar 2.21 Code Button Menu Utama

Fragment *Button* memuat halaman menu *Home, Forum, Information, dan Reservation*.

Dalam fragment ini terdapat code *ButtonNavigationView* yang berfungsi untuk menampilkan tampilan *button* pada bagian bawah aplikasi, dimana saat pengguna berada di halaman menu tertentu seperti *Home, button* itulah yang akan lebih *pop up* atau timbul dibanding button yang

lain. Selain khususnya pada halaman menu *Home* terdapat *ViewPager* yang digunakan untuk berpindah dari masing-masing halaman menu. *ViewPager* diset untuk mengembalikan atau *return* dengan nilai 3, karena menu home memuat tampilan terkait data mata kuliah bimbel dan *student notes* pada *viewpager* ke-1, elearning pada *viewpager* ke-2, dan myumn pada *viewpager* ke-3.

6. View Pager Elearning



```
1 package projectc1.com.viewpager;
2
3 import ...
4
5 public class ViewPagerElearning extends Fragment {
6
7     WebView webb;
8     @Override
9     public View onCreateView(LayoutInflater inflater, ViewGroup container,
10                             Bundle savedInstanceState) {
11
12         View view = inflater.inflate(R.layout.fragment_viewpager_elearning, container, attachToRoot: false);
13         webb = view.findViewById(R.id.webVIEW);
14         webb.getSettings().setJavaScriptEnabled(true);
15         webb.setWebViewClient(new WebViewClient());
16         webb.loadUrl("https://elearning.umn.ac.id/login/index.php");
17         return view;
18
19     }
20
21 }
22
23
24
25
26
27
28
29
30
31
32
33 }
```

Gambar 2.22 Code View Pager E-learning

Code diatas ini merupakan *code* untuk menampilkan *website* <https://elearning.umn.ac.id/my/>, yang menggunakan fungsi *WebView* dengan menyatakan *link* dari *website* yang diinginkan. Agar *WebView* dapat diakses, pengguna dapat menggeser layar tampilan *Home*, dalam kode menggunakan fungsi *ViewPager*. *ViewPager* disini hanya berbentuk *fragment*, karena *code ViewPager* yang sebenarnya berada di *class Button*.

7. View Pager MyUMN

```
1 package projectc1.com.viewpager;
2
3 import ...
4
5
6 public class ViewPagerAdapterMyumn extends Fragment {
7     ...
8     WebView webb;
9     @Override
10    public View onCreateView(LayoutInflater inflater, ViewGroup container,
11        Bundle savedInstanceState) {
12
13        View view = inflater.inflate(R.layout.fragment_view_pager_myumn, container, false);
14        webb = view.findViewById(R.id.webVIEW);
15        webb.getSettings().setJavaScriptEnabled(true);
16        webb.setWebViewClient(new WebViewClient());
17        webb.loadUrl("https://my.umn.ac.id/");
18        return view;
19
20    }
21
22 }
23
24
25
26
27
28
29
30
31
32
33
34 }
```

Gambar 2.23 Code View Pager My UMN

Code diatas ini merupakan *code* untuk menampilkan website <https://my.umn.ac.id/psp/ps/?cmd=login&languageCd=ENG&>, yang menggunakan fungsi *WebView* dengan menyatakan *link* dari *website* yang diinginkan. Agar *WebView* dapat diakses, pengguna dapat menggeser layar tampilan *Home*, dalam kode menggunakan fungsi *ViewPager*. Sama seperti *ViewPager elearning*, *ViewPager* disini hanya berbentuk *fragment*, karena *code ViewPager* yang sebenarnya berada di *class Button*.

8. Mata Kuliah Data Analysis

```
1 package projectc1.com.Matakuliah;
2
3 import ...
4
5
6 public class Dataanalisis extends AppCompatActivity {
7     ...
8     @Override
9     protected void onCreate(Bundle savedInstanceState) {
10         super.onCreate(savedInstanceState);
11         setContentView(R.layout.activity_dataanalisis);
12         final Spinner spinJadwal = findViewById(R.id.spinJadwal);
13         Button reserve = findViewById(R.id.reserve);
14
15         reserve.setOnClickListener(new View.OnClickListener() {
16             ...
17             @Override
18             public void onClick(View view) {
19                 Toast.makeText(getApplicationContext(), "Pindah Invoice untuk Reserve " + spinJadwal.getSelectedItem(), Toast.LENGTH_SHORT).show();
20                 startActivity(new Intent(getApplicationContext(), Reservation.class));
21             }
22         });
23     }
24 }
25
26
27
28
29
30
31
32 }
```

Gambar 2.24 Code Mata Kuliah Data Analysis

Code diatas merupakan *code* pada tampilan deskripsi dan jadwal bimbel mata kuliah Data Analysis, dimana pada tampilan ini deskripsi tentang mata kuliah Data Analysis yang telah di input pada *layout activity_dataanalysis.xml*. Dalam *code* diatas digunakan fungsi

Spinner agar tampilan pada jadwal bimbel dapat ditampilkan secara menurun kebawah dan juga menginisialisasi nama *button Reserve*. Fungsi *Toast* dalam *.setOnClickListener* digunakan untuk memberikan informasi "Pindah Invoice untuk Reserve" dan jadwal dari *Spinner* yang telah dipilih, ketika *button Reserve* ditekan. Dalam fungsi *.setOnClickListener* juga dinyatakan *startActivity* ke *class* lain sehingga saat *button* ditekan maka tampilan akan berpindah ke menu *Reservation*.

9. Mata Kuliah Accounting



```

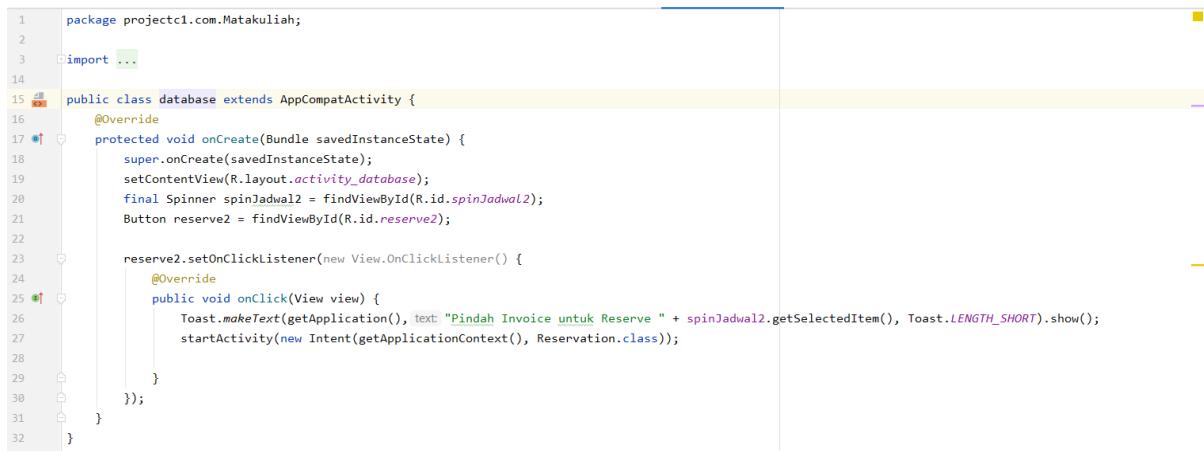
1 package projectc1.com.Matakuliah;
2
3 import ...
4
5 public class accounting extends AppCompatActivity {
6
7     @Override
8     protected void onCreate(Bundle savedInstanceState) {
9         super.onCreate(savedInstanceState);
10        setContentView(R.layout.activity_accounting);
11        final Spinner spinJadwal5 = findViewById(R.id.spinJadwal5);
12        Button reserve5 = findViewById(R.id.reserve5);
13
14        reserve5.setOnClickListener(new View.OnClickListener() {
15            @Override
16            public void onClick(View view) {
17                Toast.makeText(getApplicationContext(), text: "Pindah Invoice untuk Reserve " + spinJadwal5.getSelectedItem(), Toast.LENGTH_SHORT).show();
18                startActivity(new Intent(getApplicationContext(), Reservation.class));
19            }
20        });
21    }
22}

```

Gambar 2.25 Code Mata Kuliah Accounting

Code diatas merupakan *code* pada tampilan deskripsi dan jadwal bimbel mata kuliah Accounting, dimana pada tampilan ini deskripsi tentang mata kuliah Accounting yang telah di input pada *layout activity_accounting.xml*. Dalam *code* diatas digunakan fungsi *Spinner* agar tampilan pada jadwal bimbel dapat ditampilkan secara menurun kebawah dan juga menginisialisasi nama *button Reserve*. Fungsi *Toast* dalam *.setOnClickListener* digunakan untuk memberikan informasi "Pindah Invoice untuk Reserve" dan jadwal dari *Spinner* yang telah dipilih, ketika *button Reserve* ditekan. Dalam fungsi *.setOnClickListener* juga dinyatakan *startActivity* ke *class* lain sehingga saat *button* ditekan maka tampilan akan berpindah ke menu *Reservation*.

10. Mata Kuliah Database



```
1 package projectc1.com.Matakuliah;
2
3 import ...
4
5 public class database extends AppCompatActivity {
6     @Override
7     protected void onCreate(Bundle savedInstanceState) {
8         super.onCreate(savedInstanceState);
9         setContentView(R.layout.activity_database);
10        final Spinner spinJadwal2 = findViewById(R.id.spinJadwal2);
11        Button reserve2 = findViewById(R.id.reserve2);
12
13        reserve2.setOnClickListener(new View.OnClickListener() {
14            @Override
15            public void onClick(View view) {
16                Toast.makeText(getApplicationContext(), text: "Pindah Invoice untuk Reserve " + spinJadwal2.getSelectedItem(), Toast.LENGTH_SHORT).show();
17                startActivity(new Intent(getApplicationContext(), Reservation.class));
18            }
19        });
20    }
21
22 }
```

Gambar 2.26 Code Mata Kuliah Database

Code diatas merupakan *code* pada tampilan deskripsi dan jadwal bimbel mata kuliah Database, dimana pada tampilan ini deskripsi tentang mata kuliah Database yang telah di input pada *layout activity_database.xml*. Dalam *code* diatas digunakan fungsi *Spinner* agar tampilan pada jadwal bimbel dapat ditampilkan secara menurun kebawah dan juga menginisialisasi nama *button Reserve*. Fungsi *Toast* dalam *.setOnClickListener* digunakan untuk memberikan informasi "*Pindah Invoice untuk Reserve*" dan jadwal dari *Spinner* yang telah dipilih, ketika *button Reserve* ditekan. Dalam fungsi *.setOnClickListener* juga dinyatakan *startActivity* ke *class* lain sehingga saat *button* ditekan maka tampilan akan berpindah ke menu *Reservation*.

11. Mata Kuliah Algorithm



```
1 package projectc1.com.Matakuliah;
2
3 import ...
4
5 public class algorithm extends AppCompatActivity {
6     @Override
7     protected void onCreate(Bundle savedInstanceState) {
8         super.onCreate(savedInstanceState);
9         setContentView(R.layout.activity_algorithm);
10        final Spinner spinJadwal4 = findViewById(R.id.spinJadwal4);
11        Button reserve4 = findViewById(R.id.reserve4);
12
13        reserve4.setOnClickListener(new View.OnClickListener() {
14            @Override
15            public void onClick(View view) {
16                Toast.makeText(getApplicationContext(), text: "Pindah Invoice untuk Reserve " + spinJadwal4.getSelectedItem(), Toast.LENGTH_SHORT).show();
17                startActivity(new Intent(getApplicationContext(), Reservation.class));
18            }
19        });
20    }
21
22 }
```

Gambar 2.27 Code Mata Kuliah Algorithm

Code diatas merupakan *code* pada tampilan deskripsi dan jadwal bimbel mata kuliah Algorithm and Data Structure, dimana pada tampilan ini deskripsi tentang mata kuliah

Algorithm and Data Structure yang telah di input pada *layout activity_algorithm.xml*. Dalam *code* diatas digunakan fungsi *Spinner* agar tampilan pada jadwal bimbel dapat ditampilkan secara menurun kebawah dan juga menginisialisasi nama *button Reserve*. Fungsi *Toast* dalam *.setOnClickListener* digunakan untuk memberikan informasi "Pindah Invoice untuk Reserve" dan jadwal dari *Spinner* yang telah dipilih, ketika *button Reserve* ditekan. Dalam fungsi *.setOnClickListener* juga dinyatakan *startActivity* ke *class* lain sehingga saat *button* ditekan maka tampilan akan berpindah ke menu *Reservation*.

12. Mata Kuliah Mobile App



```

1 package projectc1.com.Matakuliah;
2
3 import ...
4
5 public class mobileapp extends AppCompatActivity {
6     @Override
7     protected void onCreate(Bundle savedInstanceState) {
8         super.onCreate(savedInstanceState);
9         setContentView(R.layout.activity_mobile_app);
10        final Spinner spinJadwal3 = findViewById(R.id.spinJadwal3);
11        Button reserve3 = findViewById(R.id.reserve3);
12
13        reserve3.setOnClickListener(new View.OnClickListener() {
14            @Override
15            public void onClick(View view) {
16                Toast.makeText(getApplicationContext(), text: "Pindah Invoice untuk Reserve " + spinJadwal3.getSelectedItem(), Toast.LENGTH_SHORT).show();
17                startActivity(new Intent(getApplicationContext(), Reservation.class));
18            }
19        });
20    }
21
22 }
23
24
25
26
27
28
29
30
31
32

```

Gambar 2.28 *Code Mata Kuliah Mobile App*

Code diatas merupakan *code* pada tampilan deskripsi dan jadwal bimbel mata kuliah Mobile App, dimana pada tampilan ini deskripsi tentang mata kuliah Mobile App yang telah di input pada *layout activity_mobile_app.xml*. Dalam *code* diatas digunakan fungsi *Spinner* agar tampilan pada jadwal bimbel dapat ditampilkan secara menurun kebawah dan juga menginisialisasi nama *button Reserve*. Fungsi *Toast* dalam *.setOnClickListener* digunakan untuk memberikan informasi "Pindah Invoice untuk Reserve" dan jadwal dari *Spinner* yang telah dipilih, ketika *button Reserve* ditekan. Dalam fungsi *.setOnClickListener* juga dinyatakan *startActivity* ke *class* lain sehingga saat *button* ditekan maka tampilan akan berpindah ke menu *Reservation*.

13. dbHandlerNote

```
1 package projectc1.com.sqllite;
2
3 import ...
4
5
6 public class dbHandlerNote extends SQLiteOpenHelper {
7
8     //Deklarasi variabel untuk pembuatan database, dengan memuat tabel dan kolom yang diperlukan
9     private static final int DATABASE_VERSION = 1;
10    private static final String DATABASE_NAME = "studentNote.db";
11    private static final String TABLE_NAME = "studentNote";
12
13    private static final String COLUMN_ID = "_id";
14    private static final String COLUMN_NAMAPENGINGAT = "_namaPengingat";
15    private static final String COLUMN_PENGINGAT = "_pengingat";
16
17    public dbHandlerNote(Context context) { super(context, DATABASE_NAME, null, DATABASE_VERSION); }
18
19    //Method untuk Create Database
20    @Override
21    public void onCreate(SQLiteDatabase sqLiteDatabase) {
22        String CREATE_TABLE_STUDENTNOTE = "CREATE TABLE " + TABLE_NAME + "(" +
23            COLUMN_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, " +
24            COLUMN_NAMAPENGINGAT + " VARCHAR(50) NOT NULL, " +
25            COLUMN_PENGINGAT + " VARCHAR(500) NOT NULL)";
26
27        sqLiteDatabase.execSQL(CREATE_TABLE_STUDENTNOTE);
28    }
29
30    //Method yang dipakai untuk upgrade tabel
31    @Override
32    public void onUpgrade(SQLiteDatabase sqLiteDatabase, int i, int ii) {
33        sqLiteDatabase.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME);
34        onCreate(sqLiteDatabase);
35    }
36
37
38
39
40
41
42
43
44
45    //Insert, Select, Update, Delete //
46
47    private SQLiteDatabase database;
48
49    //Method untuk open database connection
50    public void open() throws SQLException {
51        database = this.getWritableDatabase();
52    }
53
54    //Inisialisasi semua kolom di tabel database
55    private String[] allColumns =
56        {COLUMN_ID, COLUMN_NAMAPENGINGAT, COLUMN_PENGINGAT};
57
58    //Method untuk memindahkan isi cursor ke database sql lite
59    private StudentNote cursorToNote(Cursor cursor) {
60        StudentNote studentNote = new StudentNote();
61
62        studentNote.set_id(cursor.getLong(0));
63        studentNote.set_namaPengingat(cursor.getString(1));
64        studentNote.set_pengingat(cursor.getString(2));
65
66        return studentNote;
67    }
68
69    //Method untuk menambahkan Student Note baru
70    public void createNote(String namaPengingat, String pengingat) {
71        ContentValues values = new ContentValues();
72        values.put(COLUMN_NAMAPENGINGAT, namaPengingat);
73        values.put(COLUMN_PENGINGAT, pengingat);
74
75        database.insert(TABLE_NAME, nullColumnHack, values);
76    }
```

```

76
77
78 //Method untuk mendapatkan detail Mengenai pengingat yang ditambahkan
79 public StudentNote getNote(long id) {
80     StudentNote studentNote = new StudentNote();
81
82     Cursor cursor =
83         database.query(TABLE_NAME, allColumns, selection: "_id=" + id, selectionArgs: null, groupBy: null, having: null, orderBy: null);
84     cursor.moveToFirst();
85     studentNote = cursorToNote(cursor);
86     cursor.close();
87
88     return studentNote;
89 }
90
91 //Method untuk mendapatkan data - data yang diinput kedalam database sql Lite
92 public ArrayList<StudentNote> getStudentNote() {
93     ArrayList<StudentNote> daftarNote = new ArrayList<>();
94
95     Cursor cursor =
96         database.query(TABLE_NAME, allColumns, selection: null, selectionArgs: null, groupBy: null, having: null, orderBy: null);
97     cursor.moveToFirst();
98
99     while (!cursor.isAfterLast()) {
100         StudentNote Notes = cursorToNote(cursor);
101         daftarNote.add(Notes);
102         cursor.moveToNext();
103     }
104
105     cursor.close();
106     return daftarNote;
107 }
108
109 //Method untuk melakukan update pada data student note
110 @
111 public void updatePengingat(StudentNote Notes) {
112     String filter = "_id=" + Notes.get_id();
113     ContentValues args = new ContentValues();
114     args.put(COLUMN_NAMAPENGINGAT, Notes.get_namaPengingat());
115     args.put(COLUMN_PENGINGAT, Notes.get_pengingat());
116
117     database.update(TABLE_NAME, args, filter, whereArgs: null);
118 }
119
120 //Method untuk menghapus data pada baris yang diinput pada student note
121 public void deleteNote(long id) {
122     String filter = "_id=" + id;
123
124     database.delete(TABLE_NAME, filter, whereArgs: null);
125 }
126

```

Gambar 2.29 dbHandlerNote

Code diatas merupakan *handler* yang digunakan dalam membuat tabel database bernama studentNote.db menggunakan database lokal SQLite pada Android Studio. Dalam database studentNote.db, terdapat tabel studentNote yang memuat atribut id, namapengingat, dan pengingat. Pada *class* yang sama, beberapa fungsi operasi *back-end* juga dimuat untuk melakukan memasukkan data baru, mendapatkan data dari database SQLite dengan penggunaan *cursor*, melakukan pembaruan data, dan melakukan penghapusan *notes* atau pengingat berdasarkan id.

14. Edit Notes

```
1 package projectc1.com.sqllite;
2
3 import ...
4
5
6 public class EditNotes extends AppCompatActivity {
7     private long id;
8     private String namaPengingat;
9     private String pengingat;
10
11     @Override
12     protected void onCreate(Bundle savedInstanceState) {
13         super.onCreate(savedInstanceState);
14         setContentView(R.layout.activity_edit_notes);
15         final EditText edtNamaPengingat = (EditText)findViewById(R.id.edtNamaPengingat);
16         final EditText edtPengingat = (EditText)findViewById(R.id.edtPengingat);
17         Button btnKembali = (Button)findViewById(R.id.btnBatal);
18         Button btnEditNotes = (Button)findViewById(R.id.btnEditNotes);
19
20
21         //Buat objek untuk Class MyDBHandler
22         final dbHandlerNote MydbHandler = new dbHandlerNote( context: this);
23
24         //Membuka koneksi database
25         try{
26             MydbHandler.open();
27         }catch (SQLException e){
28             e.printStackTrace();
29         }
30
31
32         Bundle bundle = this.getIntent().getExtras();
33         id = bundle.getLong( key: "id");
34         namaPengingat = bundle.getString( key: "namaPengingat");
35         pengingat = bundle.getString( key: "Pengingat");
36
37         this.setTitle("Edit Notes ID: "+id);
38         edtNamaPengingat.setText(namaPengingat);
39         edtPengingat.setText(pengingat);
40
41
42         btnEditNotes.setOnClickListener(new View.OnClickListener() {
43             @Override
44             public void onClick(View view) {
45                 StudentNote StudentNote = new StudentNote();
46                 StudentNote.set_id(id);
47                 StudentNote.set_namaPengingat(edtNamaPengingat.getText().toString());
48                 StudentNote.set_pengingat(edtPengingat.getText().toString());
49                 MydbHandler.updatePengingat(StudentNote);
50
51                 Toast.makeText( context: EditNotes.this, text: "Student Note berhasil diedit", Toast.LENGTH_LONG).show();
52
53                 Intent i = new Intent( packageContext: EditNotes.this, StudentNotes.class);
54                 startActivity(i);
55                 EditNotes.this.finish();
56                 MydbHandler.close();
57             }
58         });
59
60         btnKembali.setOnClickListener(new View.OnClickListener() {
61             @Override
62             public void onClick(View view) {
63                 Intent i = new Intent( packageContext: EditNotes.this, StudentNotes.class);
64                 startActivity(i);
65                 EditNotes.this.finish();
66                 MydbHandler.close();
67             }
68         });
69
70     }
71
72 }
73
74
75
76 }
77 }
```

Gambar 2.30 Code Edit Notes

Code diatas merupakan *class EditNotes*, dimana terdapat beberapa elemen yang digunakan pada *class* ini, yaitu *button*, *EditText* dan *dbHandler*. Menu *EditNotes* digunakan

untuk mengubah catatan yang telah dibuat sebelumnya. Terdapat *code* yang digunakan untuk menghubungkan aplikasi dengan *database* lokal, yaitu *SQLLite*. Ketika pengguna menekan *button edit*, aplikasi akan menyimpan data yang telah diinput oleh pengguna dan muncul *toast* “*Student Note berhasil di edit*”. *Toast* ini berarti isi catatan akan berubah sesuai dengan input pengguna dan menggantikan catatan sebelumnya. Data yang diinput oleh pengguna akan ditampung pada *SQLLite*, sehingga data tersebut hanya bisa dilihat pada perangkat tersebut dan tidak dapat dilihat pada perangkat lain. Saat pengguna menekan *button cancel*, aplikasi akan mengarahkan kembali pengguna ke halaman *Student Notes*.

15. StudentNote



```
1 package projectc1.com.sqlite;
2
3 import ...
4
5 public class StudentNote {
6     private long _id;
7     private String _namaPengingat;
8     private String _pingingat;
9
10    public StudentNote(){
11        ...
12    }
13
14    public long get_id() { return _id; }
15
16    public void set_id(long _id) { this._id = _id; }
17
18    public String get_namaPengingat() { return _namaPengingat; }
19
20    public void set_namaPengingat(String _namaPengingat) { this._namaPengingat = _namaPengingat; }
21
22    public String get_pingingat() { return _pingingat; }
23
24    public void set_pingingat(String _pingingat) { this._pingingat = _pingingat; }
25
26    @Override
27    public String toString() {
28        return "Nama Pengingat\t: " + _namaPengingat + "\nNotes\t: " + _pingingat ;
29    }
30
31}
```

Gambar 2.31 *Code Student Note*

Pada *code class StudentNote*, memuat metode *get* dan *set* berdasarkan variabel yang terdapat pada database *studentNote*. *Get* berfungsi untuk mendapatkan data. *Return* berfungsi untuk mengembalikan nilai dari suatu variabel. Sedangkan *set* berfungsi untuk memberikan nilai pada variabel yang diinginkan.

16. StudentNotes

```
1 package projectc1.com.sqlite;
2
3 import ...
4
5
6 public class StudentNotes extends ListActivity implements AdapterView.OnItemLongClickListener{
7     private dbHandlerNote MydbHandler;
8     private ArrayList<StudentNote> values;
9     private Button btnEditNotes;
10    private Button btnDeleteNotes;
11    private ListView list;
12    Context context = this;
13    ImageButton back;
14
15    @Override
16    protected void onCreate(Bundle savedInstanceState) {
17        super.onCreate(savedInstanceState);
18        setContentView(R.layout.activity_student_notes);
19
20        back = (ImageButton) findViewById(R.id.buttonBack);
21        back.setOnClickListener(new View.OnClickListener() {
22            @Override
23            public void onClick(View v) {
24                Intent Back = new Intent(getApplicationContext(), projectc1.com.Button.class);
25                startActivity(Back);
26            }
27        });
28
29
30        //Buat objek untuk Class MyDBHandler
31        MydbHandler = new dbHandlerNote(context: this);
32
33        //Membuka koneksi database
34        try{
35            ...
36
37            try{
38                MydbHandler.open();
39            }catch (SQLException e){
40                e.printStackTrace();
41            }
42
43            values = MydbHandler.getStudentNote();
44
45            ArrayAdapter<StudentNote> adapter = new ArrayAdapter<>(context: this, android.R.layout.simple_list_item_1, values);
46
47            setListAdapter(adapter);
48
49            list = (ListView)findViewById(android.R.id.list);
50            list.setOnItemLongClickListener(this);
51
52
53            FloatingActionButton fab = (FloatingActionButton)findViewById(R.id.floatingActionButton);
54            fab.setOnClickListener(new View.OnClickListener() {
55                @Override
56                public void onClick(View view) {
57                    Intent i = new Intent(getApplicationContext(), tambah_studentnote.class);
58                    startActivity(i);
59                }
60            });
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
//Method yang digunakan jika ListView ditekan agak lama
@Override
public boolean onItemLongClick(AdapterView<> adapterView, View view, int i, long l) {
    //Menampilkan dialog dan mengambil Layout dari dialog.xml
    final Dialog dialog = new Dialog(context: this);
    dialog.setContentView(R.layout.dialog);
    dialog.setTitle("Choose one");
    dialog.show();

    final StudentNote studentNote = (StudentNote) getListAdapter().getItem(i);
    final long id = studentNote.get_id();

    btnEditNotes = dialog.findViewById(R.id.btnEditDialog);
    btnDeleteNotes = dialog.findViewById(R.id.btnDelete);

    //Method yang digunakan jika Button Edit pada dialog.xml ditekan
```

```
106
107     btnEditNotes.setOnClickListener(new View.OnClickListener() {
108         @Override
109         public void onClick(View view) {
110             StudentNote Note = MydbHandler.getNote(id);
111             Intent i = new Intent(getApplicationContext(), EditNotes.class);
112             Bundle bundle = new Bundle();
113             bundle.putLong("id", Note.get_id());
114             bundle.putString("namapengingat", Note.get_namaPengingat());
115             bundle.putString("Pengingat", Note.get_pengingat());
116             i.putExtras(bundle);
117             startActivity(i);
118             dialog.dismiss();
119         }
120     });
121
122 //Method yang digunakan jika Button Delete pada dialog.xml ditekan
123 btnDeleteNotes.setOnClickListener(new View.OnClickListener() {
124     @Override
125     public void onClick(View view) {
126         AlertDialog.Builder konfirm = new AlertDialog.Builder(context);
127         konfirm.setTitle("Hapus Student Note");
128         konfirm.setMessage("Anda yakin akan menghapus Student Note ini?");
129         konfirm.setPositiveButton(text: "Ya", new DialogInterface.OnClickListener() {
130             @Override
131             public void onClick(DialogInterface dialog, int which) {
132                 MydbHandler.deleteNote(id);
133             }
134         });
135         finish();
136         startActivity(getIntent());
137
138         Toast.makeText(context: StudentNotes.this, text: "Student Note berhasil dihapus", Toast.LENGTH_LONG).show();
139     }
140 });
141 konfirm.setNegativeButton(text: "Tidak", new DialogInterface.OnClickListener() {
142     @Override
143     public void onClick(DialogInterface dialog, int which) {
144
145     }
146 });
147 konfirm.show();
148 dialog.dismiss();
149
149
150     });
151     });
152     });
153 }
```

Gambar 2.32 *Code Student Notes*

Pada menu *StudentNotes*, pengguna aplikasi dapat melakukan fungsi CRUD (*Create, Read, Update, dan Delete*). *Class* ini berfungsi untuk membuat catatan atau pengingat bagi pengguna. Untuk menambahkan *note* atau catatan, pengguna dapat menekan tombol *floating action button* pada kanan bawah *layout* lalu pengguna akan langsung berpindah ke *class TambahStudentNote*. Fungsi *ListView* digunakan untuk menampilkan *note* yang sudah ditambahkan sebelumnya dalam bentuk daftar. Untuk mengubah atau menghapus catatan yang sudah ditambahkan, pengguna dapat menekan lama *note* yang ingin diubah lalu akan muncul dialog yang berisi *button* bernama *btnEditNotes* dan *btnDeleteNotes*. *btnEditNotes* berfungsi untuk mengubah catatan sedangkan *btnDeleteNotes* berfungsi untuk menghapus catatan yang telah ditambahkan sebelumnya. Saat pengguna menekan tombol *delete*, akan sebuah dialog

yang berbunyi “Anda yakin akan menghapus Student Note ini?” dan *Toast* yang berbunyi “Student Note berhasil dihapus” akan muncul ketika pengguna berhasil menghapus catatan.

17. Tambah Student Note

```

1 package projectc1.com.sqllite;
2
3 import ...
4
5
6 public class tambah_studentnote extends AppCompatActivity {
7     public static final int REQUEST_ADDNote = 999;
8     @Override
9     protected void onCreate(Bundle savedInstanceState) {
10         super.onCreate(savedInstanceState);
11         setContentView(R.layout.activity_tambah_studentnote);
12
13         final EditText edtPengingat = (EditText)findViewById(R.id.edtPengingat);
14         final EditText edtPesanan = (EditText)findViewById(R.id.edtPesanan);
15         Button btnCancel = (Button)findViewById(R.id.btnCancel);
16         Button btnAdd = (Button)findViewById(R.id.btnAdd);
17
18         //Buat objek untuk Class MyDBHandler
19         final dbHandlerNote MydbHandler = new dbHandlerNote( context: this );
20
21         //Membuka koneksi database
22         try{
23             MydbHandler.open();
24         }catch (SQLException e){
25             e.printStackTrace();
26         }
27
28         btnAdd.setOnClickListener(new View.OnClickListener() {
29             @Override
30             public void onClick(View view) {
31                 StudentNote studentNote = new StudentNote();
32                 String pengingat = edtPengingat.getText().toString();
33                 String pesan = edtPesanan.getText().toString();
34                 MydbHandler.createReview(pengingat,pesan);
35
36                 Toast.makeText( context: tambah_studentnote.this, text: "Student Note berhasil ditambahkan", Toast.LENGTH_LONG).show();
37                 edtPengingat.setText("");
38                 edtPesanan.setText("");
39                 edtPengingat.requestFocus();
40
41                 Intent i = new Intent( packageContext: tambah_studentnote.this, StudentNotes.class );
42                 startActivity(i);
43                 tambah_studentnote.this.finish();
44                 MydbHandler.close();
45             });
46
47         btnCancel.setOnClickListener(new View.OnClickListener() {
48             @Override
49             public void onClick(View view) {
50                 edtPengingat.setText("");
51                 edtPengingat.setText("");
52                 edtPengingat.requestFocus();
53
54                 Intent i = new Intent( packageContext: tambah_studentnote.this, StudentNotes.class );
55                 startActivity(i);
56                 tambah_studentnote.this.finish();
57                 MydbHandler.close();
58             });
59
60         });
61
62         btnAdd.setOnClickListener(new View.OnClickListener() {
63             @Override
64             public void onClick(View view) {
65                 StudentNote studentNote = new StudentNote();
66                 String pengingat = edtPengingat.getText().toString();
67                 String pesan = edtPesanan.getText().toString();
68                 MydbHandler.createReview(pengingat,pesan);
69
70             });
71     }
72 }
```

Gambar 2.33 Code Tambah Student Note

Code pada class *tambah_studentnote.java* ini digunakan untuk mendapatkan nilai yang dikirim oleh pengguna ke database SQLite lokal yang terdapat pada perangkatnya. Aktivitas pengguna dalam menambahkan judul pengingat dan pesan kemudian akan ditampung ke dalam *plain text* dengan id *edtPengingat* dan *edtPesanan*. Agar dapat melakukan fungsi *insert* ke dalam

database SQLite, kita perlu membuka koneksi ke databasenya terlebih dahulu dengan fungsi *MydbHandler.open()*. *Button* dengan id *btnAdd* digunakan untuk menambah pengingat dengan mendapatkan dan mengirim data ke dalam databasenya, sedangkan *btnCancel* digunakan untuk membatalkan aktivitas menambah pengingat. Saat pengguna berhasil menambahkan suatu catatan, akan muncul sebuah *Toast* yang berbunyi “Student Note berhasil ditambahkan.”

18. Forum

```
1 package projectc1.com.forum;
2
3 import ...
4
5
6 public class Forum extends AppCompatActivity {
7
8     FloatingActionButton floatingActionButton;
9     ProgressBar progressBar2;
10    ListView lv;
11    SwipeRefreshLayout refreshLayout;
12    ArrayList<HashMap<String, String>> list_forum;
13    String URL_GET_FORUM="https://bimbelhimsi.000webhostapp.com/BIMBEL/getForum.php";
14
15    private static final String TAG_FORUM="data";
16    private static final String TAG_ID="id";
17    private static final String TAG_NAMA="nama";
18    private static final String TAG_POST = "post";
19
20    @Override
21    protected void onCreate(Bundle savedInstanceState) {
22        setTitle("Forum");
23        super.onCreate(savedInstanceState);
24        setContentView(R.layout.activity_forum);
25        BottomNavigationView bottomNavigationView = findViewById(R.id.bottomNavigation);
26
27        bottomNavigationView.setSelectedItemId(R.id.forum);
28        bottomNavigationView.setOnNavigationItemSelectedListener(new BottomNavigationView.OnNavigationItemSelectedListener() {
29            @Override
30            public boolean onNavigationItemSelected(@NonNull MenuItem item) {
31                switch (item.getItemId()){
32                    case R.id.forum:
33                        return true;
34
35                    case R.id.home:
36                        startActivity(new Intent(getApplicationContext(), Button.class));
37                        overridePendingTransition(enterAnim: 0, exitAnim: 0);
38                        return true;
39
40                    case R.id.information:
41                        startActivity(new Intent(getApplicationContext(), Informasi.class));
42                        overridePendingTransition(enterAnim: 0, exitAnim: 0);
43                        return true;
44
45                    case R.id.registrasi:
46                        startActivity(new Intent(getApplicationContext(), Reservation.class));
47                        overridePendingTransition(enterAnim: 0, exitAnim: 0);
48                        return true;
49
50                }
51                return false;
52            }
53        });
54
55        list_forum=new ArrayList<>();
56        lv=findViewById(R.id.listView);
57        progressBar2=findViewById(R.id.progressBar2);
58        floatingActionButton = findViewById(R.id.fab);
59        progressBar2.setVisibility(View.VISIBLE);
60
61    }
62
63    @Override
64    public boolean onCreateOptionsMenu(Menu menu) {
65        getMenuInflater().inflate(R.menu.menu_main, menu);
66        return true;
67    }
68
69    @Override
70    public boolean onOptionsItemSelected(MenuItem item) {
71        int id = item.getItemId();
72
73        if (id == R.id.action_settings) {
74            Intent intent = new Intent(this, SettingsActivity.class);
75            startActivity(intent);
76        }
77
78        return super.onOptionsItemSelected(item);
79    }
80
81    @Override
82    protected void onStart() {
83        super.onStart();
84
85        refreshLayout.setOnRefreshListener(new SwipeRefreshLayout.OnRefreshListener() {
86            @Override
87            public void onRefresh() {
88                refresh();
89            }
90        });
91
92        refresh();
93    }
94
95    private void refresh() {
96        list_forum.clear();
97        lv.setAdapter(null);
98        progressBar2.setVisibility(View.VISIBLE);
99
100        new Thread(new Runnable() {
101            @Override
102            public void run() {
103                try {
104                    URL url = new URL(URL_GET_FORUM);
105                    HttpURLConnection connection = (HttpURLConnection) url.openConnection();
106                    connection.setRequestMethod("GET");
107                    connection.setReadTimeout(5000);
108                    connection.setConnectTimeout(5000);
109
110                    BufferedReader reader = new BufferedReader(new InputStreamReader(connection.getInputStream()));
111                    String response = reader.readLine();
112
113                    Gson gson = new Gson();
114                    Type type = new TypeToken<ArrayList<HashMap<String, String>>(){};
115                    list_forum = gson.fromJson(response, type);
116
117                    runOnUiThread(new Runnable() {
118                        @Override
119                        public void run() {
120                            lv.setAdapter(new ForumAdapter(list_forum));
121                            progressBar2.setVisibility(View.GONE);
122                        }
123                    });
124
125                } catch (Exception e) {
126                    e.printStackTrace();
127                }
128            }
129        }).start();
130    }
131
132    private void insertData() {
133        String nama = namaEditText.getText().toString();
134        String post = postEditText.getText().toString();
135
136        if (!nama.isEmpty() && !post.isEmpty()) {
137            MydbHandler db = new MydbHandler(this);
138            db.insertData(nama, post);
139
140            Toast.makeText(this, "Student Note berhasil ditambahkan.", Toast.LENGTH_SHORT).show();
141
142            refresh();
143        } else {
144            Toast.makeText(this, "Nama dan Post tidak boleh kosong!", Toast.LENGTH_SHORT).show();
145        }
146    }
147
148    private void updateData() {
149        String nama = namaEditText.getText().toString();
150        String post = postEditText.getText().toString();
151
152        if (!nama.isEmpty() && !post.isEmpty()) {
153            MydbHandler db = new MydbHandler(this);
154            db.updateData(nama, post);
155
156            Toast.makeText(this, "Student Note berhasil diperbarui.", Toast.LENGTH_SHORT).show();
157
158            refresh();
159        } else {
160            Toast.makeText(this, "Nama dan Post tidak boleh kosong!", Toast.LENGTH_SHORT).show();
161        }
162    }
163
164    private void deleteData() {
165        String nama = namaEditText.getText().toString();
166
167        if (!nama.isEmpty()) {
168            MydbHandler db = new MydbHandler(this);
169            db.deleteData(nama);
170
171            Toast.makeText(this, "Student Note berhasil dihapus.", Toast.LENGTH_SHORT).show();
172
173            refresh();
174        } else {
175            Toast.makeText(this, "Nama tidak boleh kosong!", Toast.LENGTH_SHORT).show();
176        }
177    }
178
179    private void cancelData() {
180        namaEditText.setText("");
181        postEditText.setText("");
182    }
183
184    private void clearData() {
185        namaEditText.setText("");
186        postEditText.setText("");
187    }
188
189    private void showData() {
190        namaEditText.setText(list_forum.get(position).get(TAG_NAMA));
191        postEditText.setText(list_forum.get(position).get(TAG_POST));
192    }
193
194    private void editData() {
195        namaEditText.setEnabled(true);
196        postEditText.setEnabled(true);
197    }
198
199    private void cancelEditData() {
200        namaEditText.setEnabled(false);
201        postEditText.setEnabled(false);
202    }
203
204    private void showEditData() {
205        namaEditText.setText(list_forum.get(position).get(TAG_NAMA));
206        postEditText.setText(list_forum.get(position).get(TAG_POST));
207    }
208
209    private void cancelEdit() {
210        namaEditText.setText("");
211        postEditText.setText("");
212    }
213
214    private void showDeleteData() {
215        namaEditText.setText(list_forum.get(position).get(TAG_NAMA));
216    }
217
218    private void cancelDeleteData() {
219        namaEditText.setText("");
220    }
221
222    private void showCancelData() {
223        namaEditText.setText("");
224    }
225
226    private void cancelCancelData() {
227        namaEditText.setText("");
228    }
229
230    private void showClearData() {
231        namaEditText.setText("");
232    }
233
234    private void cancelClearData() {
235        namaEditText.setText("");
236    }
237
238    private void showEditCancelData() {
239        namaEditText.setText("");
240    }
241
242    private void cancelEditCancelData() {
243        namaEditText.setText("");
244    }
245
246    private void showEditCancelDeleteData() {
247        namaEditText.setText("");
248    }
249
250    private void cancelEditCancelDeleteData() {
251        namaEditText.setText("");
252    }
253
254    private void showEditCancelEditData() {
255        namaEditText.setText("");
256    }
257
258    private void cancelEditCancelEditData() {
259        namaEditText.setText("");
260    }
261
262    private void showEditCancelCancelData() {
263        namaEditText.setText("");
264    }
265
266    private void cancelEditCancelCancelData() {
267        namaEditText.setText("");
268    }
269
270    private void showEditCancelClearData() {
271        namaEditText.setText("");
272    }
273
274    private void cancelEditCancelClearData() {
275        namaEditText.setText("");
276    }
277
278    private void showEditCancelEditCancelData() {
279        namaEditText.setText("");
280    }
281
282    private void cancelEditCancelEditCancelData() {
283        namaEditText.setText("");
284    }
285
286    private void showEditCancelCancelCancelData() {
287        namaEditText.setText("");
288    }
289
290    private void cancelEditCancelCancelCancelData() {
291        namaEditText.setText("");
292    }
293
294    private void showEditCancelCancelEditData() {
295        namaEditText.setText("");
296    }
297
298    private void cancelEditCancelCancelEditData() {
299        namaEditText.setText("");
300    }
301
302    private void showEditCancelCancelEditCancelData() {
303        namaEditText.setText("");
304    }
305
306    private void cancelEditCancelCancelEditCancelData() {
307        namaEditText.setText("");
308    }
309
310    private void showEditCancelCancelCancelEditData() {
311        namaEditText.setText("");
312    }
313
314    private void cancelEditCancelCancelCancelEditData() {
315        namaEditText.setText("");
316    }
317
318    private void showEditCancelCancelCancelCancelData() {
319        namaEditText.setText("");
320    }
321
322    private void cancelEditCancelCancelCancelCancelData() {
323        namaEditText.setText("");
324    }
325
326    private void showEditCancelCancelCancelCancelEditData() {
327        namaEditText.setText("");
328    }
329
330    private void cancelEditCancelCancelCancelCancelEditData() {
331        namaEditText.setText("");
332    }
333
334    private void showEditCancelCancelCancelCancelCancelData() {
335        namaEditText.setText("");
336    }
337
338    private void cancelEditCancelCancelCancelCancelCancelData() {
339        namaEditText.setText("");
340    }
341
342    private void showEditCancelCancelCancelCancelCancelEditData() {
343        namaEditText.setText("");
344    }
345
346    private void cancelEditCancelCancelCancelCancelCancelEditData() {
347        namaEditText.setText("");
348    }
349
350    private void showEditCancelCancelCancelCancelCancelCancelData() {
351        namaEditText.setText("");
352    }
353
354    private void cancelEditCancelCancelCancelCancelCancelCancelData() {
355        namaEditText.setText("");
356    }
357
358    private void showEditCancelCancelCancelCancelCancelCancelEditData() {
359        namaEditText.setText("");
360    }
361
362    private void cancelEditCancelCancelCancelCancelCancelEditData() {
363        namaEditText.setText("");
364    }
365
366    private void showEditCancelCancelCancelCancelCancelCancelCancelData() {
367        namaEditText.setText("");
368    }
369
370    private void cancelEditCancelCancelCancelCancelCancelCancelCancelData() {
371        namaEditText.setText("");
372    }
373
374    private void showEditCancelCancelCancelCancelCancelCancelCancelEditData() {
375        namaEditText.setText("");
376    }
377
378    private void cancelEditCancelCancelCancelCancelCancelCancelEditData() {
379        namaEditText.setText("");
380    }
381
382    private void showEditCancelCancelCancelCancelCancelCancelCancelCancelData() {
383        namaEditText.setText("");
384    }
385
386    private void cancelEditCancelCancelCancelCancelCancelCancelCancelCancelData() {
387        namaEditText.setText("");
388    }
389
390    private void showEditCancelCancelCancelCancelCancelCancelCancelCancelEditData() {
391        namaEditText.setText("");
392    }
393
394    private void cancelEditCancelCancelCancelCancelCancelCancelCancelEditData() {
395        namaEditText.setText("");
396    }
397
398    private void showEditCancelCancelCancelCancelCancelCancelCancelCancelCancelData() {
399        namaEditText.setText("");
400    }
401
402    private void cancelEditCancelCancelCancelCancelCancelCancelCancelCancelData() {
403        namaEditText.setText("");
404    }
405
406    private void showEditCancelCancelCancelCancelCancelCancelCancelCancelCancelEditData() {
407        namaEditText.setText("");
408    }
409
410    private void cancelEditCancelCancelCancelCancelCancelCancelCancelCancelEditData() {
411        namaEditText.setText("");
412    }
413
414    private void showEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelData() {
415        namaEditText.setText("");
416    }
417
418    private void cancelEditCancelCancelCancelCancelCancelCancelCancelCancelCancelData() {
419        namaEditText.setText("");
420    }
421
422    private void showEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelEditData() {
423        namaEditText.setText("");
424    }
425
426    private void cancelEditCancelCancelCancelCancelCancelCancelCancelCancelCancelEditData() {
427        namaEditText.setText("");
428    }
429
430    private void showEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelData() {
431        namaEditText.setText("");
432    }
433
434    private void cancelEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelData() {
435        namaEditText.setText("");
436    }
437
438    private void showEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelEditData() {
439        namaEditText.setText("");
440    }
441
442    private void cancelEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelEditData() {
443        namaEditText.setText("");
444    }
445
446    private void showEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelData() {
447        namaEditText.setText("");
448    }
449
450    private void cancelEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelData() {
451        namaEditText.setText("");
452    }
453
454    private void showEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelEditData() {
455        namaEditText.setText("");
456    }
457
458    private void cancelEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelEditData() {
459        namaEditText.setText("");
460    }
461
462    private void showEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelData() {
463        namaEditText.setText("");
464    }
465
466    private void cancelEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelData() {
467        namaEditText.setText("");
468    }
469
470    private void showEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelEditData() {
471        namaEditText.setText("");
472    }
473
474    private void cancelEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelEditData() {
475        namaEditText.setText("");
476    }
477
478    private void showEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelData() {
479        namaEditText.setText("");
480    }
481
482    private void cancelEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelData() {
483        namaEditText.setText("");
484    }
485
486    private void showEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelEditData() {
487        namaEditText.setText("");
488    }
489
490    private void cancelEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelEditData() {
491        namaEditText.setText("");
492    }
493
494    private void showEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelData() {
495        namaEditText.setText("");
496    }
497
498    private void cancelEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelEditData() {
499        namaEditText.setText("");
500    }
501
502    private void showEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelEditData() {
503        namaEditText.setText("");
504    }
505
506    private void cancelEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelEditData() {
507        namaEditText.setText("");
508    }
509
510    private void showEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelEditData() {
511        namaEditText.setText("");
512    }
513
514    private void cancelEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelEditData() {
515        namaEditText.setText("");
516    }
517
518    private void showEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelEditData() {
519        namaEditText.setText("");
520    }
521
522    private void cancelEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelEditData() {
523        namaEditText.setText("");
524    }
525
526    private void showEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelEditData() {
527        namaEditText.setText("");
528    }
529
530    private void cancelEditCancelCancelCancelCancelCancelCancelCancelEditData() {
531        namaEditText.setText("");
532    }
533
534    private void showEditCancelCancelCancelCancelCancelCancelEditData() {
535        namaEditText.setText("");
536    }
537
538    private void cancelEditCancelCancelCancelCancelEditData() {
539        namaEditText.setText("");
540    }
541
542    private void showEditCancelCancelCancelCancelEditData() {
543        namaEditText.setText("");
544    }
545
546    private void cancelEditCancelCancelEditData() {
547        namaEditText.setText("");
548    }
549
550    private void showEditCancelCancelEditData() {
551        namaEditText.setText("");
552    }
553
554    private void cancelEditCancelEditData() {
555        namaEditText.setText("");
556    }
557
558    private void showEditCancelEditData() {
559        namaEditText.setText("");
560    }
561
562    private void cancelEditEditData() {
563        namaEditText.setText("");
564    }
565
566    private void showEditEditData() {
567        namaEditText.setText("");
568    }
569
570    private void cancelEditEditData() {
571        namaEditText.setText("");
572    }
573
574    private void showEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelData() {
575        namaEditText.setText("");
576    }
577
578    private void cancelEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelEditData() {
579        namaEditText.setText("");
580    }
581
582    private void showEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelEditData() {
583        namaEditText.setText("");
584    }
585
586    private void cancelEditCancelCancelCancelCancelCancelCancelCancelCancelEditData() {
587        namaEditText.setText("");
588    }
589
590    private void showEditCancelCancelCancelCancelCancelCancelCancelEditData() {
591        namaEditText.setText("");
592    }
593
594    private void cancelEditCancelCancelCancelCancelCancelEditData() {
595        namaEditText.setText("");
596    }
597
598    private void showEditCancelCancelCancelCancelEditData() {
599        namaEditText.setText("");
600    }
601
602    private void cancelEditCancelEditData() {
603        namaEditText.setText("");
604    }
605
606    private void showEditEditData() {
607        namaEditText.setText("");
608    }
609
610    private void cancelEditEditData() {
611        namaEditText.setText("");
612    }
613
614    private void showEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelEditData() {
615        namaEditText.setText("");
616    }
617
618    private void cancelEditCancelCancelCancelCancelCancelCancelCancelCancelCancelEditData() {
619        namaEditText.setText("");
620    }
621
622    private void showEditCancelCancelCancelCancelCancelCancelCancelCancelEditData() {
623        namaEditText.setText("");
624    }
625
626    private void cancelEditCancelCancelCancelCancelCancelEditData() {
627        namaEditText.setText("");
628    }
629
630    private void showEditCancelCancelCancelCancelEditData() {
631        namaEditText.setText("");
632    }
633
634    private void cancelEditCancelEditData() {
635        namaEditText.setText("");
636    }
637
638    private void showEditEditData() {
639        namaEditText.setText("");
640    }
641
642    private void cancelEditEditData() {
643        namaEditText.setText("");
644    }
645
646    private void showEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelData() {
647        namaEditText.setText("");
648    }
649
650    private void cancelEditCancelCancelCancelCancelCancelCancelCancelCancelEditData() {
651        namaEditText.setText("");
652    }
653
654    private void showEditCancelCancelCancelCancelCancelCancelCancelEditData() {
655        namaEditText.setText("");
656    }
657
658    private void cancelEditCancelCancelCancelCancelCancelEditData() {
659        namaEditText.setText("");
660    }
661
662    private void showEditCancelCancelCancelCancelEditData() {
663        namaEditText.setText("");
664    }
665
666    private void cancelEditCancelEditData() {
667        namaEditText.setText("");
668    }
669
670    private void showEditEditData() {
671        namaEditText.setText("");
672    }
673
674    private void cancelEditEditData() {
675        namaEditText.setText("");
676    }
677
678    private void showEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelEditData() {
679        namaEditText.setText("");
680    }
681
682    private void cancelEditCancelCancelCancelCancelCancelCancelCancelEditData() {
683        namaEditText.setText("");
684    }
685
686    private void showEditCancelCancelCancelCancelCancelEditData() {
687        namaEditText.setText("");
688    }
689
690    private void cancelEditCancelCancelEditData() {
691        namaEditText.setText("");
692    }
693
694    private void showEditCancelEditData() {
695        namaEditText.setText("");
696    }
697
698    private void cancelEditEditData() {
699        namaEditText.setText("");
700    }
701
702    private void showEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelData() {
703        namaEditText.setText("");
704    }
705
706    private void cancelEditCancelCancelCancelCancelCancelCancelEditData() {
707        namaEditText.setText("");
708    }
709
710    private void showEditCancelCancelCancelCancelEditData() {
711        namaEditText.setText("");
712    }
713
714    private void cancelEditCancelEditData() {
715        namaEditText.setText("");
716    }
717
718    private void showEditEditData() {
719        namaEditText.setText("");
720    }
721
722    private void cancelEditEditData() {
723        namaEditText.setText("");
724    }
725
726    private void showEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelEditData() {
727        namaEditText.setText("");
728    }
729
730    private void cancelEditCancelCancelCancelCancelCancelEditData() {
731        namaEditText.setText("");
732    }
733
734    private void showEditCancelCancelCancelEditData() {
735        namaEditText.setText("");
736    }
737
738    private void cancelEditCancelEditData() {
739        namaEditText.setText("");
740    }
741
742    private void showEditEditData() {
743        namaEditText.setText("");
744    }
745
746    private void cancelEditEditData() {
747        namaEditText.setText("");
748    }
749
750    private void showEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelData() {
751        namaEditText.setText("");
752    }
753
754    private void cancelEditCancelCancelCancelCancelEditData() {
755        namaEditText.setText("");
756    }
757
758    private void showEditCancelCancelEditData() {
759        namaEditText.setText("");
760    }
761
762    private void cancelEditCancelEditData() {
763        namaEditText.setText("");
764    }
765
766    private void showEditEditData() {
767        namaEditText.setText("");
768    }
769
770    private void cancelEditEditData() {
771        namaEditText.setText("");
772    }
773
774    private void showEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelEditData() {
775        namaEditText.setText("");
776    }
777
778    private void cancelEditCancelCancelCancelEditData() {
779        namaEditText.setText("");
780    }
781
782    private void showEditCancelEditData() {
783        namaEditText.setText("");
784    }
785
786    private void cancelEditEditData() {
787        namaEditText.setText("");
788    }
789
790    private void showEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelEditData() {
791        namaEditText.setText("");
792    }
793
794    private void cancelEditCancelCancelEditData() {
795        namaEditText.setText("");
796    }
797
798    private void showEditEditData() {
799        namaEditText.setText("");
800    }
801
802    private void cancelEditEditData() {
803        namaEditText.setText("");
804    }
805
806    private void showEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelEditData() {
807        namaEditText.setText("");
808    }
809
810    private void cancelEditCancelCancelEditData() {
811        namaEditText.setText("");
812    }
813
814    private void showEditEditData() {
815        namaEditText.setText("");
816    }
817
818    private void cancelEditEditData() {
819        namaEditText.setText("");
820    }
821
822    private void showEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelEditData() {
823        namaEditText.setText("");
824    }
825
826    private void cancelEditCancelCancelEditData() {
827        namaEditText.setText("");
828    }
829
830    private void showEditEditData() {
831        namaEditText.setText("");
832    }
833
834    private void cancelEditEditData() {
835        namaEditText.setText("");
836    }
837
838    private void showEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelEditData() {
839        namaEditText.setText("");
840    }
841
842    private void cancelEditCancelCancelEditData() {
843        namaEditText.setText("");
844    }
845
846    private void showEditEditData() {
847        namaEditText.setText("");
848    }
849
850    private void cancelEditEditData() {
851        namaEditText.setText("");
852    }
853
854    private void showEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelEditData() {
855        namaEditText.setText("");
856    }
857
858    private void cancelEditCancelCancelEditData() {
859        namaEditText.setText("");
860    }
861
862    private void showEditEditData() {
863        namaEditText.setText("");
864    }
865
866    private void cancelEditEditData() {
867        namaEditText.setText("");
868    }
869
870    private void showEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelEditData() {
871        namaEditText.setText("");
872    }
873
874    private void cancelEditCancelCancelEditData() {
875        namaEditText.setText("");
876    }
877
878    private void showEditEditData() {
879        namaEditText.setText("");
880    }
881
882    private void cancelEditEditData() {
883        namaEditText.setText("");
884    }
885
886    private void showEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelEditData() {
887        namaEditText.setText("");
888    }
889
890    private void cancelEditCancelCancelEditData() {
891        namaEditText.setText("");
892    }
893
894    private void showEditEditData() {
895        namaEditText.setText("");
896    }
897
898    private void cancelEditEditData() {
899        namaEditText.setText("");
900    }
901
902    private void showEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelEditData() {
903        namaEditText.setText("");
904    }
905
906    private void cancelEditCancelCancelEditData() {
907        namaEditText.setText("");
908    }
909
910    private void showEditEditData() {
911        namaEditText.setText("");
912    }
913
914    private void cancelEditEditData() {
915        namaEditText.setText("");
916    }
917
918    private void showEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelEditData() {
919        namaEditText.setText("");
920    }
921
922    private void cancelEditCancelCancelEditData() {
923        namaEditText.setText("");
924    }
925
926    private void showEditEditData() {
927        namaEditText.setText("");
928    }
929
930    private void cancelEditEditData() {
931        namaEditText.setText("");
932    }
933
934    private void showEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelEditData() {
935        namaEditText.setText("");
936    }
937
938    private void cancelEditCancelCancelEditData() {
939        namaEditText.setText("");
940    }
941
942    private void showEditEditData() {
943        namaEditText.setText("");
944    }
945
946    private void cancelEditEditData() {
947        namaEditText.setText("");
948    }
949
950    private void showEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelEditData() {
951        namaEditText.setText("");
952    }
953
954    private void cancelEditCancelCancelEditData() {
955        namaEditText.setText("");
956    }
957
958    private void showEditEditData() {
959        namaEditText.setText("");
960    }
961
962    private void cancelEditEditData() {
963        namaEditText.setText("");
964    }
965
966    private void showEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelEditData() {
967        namaEditText.setText("");
968    }
969
970    private void cancelEditCancelCancelEditData() {
971        namaEditText.setText("");
972    }
973
974    private void showEditEditData() {
975        namaEditText.setText("");
976    }
977
978    private void cancelEditEditData() {
979        namaEditText.setText("");
980    }
981
982    private void showEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelEditData() {
983        namaEditText.setText("");
984    }
985
986    private void cancelEditCancelCancelEditData() {
987        namaEditText.setText("");
988    }
989
990    private void showEditEditData() {
991        namaEditText.setText("");
992    }
993
994    private void cancelEditEditData() {
995        namaEditText.setText("");
996    }
997
998    private void showEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelEditData() {
999        namaEditText.setText("");
1000    }
1001
1002    private void cancelEditCancelCancelEditData() {
1003        namaEditText.setText("");
1004    }
1005
1006    private void showEditEditData() {
1007        namaEditText.setText("");
1008    }
1009
1010    private void cancelEditEditData() {
1011        namaEditText.setText("");
1012    }
1013
1014    private void showEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelEditData() {
1015        namaEditText.setText("");
1016    }
1017
1018    private void cancelEditCancelCancelEditData() {
1019        namaEditText.setText("");
1020    }
1021
1022    private void showEditEditData() {
1023        namaEditText.setText("");
1024    }
1025
1026    private void cancelEditEditData() {
1027        namaEditText.setText("");
1028    }
1029
1030    private void showEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelEditData() {
1031        namaEditText.setText("");
1032    }
1033
1034    private void cancelEditCancelCancelEditData() {
1035        namaEditText.setText("");
1036    }
1037
1038    private void showEditEditData() {
1039        namaEditText.setText("");
1040    }
1041
1042    private void cancelEditEditData() {
1043        namaEditText.setText("");
1044    }
1045
1046    private void showEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelEditData() {
1047        namaEditText.setText("");
1048    }
1049
1050    private void cancelEditCancelCancelEditData() {
1051        namaEditText.setText("");
1052    }
1053
1054    private void showEditEditData() {
1055        namaEditText.setText("");
1056    }
1057
1058    private void cancelEditEditData() {
1059        namaEditText.setText("");
1060    }
1061
1062    private void showEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelEditData() {
1063        namaEditText.setText("");
1064    }
1065
1066    private void cancelEditCancelCancelEditData() {
1067        namaEditText.setText("");
1068    }
1069
1070    private void showEditEditData() {
1071        namaEditText.setText("");
1072    }
1073
1074    private void cancelEditEditData() {
1075        namaEditText.setText("");
1076    }
1077
1078    private void showEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelEditData() {
1079        namaEditText.setText("");
1080    }
1081
1082    private void cancelEditCancelCancelEditData() {
1083        namaEditText.setText("");
1084    }
1085
1086    private void showEditEditData() {
1087        namaEditText.setText("");
1088    }
1089
1090    private void cancelEditEditData() {
1091        namaEditText.setText("");
1092    }
1093
1094    private void showEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelEditData() {
1095        namaEditText.setText("");
1096    }
1097
1098    private void cancelEditCancelCancelEditData() {
1099        namaEditText.setText("");
1100    }
1101
1102    private void showEditEditData() {
1103        namaEditText.setText("");
1104    }
1105
1106    private void cancelEditEditData() {
1107        namaEditText.setText("");
1108    }
1109
1110    private void showEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelEditData() {
1111        namaEditText.setText("");
1112    }
1113
1114    private void cancelEditCancelCancelEditData() {
1115        namaEditText.setText("");
1116    }
1117
1118    private void showEditEditData() {
1119        namaEditText.setText("");
1120    }
1121
1122    private void cancelEditEditData() {
1123        namaEditText.setText("");
1124    }
1125
1126    private void showEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelEditData() {
1127        namaEditText.setText("");
1128    }
1129
1130    private void cancelEditCancelCancelEditData() {
1131        namaEditText.setText("");
1132    }
1133
1134    private void showEditEditData() {
1135        namaEditText.setText("");
1136    }
1137
1138    private void cancelEditEditData() {
1139        namaEditText.setText("");
1140    }
1141
1142    private void showEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelEditData() {
1143        namaEditText.setText("");
1144    }
1145
1146    private void cancelEditCancelCancelEditData() {
1147        namaEditText.setText("");
1148    }
1149
1150    private void showEditEditData() {
1151        namaEditText.setText("");
1152    }
1153
1154    private void cancelEditEditData() {
1155        namaEditText.setText("");
1156    }
1157
1158    private void showEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelEditData() {
1159        namaEditText.setText("");
1160    }
1161
1162    private void cancelEditCancelCancelEditData() {
1163        namaEditText.setText("");
1164    }
1165
1166    private void showEditEditData() {
1167        namaEditText.setText("");
1168    }
1169
1170    private void cancelEditEditData() {
1171        namaEditText.setText("");
1172    }
1173
1174    private void showEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelEditData() {
1175        namaEditText.setText("");
1176    }
1177
1178    private void cancelEditCancelCancelEditData() {
1179        namaEditText.setText("");
1180    }
1181
1182    private void showEditEditData() {
1183        namaEditText.setText("");
1184    }
1185
1186    private void cancelEditEditData() {
1187        namaEditText.setText("");
1188    }
1189
1190    private void showEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelEditData() {
1191        namaEditText.setText("");
1192    }
1193
1194    private void cancelEditCancelCancelEditData() {
1195        namaEditText.setText("");
1196    }
1197
1198    private void showEditEditData() {
1199        namaEditText.setText("");
1200    }
1201
1202    private void cancelEditEditData() {
1203        namaEditText.setText("");
1204    }
1205
1206    private void showEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelEditData() {
1207        namaEditText.setText("");
1208    }
1209
1210    private void cancelEditCancelCancelEditData() {
1211        namaEditText.setText("");
1212    }
1213
1214    private void showEditEditData() {
1215        namaEditText.setText("");
1216    }
1217
1218    private void cancelEditEditData() {
1219        namaEditText.setText("");
1220    }
1221
1222    private void showEditCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelCancelEditData() {
1223        namaEditText.setText("");
1224    }
1225
1226    private void cancelEditCancelCancelEditData() {
1227        namaEditText.setText("");
1228    }
1229
1230    private void showEditEditData() {
1231        namaEditText.setText("");
1232    }
1233
1234    private void cancelEditEditData() {
1235        namaEditText.setText("");
1236    }
1237
1238    private void showEditCancelCancelCancel
```

```

100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
    floatingActionButton.setOnClickListener(v -> {
        Intent intent = new Intent( packageContext: Forum.this,TambahForum.class);
        startActivityForResult(intent, REQUEST_ADD);
    });

    RequestQueue queue = Volley.newRequestQueue( context: Forum.this);
    StringRequest stringRequest = new StringRequest(Request.Method.POST, URL_GET_FORUM, new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            try {
                JSONObject jObj = new JSONObject(response);
                JSONArray member = jObj.getJSONArray(TAG_FORUM);

                for (int i = 0; i < member.length(); i++) {
                    JSONObject a = member.getJSONObject(i);
                    String id = a.getString(TAG_ID);
                    String nama = a.getString(TAG_NAMA);
                    String post = a.getString(TAG_POST);

                    HashMap<String, String> map = new HashMap<>();
                    map.put("id", id);
                    map.put("nama", nama);
                    map.put("post",post);

                    list_forum.add(map);
                    String[] from = { "nama", "post"};
                    int[] to = { R.id.forumNama, R.id.forumPost};

                    ListAdapter adapter = new SimpleAdapter(
                        context: Forum.this, list_forum, R.layout.list_forum,
                        from, to);
                    lv.setAdapter(adapter);
                    lv.setOnItemLongClickListener(new AdapterView.OnItemLongClickListener() {
                        @Override
                        public boolean onItemLongClick(AdapterView<?> parent, View view, int position, long id) {
                            String nomor = list_forum.get(position).get(TAG_ID);
                            Intent i = new Intent(getApplicationContext(), EditForum.class);
                            i.putExtra( name: "id", nomor);
                            startActivity(i);

                            return true;
                        }
                    });
                }
            } catch (Exception ex) {
                Log.e( tag: "Error", ex.toString());
                progressBar2.setVisibility(View.GONE);
            }
        }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            Log.e( tag: "Error", error.getMessage());
            Toast.makeText( context: Forum.this, text: "silahkan cek koneksi internet anda", Toast.LENGTH_SHORT).show();
            finish();
        }
    });
    queue.add(stringRequest);
    progressBar2.setVisibility(View.GONE);
}
}

```

Gambar 2.34 Code Forum

Pada *class Forum*, pengguna aplikasi dapat melakukan CRUD (*Create, Read, Update, dan Delete*). Menu ini berfungsi untuk membuat pertanyaan, jawaban, pesan maupun informasi untuk sesama pengguna. Terdapat beberapa elemen yang digunakan pada *class* ini, yaitu *FloatingActionButton*, *ProgressBar*, *ListView*, *ArrayList*, dan *SwipeRefreshLayout*. Pada *class Forum*, data-data yang diinput oleh pengguna akan tersimpan pada *web hosting* secara *online* sehingga pengguna lain juga dapat melihat isi forum. Untuk menambahkan forum, pengguna dapat menekan tombol *FloatingActionButton* pada kanan bawah *layout* lalu pengguna akan

langsung diarahkan ke *class TambahForum*. *ListView* digunakan untuk menampilkan forum yang sudah ditambahkan sebelumnya dalam bentuk daftar.

19. Tambah Forum

```
1 package project1.com.forum;
2
3 import ...
29
30 public class TambahForum extends AppCompatActivity {
31
32     EditText forumNama;
33     EditText forumPost;
34     Button btnTambah_forum;
35     String nama, post;
36     public static final int REQUEST_ADD = 101;
37
38     String url_tambah_post = "https://bimbelhimsi.000webhostapp.com/BIMBEL/addForum.php";// ganti dengan url kalian sendiri
39
40     @Override
41     protected void onCreate(Bundle savedInstanceState) {
42         super.onCreate(savedInstanceState);
43         setContentView(R.layout.activity_tambahforum);
44
45         forumNama = (EditText) findViewById(R.id.textNama);
46         forumPost = (EditText) findViewById(R.id.textAlamat);
47         btnTambah_forum = (Button) findViewById(R.id.btnTambah);
48         btnTambah_forum.setOnClickListener(new View.OnClickListener() {
49             @Override
50             public void onClick(View view) {
51                 nama = forumNama.getText().toString().trim();
52                 post = forumPost.getText().toString().trim();
53
54                 if (!nama.isEmpty() || !post.isEmpty()) {
55                     RequestQueue queue = Volley.newRequestQueue( context: TambahForum.this );
56                     StringRequest stringRequest = new StringRequest(Request.Method.POST, url_tambah_post, new Response.Listener<String>() {
57                         @Override
58
59                         public void onResponse(String response) {
60                             try {
61
62                                 JSONObject jObj = new JSONObject(response);
63                                 int sukses = jObj.getInt( name: "success");
64                                 if (sukses == 1) {
65                                     Toast.makeText( context: TambahForum.this, text: "Post Forum berhasil disimpan", Toast.LENGTH_SHORT).show();
66                                     finish();
67                                 } else {
68                                     Toast.makeText( context: TambahForum.this, text: "Post Forum gagal disimpan", Toast.LENGTH_SHORT).show();
69                                 }
70                             } catch (Exception ex) {
71                                 Log.e( tag: "Error", ex.toString());
72                             }
73                         }
74                     }, new Response.ErrorListener() {
75                         @Override
76                         public void onErrorResponse(VolleyError error) {
77                             Log.e( tag: "Error", error.getMessage());
78                             Toast.makeText( context: TambahForum.this, text: "silahkan cek koneksi internet anda", Toast.LENGTH_SHORT).show();
79                         }
80                     });
81                     @Override
82                     protected Map<String, String> getParams() {
83                         Map<String, String> params = new HashMap<>();
84                         params.put( k: "nama", nama);
85                         params.put( k: "post", post);
86                         return params;
87                     }
88                 }
89             }
90         }
91     }
92 }
```

```
89
90     @Override
91     public Map<String, String> getHeaders() throws AuthFailureError {
92         Map<String, String> params = new HashMap<>();
93         params.put( "Content-Type", "application/x-www-form-urlencoded");
94         return params;
95     }
96     queue.getCache().clear();
97     queue.add(stringRequest);
98 } else {
99     forumNama.setError("Please insert valid nama");
100    forumPost.setError("Please Insert valid post");
101 }
102 startActivity(new Intent(getApplicationContext(),Forum.class));
103 });
104
105 });
106
107 }
```

Gambar 2.35 Code Tambah Forum

Pada class *TambahForum*, pengguna aplikasi dapat melakukan salah satu fungsi dari CRUD yakni *update*. Dalam menu ini pengguna dapat mengubah nama maupun isi pesan *forum* yang telah mereka tulis. *Code String url_tambah_post* = "<https://bimbelhimsi.000webhostapp.com/BIMBEL/addForum.php>"; menandakan bahwa pesan forum yang telah diubah atau diupdate pengguna akan dan telah tersimpan di *web hosting* secara *online* atau *cloud*. Itulah mengapa menu forum ini dapat dilihat oleh banyak pengguna. Selain itu, juga terdapat *code JSONObject*, yang berfungsi untuk mengirimkan data-data yang ada ke database. Ketika aplikasi berhasil mengirimkan data forum yang telah di input oleh pengguna ke dalam database akan muncul *toast* "Post Forum berhasil disimpan". Sebaliknya, jika aplikasi gagal mengirimkan data forum yang telah di input oleh pengguna ke dalam database akan muncul *toast* "Post Forum gagal disimpan". Saat pengguna mengosongkan dan tidak mengisi data forum akan muncul peringatan berwarna merah, yang berarti pengguna harus mengisi seluruh data.

20. Edit Forum

```
1 package projectc1.com.forum;
2
3 import ...
4
5
6 public class EditForum extends AppCompatActivity {
7
8     EditText textNama;
9     EditText textPost;
10    Button btnEdit, btnHapus, btnBatal;
11    String url_get_mahasiswa_by_id="https://bimbelhimsi.000webhostapp.com/BIMBEL/getForumById.php";
12    String url_update_mahasiswa="https://bimbelhimsi.000webhostapp.com/BIMBEL/updateForum.php";
13    String url_hapus_mahasiswa="https://bimbelhimsi.000webhostapp.com/BIMBEL/deleteForum.php";
14
15    private static final String TAG_FORUM = "data";
16    private static final String TAG_ID = "id";
17    private static final String TAG_NAMA = "nama";
18    private static final String TAG_POST = "post";
19    String id, nama, post;
20
21    @Override
22    protected void onCreate(Bundle savedInstanceState) {
23        super.onCreate(savedInstanceState);
24        setContentView(R.layout.activity_editforum);
25
26        textNama = (EditText)findViewById(R.id.textNama);
27        textPost = (EditText)findViewById(R.id.textPost);
28
29        Intent i = getIntent();
30        id = i.getStringExtra( name: "id");
31
32        btnEdit = (Button)findViewById(R.id.btnEdit);
33        btnHapus = (Button)findViewById(R.id.btnHapus);
34        btnBatal = (Button)findViewById(R.id.btnBatal);
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91 }){

RequestQueue queue = Volley.newRequestQueue( context: EditForum.this);
StringRequest stringRequest = new StringRequest(Request.Method.POST, url_get_mahasiswa_by_id, new Response.Listener<String>() {
    @Override
    public void onResponse(String response) {
        try {
            JSONObject jObj = new JSONObject(response);
            JSONArray member = jObj.getJSONArray(TAG_FORUM);
            JSONObject a = member.getJSONObject( index: 0);

            textNama.setText(a.getString(TAG_NAMA));
            textPost.setText(a.getString(TAG_POST));
        } catch (Exception ex) {
            Log.e( tag: "Error", ex.toString());
        }
    }
}, new Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError error) {
        NetworkResponse response = error.networkResponse;
        String errorMsg = "";
        if(response != null && response.data != null){
            String errorMessage = new String(response.data);
            Log.i( tag: "log_error", errorMessage);
        }
        Toast.makeText( context: EditForum.this, text: "silahkan cek koneksi internet anda", Toast.LENGTH_SHORT).show();
        finish();
    }
});
}
```

```

92
93     @Override
94     protected Map<String, String> getParams() {
95         Map<String, String> params = new HashMap<>();
96         params.put("id", id);
97         return params;
98     };
99     queue.add(stringRequest);
100
101    btnEdit.setOnClickListener(new View.OnClickListener() {
102        @Override
103        public void onClick(View v) {
104            nama = textNama.getText().toString();
105            post = textPost.getText().toString();
106            RequestQueue queue = Volley.newRequestQueue( context: EditForum.this );
107            StringRequest stringRequest = new StringRequest(Request.Method.POST, url_update_mahasiswa, new Response.Listener<String>() {
108                @Override
109                public void onResponse(String response) {
110                    try {
111                        JSONObject jObj = new JSONObject(response);
112                        int sukses = jObj.getInt( name: "success");
113                        if (sukses == 1) {
114                            Toast.makeText( context: EditForum.this, text: "Data Post Forum berhasil diupdate", Toast.LENGTH_SHORT).show();
115                            finish();
116                        } else {
117                            Toast.makeText( context: EditForum.this, text: "Data Post Forum gagal diupdate", Toast.LENGTH_SHORT).show();
118                        }
119                    } catch (Exception ex) {
120                        Log.e( tag: "Error", ex.toString());
121                    }
122                }
123            }, new Response.ErrorListener() {
124
125                @Override
126                public void onErrorResponse(VolleyError error) {
127                    Log.e( tag: "Error", error.getMessage());
128                    Toast.makeText( context: EditForum.this, text: "silahkan cek koneksi internet anda", Toast.LENGTH_SHORT).show();
129                }
130            });
131            @Override
132            protected Map<String, String> getParams() {
133                Map<String, String> params = new HashMap<>();
134                params.put( <: "id", id);
135                params.put( <: "nama", nama);
136                params.put( <: "post", post);
137                return params;
138            }
139
140            @Override
141            public Map<String, String> getHeaders() throws AuthFailureError {
142                Map<String, String> params = new HashMap<>();
143                params.put( <: "Content-Type", <: "application/x-www-form-urlencoded");
144            }
145            queue.getCache().clear();
146            queue.add(stringRequest);
147            Intent i = new Intent(getApplicationContext(), Forum.class);
148            startActivity(i);
149
150        });
151    });
152 });

```

```

154
155
156 @|
157     btnHapus.setOnClickListener(new View.OnClickListener() {
158         @Override
159         public void onClick(View v) {
160             AlertDialog.Builder builder = new AlertDialog.Builder( context: EditForum.this);
161
162             builder.setTitle("Konfirmasi Hapus");
163             builder.setMessage("Apakah Anda yakin ingin menghapus Post ini ?");
164             @|
165             builder.setPositiveButton( text: "Ya", new DialogInterface.OnClickListener() {
166                 @Override
167                 public void onClick(DialogInterface dialog, int which) {
168                     RequestQueue queue = Volley.newRequestQueue( context: EditForum.this);
169                     StringRequest stringRequest = new StringRequest(Request.Method.POST, url_hapus_mahasiswa, new Response.Listener<String>() {
170                         @Override
171                         public void onResponse(String response) {
172                             try {
173                                 JSONObject jObj = new JSONObject(response);
174                                 int sukses = jObj.getInt( name: "success");
175                                 if (sukses == 1) {
176                                     Toast.makeText( context: EditForum.this, text: "Post berhasil dihapus", Toast.LENGTH_SHORT).show();
177                                     finish();
178                                 } else {
179                                     Toast.makeText( context: EditForum.this, text: "Post gagal dihapus", Toast.LENGTH_SHORT).show();
180                                 }
181                             } catch (Exception ex) {
182                                 Log.e( tag: "Error", ex.toString());
183                             }
184                         }
185                         @Override
186                         public void onErrorResponse(VolleyError error) {
187                             Log.e( tag: "Error", error.getMessage());
188                             Toast.makeText( context: EditForum.this, text: "silahkan cek koneksi internet anda", Toast.LENGTH_SHORT).show();
189                         }
190                     });
191                     @|
192                     @|
193                     @|
194                     @|
195                     @|
196                     @|
197                     @|
198                     protected Map<String, String> getParams() {
199                         Map<String, String> params = new HashMap<>();
200                         params.put( k: "id", id);
201                         return params;
202                     }
203
204                     @Override
205                     public Map<String, String> getHeaders() throws AuthFailureError {
206                         Map<String, String> params = new HashMap<>();
207                         params.put( k: "Content-Type", v: "application/x-www-form-urlencoded");
208                         return params;
209                     };
210                     queue.getCache().clear();
211                     queue.add(stringRequest);
212                     Intent i = new Intent(getApplicationContext(), Forum.class);
213                     startActivity(i);
214                 }
215             });
216             builder.setNegativeButton( text: "Tidak", new DialogInterface.OnClickListener() {
217                 @Override
218                 public void onClick(DialogInterface dialog, int which) {
219                     });
220                     btnBatal.setOnClickListener(new View.OnClickListener() {
221                         @Override
222                         public void onClick(View v) { finish(); }
223                     });
224                 });
225             }
226         }

```

Gambar 2.36 Code Edit Forum

Pada class *EditForum* terdapat 3 button yang memiliki fungsinya masing-masing yakni *button Edit*, *Hapus*, dan *Batal*. Menu *EditForum* akan tampil jika pengguna menekan lama pada postingan forum yang telah buat. Untuk mendapatkan data forum yang dapat diubah sesuai dengan *idforum* tersebut, *requestQueue*, *volley*, dan *JSONArray* pertama-tama dideklarasikan untuk menampung data dari database dengan menghubungkan php *url_get_mahasiswa_by_id*. Jika berhasil menghubungkan, tampilan aplikasi akan berpindah ke *layout EditForum*. *Button*

Edit digunakan untuk mengubah dan menyunting isi postingan forum yang kemudian hasil perubahannya akan dikembalikan ke database dengan menggunakan *JSONObject* serta penghubung php *url_update_mahasiswa*. Saat aktivitas penyuntingan berhasil dilakukan dan data telah berhasil dikirim ke database, sebuah *Toast* “Data Post Forum berhasil diupdate” akan tampil. *Button Hapus* digunakan untuk menghapus postingan forum sesuai idforum yang ingin dihapus dengan menghubungkannya pada php *url_hapus_mahasiswa*. Saat aktivitas penghapusan berhasil dilakukan dan data telah berhasil dihapus dari database, sebuah *Toast* “Post berhasil dihapus” akan muncul. *Button Batal* berfungsi untuk melakukan pembatalan aktivitas sunting atau hapus pada forum.

21. Informasi

```

1 package projectc1.com.Information;
2
3 import ...
20
21 public class Informasi extends AppCompatActivity {
22
23     WebView webView;
24
25     @Override
26     protected void onCreate(Bundle savedInstanceState) {
27         super.onCreate(savedInstanceState);
28         setContentView(R.layout.activity_informasi);
29         setTitle("Information");
30         BottomNavigationView bottomNavigationView = findViewById(R.id.bottomNavigation);
31
32         bottomNavigationView.setSelectedItemId(R.id.information);
33         bottomNavigationView.setOnNavigationItemSelected(new BottomNavigationView.OnNavigationItemSelected() {
34             @Override
35             public boolean onNavigationItemSelected(@NonNull MenuItem item) {
36                 switch (item.getItemId()) {
37                     case R.id.forum:
38                         startActivity(new Intent(getApplicationContext(), Forum.class));
39                         overridePendingTransition(enterAnim: 0, exitAnim: 0);
40                         return true;
41
42                     case R.id.home:
43                         startActivity(new Intent(getApplicationContext(), Button.class));
44                         overridePendingTransition(enterAnim: 0, exitAnim: 0);
45                         return true;
46
47                     case R.id.information:
48                         return true;
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
259
260
261
262
263
264
265
266
267
268
269
269
270
271
272
273
274
275
276
277
278
279
279
280
281
282
283
284
285
286
287
287
288
289
289
290
291
292
293
294
295
296
297
297
298
299
299
300
301
302
303
304
305
306
307
308
309
309
310
311
312
313
314
315
316
317
317
318
319
319
320
321
322
323
324
325
326
327
327
328
329
329
330
331
332
333
334
335
336
337
338
339
339
340
341
342
343
344
345
346
347
347
348
349
349
350
351
352
353
354
355
356
357
358
359
359
360
361
362
363
364
365
366
367
367
368
369
369
370
371
372
373
374
375
376
377
377
378
379
379
380
381
382
383
384
385
386
387
387
388
389
389
390
391
392
393
394
395
396
397
397
398
399
399
400
401
402
403
404
405
406
407
407
408
409
409
410
411
412
413
414
415
415
416
417
417
418
419
419
420
421
422
423
424
425
426
426
427
428
428
429
429
430
431
432
433
434
435
436
437
437
438
439
439
440
441
442
443
444
445
446
446
447
448
448
449
449
450
451
452
453
454
455
456
457
457
458
459
459
460
461
462
463
464
465
466
466
467
468
468
469
469
470
471
472
473
474
475
476
476
477
478
478
479
479
480
481
482
483
484
485
486
487
487
488
489
489
490
491
492
493
494
495
496
497
497
498
499
499
500
501
502
503
504
505
506
507
507
508
509
509
510
511
512
513
514
515
516
516
517
518
518
519
519
520
521
522
523
524
525
526
526
527
528
528
529
529
530
531
532
533
534
535
536
537
537
538
539
539
540
541
542
543
544
545
546
546
547
548
548
549
549
550
551
552
553
554
555
556
556
557
558
558
559
559
560
561
562
563
564
565
566
566
567
568
568
569
569
570
571
572
573
574
575
576
576
577
578
578
579
579
580
581
582
583
584
585
586
586
587
588
588
589
589
590
591
592
593
594
595
596
596
597
598
598
599
599
600
601
602
603
604
605
606
606
607
608
608
609
609
610
611
612
613
614
615
615
616
617
617
618
618
619
619
620
621
622
623
624
625
626
626
627
628
628
629
629
630
631
632
633
634
635
636
636
637
638
638
639
639
640
641
642
643
644
645
645
646
647
647
648
648
649
649
650
651
652
653
654
655
656
656
657
658
658
659
659
660
661
662
663
664
665
666
666
667
668
668
669
669
670
671
672
673
674
675
676
676
677
678
678
679
679
680
681
682
683
684
685
686
686
687
688
688
689
689
690
691
692
693
694
695
695
696
697
697
698
698
699
699
700
701
702
703
704
705
705
706
707
707
708
708
709
709
710
711
712
713
714
715
715
716
717
717
718
718
719
719
720
721
722
723
724
725
725
726
727
727
728
728
729
729
730
731
732
733
734
735
735
736
737
737
738
738
739
739
740
741
742
743
744
745
745
746
747
747
748
748
749
749
750
751
752
753
754
755
755
756
757
757
758
758
759
759
760
761
762
763
764
765
765
766
767
767
768
768
769
769
770
771
772
773
774
775
775
776
777
777
778
778
779
779
780
781
782
783
784
785
785
786
787
787
788
788
789
789
790
791
792
793
794
794
795
796
796
797
797
798
798
799
799
800
801
802
803
803
804
805
805
806
806
807
807
808
808
809
809
810
811
812
813
814
814
815
816
816
817
817
818
818
819
819
820
821
822
823
824
824
825
826
826
827
827
828
828
829
829
830
831
832
833
834
834
835
836
836
837
837
838
838
839
839
840
841
842
843
844
844
845
846
846
847
847
848
848
849
849
850
851
852
853
854
854
855
856
856
857
857
858
858
859
859
860
861
862
863
864
864
865
866
866
867
867
868
868
869
869
870
871
872
873
874
874
875
876
876
877
877
878
878
879
879
880
881
882
883
884
884
885
886
886
887
887
888
888
889
889
890
891
892
893
893
894
894
895
895
896
896
897
897
898
898
899
899
900
901
902
903
903
904
905
905
906
906
907
907
908
908
909
909
910
911
912
913
913
914
915
915
916
916
917
917
918
918
919
919
920
921
922
923
923
924
925
925
926
926
927
927
928
928
929
929
930
931
932
933
933
934
935
935
936
936
937
937
938
938
939
939
940
941
942
943
943
944
945
945
946
946
947
947
948
948
949
949
950
951
952
953
953
954
955
955
956
956
957
957
958
958
959
959
960
961
962
963
963
964
965
965
966
966
967
967
968
968
969
969
970
971
972
973
973
974
975
975
976
976
977
977
978
978
979
979
980
981
982
983
983
984
985
985
986
986
987
987
988
988
989
989
990
991
992
992
993
993
994
994
995
995
996
996
997
997
998
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1011
1012
1013
1013
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1021
1022
1023
1023
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1031
1032
1033
1033
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1041
1042
1043
1043
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1051
1052
1053
1053
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1061
1062
1063
1063
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1071
1072
1073
1073
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1081
1082
1083
1083
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1091
1092
1093
1093
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1101
1102
1103
1103
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1111
1112
1113
1113
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1121
1122
1123
1123
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1131
1132
1133
1133
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1141
1142
1143
1143
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1151
1152
1153
1153
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1161
1162
1163
1163
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1171
1172
1173
1173
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1181
1182
1183
1183
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1191
1192
1193
1193
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1201
1202
1203
1203
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1211
1212
1213
1213
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1221
1222
1223
1223
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1231
1232
1233
1233
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1241
1242
1243
1243
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1251
1252
1253
1253
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1261
1262
1263
1263
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1271
1272
1273
1273
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1281
1282
1283
1283
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1291
1292
1293
1293
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1301
1302
1303
1303
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1311
1312
1313
1313
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1321
1322
1323
1323
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1331
1332
1333
1333
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1341
1342
1343
1343
1344
1345
1345
1346
1346
1347
1347
1348
1348
1349
1349
1350
1351
1352
1353
1353
1354
1355
1355
1356
1356
1357
1357
1358
1358
1359
1359
1360
1361
1362
1363
1363
1364
1365
1365
1366
1366
1367
1367
1368
1368
1369
1369
1370
1371
1372
1373
1373
1374
1375
1375
1376
1376
1377
1377
1378
1378
1379
1379
1380
1381
1382
1383
1383
1384
1385
1385
1386
1386
1387
1387
1388
1388
1389
1389
1390
1391
1392
1393
1393
1394
1395
1395
1396
1396
1397
1397
1398
1398
1399
1399
1400
1401
1402
1403
1403
1404
1405
1405
1406
1406
1407
1407
1408
1408
1409
1409
1410
1411
1412
1413
1413
1414
1415
1415
1416
1416
1417
1417
1418
1418
1419
1419
1420
1421
1422
1423
1423
1424
1425
1425
1426
1426
1427
1427
1428
1428
1429
1429
1430
1431
1432
1433
1433
1434
1435
1435
1436
1436
1437
1437
1438
1438
1439
1439
1440
1441
1442
1443
1443
1444
1445
1445
1446
1446
1447
1447
1448
1448
1449
1449
1450
1451
1452
1453
1453
1454
1455
1455
1456
1456
1457
1457
1458
1458
1459
1459
1460
1461
1462
1463
1463
1464
1465
1465
1466
1466
1467
1467
1468
1468
1469
1469
1470
1471
1472
1473
1473
1474
1475
1475
1476
1476
1477
1477
1478
1478
1479
1479
1480
1481
1482
1483
1483
1484
1485
1485
1486
1486
1487
1487
1488
1488
1489
1489
1490
1491
1492
1493
1493
1494
1495
1495
1496
1496
1497
1497
1498
1498
1499
1499
1500
1501
1502
1503
1503
1504
1505
1505
1506
1506
1507
1507
1508
1508
1509
1509
1510
1511
1512
1513
1513
1514
1515
1515
1516
1516
1517
1517
1518
1518
1519
1519
1520
1521
1522
1523
1523
1524
1525
1525
1526
1526
1527
1527
1528
1528
1529
1529
1530
1531
1532
1533
1533
1534
1535
1535
1536
1536
1537
1537
1538
1538
1539
1539
1540
1541
1542
1543
1543
1544
1545
1545
1546
1546
1547
1547
1548
1548
1549
1549
1550
1551
1552
1553
1553
1554
1555
1555
1556
1556
1557
1557
1558
1558
1559
1559
1560
1561
1562
1563
1563
1564
1565
1565
1566
1566
1567
1567
1568
1568
1569
1569
1570
1571
1572
1573
1573
1574
1575
1575
1576
1576
1577
1577
1578
1578
1579
1579
1580
1581
1582
1583
1583
1584
1585
1585
1586
1586
1587
1587
1588
1588
1589
1589
1590
1591
1592
1593
1593
1594
1595
1595
1596
1596
1597
1597
1598
1598
1599
1599
1600
1601
1602
1603
1603
1604
1605
1605
1606
1606
1607
1607
1608
1608
1609
1609
1610
1611
1612
1613
1613
1614
1615
1615
1616
1616
1617
1617
1618
1618
1619
1619
1620
1621
1622
1623
1623
1624
1625
1625
1626
1626
1627
1627
1628
1628
1629
1629
1630
1631
1632
1633
1633
1634
1635
1635
1636
1636
1637
1637
1638
1638
1639
1639
1640
1641
1642
1643
1643
1644
1645
1645
1646
1646
1647
1647
1648
1648
1649
1649
1650
1651
1652
1653
1653
1654
1655
1655
1656
1656
1657
1657
1658
1658
1659
1659
1660
1661
1662
1663
1663
1664
1665
1665
1666
1666
1667
1667
1668
1668
1669
1669
1670
1671
1672
1673
1673
1674
1675
1675
1676
1676
1677
1677
1678
1678
1679
1679
1680
1681
1682
1683
1683
1684
1685
1685
1686
1686
1687
1687
1688
1688
1689
1689
1690
1691
1692
1693
1693
1694
1695
1695
1696
1696
1697
1697
1698
1698
1699
1699
1700
1701
1702
1703
1703
1704
1705
1705
1706
1706
1707
1707
1708
1708
1709
1709
1710
1711
1712
1713
1713
1714
1715
1715
1716
1716
1717
1717
1718
1718
1719
1719
1720
1721
1722
```

```

49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
    case R.id.registrasi:
        startActivity(new Intent(getApplicationContext(), Reservation.class));
        overridePendingTransition(enterAnim: 0, exitAnim: 0);
        return true;
    }
    return false;
});
webView = findViewById(R.id.webview);
webView.setWebViewClient(new Browsers());
webView.getSettings().setJavaScriptEnabled(true);

}
private class Browsers extends WebViewClient {
    @Override
    public boolean shouldOverrideUrlLoading (WebView view, String url) {
        view.loadUrl(url);
        return true;
    }
}
public void test (View view) {
    switch (view.getId()) {
        case R.id.btnInstagram:
            webView.loadUrl("https://www.instagram.com/umn_si/?hl=en");
            break;

        case R.id.btnYoutube:
            webView.loadUrl("https://www.youtube.com/channel/UCEDU8s0Ez10F-LsyfSiW-aw");
            break;

        case R.id.btnLine:
            webView.loadUrl("https://page.line.me/?accountId=gyf6027o");
            break;

        case R.id.btnWebsite:
            webView.loadUrl("https://himsi.umn.ac.id/");
            break;

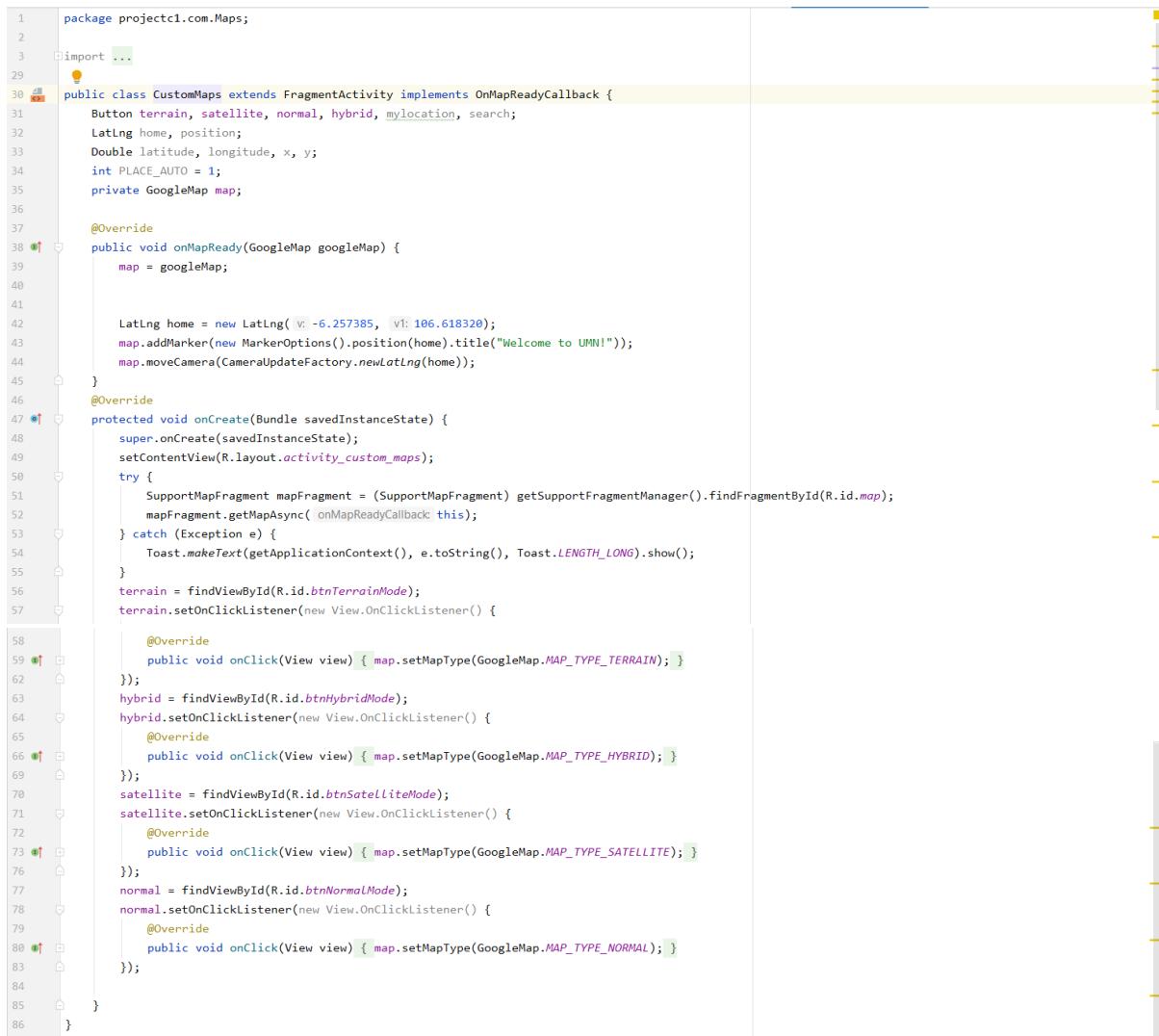
        case R.id.btnMaps:
            startActivity(new Intent(getApplicationContext(), CustomMaps.class));
            break;
    }
}

```

Gambar 2.37 Code Informasi

Fragment Informasi berisikan elemen *WebView*, dimana pengguna nantinya akan diarahkan ke masing-masing media sosial yang dimiliki oleh HIMSI, seperti Instagram, Youtube, Official Line dan Official Website serta mengetahui lokasi gedung Universitas Multimedia Nusantara. Untuk menggunakan *WebView* diperlukan *link* atau URL dari halaman *web*.

22. CustomMaps



```
1 package projectc1.com.Maps;
2
3 import ...
4
5
6 public class CustomMaps extends FragmentActivity implements OnMapReadyCallback {
7     Button terrain, satellite, normal, hybrid, mylocation, search;
8     LatLng home, position;
9     Double latitude, longitude, x, y;
10    int PLACE_AUTO = 1;
11    private GoogleMap map;
12
13    @Override
14    public void onMapReady(GoogleMap googleMap) {
15        map = googleMap;
16
17        LatLng home = new LatLng( -6.257385, 106.618320 );
18        map.addMarker(new MarkerOptions().position(home).title("Welcome to UMN!"));
19        map.moveCamera(CameraUpdateFactory.newLatLng(home));
20    }
21
22    @Override
23    protected void onCreate(Bundle savedInstanceState) {
24        super.onCreate(savedInstanceState);
25        setContentView(R.layout.activity_custom_maps);
26        try {
27            SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager().findFragmentById(R.id.map);
28            mapFragment.getMapAsync( onMapReadyCallback );
29        } catch (Exception e) {
30            Toast.makeText(getApplicationContext(), e.toString(), Toast.LENGTH_LONG).show();
31        }
32        terrain = findViewById(R.id.btnTerrainMode);
33        terrain.setOnClickListener(new View.OnClickListener() {
34
35            @Override
36            public void onClick(View view) { map.setMapType(GoogleMap.MAP_TYPE_TERRAIN); }
37        });
38        hybrid = findViewById(R.id.btnHybridMode);
39        hybrid.setOnClickListener(new View.OnClickListener() {
40
41            @Override
42            public void onClick(View view) { map.setMapType(GoogleMap.MAP_TYPE_HYBRID); }
43        });
44        satellite = findViewById(R.id.btnSatelliteMode);
45        satellite.setOnClickListener(new View.OnClickListener() {
46
47            @Override
48            public void onClick(View view) { map.setMapType(GoogleMap.MAP_TYPE_SATELLITE); }
49        });
50        normal = findViewById(R.id.btnNormalMode);
51        normal.setOnClickListener(new View.OnClickListener() {
52
53            @Override
54            public void onClick(View view) { map.setMapType(GoogleMap.MAP_TYPE_NORMAL); }
55        });
56    }
57
58 }
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86 }
```

Gambar 2.38 Code Custom Maps

Code diatas merupakan *code* untuk menampilkan menu *maps*. Ketika pengguna menekan tombol *maps* pada menu *Information*, pengguna akan langsung berpindah ke halaman baru yang menampilkan lokasi dari gedung Universitas Multimedia Nusantara. Fungsi *LatLng* digunakan untuk menyatakan lokasi gedung Universitas Multimedia Nusantara berdasarkan koordinatnya secara geografis. Pada *code* diatas juga terdapat fungsi *addMarker* yang berfungsi untuk menampilkan simbol serta tulisan singkat pada saat *marker* tersebut ditekan. Dalam *code* diatas juga terdapat fungsi *.setOnClickListener* pada button button yang telah diinisialisasikan idnya, yang didalam fungsi tersebut terdapat fungsi *.setMapTypes* sesuai dengan button yang ditekan yang berfungsi agar dapat menampilkan tipe *map* tertentu sesuai dengan button yang ditekan.

23. Reservation

```
1 package projectc1.com.Reservation;
2
3 import ...
45
46 public class Reservation extends AppCompatActivity {
47
48     private RecyclerView TutorRecyclerView;
49     private projectc1.com.adapter.TutorAdapter TutorAdapter;
50     private List<Tutor> TutorList;
51     private RecyclerView.LayoutManager manager;
52
53     ImageView dataanalis, accountingg, databasee, algoritma, mobile;
54     private static final String BASE_URL = "https://bimbelhimsi.000webhostapp.com/BIMBEL/tutor.php";
55
56     @Override
57     protected void onCreate(Bundle savedInstanceState) {
58         super.onCreate(savedInstanceState);
59         setContentView(R.layout.activity_invoice);
60         setTitle("BIMSI Reservation");
61
62         dataanalis = findViewById(R.id.imageDataanal);
63         accountingg = findViewById(R.id.imageAccounting);
64         databasee = findViewById(R.id.imageDatabase);
65         algoritma = findViewById(R.id.imageAlgorithm);
66         mobile = findViewById(R.id.imageMobileapp);
67
68         TutorRecyclerView = findViewById(R.id.ProductRec);
69         TutorList = new ArrayList<>();
70         getTutor();
71
72         dataanalis.setOnClickListener(new View.OnClickListener() {
73             @Override
74             public void onClick(View v) {
75                 TutorRecyclerView.setAdapter(null);
76                 TutorList.clear();
77
78                 getTutorDetail(choose: "MK001");
79             }
80         });
81         accountingg.setOnClickListener(new View.OnClickListener() {
82             @Override
83             public void onClick(View v) {
84                 TutorRecyclerView.setAdapter(null);
85                 TutorList.clear();
86
87                 getTutorDetail(choose: "MK002");
88             }
89         });
90         databasee.setOnClickListener(new View.OnClickListener() {
91             @Override
92             public void onClick(View v) {
93                 TutorRecyclerView.setAdapter(null);
94                 TutorList.clear();
95
96                 getTutorDetail(choose: "MK003");
97             }
98         });
99
100        algoritma.setOnClickListener(new View.OnClickListener() {
101            @Override
102            public void onClick(View v) {
103                TutorRecyclerView.setAdapter(null);
104                TutorList.clear();
105
106                getTutorDetail(choose: "MK004");
107            }
108        });
109        mobile.setOnClickListener(new View.OnClickListener() {
110            @Override
111            public void onClick(View v) {
112                TutorRecyclerView.setAdapter(null);
113                TutorList.clear();
114
115                getTutorDetail(choose: "MK005");
116            }
117        });
118        BottomNavigationView bottomNavigationView = findViewById(R.id.bottomNavigation);
119
120        bottomNavigationView.setSelectedItemId(R.id.registrasi);
121        bottomNavigationView.setOnNavigationItemSelectedListener(new BottomNavigationView.OnNavigationItemSelectedListener() {
122            @Override
123            public boolean onNavigationItemSelected(@NonNull MenuItem item) {
124                switch (item.getItemId()) {
125                    case R.id.forum:
126                        startActivity(new Intent(getApplicationContext(), Forum.class));
127                        overridePendingTransition(enterAnim: 0, exitAnim: 0);
128                        return true;
129                }
130            }
131        });
132    }
133}
```

```

128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211

        case R.id.home:
            startActivity(new Intent(getApplicationContext(), Button.class));
            overridePendingTransition(enterAnim: 0, exitAnim: 0);
            return true;

        case R.id.information:
            startActivity(new Intent(getApplicationContext(), Informasi.class));
            overridePendingTransition(enterAnim: 0, exitAnim: 0);
            return true;

        case R.id.registrasi:
            return true;
    }
    return false;
}
});

private void getTutor() {
    StringRequest stringRequest = new StringRequest(Request.Method.GET, BASE_URL,
        new Response.Listener<String>() {
            @Override
            public void onResponse(String response) {

                try {
                    JSONArray array = new JSONArray(response);
                    for (int i = 0; i < array.length(); i++) {

                        JSONObject object = array.getJSONObject(i);

                        String namaTutor = object.getString(name: "namaTutor");
                        String kode_kelas = object.getString(name: "kode_kelas");
                        String jadwal = object.getString(name: "jadwal");
                        String kode_namasubject = object.getString(name: "kode_namasubject");
                        String stok = object.getString(name: "stok");
                        String harga = object.getString(name: "harga");
                        String gambar = object.getString(name: "gambar");
                        String note = object.getString(name: "note");
                        ;

                        Tutor tutor = new Tutor(namaTutor, kode_kelas, jadwal, stok, harga, gambar, note, kode_namasubject);
                        TutorList.add(tutor);
                    }
                } catch (Exception e) {
                }

                TutorAdapter = new TutorAdapter(context: Reservation.this, TutorList);
                TutorRecyclerView.setAdapter(TutorAdapter);
            }
        }, new Response.ErrorListener() {

            @Override
            public void onErrorResponse(VolleyError error) {
                Toast.makeText(context: Reservation.this, error.toString(), Toast.LENGTH_LONG).show();
            }
        });
    Volley.newRequestQueue(context: Reservation.this).add(stringRequest);
    setProductRec(TutorList);
}

private void getTutorDetail(String choose) {
    StringRequest stringRequest = new StringRequest(Request.Method.GET, BASE_URL,
        new Response.Listener<String>() {
            @Override
            public void onResponse(String response) {

                try {
                    JSONArray array = new JSONArray(response);
                    for (int i = 0; i < array.length(); i++) {

                        JSONObject object = array.getJSONObject(i);

```

```

213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257

    String namaTutor = object.getString( name: "namaTutor");
    String kode_kelas = object.getString( name: "kode_kelas");
    String jadwal = object.getString( name: "jadwal");
    String kode_namasubject = object.getString( name: "kode_namasubject");
    String stok = object.getString( name: "stok");
    String harga = object.getString( name: "harga");
    String gambar = object.getString( name: "gambar");
    String note = object.getString( name: "note");
    Tutor tutor = new Tutor(namaTutor, kode_kelas, jadwal, stok, harga, gambar, note, kode_namasubject);
    if (kode_namasubject.equalsIgnoreCase(choose)) {
        TutorList.add(tutor);
    }
}
} catch (Exception e) {

}

TutorAdapter = new TutorAdapter( context: Reservation.this, TutorList);
TutorRecyclerView.setAdapter(TutorAdapter);

}, new Response.ErrorListener() {
@Override
public void onErrorResponse(VolleyError error) {
    Toast.makeText( context: Reservation.this, error.toString(), Toast.LENGTH_LONG).show();
});

Volley.newRequestQueue( context: Reservation.this).add(stringRequest);
setProductRec(TutorList);

}

private void setProductRec(List<Tutor> ProductList) {
    manager = new GridLayoutManager( context: Reservation.this, spanCount: 2);
    TutorRecyclerView.setLayoutManager(manager);
}

private String formatRupiah(int number) {
    Locale localeID = new Locale( language: "in", country: "ID");
    NumberFormat formatRupiah = NumberFormat.getCurrencyInstance(localeID);
    return formatRupiah.format(number);
}
}

```

Gambar 2.39 *Code Reservation*

Class Reservation berisikan elemen *RecyclerView*, *Adapter*, *ArrayList*, *ImageView*, dan URL. Pada menu ini, pengguna dapat melakukan reservasi kuota kelas bimbel, dimana pengguna akan memilih mata kuliah dengan menekan *ImageButton* wajah tutor. Saat pengguna menekan *ImageButton* akan langsung diarahkan ke halaman *TutorBIMSI* yang terdapat dalam *class Detail*, yang berisikan data dan profil singkat mengenai tutor mata kuliah terkait. *RecyclerView* dapat digunakan untuk melihat tutor yang mengajar mata kuliah terkait, ketika pengguna menekan mata kuliah maka akan muncul wajah tutor yang mengajar. Selain itu, juga terdapat *code JSONArray*, yang berfungsi untuk menampung data-data terkait tutor dan kelas dari database. Pada halaman *Tutor BIMSI* juga terdapat *button Reserve Now* yang akan mengarahkan pengguna ke halaman baru, yaitu *Reservation* atau *class FormPembelian*.

24. Detail

```
1 package projectc1.com.Admin;
2
3 import ...
4
5
6 public class Detail extends AppCompatActivity {
7
8     private TextView mNamaTutor, mKode_kelas, mStok, mHarga, mNote, mJadwal;
9     private ImageView mGambar;
10    private Button btnBuy;
11
12
13    @Override
14    protected void onCreate(Bundle savedInstanceState) {
15        super.onCreate(savedInstanceState);
16        setContentView(R.layout.activity_detail);
17        setTitle("Tutor BIMSI");
18
19        mGambar = findViewById(R.id.gambar);
20        mNamaTutor = findViewById(R.id.namaTutor);
21        mKode_kelas = findViewById(R.id.kode_kelas);
22        mJadwal = findViewById(R.id.jadwal);
23        mStok = findViewById(R.id.stok);
24        mHarga = findViewById(R.id.harga);
25        mNote = findViewById(R.id.note);
26
27        btnBuy = findViewById(R.id.btnBuy);
28
29        Intent i = getIntent();
30        String gambar = i.getStringExtra( name: "gambar");
31        String namaTutor = i.getStringExtra( name: "namaTutor");
32        String jadwal = i.getStringExtra( name: "jadwal");
33        String kode_kelas = i.getStringExtra( name: "kode_kelas");
34        String stok = i.getStringExtra( name: "stok");
35        String harga = i.getStringExtra( name: "harga");
36
37        String note = i.getStringExtra( name: "note");
38
39        if ( i !=null){
40
41            mNamaTutor.setText(namaTutor);
42            mKode_kelas.setText(kode_kelas);
43            mJadwal.setText(jadwal);
44            mStok.setText(stok);
45            String detailHarga = (formatRupiah(Double.parseDouble(String.valueOf(harga))));
46            mHarga.setText(detailHarga);
47            mNote.setText(note);
48            Glide.with( activity: Detail.this).load(gambar).into(mGambar);
49
50        }
51
52
53        btnBuy.setOnClickListener(new View.OnClickListener() {
54            @Override
55            public void onClick(View v) {
56                Intent i = new Intent( packageContext: Detail.this, FormPembelian.class);
57
58                i.putExtra( name: "kode_kelas", kode_kelas);
59                i.putExtra( name: "namaTutor", namaTutor);
60                i.putExtra( name: "jadwal", jadwal);
61                i.putExtra( name: "stok", stok);
62                i.putExtra( name: "harga", harga);
63
64                startActivity(i);
65            }
66        });
67    }
68
69
70    private String formatRupiah(Double number) {
71        Locale localeID = new Locale( language: "in", country: "ID");
72        NumberFormat formatRupiah = NumberFormat.getCurrencyInstance(localeID);
73        return formatRupiah.format(number);
74    }
75
76
77
78
79
80
81    }
82
83
84
85
86
87
88 }
```

Gambar 2.40 Code Detail

Class Detail berisikan elemen *TextView*, *ImageView*, dan *Button*. Pada menu ini, pengguna dapat membaca data dan profil singkat mengenai tutor serta informasi mengenai kuota kelas dan mata kuliah yang diajarkan oleh tutor terkait. Informasi mata kuliah yang terdapat pada *class* ini yaitu kode kelas, stok, harga, kuota, dan jadwal kelas. Pengguna juga

dapat melihat wajah tutor yang mengajar mata kuliah tersebut lewat *ImageView* yang terdapat pada *class* ini. Sedangkan, semua data mengenai informasi yang tertulis ditampung dalam *TextView* dengan untuk kemudian ditampilkan. Terakhir, pada *class* ini juga terdapat *button Reserve Now* yang akan mengarahkan pengguna ke halaman baru, yaitu *Reservation* atau *class* *FormPembelian*.

25. Form Pembelian

```

1 package projectc1.com.Reservation;
2
3 import ...
4
5 public class FormPembelian extends AppCompatActivity {
6
7     TextView txtNamaTutor, txtHarga, txtTotal, txtKode_kelas;
8     String total_harga;
9
10    EditText EdtNama_mahasiswa, EdtLine, EdtJumlah, EdtNotes;
11
12    android.widget.Button btnConfirm, btnHitung;
13
14    String hargaa, jumlah_pesan;
15
16    String nama_mahasiswa, line, kode_kelas, harga, jangka_waktu, total, notes;
17
18    String url_tambah_penjualan = "https://bimbelhimsi.000webhostapp.com/BIMBEL/pemesanan.php";
19
20    @Override
21    protected void onCreate(Bundle savedInstanceState) {
22        super.onCreate(savedInstanceState);
23        setContentView(R.layout.activity_form_pembelian);
24        setTitle("Reservation Page");
25
26        txtKode_kelas = findViewById(R.id.kodekelas);
27        txtNamaTutor = findViewById(R.id.namaTutor);
28        txtHarga = findViewById(R.id.hargaPerminggu);
29        btnHitung = findViewById(R.id.btnHitung);
30        btnConfirm = findViewById(R.id.btnConfirm);
31        EdtNama_mahasiswa = findViewById(R.id.nama);
32        EdtLine = findViewById(R.id.line);
33        EdtJumlah = findViewById(R.id.jumlah);
34        EdtNotes = findViewById(R.id.notes); //harusnya gambar
35
36
37        txtTotal = findViewById(R.id.txtTotal);
38
39        Intent i = getIntent();
40        String getNamaTutor = i.getStringExtra( name: "namaTutor");
41        String getKode_kelas = i.getStringExtra( name: "kode_kelas");
42        hargaa = i.getStringExtra( name: "harga");
43        jumlah_pesan = i.getStringExtra( name: "jangka_waktu");
44        txtNamaTutor.setText(getNamaTutor);
45        txtKode_kelas.setText(getKode_kelas);
46        String hargaDatabase = harga;
47        txtHarga.setText(formatRupiah(Double.parseDouble(hargaDatabase)));
48
49
50        btnHitung.setOnClickListener(new View.OnClickListener() {
51            @Override
52            public void onClick(View v) {
53                int subtotal, jumlahh, hargaax;
54                jumlahh = Integer.valueOf(EdtJumlah.getText().toString().trim());
55                hargaax = Integer.valueOf(hargaDatabase.trim());
56                subtotal = jumlahh * hargaax;
57                if (jumlahh==4){
58                    String totalHarga = (formatRupiah(Double.parseDouble(valueOf(subtotal))));
59                    txtTotal.setText(totalHarga);
60                    total_harga = valueOf(Integer.valueOf(subtotal));
61                }else {
62                    Toast.makeText( context: FormPembelian.this, text: "Jumlah Reserve Harus 4", Toast.LENGTH_SHORT).show();
63                }
64            }
65        });
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91

```

```

93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118

    btnConfirm.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            nama_mahasiswa = EdtNama_mahasiswa.getText().toString();
            line = Edtline.getText().toString();
            kode_kelas = txtKode_kelas.getText().toString();
            harga = hargaDatabase;
            jangka_waktu = EdtJumlah.getEditText().toString();
            total = total_harga;
            notes = EdtNotes.getText().toString();

            if (EdtNama_mahasiswa.length() > 0 && Edtline.length() > 0 && txtKode_kelas.length() > 0 && hargaDatabase.length() > 0 && EdtJumlah.length() > 0 && total_harga.length() > 0 && EdtNotes.length() > 0 ) {
                tambahPemesanan(nama_mahasiswa, line, kode_kelas, harga, jangka_waktu, total, notes);
            } else {
                Toast.makeText( context: FormPembelian.this, text: "Data tidak boleh kosong!", Toast.LENGTH_SHORT).show();
            }
        }
    });

    private String formatRupiah(Double number) {
        Locale localeID = new Locale( language: "in", country: "ID");
        NumberFormat formatRupiah = NumberFormat.getCurrencyInstance(localeID);
        return formatRupiah.format(number);
    }

    private void tambahPemesanan(String nama_mahasiswa, String line, String kode_kelas, String harga, String jangka_waktu, String total, String notes) {
        RequestQueue requestQueue = Volley.newRequestQueue( context: FormPembelian.this);
        StringRequest stringRequest = new StringRequest(Request.Method.POST, url_tambah_penjualan, new Response.Listener<String>() {
            @Override
            public void onResponse(String response) {
                try {
                    JSONObject jsonObject = new JSONObject(response);
                    int success = jsonObject.getInt( name: "success");
                    if (success == 1) {
                        Toast.makeText( context: FormPembelian.this, text: "Invoice Berhasil dibuat", Toast.LENGTH_SHORT).show();
                        Intent i = new Intent( packageContext: FormPembelian.this, Reservation.class);
                        startActivity(i);
                        finish();
                    } else {
                        Toast.makeText( context: FormPembelian.this, text: "Invoice gagal dibuat", Toast.LENGTH_SHORT).show();
                    }
                } catch (Exception ex) {
                    Log.e( tag: "Error", ex.toString());
                }
            }
        }, new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError error) {
                Toast.makeText( context: FormPembelian.this, text: "Cek koneksi anda terlebih dahulu!", Toast.LENGTH_SHORT).show();
            }
        });
    }

    @Override
    protected Map<String, String> getParams() throws AuthFailureError {
        Map<String, String> params = new HashMap<>();
        params.put( k: "nama_mahasiswa", nama_mahasiswa);
        params.put( k: "line", line);
        params.put( k: "kode_kelas", kode_kelas);
        params.put( k: "harga", harga);
        params.put( k: "jangka_waktu", jangka_waktu);
        params.put( k: "total", total);
        params.put( k: "notes", notes);
        return params;
    }
};

requestQueue.getCache().clear();
requestQueue.add(stringRequest);
}

```

Gambar 2.41 Code Form Pembelian

Class FormPembelian diperuntukkan untuk mahasiswa yang telah menentukan mata kuliah dan tutor yang ingin direservasi. Setelah menentukan kedua hal tersebut, akan tampil Nama Tutor sesuai dengan pilihan, dimana nama tutor didapatkan dari hasil parsing data. Lalu mahasiswa perlu menginput nama, id line, jumlah reservasi, dan catatan (*note*) pembayaran. Jumlah reservasi disini sudah diatur pada *code java* untuk harus berjumlah 4 karena reservasi hanya boleh dilakukan sebanyak empat minggu atau satu bulan. Setelah sudah selesai mengisi,

pengguna dapat menekan *Button Calculate* untuk melihat hasil perkalian atau total pembayarannya. Dibawahnya, terdapat *note* pembayaran yakni pesan mengenai metode pembayaran yang dipilih pengguna dan nama pengguna yang mentransfer. Semua *fields* yang harus terpenuhi wajib disi, karena jika tidak akan muncul *Toast* “Data tidak boleh kosong”. Ketika data yang diminta sudah sesuai dan terisi semua, *requestQueue* dan *olley* akan membantu memarsing data dengan metode POST ke dalam database dan dibantu juga oleh php *url_tambah_penjualan* dan *JSONObject* untuk mengirim datanya. Saat data berhasil dikirim dan dibuat pada database, maka akan muncul *Toast* “Invoice Berhasil dibuat”.

26. Read Admin

```

90     getReadDetail( choose: "MK002");
91   });
92 }
93 databasee.setOnClickListener(new View.OnClickListener() {
94   @Override
95   public void onClick(View v) {
96     ReadRecyclerView.setAdapter(null);
97     ReadList.clear();
98
99     getReadDetail( choose: "MK003");
100  });
101 });
102 algoritma.setOnClickListener(new View.OnClickListener() {
103   @Override
104   public void onClick(View v) {
105     ReadRecyclerView.setAdapter(null);
106     ReadList.clear();
107
108     getReadDetail( choose: "MK004");
109  });
110 });
111 mobile.setOnClickListener(new View.OnClickListener() {
112   @Override
113   public void onClick(View v) {
114     ReadRecyclerView.setAdapter(null);
115     ReadList.clear();
116
117     getReadDetail( choose: "MK005");
118  });
119 });
120 }
121 }

123 private void getRead() {
124
125   StringRequest stringRequest = new StringRequest(Request.Method.GET, BASE_URL,
126     new Response.Listener<String>() {
127       @Override
128       public void onResponse(String response) {
129
130         try {
131
132           JSONArray array = new JSONArray(response);
133           for (int i = 0; i < array.length(); i++) {
134
135             JSONObject object = array.getJSONObject(i);
136             String kode_kelas = object.getString( name: "kode_kelas");
137             String kode_namasubject = object.getString( name: "kode_namasubject");
138             String kode_tutor = object.getString( name: "kode_tutor");
139             String namaTutor = object.getString( name: "namaTutor");
140             String jadwal = object.getString( name: "jadwal");
141             String stok = object.getString( name: "stok");
142             String harga = object.getString( name: "harga");
143             String gambar = object.getString( name: "gambar");
144             String note = object.getString( name: "note");
145             Read read = new Read(kode_kelas, kode_namasubject, kode_tutor, namaTutor, jadwal, stok, harga, gambar, note);
146             ReadList.add(read);
147           }
148         } catch (Exception e) {
149
150         }
151
152         ReadAdapter = new ReadAdapter( context: ReadAdmin.this, ReadList);
153         ReadRecyclerView.setAdapter(ReadAdapter);
154     }
155
156   }, new Response.ErrorListener() {
157     @Override
158     public void onErrorResponse(VolleyError error) {
159
160       Toast.makeText( context: ReadAdmin.this, error.toString(), Toast.LENGTH_LONG).show();
161
162     });
163
164   Volley.newRequestQueue( context: ReadAdmin.this).add(stringRequest);
165   setReadRec(ReadList);
166
167 }
168 private void getReadDetail(String choose) {
169
170   StringRequest stringRequest = new StringRequest(Request.Method.GET, BASE_URL,
171     new Response.Listener<String>() {
172       @Override
173       public void onResponse(String response) {
174
175         try {
176
177           JSONArray array = new JSONArray(response);
178           for (int i = 0; i < array.length(); i++) {

```

```

180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220

        JSONObject object = array.getJSONObject(i);
        String kode_kelas = object.getString( name: "kode_kelas");
        String kode_namasubject = object.getString( name: "kode_namasubject");
        String kode_tutor = object.getString( name: "kode_tutor");
        String namaTutor = object.getString( name: "namaTutor");
        String jadwal = object.getString( name: "jadwal");
        String stok = object.getString( name: "stok");
        String harga = object.getString( name: "harga");
        String gambar = object.getString( name: "gambar");
        String note = object.getString( name: "note");
        Read read = new Read(kode_kelas, kode_namasubject, kode_tutor, namaTutor, jadwal, stok, harga, gambar, note);
        if (kode_namasubject.equalsIgnoreCase(choose)) {
            ReadList.add(read);
        }
    } catch (Exception e) {

}

ReadAdapter = new ReadAdapter( context: ReadAdmin.this, ReadList);
ReadRecyclerView.setAdapter(ReadAdapter);

}, new Response.ErrorListener() {
@Override
public void onErrorResponse(VolleyError error) {

    Toast.makeText( context: ReadAdmin.this, error.toString(), Toast.LENGTH_LONG).show();
}
});

Volley.newRequestQueue( context: ReadAdmin.this).add(stringRequest);
setReadRec(ReadList);

private void setReadRec(List<Read> ReadList) {
    manager = new GridLayoutManager( Context: ReadAdmin.this, spanCount: 2);
    ReadRecyclerView.setLayoutManager(manager);
}
}

```

Gambar 2.42 Code Read Admin

Class Read Admin berisikan elemen *RecyclerView*, *Adapter*, *ArrayList*, *ImageView*, dan URL. Menu ini hanya dikhususkan untuk admin dan tidak dapat dilihat oleh pengguna, dimana admin dapat melihat ikon setiap mata kuliah yang diajarkan oleh masing-masing tutor dan daftar dari setiap tutor yang aktif mengajar di bimbingan belajar BIMSI. *Code RecyclerView* digunakan untuk melihat tutor yang mengajar mata kuliah terkait, ketika admin menekan mata kuliah maka akan muncul wajah tutor yang mengajar. Saat admin menekan *ImageButton* wajah tutor, admin akan diarahkan ke kelas BIMSI, halaman ini berisi kode kelas, kode mata kuliah, kode tutor, nama tutor, jadwal, kuota, harga, wajah tutor, dan catatan, dimana data-data tersebut ditampung dari database dengan menggunakan *code JSONArray*.

27. Update Admin

```
1 package project1.com.Admin;
2
3 import ...
4
5 public class UpdateAdmin extends AppCompatActivity {
6
7     TextView txtKodeKelas, txtKodeNamasubject, txtKodeTutor, txtNamaTutor, txtNote, txtJadwal; //kemungkinan salah di sini namanya sama
8     EditText EdtHarga, EdtStok;
9     ImageView ImgGambar;
10    String id_tutor, kode_kelas, kode_namasubject, kode_tutor, namaTutor, stok, harga, gambar, note, jadwal;
11    private Button btnUpdate, btnDelete;
12
13
14    String url_update = "https://bimbelhimsi.000webhostapp.com/BIMBEL/updateTutorStok.php";
15    String url_delete = "https://bimbelhimsi.000webhostapp.com/BIMBEL/deleteTutor.php";
16
17    @Override
18    protected void onCreate(Bundle savedInstanceState) {
19        super.onCreate(savedInstanceState);
20        setContentView(R.layout.activity_update_admin);
21
22        txtKodeKelas = findViewById(R.id.txtKodekelas);
23        txtKodeNamasubject = findViewById(R.id.txtkodeemasubject);
24        txtKodeTutor = findViewById(R.id.txtkodetutor);
25        txtNamaTutor = findViewById(R.id.txtnamatutor);
26        txtJadwal = findViewById(R.id.txtJadwal);
27        EdtStok = findViewById(R.id.EdtStok);
28        EdtHarga = findViewById(R.id.EdtHargaSatuan);
29        ImgGambar = findViewById(R.id.ImgGambar);
30        txtNote = findViewById(R.id.txtNotes);
31
32        btnUpdate = findViewById(R.id.btnUpdate);
33        btnDelete = findViewById(R.id.btnDelete);
34
35
36        Intent i = getIntent();
37
38        id_tutor = i.getStringExtra( name: "id_tutor");
39        kode_kelas = i.getStringExtra( name: "kode_kelas");
40        kode_namasubject = i.getStringExtra( name: "kode_namasubject");
41        kode_tutor = i.getStringExtra( name: "kode_tutor");
42        namaTutor = i.getStringExtra( name: "namaTutor");
43        jadwal = i.getStringExtra( name: "jadwal");
44        stok = i.getStringExtra( name: "stok");
45        harga = i.getStringExtra( name: "harga");
46        gambar = i.getStringExtra( name: "gambar");
47        note = i.getStringExtra( name: "note");
48
49        txtKodeKelas.setText(kode_kelas);
50        txtKodeNamasubject.setText(kode_namasubject);
51        txtKodeTutor.setText(kode_tutor);
52        txtNamaTutor.setText(namaTutor);
53        txtJadwal.setText(jadwal);
54        EdtStok.setText(stok);
55        EdtHarga.setText(harga);
56        txtNote.setText(note);
57
58        Glide.with( activity: UpdateAdmin.this).load(gambar).into(ImgGambar);
59
60
61        btnDelete.setOnClickListener(new View.OnClickListener() {
62            @Override
63            public void onClick(View v) {
64                deleteTutor(kode_kelas);
65
66                Intent i = new Intent( packageContext: UpdateAdmin.this, ReadAdmin.class);
67                startActivity(i);
68                finish();
69            }
70        });
71    }
72}
```

```
96
97
98     btnUpdate.setOnClickListener(new View.OnClickListener() {
99         @Override
100        public void onClick(View v) {
101
102            kode_kelas= txtKodeKelas.getText().toString();
103            stok = EdtStok.getText().toString();
104            harga = EdtHarga.getText().toString();
105            UpdateTutor(kode_kelas, stok, harga);
106        }
107    });
108
109
110    private void UpdateTutor(String kode_kelas, String stok, String harga) {
111        RequestQueue requestQueue = Volley.newRequestQueue( context: UpdateAdmin.this );
112        StringRequest stringRequest = new StringRequest(Request.Method.POST, url_update, new Response.Listener<String>() {
113            @Override
114            public void onResponse(String response) {
115                try {
116                    JSONObject jsonObject = new JSONObject(response);
117                    int success = jsonObject.getInt( name: "success");
118                    if (success == 1) {
119                        Toast.makeText( context: UpdateAdmin.this, text: "Data tutor berhasil diupdate", Toast.LENGTH_SHORT).show();
120                        Intent i = new Intent( packageContext: UpdateAdmin.this, ReadAdmin.class);
121                        startActivity(i);
122                        finish();
123                    } else {
124                        Toast.makeText( context: UpdateAdmin.this, text: "Data tutor gagal diupdate", Toast.LENGTH_SHORT).show();
125                    }
126                } catch (JSONException e) {
127                    e.printStackTrace();
128                }
129            }
130        }, new Response.ErrorListener() {
131            @Override
132            public void onErrorResponse(VolleyError error) {
133                Toast.makeText( context: UpdateAdmin.this, text: "Connection Error!", Toast.LENGTH_SHORT).show();
134            }
135        } {
136            @Nullable
137            @Override
138            protected Map<String, String> getParams() throws AuthFailureError {
139                Map<String, String> params = new HashMap<>();
140                params.put( key: "kode_kelas", kode_kelas);
141                params.put( key: "stok", stok);
142                params.put( key: "harga", harga);
143                return params;
144            }
145        };
146        requestQueue.getCache().clear();
147        requestQueue.add(stringRequest);
148    }
149
150    private void deleteTutor(String mKode_kelas) {
151        RequestQueue requestQueue = Volley.newRequestQueue( context: UpdateAdmin.this );
152        StringRequest stringRequest = new StringRequest(Request.Method.POST, url_delete,
153            new Response.Listener<String>() {
154                @Override
155                public void onResponse(String response) {
156                    try {
157                        JSONObject jsonObject = new JSONObject(response);
158                        int sukses = jsonObject.getInt( name: "success");
159                        if (sukses == 1) {
160                            Toast.makeText( context: UpdateAdmin.this,
161                                text: "Data Tutor berhasil dihapus", Toast.LENGTH_SHORT).show();
162                        }
163                    } catch (JSONException e) {
164                        e.printStackTrace();
165                    }
166                }
167            }
168        );
169    }
170}
```

```
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
```



```
    } else {
        Toast.makeText( context: UpdateAdmin.this,
                        text: "Data Tutor berhasil dihapus", Toast.LENGTH_SHORT).show();
    }
} catch (JSONException e) {
    Toast.makeText( context: UpdateAdmin.this,
                    e.getMessage(), Toast.LENGTH_SHORT).show();
}
}
}, new Response.ErrorListener() {
@Override
public void onErrorResponse(VolleyError error) {
    Toast.makeText( context: UpdateAdmin.this,
                    text: "Connection Error!", Toast.LENGTH_SHORT).show();
}
}
{
    @Nullable
    @Override
    protected Map<String, String> getParams() throws AuthFailureError {
        Map<String, String> params = new HashMap<>();
        params.put( k: "kode_kelas", mKode_kelas);
        return params;
    }
};
requestQueue.getCache().clear();
requestQueue.add(stringRequest);
}
```

Gambar 2. 43 Code Update Admin

Code *Update Admin* merupakan *code* pada tampilan admin untuk melakukan *update* terhadap data-data bimbel. Terdapat beberapa elemen dalam *codingan* ini, seperti *TextView*, *EditText*, *ImageView*, dan *Button*. Pada *class* ini, penulisan *code* diawali dengan penginisialisasi variabel-variabel, lalu dilanjutkan dengan penggunaan fungsi *findViewById* untuk menemukan tampilan yang diidentifikasi oleh atribut XML *android:id*. Kemudian, terdapat penggunaan *intent* dan *.getStringExtra* yang berfungsi untuk mengambil data yang telah di *parsing* dari kelas lain. Admin dapat melihat data terkait pemesanan kelas, seperti kode kelas, kode mata kuliah, kode tutor, nama tutor, jadwal, kuota, harga, foto tutor, dan catatan.

Admin dapat melakukan *update* data pada bagian kuota kelas dan harga serta menghapus tutor apabila tutor sudah tidak lagi mengajar di BIMSI. Fungsi *.setOnClickListener* adalah untuk merujukkan satu *button* ke fungsi lain, sehingga saat mengklik *button Delete*, itu juga mengacu pada *private void deleteTutor*. Di dalam *deleteTutor*, terdapat fungsi *parsing* data dari *Intent* dan *private void deleteTutor* yang akan meng-*update* data tutor, yang telah dihapus *php deleteTutor* dengan dibantu oleh *JSONObject*. Ketika admin ingin menghapus data tutor yang sudah tidak lagi mengajar, akan muncul *toast* “Data tutor berhasil dihapus” yang berarti aplikasi berhasil menghapus data tutor dari dalam database. Sebaliknya, jika aplikasi gagal menghapus data tutor dari dalam database akan muncul *toast* “Data tutor gagal dihapus”. Sedangkan, untuk *.setOnClickListener button Update* akan mengacu pada *private void updateTutor*. *Private void updateTutor* akan meng-*update* data tutor yang telah diganti yang juga menggunakan *JSONObject*. Ketika aplikasi berhasil mengirimkan perubahan data terkait kuota atau harga ke dalam database akan muncul *toast* “Data tutor berhasil disimpan”.

Sebaliknya, jika aplikasi gagal mengirimkan perubahan data terkait kuota atau harga ke dalam database akan muncul *toast* “Data tutor gagal disimpan”.

28. Admin Order

```
1 package projectc1.com.Admin;
2
3 import ...
4
5
6 public class AdminOrder extends AppCompatActivity {
7
8
9     private RecyclerView OrderRecyclerView;
10    private projectc1.com.adapter.OrderAdapter OrderAdapter;
11    private List<Order> OrderList;
12    private RecyclerView.LayoutManager manager;
13
14
15    private static final String url_order = "https://bimbelhimsi.000webhostapp.com/BIMBEL/lihatOrder.php";
16
17
18    @Override
19    protected void onCreate(Bundle savedInstanceState) {
20        setTitle("Student Reservation Data");
21        super.onCreate(savedInstanceState);
22        setContentView(R.layout.activity_admin_order);
23        OrderRecyclerView = findViewById(R.id.OrderRec);
24        OrderList = new ArrayList<>();
25        getOrder();
26
27
28        BottomNavigationView bottomNavigationView = findViewById(R.id.bottomNavigationadmin);
29        bottomNavigationView.setSelectedItemId(R.id.invoice);
30        bottomNavigationView.setOnNavigationItemSelectedListener(new BottomNavigationView.OnNavigationItemSelectedListener() {
31            @Override
32            public boolean onNavigationItemSelected(@NonNull MenuItem item) {
33                switch (item.getItemId()){
34                    case R.id.home:
35                        startActivity(new Intent(getApplicationContext(), ReadAdmin.class));
36                        overridePendingTransition(enterAnim: 0, exitAnim: 0);
37                        return true;
38
39
40                    case R.id.invoice:
41                        return true;
42
43                }
44                return false;
45            });
46
47
48
49    }
50
51
52    private void getOrder() {
53
54        StringRequest stringRequest = new StringRequest(Request.Method.GET, url_order,
55            new Response.Listener<String>() {
56                @Override
57                public void onResponse(String response) {
58
59                    try {
60
61                        JSONArray array = new JSONArray(response);
62                        for (int i = 0; i < array.length(); i++) {
63
64                            JSONObject object = array.getJSONObject(i);
65
66
67                            String tanggal_pemesanan = object.getString( name: "tanggal_pemesanan");
68                            String nama_mahasiswa = object.getString( name: "nama_mahasiswa");
69                            String line = object.getString( name: "line");
70                            String kode_kelas = object.getString( name: "kode_kelas");
71                            String harga = object.getString( name: "harga");
72                            String jangka_waktu = object.getString( name: "jangka_waktu");
73                            String total = object.getString( name: "total");
74                            String notes = object.getString( name: "notes");
75                            Order order = new Order(tanggal_pemesanan, nama_mahasiswa, line, kode_kelas, harga, jangka_waktu, total, notes);
76                            OrderList.add(order);
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94 }
```

```

96     }
97     }
98     }
99     }
100    }
101   }, new Response.ErrorListener() {
102     @Override
103     public void onErrorResponse(VolleyError error) {
104       Toast.makeText(context, AdminOrder.this, error.toString(), Toast.LENGTH_LONG).show();
105     }
106   });
107 }
108
109 private void setOrderRec(List<Order> OrderList) {
110
111   manager = new GridLayoutManager(context, AdminOrder.this, spanCount: 1);
112   OrderRecyclerView.setLayoutManager(manager);
113 }
114
115 }
116
117 private void setOrderRec(List<Order> OrderList) {
118
119   manager = new GridLayoutManager(context, AdminOrder.this, spanCount: 1);
120   OrderRecyclerView.setLayoutManager(manager);
121 }
122
123 }

```

Gambar 2.44 Code Admin Order

Class *Admin Order* berisikan elemen *RecyclerView*, *Adapter*, *List*, *ButtonNavigationView* dan URL. Melalui *RecyclerView*, admin dapat melihat data terkait pemesanan kelas yang dilakukan mahasiswa, seperti tanggal pemesanan, nama mahasiswa, id line, kode kelas, harga, jangka waktu, total, dan *notes*, data-data tersebut ditampung pada class *OrderAdapter* yang didapat dari database dengan menggunakan code *JSONArray*.

29. Order

```

1 package projectc1.com.Item;
2
3 public class Order {
4
5
6   private String tanggal_pemesanan;
7   private String nama_mahasiswa;
8   private String line;
9   private String kode_kelas;
10  private String harga;
11  private String jangka_waktu;
12  private String total;
13  private String notes;
14
15  public Order( String tanggal_pemesanan, String nama_mahasiswa, String line, String kode_kelas, String harga, String jangka_waktu, String total, String notes
16  {
17
18    this.tanggal_pemesanan = tanggal_pemesanan;
19    this.nama_mahasiswa = nama_mahasiswa;
20    this.line = line;
21    this.kode_kelas = kode_kelas;
22    this.harga = harga;
23    this.jangka_waktu = jangka_waktu;
24    this.total = total;
25    this.notes = notes;
26  }
27
28
29
30  public String getTanggal_pemesanan() { return tanggal_pemesanan; }
31
32  public void setTanggal_pemesanan(String tanggal_pemesanan) {
33    this.tanggal_pemesanan = tanggal_pemesanan;
34  }
35
36

```

```
38     public String getNama_mahasiswa() { return nama_mahasiswa; }
41     public void setNama_mahasiswa(String nama_mahasiswa) { this.nama_mahasiswa = nama_mahasiswa; }
45
46     public String getLine() { return line; }
49
50     public void setLine(String line) { this.line = line; }
53
54
55     public String getKode_kelas() { return kode_kelas; }
58     public void setKode_kelas(String kode_kelas) { this.kode_kelas = kode_kelas; }
62
63     public String getHarga() { return harga; }
66
67     public void setHarga(String harga) { this.harga = harga; }
70
71     public String getJangka_waktu() { return jangka_waktu; }
74
75     public void setJangka_waktu(String jangka_waktu) { this.jangka_waktu = jangka_waktu; }
78
79     public String getTotal() { return total; }
82
83     public void setTotal(String total) { this.total = total; }
86
87     public String getNotes() { return notes; }
90
91     public void setNotes(String notes) { this.notes = notes; }
94
95 }
```

Gambar 2.45 *Code Order*

Code class Order memuat metode *get* dan *set* berdasarkan variabel yang terdapat pada database. *Get* berfungsi untuk mendapatkan data, *return* berfungsi untuk mengembalikan nilai dari suatu variabel, sedangkan *set* berfungsi untuk memberikan nilai pada variabel yang diinginkan. Seluruh variabel pada *class* ini berperan untuk mengambil dan mengatur nilai variabel dalam menampilkan pemesanan mahasiswa dalam bentuk daftar *invoice* yang dapat dilihat oleh admin. *Class* ini kemudian akan melakukan *parsing data* ke *class Order Adapter*.

30. Read

```
1 package project1.com.Item;
2
3 public class Read {
4
5     private String kode_kelas;
6     private String kode_namasubject;
7     private String kode_tutor;
8     private String namaTutor;
9     private String jadwal;
10    private String stok;
11    private String harga;
12    private String gambar;
13    private String note;
14
15    public Read(String kode_kelas, String kode_namasubject, String kode_tutor, String namaTutor, String jadwal,
16                String stok, String harga, String gambar, String note) {
17
18        this.kode_kelas = kode_kelas;
19        this.kode_namasubject = kode_namasubject;
20        this.kode_tutor = kode_tutor;
21        this.namaTutor = namaTutor;
22        this.jadwal = jadwal;
23        this.stok = stok;
24        this.harga = harga;
25        this.gambar = gambar;
26        this.note = note;
27    }
28
29    public String getKode_kelas() { return kode_kelas; }
30
31    public void setKode_kelas(String kode_kelas) { this.kode_kelas = kode_kelas; }
32
33    public String getKode_namasubject() { return kode_namasubject; }
34
35    public void setKode_namasubject(String kode_namasubject) {
36        this.kode_namasubject = kode_namasubject;
37    }
38
39    public String getKode_tutor() { return kode_tutor; }
40
41    public void setKode_tutor(String kode_tutor) { this.kode_tutor = kode_tutor; }
42
43    public String getNamaTutor() { return namaTutor; }
44
45    public void setNamaTutor(String namaTutor) { this.namaTutor = namaTutor; }
46
47    public String getJadwal() { return jadwal; }
48
49    public void setJadwal(String jadwal) { this.namaTutor = jadwal; }
50
51    public String getStok() { return stok; }
52
53    public void setStok(String stok) { this.stok = stok; }
54
55    public String getHarga() { return harga; }
56
57    public void setHarga(String harga) { this.harga = harga; }
58
59    public String getGambar() { return gambar; }
60
61    public void setGambar(String gambar) { this.gambar = gambar; }
62
63    public String getNote() { return note; }
64
65    public void setNote(String note) { this.note = note; }
66
67}
```

Gambar 2.46 Code Read

Code class Read, memuat metode *get* dan *set* berdasarkan variabel yang terdapat pada database. *Get* berfungsi untuk mendapatkan data, *return* berfungsi untuk mengembalikan nilai dari suatu variabel, sedangkan *set* berfungsi untuk memberikan nilai pada variabel yang diinginkan. Seluruh variabel pada *class* ini berperan untuk mengambil dan mengatur nilai

variabel dalam menampilkan informasi tutor beserta mata kuliah yang dapat dilihat oleh admin.

Class ini kemudian akan melakukan *parsing data* ke class *Read Adapter*.

31. Tutor

```
1 package projectc1.com.Item;
2
3 public class Tutor {
4
5     private String namaTutor;
6     private String kode_kelas;
7     private String jadwal;
8     private String stok;
9     private String harga;
10    private String gambar;
11    private String note;
12    private String kode_namasubject;
13
14    public Tutor(String namaTutor, String kode_kelas, String jadwal, String stok, String harga, String gambar, String note, String kode_namasubject ) {
15
16        this.namaTutor = namaTutor;
17        this.kode_kelas = kode_kelas;
18        this.jadwal = jadwal;
19        this.stok = stok;
20        this.harga = harga;
21        this.gambar = gambar;
22        this.note = note;
23        this.kode_namasubject = kode_namasubject;
24    }
25
26    public String getNamaTutor() { return namaTutor; }
27
28    public void setNamaTutor(String namaTutor) { this.namaTutor = namaTutor; }
29
30    public String getJadwal() { return jadwal; }
31
32    public void setJadwal(String jadwal) { this.jadwal = jadwal; }
33
34    public String getStok() { return stok; }
35
36    public void setStok(String stok) { this.stok = stok; }
37
38    public String getHarga() { return harga; }
39
40    public void setHarga(String harga) { this.harga = harga; }
41
42    public String getGambar() { return gambar; }
43
44    public void setGambar(String gambar) { this.gambar = gambar; }
45
46    public String getNote() { return note; }
47
48    public void setNote(String note) { this.note = note; }
49
50    public String getKode_namasubject() { return kode_namasubject; }
51
52    public void setKode_namasubject(String kode_merk) { this.kode_namasubject = kode_namasubject; }
53
54    public String getCode_kelas() { return kode_kelas; }
55
56    public void setCode_kelas(String kode_kelas) { this.kode_kelas = kode_kelas; }
57
58}
```

Gambar 2.47 Code Tutor

Code class *Tutor*, memuat metode *get* dan *set* berdasarkan variabel yang terdapat pada database. *Get* berfungsi untuk mendapatkan data, *return* berfungsi untuk mengembalikan nilai dari suatu variabel, sedangkan *set* berfungsi untuk memberikan nilai pada variabel yang diinginkan. Seluruh variabel pada class ini berperan untuk mengambil dan mengatur nilai

variabel dalam menampilkan informasi tutor beserta mata kuliah yang dapat dilihat oleh mahasiswa. *Class* ini kemudian akan melakukan *parsing data* ke *class Tutor Adapter*.

32. Order Adapter

```

1 package projectc1.com.adapter;
2
3 import ...
21
22 public class ReadAdapter extends RecyclerView.Adapter<ReadAdapter.ReadViewHolder>{
23
24     private Context context;
25     private List<Read> ReadList;
26
27     public ReadAdapter(Context context, List<Read> ReadList) {
28
29         this.context = context;
30         this.ReadList = ReadList;
31     }
32
33     @NonNull
34     @Override
35     public ReadViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
36         View view = LayoutInflater.from(context).inflate(R.layout.read_rowtutor, parent, attachToRoot: false);
37
38         return new ReadViewHolder(view);
39     }
40
41     @Override
42     public void onBindViewHolder(@NonNull ReadViewHolder holder, int position) {
43
44         final Read read = ReadList.get(position);
45
46         holder.namaTutor.setText(ReadList.get(position).getNamaTutor());
47         Glide.with(context)
48             .load(read.getGambar())
49             .into(holder.gambar);
50
51         holder.itemView.setOnClickListener(new View.OnClickListener() {
52             @Override
53             public void onClick(View v) {
54
55                 Intent i = new Intent(context, UpdateAdmin.class);
56
57                 i.putExtra(name: "kode_kelas", ReadList.get(position).getKode_kelas());
58                 i.putExtra(name: "kode_namasubject", ReadList.get(position).getKode_namasubject());
59                 i.putExtra(name: "kode_tutor", ReadList.get(position).getKode_tutor());
60                 i.putExtra(name: "namatutor", ReadList.get(position).getNamaTutor());
61                 i.putExtra(name: "jadwal", ReadList.get(position).getJadwal());
62                 i.putExtra(name: "stok", ReadList.get(position).getStok());
63                 i.putExtra(name: "harga", ReadList.get(position).getHarga());
64                 i.putExtra(name: "gambar", ReadList.get(position).getGambar());
65                 i.putExtra(name: "note", ReadList.get(position).getNote());
66
67                 context.startActivity(i);
68             }
69         });
70     }
71
72     @Override
73     public int getItemCount() { return ReadList.size(); }
74
75     public class ReadViewHolder extends RecyclerView.ViewHolder {
76         private TextView namaTutor;
77         private ImageView gambar;
78
79         public ReadViewHolder(@NonNull View view) {
80             super(view);
81             namaTutor = view.findViewById(R.id.namatutor);
82             gambar = view.findViewById(R.id.gambar);
83         }
84     }
85 }
86
87

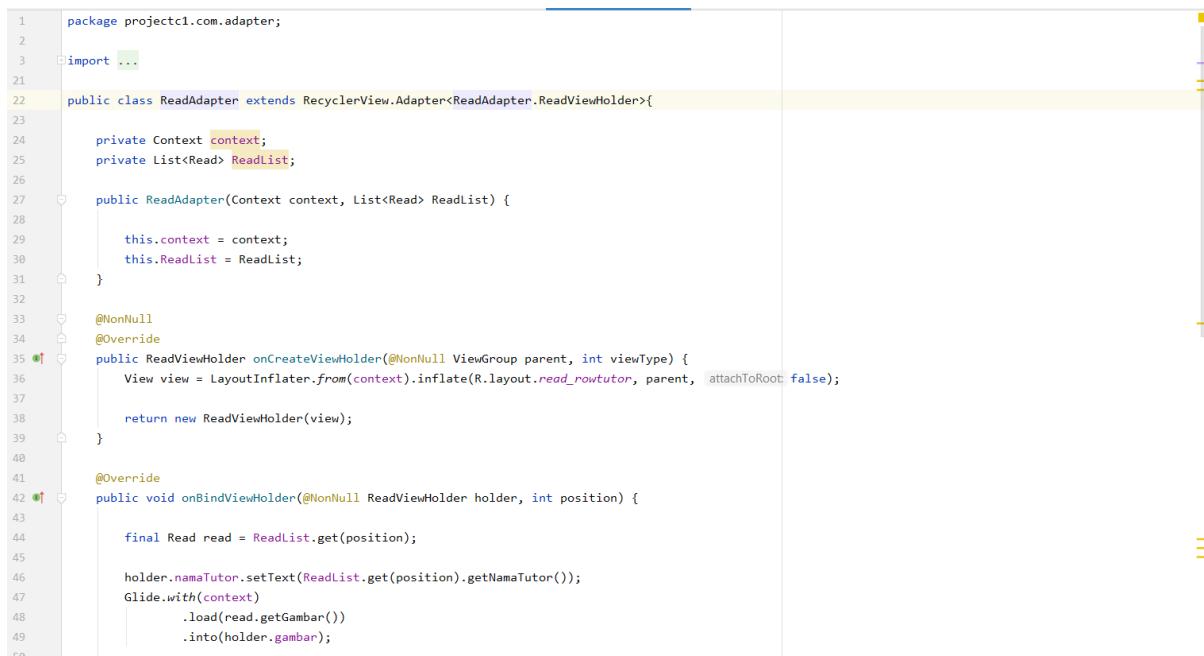
```

Gambar 2.48 Code Order Adapter

Class Order Adapter adalah *class* yang akan menampung dan mengirimkan variabel-variabel yang diterimanya dari *class Order*. Seluruh variabel pada *class* ini digunakan untuk

menampilkan pemesanan mahasiswa dalam bentuk daftar *invoice* yang dapat dilihat oleh admin pada class *Admin Order*. Code *RecyclerView* digunakan untuk menampilkan pemesanan yang terdiri dari tanggal pemesanan, kode kelas, nama mahasiswa, line id, jangka waktu, total, dan *notes* yang sebelumnya tampilannya sudah kita buat di *xml* lain. Data dari *RecyclerView* didapat dari *arraylist* dengan bantuan *onBindViewHolder* yang mengambil data dari class *Order* yang akan ditampilkan pada *RecyclerView*.

33. Read Adapter



```
1 package projectc1.com.adapter;
2
3 import ...
4
5 public class ReadAdapter extends RecyclerView.Adapter<ReadAdapter.ReadViewHolder>{
6
7     private Context context;
8     private List<Read> ReadList;
9
10    public ReadAdapter(Context context, List<Read> ReadList) {
11
12        this.context = context;
13        this.ReadList = ReadList;
14    }
15
16    @NonNull
17    @Override
18    public ReadViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
19        View view = LayoutInflater.from(context).inflate(R.layout.read_rowtutor, parent, attachToRoot: false);
20
21        return new ReadViewHolder(view);
22    }
23
24    @Override
25    public void onBindViewHolder(@NonNull ReadViewHolder holder, int position) {
26
27        final Read read = ReadList.get(position);
28
29        holder.namaTutor.setText(ReadList.get(position).getNamaTutor());
30        Glide.with(context)
31            .load(read.getGambar())
32            .into(holder.gambar);
33    }
34
35 }
```

```
51
52
53 holder.itemView.setOnClickListener(new View.OnClickListener() {
54     @Override
55     public void onClick(View v) {
56
57         Intent i = new Intent(context, UpdateAdmin.class);
58
59         i.putExtra( name: "kode_kelas", ReadList.get(position).getKode_kelas());
60         i.putExtra( name: "kode_namasubject", ReadList.get(position).getKode_namasubject());
61         i.putExtra( name: "kode_tutor", ReadList.get(position).getKode_tutor());
62         i.putExtra( name: "namatutor", ReadList.get(position).getNamaTutor());
63         i.putExtra( name: "jadwal", ReadList.get(position).getJadwal());
64         i.putExtra( name: "stok", ReadList.get(position).getStok());
65         i.putExtra( name: "harga", ReadList.get(position).getHarga());
66         i.putExtra( name: "gambar", ReadList.get(position).getGambar());
67         i.putExtra( name: "note", ReadList.get(position).getNote());
68
69         context.startActivity(i);
70     }
71 }
72
73 @Override
74 public int getItemCount() { return ReadList.size(); }
75
76 public class ReadViewHolder extends RecyclerView.ViewHolder {
77     private TextView namatutor;
78     private ImageView gambar;
79
80     public ReadViewHolder(@NotNull View view) {
81         super(view);
82         namatutor = view.findViewById(R.id.namatutor);
83         gambar = view.findViewById(R.id.gambar);
84     }
85 }
86
87 }
```

Gambar 2.49 Code Read Adapter

Class Read Adapter adalah *class* yang akan menampung dan mengirimkan variabel-variabel yang diterimanya dari *class Read*. Seluruh variabel pada *class* ini digunakan untuk menampilkan informasi mengenai daftar tutor yang dapat dilihat oleh admin pada *class Read Admin*. *Code recyclerView* digunakan untuk menampilkan foto dan nama tutor yang mengajar pada BIMSI yang ketika diklik pada foto tutor tersebut, admin akan diarahkan aplikasi ke dalam *class Update Admin*. Data dari *RecyclerView* didapat dengan menggunakan *arraylist* dan bantuan *onBindViewHolder*. Fungsi *onBindViewHolder* adalah mengambil dan menampilkan data mengenai nama dan foto tutor dari *class Read*.

34. Tutor Adapter

```
1 package projectc1.com.adapter;
2
3 import ...
21
22 public class TutorAdapter extends RecyclerView.Adapter<TutorAdapter.TutorViewHolder>{
23
24     private Context context;
25     private List<Tutor> tutorList;
26
27     public TutorAdapter(Context context, List<Tutor> tutorList) {
28
29         this.context = context;
30         this.tutorList = tutorList;
31     }
32
33     @NonNull
34     @Override
35     public TutorViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
36
37         View view = LayoutInflater.from(context).inflate(R.layout.tutor_row, parent, false);
38
39         return new TutorViewHolder(view);
40     }
41
42     @Override
43     public void onBindViewHolder(@NonNull TutorViewHolder holder, int position) {
44
45         final Tutor tutor = tutorList.get(position);
46
47         holder.namaTutor.setText(tutorList.get(position).getNamaTutor());
48         Glide.with(context)
49             .load(tutor.getGambar())
50             .into(holder.gambar);
51
52         holder.itemView.setOnClickListener(new View.OnClickListener() {
53             @Override
54             public void onClick(View v) {
55
56                 Intent i = new Intent(context, Detail.class);
57
58                 i.putExtra( name: "kode_kelas", tutorList.get(position).getKode_kelas());
59                 i.putExtra( name: "kode_namasubject", tutorList.get(position).getKode_namasubject());
60                 i.putExtra( name: "namatutor", tutorList.get(position).getNamaTutor());
61                 i.putExtra( name: "jadwal", tutorList.get(position).getJadwal());
62                 i.putExtra( name: "stok", tutorList.get(position).getStok());
63                 i.putExtra( name: "harga", tutorList.get(position).getHarga());
64                 i.putExtra( name: "gambar", tutorList.get(position).getGambar());
65                 i.putExtra( name: "note", tutorList.get(position).getNote());
66
67                 context.startActivity(i);
68             }
69         });
70     }
71
72     @Override
73     public int getItemCount() { return tutorList.size(); }
77
78     public static class TutorViewHolder extends RecyclerView.ViewHolder{
79
80         private TextView namaTutor;
81         private ImageView gambar;
82
83         public TutorViewHolder(@NonNull View view) {
84             super(view);
85             namaTutor = view.findViewById(R.id.namatutor);
86             gambar = view.findViewById(R.id.gambar);
87         }
88     }
89 }
90
91 }
```

Gambar 2.50 Code Tutor Adapter

Class Tutor Adapter adalah *class* yang akan menampung dan mengirimkan variabel-variabel yang diterimanya dari *class Tutor*. Seluruh variabel pada *class* ini digunakan untuk menampilkan informasi mengenai daftar tutor yang dapat dilihat oleh mahasiswa pada *class*

Reservation. Code *recyclerView* digunakan untuk menampilkan foto dan nama tutor yang mengajar pada BIMSI yang ketika diklik pada foto tutor tersebut, mahasiswa akan diarahkan aplikasi ke dalam *class Detail*. Data dari *RecyclerView* didapat dengan menggunakan *arraylist* dan bantuan *onBindViewHolder*. Fungsi *onBindViewHolder* adalah mengambil dan menampilkan data mengenai nama dan foto tutor dari *class Tutor*.

2.5.6 File PHP

1. conn.php

```
1 <?php
2
3 class conn
4 {
5     public $servername;
6     public $username;
7     public $password;
8     public $databasename;
9
10    public function __construct()
11    {
12
13        $this->servername = 'localhost';
14        $this->username = 'id17942833_user';
15        $this->password = 'BimbelJayaJaya2!';
16        $this->databasename = 'id17942833_bimbel';
17
18    }
19 }
20
21 ?>
```

Gambar 2.51 Conn.php

Pada file php ini, kami melakukan proses penghubungan aplikasi ke database BIMSI yang ada pada PHPMyAdmin yang diperuntukkan untuk proses jalannya *DataBase.php* dalam melakukan *sign up* dan *sign in*

2. db_connect.php

```
1 <?php
2 $conn = mysqli_connect("localhost","id17942833_user","BimbelJayaJaya2!"
3 , "id17942833_bimbel");
4
5 function konneksiDb(){
6     try{
7         $dsn = "mysql:host=localhost;dbname=id17942833_bimbel";
8         $pdo = new PDO($dsn, 'id17942833_user', 'BimbelJayaJaya2!');
9         return $pdo;
10    }catch(PDOException $e){
11        return $e;
12    }
13 }
14 ?>
```

Gambar 2.52 dbConnect.php

Sama seperti pada gambar sebelumnya, *code* diatas juga melakukan proses koneksi atau menghubungkan ke databaseBIMSI yaitu id17942833_bimbel.

3. DataBase.php

```
1 <?php
2 require "conn.php";
3
4 class DataBase
5 {
6     public $connect;
7     public $data;
8     private $sql;
9     protected $servername;
10    protected $username;
11    protected $password;
12    protected $databasename;
13
14    public function __construct()
15    {
16        $this->connect = null;
17        $this->data = null;
18        $this->sql = null;
19        $dbc = new conn();
20        $this->servername = $dbc->servername;
21        $this->username = $dbc->username;
22        $this->password = $dbc->password;
23        $this->databasename = $dbc->databasename;
24    }
25
26    function dbConnect()
27    {
28        $this->connect = mysqli_connect($this->servername, $this->username, $this->password, $this->databasename);
29        return $this->connect;
30    }
31
32    function prepareData($data)
33    {
34        return mysqli_real_escape_string($this->connect, stripslashes(htmlspecialchars($data)));
35    }
36
37
38    function signUp($user, $NamaFull, $nickname, $email, $password, $phone, $yoc)
39    {
40        $NamaFull = $this->prepareData($NamaFull);
41        $nickname = $this->prepareData($nickname);
42        $email = $this->prepareData($email);
43        $password = $this->prepareData($password);
44        $phone = $this->prepareData($phone);
45        $yoc = $this->prepareData($yoc);
46        $email = filter_var($email, FILTER_VALIDATE_EMAIL);
47        $password = password_hash($password, PASSWORD_BCRYPT);
48
49
50        $dbc = mysqli_connect("localhost", "id17942833_user", "BimbelJayaJay21", "id17942833_bimbel") or die ('Error connecting to MySQL server');
51        $check = mysqli_query($dbc, "select * from user where email = '$email'");
52        $checkrows = mysqli_num_rows($check);
53
54        if($checkrows > 0) {
55            echo "User already registered or email invalid, ";
56        }else {
57            $this->sql =
58                "INSERT INTO `user`(`user`, `Nickname`, `Email`, `Password`, `Phone`, `YOC`) VALUES ('" . $NamaFull . "','" . $nickname . "','" . $email . "','" . $password . "','" . $phone . "','" . $yoc . "')";
59            if (mysqli_query($this->connect, $this->sql)) {
60                return true;
61            } else return false;
62        }
63
64    }
65 }
66 >>
```

Gambar 2.53 Database.php

Code diatas merupakan file php dari database, dimana file ini terhubung dengan file *log in* dan *sign up*. Pada *function sign up*, terdapat beberapa data yang harus diisi oleh pengguna, yaitu nama lengkap, nama panggilan, email, password, nomor telepon, dan tahun angkatan. Khusus untuk variabel email, pengguna harus memasukkan email yang valid dengan syarat terdapat @gmail.com, code yang digunakan adalah *\$email = filter_var(\$email, FILTER_VALIDATE_EMAIL);* Selain itu pada variabel *password*, pengguna tidak perlu khawatir *password* akan terbaca karena telah dienkripsi sehingga tidak dapat terlihat, code yang digunakan adalah *\$password = password_hash(\$password, PASSWORD_BCRYPT);*

4. LogIn.php

```
1 <?php
2 header('Content-type:application/json;charset=utf-8');
3 require_once 'db_connect.php';
4
5 if (isset($_POST['email']) && isset($_POST['password'])) {
6     $email = $_POST['email'];
7     $password = $_POST['password'];
8
9     $sql = "SELECT * FROM user WHERE email = '$email'";
10
11    $response = mysqli_query($conn, $sql);
12
13    $result = array();
14    $result['SignIn'] = array();
15
16    if(mysqli_num_rows($response) === 1) {
17        $row = mysqli_fetch_assoc($response);
18
19
20    if (password_verify($password, $row['password'])){
21        $index['email'] = $row['email'];
22
23        array_push($result['SignIn'], $index);
24
25        $result['success'] = "1";
26        $result['message'] = "success";
27
28        echo json_encode($result);
29
30        mysqli_close($conn);
31    }else {
32        $result['success'] = "0";
33        $result['message'] = "error";
34        echo json_encode($result);
35
36        mysqli_close($conn);
37    }
38 }else {
39     $result['success'] = "0";
40     $result['message'] = "error";
41     echo json_encode($result);
42
43     mysqli_close($conn);
44 }
45
46 ?>
```

Gambar 2.54 LogIn.php

File php diatas merupakan file *LogIn.php* yang digunakan untuk memverifikasi data pengguna yang di input pada aplikasi. Jika data yang diinput terdapat pada database, maka pengguna akan diizinkan masuk untuk menggunakan aplikasi. Namun, apabila pengguna belum melakukan pendaftaran, maka pengguna dapat melakukan pendaftaran (*sign up*) yang akan terintegrasi dengan file *SignUp.php*.

5. SignUp.php

```
1 <?php
2 require "DataBase.php";
3 $db = new DataBase();
4
5 if (isset($_POST['NamaFull']) && isset($_POST['nickname']) && isset($_POST['email']) && isset($_POST['password']) &&
6   isset($_POST['phone']) && isset($_POST['yoc'])) {
7   if ($db->dbConnect()) {
8     if ($db->signUp("user", $_POST['NamaFull'], $_POST['nickname'], $_POST['email'], $_POST['password'],
9       $_POST['phone'], $_POST['yoc'])) {
10    echo "Sign Up Successfull";
11  } else echo "Sign Up Failed";
12 } else echo "Error: Database connection";
13 } else echo "All fields must be filled";
14 ?>
```

Gambar 2.55 SignUp.php

Pada code dalam file *SignUp* diatas terdapat code *isset(\$_POST)*; yang berfungsi untuk mengirimkan data-data *sign up* pengguna, seperti nama lengkap, nama panggilan, email, *password*, nomor telepon, dan tahun angkatan ke dalam database. Saat berhasil masuk ke database, akan muncul *echo* “*Sign Up Successful*”. Namun saat gagal, akan muncul *echo* “*Email Invalid, Sign Up Failed*” atau “*Error: Database connection*”. Semua *fields* yang ditentukan juga harus diisi. Jika pengguna tidak mengisi semua *fields* yang tersedia, akan muncul sebuah *echo* “*All fields must be filled*”.

6. getForum.php

```
1 <?php
2 header('Content-type:application/json;charset=utf-8');
3 include 'db_connect.php';
4
5 $q = $conn->query("SELECT * FROM forum");
6 $response = [];
7 if($q->num_rows> 0){
8   $response['data'] = [];
9   while($r = $q->fetch_array(MYSQLI_ASSOC)){
10     $response['data'][] = $r;
11   }
12   $response['success'] = 1;
13   $response['message'] = "Data mahasiswa berhasil dibaca";
14 } else {
15   $response['success'] = 0;
16   $response['message'] = 'Tidak ada data';
17 }
18 echo json_encode($response);
```

Gambar 2.56 getForum.php

File *getForum.php* ini digunakan untuk menampilkan data-data yang terdapat pada halaman forum, dengan code *SELECT * FROM forum*. Apabila aplikasi berhasil mengirim data

pengguna ke database akan muncul respon “Data mahasiswa berhasil dibaca”, jika aplikasi gagal mengirim data pengguna ke database akan muncul respon “Tidak ada data”.

7. getForumById.php

```
1 <?php
2 header('Content-type:application/json;charset=utf-8');
3 include 'db_connect.php';
4
5 $response=[];
6 if(isset($_POST['id'])){
7     | $id=$_POST['id'];
8
9     $q = $conn->query("SELECT * FROM forum WHERE id =  $id");
10    $response = [];
11    if($q->num_rows> 0){
12        $response['data'] = [];
13        while($r = $q->fetch_array(MYSQLI_ASSOC)){
14            | $response['data'][] = $r;
15        }
16        $response['success'] = 1;
17        $response['message'] = "Data mahasiswa berhasil dibaca";
18    } else {
19        | $response['success'] = 0;
20        | $response['message'] = "Data gagal Dibaca/ Tidak ada data";
21    }
22 }
23 echo json_encode($response);|
```

Gambar 2. 57 *getForumById.php*

File *getForumById.php* ini digunakan untuk mengambil id yang akan digunakan setiap mahasiswa saat mengirim pesan di forum. *Code* yang digunakan adalah *SELECT * FROM forum WHERE id = “\$id”*. Apabila aplikasi berhasil mengirim data pengguna ke database akan muncul respon “Data mahasiswa berhasil dibaca”, jika aplikasi gagal mengirim data pengguna ke database akan muncul respon “Data gagal dibaca/Tidak ada data”.

8. addForum.php

```
1 <?php
2 header('Content-type:application/json;charset=utf-8');
3 include 'db_connect.php';
4
5
6 $response = [];
7 if( isset($_POST['nama']) && isset($_POST['post'])){
8     $nama = $_POST['nama'];
9     $post = $_POST['post'];
10    $q = $conn->query("INSERT INTO forum(nama, post) VALUES ( '$nama', '$post')");
11
12    if($q){
13        $response['success'] = 1;
14        $response['message'] = "Data berhasil ditambah";
15    } else {
16        $response['success'] = 0;
17        $response['message'] = "Data gagal ditambah";
18    }
19 }
20 else {
21     $response['success'] = 1;
22     $response['message'] = "Data kosong";
23 }
24 echo json_encode($response);
25
```

Gambar 2.58 *addForum.php*

File *addForum.php* digunakan untuk menambahkan pesan atau post baru pada forum. Saat pengguna menambahkan pesan forum, php akan melakukan *ADD* dalam database dan akan muncul respon “Data berhasil ditambah” jika pesan berhasil ditambahkan. Sedangkan, jika pengguna gagal menambahkan pesan forum ke database, maka akan muncul respon “Data gagal ditambah”.

9. updateForum.php

```
1 <?php
2 header('Content-type:application/json;charset=utf-8');
3 include 'db_connect.php';
4
5 $response = [];
6 if(isset($_POST['id']) && isset($_POST['nama']) && isset($_POST['post'])){
7     $id = $_POST['id'];
8     $nama = $_POST['nama'];
9     $post = $_POST['post'];
10
11     $q = $conn->query("UPDATE forum SET nama='$nama', post='$post' WHERE id=' $id ''");
12     if($q){
13         $response['success'] = 1;
14         $response['message'] = "Data berhasil di update";
15     } else {
16         $response['success'] = 0;
17         $response['message'] = "Data gagal diupdate";
18     }
19
20 } else {
21     $response['success'] = -1;
22     $response['message'] = "Data kosong";
23 }
24 echo json_encode($response);
```

Gambar 2.59 *updateForum.php*

File *updateForum.php* digunakan untuk mengubah pesan atau post yang terdapat pada forum berdasarkan id. Jika pengguna berhasil mengubah pesan forum, php akan melakukan *UPDATE* dalam database dan akan muncul respon “Data berhasil di update”. Sedangkan, jika pengguna gagal mengubah pesan forum dalam database, akan muncul respon “Data gagal diupdate”, jika id tidak tersedia akan menampilkan “Data kosong”.

10. deleteForum.php

```
2 header('Content-type:application/json;charset=utf-8');
3 include 'db_connect.php';
4
5
6 $response=[];
7 if(isset($_POST['id'])){
8     $id=$_POST['id'];
9
10
11     if($conn->query("DELETE FROM forum WHERE id=' $id '")){
12         $response["success"]=1;
13         $response["message"]="Post berhasil di delete";
14
15     }else{
16         $response["success"]=0;
17         $response["message"]="Post gagal di delete";
18
19     }
20 }else{
21     $response["success"]=-1;
22     $response["message"]="Data kosong";
23
24 }
25 echo json_encode($response);
26 ?>
```

Gambar 2.60 *deleteForum.php*

File *deleteForum.php* ini digunakan untuk mengambil id yang akan dihapus oleh mahasiswa di forum. *Code* yang digunakan adalah *DELETE FROM* *forum WHERE id = “\$id”*. Apabila php berhasil menghapus data pengguna pada database, akan muncul respon “Post berhasil di delete”. Sedangkan, jika aplikasi gagal menghapus data pengguna pada database akan muncul respon “Post gagal di delete”, atau jika id tidak tersedia maka respon “Data kosong”

11. tutor.php

```
1 <?php
2
3     include ('db_connect.php');
4
5     $stmt = $conn->prepare("SELECT kode_kelas, kode_namasubject, kode_tutor, namaTutor, jadwal, stok, harga, gambar, note FROM tutor");
6
7     $stmt ->execute();
8     $stmt -> bind_result($kode_kelas, $kode_namasubject, $kode_tutor, $namaTutor, $jadwal, $stok, $harga, $gambar, $note);
9
10    $tutor = array();
11
12    while($stmt ->fetch()) {
13
14        $temp = array();
15
16        $temp['kode_kelas'] = $kode_kelas;
17        $temp['kode_namasubject'] = $kode_namasubject;
18        $temp['kode_tutor'] = $kode_tutor;
19        $temp['namaTutor'] = $namaTutor;
20        $temp['jadwal'] = $jadwal;
21        $temp['stok'] = $stok;
22        $temp['harga'] = $harga;
23        $temp['gambar'] = $gambar;
24        $temp['note'] = $note;
25
26        array_push($tutor,$temp);
27    }
28
29    echo json_encode($tutor);
30
31 ?>
```

Gambar 2.61 *Tutor.php*

File *tutor.php* digunakan untuk mengambil dan menampilkan data-data terkait tutor, seperti kode kelas, kode mata kuliah, nama tutor, jadwal, stok, harga, gambar, dan catatan.

12. pemesanan.php

```
1 <?php
2     header ('content-type:application/json;charset-utf-8');
3     include "db_connect.php";
4
5
6     if (isset($_POST['nama_mahasiswa']) &&
7         isset ($_POST['line']) &&
8         isset ($_POST['kode_kelas']) &&
9         isset ($_POST['harga']) &&
10        isset ($_POST['jangka_waktu']) &&
11        isset ($_POST['total']) &&
12        isset ($_POST[ 'notes']))
13
14    {
15        $nama_mahasiswa=$_POST['nama_mahasiswa'];
16        $line=$_POST['line'];
17        $kode_kelas=$_POST['kode_kelas'];
18        $harga=$_POST['harga'];
19        $jangka_waktu=$_POST['jangka_waktu'];
20        $total=$_POST['total'];
21        $notes=$_POST['notes'];
22
23        $q=mysqli_query ($conn, "INSERT INTO pemesanan(nama_mahasiswa, line, kode_kelas, harga, jangka_waktu, total, notes)
24                                VALUES ('$nama_mahasiswa','$line', '$kode_kelas','$harga', '$jangka_waktu', '$total', '$notes')");
25
26        $response=array();
27        if($q)
28        {
29            $response["success"]=1;
30            $response["message"]="Data berhasil disimpan";
31            echo json_encode($response);
32        }
33        else
34        {
35            $response["success"]=0;
36            $response["message"]="Data gagal disimpan";
37            echo json_encode($response);
38        }
39    }
40    else
41    {
42        $response["success"]=-1;
43        $response["message"]="Data kosong";
44        echo json_encode($response);
45    }
46}
47 ?>
```

Gambar 2.62 Pemesanan.php

File *pemesanan.php* digunakan untuk menambahkan data terkait pemesanan, seperti nama mahasiswa, id line, kode kelas, harga, jangka waktu, total dan catatan. Saat pengguna menambahkan data-data pemesanan, php akan melakukan *ADD* dalam database dan akan muncul respon “Data berhasil disimpan” jika data berhasil ditambahkan. Sedangkan, jika pengguna gagal menambahkan data pemesanan ke database, maka akan muncul respon “Data gagal disimpan”.

13. bacaAdmin.php

```
1 <?php
2     header ('content-type:application/json;charset-utf-8');
3     include "db_connect.php";
4
5     $stmt = $conn->prepare("SELECT kode_kelas, kode_namasubject, kode_tutor, namaTutor , jadwal, stok, harga, gambar, note FROM tutor");
6
7     $stmt ->execute();
8     $stmt -> bind_result($kode_kelas, $kode_namasubject, $kode_tutor, $namaTutor , $jadwal, $stok, $harga, $gambar, $note);
9
10    $tutor = array();
11
12    while($stmt ->fetch()) {
13
14        $temp = array();
15
16        $temp['kode_kelas'] = $kode_kelas;
17        $temp['kode_namasubject'] = $kode_namasubject;
18        $temp['kode_tutor'] = $kode_tutor;
19        $temp['namaTutor'] = $namaTutor;
20        $temp['jadwal'] = $jadwal;
21        $temp['stok'] = $stok;
22        $temp['harga'] = $harga;
23        $temp['gambar'] = $gambar;
24        $temp['note'] = $note;
25
26        array_push($tutor,$temp);
27    }
28
29    echo json_encode($tutor);
?>
```

Gambar 2.63 *bacaAdmin.php*

File *bacaAdmin.php* digunakan admin untuk mengambil dan menampilkan data-data terkait tutor, seperti kode kelas, kode mata kuliah, nama tutor, jadwal, stok, harga, gambar, dan catatan.

14. updateTutorStok.php

```
1 <?php
2     header ('content-type:application/json;charset-utf-8');
3     include "db_connect.php";
4
5
6     if (isset($_POST['kode_kelas']) &&
7         isset ($_POST['stok']) &&
8         isset ($_POST['harga']))
9     {
10         $kode_kelas=$_POST['kode_kelas'];
11         $stok=$_POST['stok'];
12         $harga=$_POST['harga'];
13
14         $q=mysqli_query($conn, "UPDATE tutor SET stok='$stok', harga='$harga' WHERE kode_kelas= '$kode_kelas'");
15         $response=array();
16         if($q)
17         {
18             $response["success"]=1;
19             $response["message"]="Data berhasil diupdate";
20             echo json_encode($response);
21         }
22         else
23         {
24             $response["success"]=0;
25             $response["message"]="Data gagal diupdate";
26             echo json_encode($response);
27         }
28     }
29     else
30     {
31         $response["success"]=-1;
32         $response["message"]="Data kosong";
33         echo json_encode($response);
34     }
35
```

Gambar 2.64 *updateTutorStok.php*

File *updateTutorStok.php* digunakan untuk mengubah stok dan harga yang terdapat pada tabel tutor berdasarkan kode_kelas. Jika admin berhasil mengubah stok atau harga pada

aplikasi, php akan melakukan *UPDATE* dalam database dan akan muncul respon “Data berhasil di update”. Sedangkan, jika pengguna gagal mengubah stok dan harga dalam database, akan muncul respon “Data gagal diupdate”.

15. lihatOrder.php

```
1 <?php
2     header('content-type:application/json;charset=utf-8');
3     include "db_connect.php";
4
5     $stmt = $conn->prepare("SELECT tanggal_pemesanan, nama_mahasiswa, line, kode_kelas, harga, jangka_waktu, total, notes FROM pemesanan");
6
7     $stmt ->execute();
8     $stmt ->bind_result($tanggal_pemesanan, $nama_mahasiswa, $line, $kode_kelas, $harga, $jangka_waktu, $total, $notes);
9
10    $pemesanan = array();
11
12    while($stmt ->fetch()) {
13        $temp = array();
14
15        $temp['tanggal_pemesanan']=$tanggal_pemesanan;
16        $temp['nama_mahasiswa']=$nama_mahasiswa;
17        $temp['line']=$line;
18        $temp['kode_kelas']=$kode_kelas;
19        $temp['harga']=$harga;
20        $temp['jangka_waktu']=$jangka_waktu;
21        $temp['total']=$total;
22        $temp['notes']=$notes;
23        array_push($pemesanan,$temp);
24
25    }
26    echo json_encode($pemesanan);
27 ?>
```

Gambar 2.65 *lihatOrder.php*

File *lihatOrder.php* digunakan admin untuk mengambil dan menampilkan data-data terkait pemesanan, seperti tanggal pemesanan, nama mahasiswa, line, kode kelas, harga, jangka waktu, total dan *notes* atau catatan.

16. deleteTutor.php

```
1 <?php
2     header ('content-type:application/json;charset-utf-8');
3     include "db_connect.php";
4
5
6 if (isset($_POST['kode_kelas'])) {
7     $kode_kelas=$_POST['kode_kelas'];
8
9     $q=mysqli_query($conn, "DELETE FROM tutor WHERE kode_kelas= '$kode_kelas'");
10    $response=array();
11
12 if($q){
13     $response["success"]=1;
14     $response["message"]="Data berhasil didelete";
15     echo json_encode($response);
16 }else{
17     $response["success"]=0;
18     $response["message"]="Data gagal didelete";
19     echo json_encode($response);
20 }
21 }else{
22     $response["success"]=-1;
23     $response["message"]="Data kosong";
24     echo json_encode($response);
25 }
26 ?>
```

Gambar 2.66 *deleteTutor.php*

File *deleteTutor.php* ini digunakan untuk mengambil kode kelas yang akan dihapus oleh admin di menu admin. *Code* yang digunakan adalah *DELETE FROM forum WHERE KODE_KELAS= “\$kode_kelas”*. Apabila *php* berhasil menghapus data pada database akan muncul *response* “Data berhasil didelete”, jika aplikasi gagal menghapus data pada database akan muncul *response* “Data gagal didelete” atau jika tidak ada data muncul “Data kosong”.

BAB III

PENUTUP

3.1 Kesimpulan

BIMSI merupakan aplikasi mengenai bimbingan belajar yang dikelola oleh Himpunan Mahasiswa Sistem Informasi (HIMSI) Universitas Multimedia Nusantara. Pembuatan aplikasi BIMSI bertujuan untuk mempermudah proses dan alur registrasi pada sistem bimbingan belajar HIMSI. Sasaran utama dari aplikasi BIMSI adalah para mahasiswa yang membutuhkan bimbingan belajar untuk memperoleh penjelasan lebih mengenai mata kuliah yang kurang dimengerti atau ingin didalami. Para mahasiswa yang ingin mendaftar bimbingan belajar dapat langsung mengunduh aplikasi BIMSI dan memesan kuota kelas mata kuliah yang diinginkan cukup dengan sentuhan jari. Tak hanya itu, program BIMSI juga dapat diunduh oleh para admin yang mengelola program bimbingan belajar HIMSI untuk memperbarui data tutor setiap bulannya maupun melihat daftar data mahasiswa yang mendaftar melalui aplikasi.

Aplikasi BIMSI memiliki dua fungsi utama yakni untuk menampilkan informasi seputar program bimbingan belajar yang disediakan HIMSI dan memudahkan proses pendaftaran bimbingan belajar HIMSI bagi mahasiswa. Dalam menunjang dan melengkapi fungsi aplikasinya, BIMSI memiliki empat menu utama yaitu *Home*, *Forum*, *Information*, dan *Reservation*. Untuk mengawali alur aplikasi BIMSI, terdapat fitur *Log In* untuk masuk ke dalam aplikasi bagi pengguna yang sudah mendaftar dan *Sign Up* bagi pengguna baru yang ingin mendaftarkan akunnya. Dalam menu *Home*, pengguna dapat melihat informasi dan jadwal bimbingan belajar dari setiap mata kuliah yang disediakan, mengakses sub menu *Student Note*, serta mengakses laman e-learning UMN maupun myUMN. Sub menu *Student Note* berfungsi untuk menambahkan, memperbarui, dan menghapus catatan atau pengingat yang disimpan ke database lokal perangkat pengguna menggunakan SQLite database. Dalam menu *Forum*, pengguna dapat menambahkan, memperbarui, dan menghapus posting forum dalam berinteraksi dengan pengguna lainnya. Data dari menu *Forum* akan dikirim dan disimpan ke dalam database *cloud* atau *online* dengan bantuan database PHPMyAdmin dan server *webhosting*. Pengguna dapat melihat informasi seputar HIMSI UMN dalam menu *Information* yang ditampilkan dalam bentuk *webview* berupa akun Instagram, Youtube, Line, Website resmi HIMSI, serta lokasi gedung UMN yang ditampilkan dengan fitur API Google Maps. Terakhir, pengguna dapat melihat informasi dan data mengenai tutor dan kuota kelas bimbingan belajar HIMSI pada menu *Reservation* serta melakukan pemesanan kelas

bimbingan belajar. Selain itu, terdapat fitur tambahan berupa *button Log Out* untuk keluar dari aplikasi yang terletak pada pojok kiri atas menu *Home*.

Selain dua fungsi utama yang disebutkan sebelumnya, BIMSI juga memiliki fitur tambahan untuk memudahkan admin dalam melakukan pemantauan terhadap tutor, kuota kelas, dan data registrasi mahasiswa. Admin dapat masuk ke dalam aplikasi BIMSI menggunakan *e-mail* dan *password* khusus yang telah diatur sebelumnya untuk mengakses fitur admin dari aplikasi BIMSI. Fitur yang ditawarkan dalam BIMSI versi admin adalah penghapusan tutor aktif, pembaruan kuota dan harga kelas bimbingan belajar, serta pendataan *invoice* dari setiap registrasi yang dilakukan mahasiswa. Pembuatan dan pengembangan aplikasi seluler BIMSI seluruhnya dilakukan menggunakan *software* Android Studio.

3.2 Peran Anggota

Berikut merupakan peran dari masing-masing anggota Kelompok 1 IS534-C dalam pembuatan aplikasi BIMSI:

1. Amelia Rahmanita (00000044296)

Bertugas dalam pembuatan dan penyusunan laporan, pembuatan *layout* aplikasi, penjelasan proses bisnis, pembuatan dan penjelasan *coding*

2. Bill Kiki (00000043854)

Bertugas dalam pembuatan dan penyusunan laporan, pembuatan *layout* aplikasi, pembuatan *flowchart*, penjelasan *coding*, pembuatan database

3. Meilani Yaputri (00000045878)

Bertugas dalam pembuatan dan penyusunan laporan, mencari dan mengumpulkan data organisasi, pembuatan dan penjelasan *layout* aplikasi, penjelasan *coding*, pembuatan ERD

4. Mettadewi (00000045973)

Bertugas dalam pembuatan dan penyusunan laporan, mencari dan mengumpulkan data organisasi, pembuatan dan penjelasan *layout* aplikasi, penjelasan *coding*, penjelasan ERD

5. Reynardthan Handoko (00000044437)

Bertugas dalam pembuatan dan penyusunan laporan, pembuatan *database* aplikasi, pembuatan dan penjelasan *coding* aplikasi, pembuatan file .php

3.3 Link Video Presentasi Project

Berikut merupakan *link* dari video presentasi *project Mobile Application Development* Kelompok 1 IS534-C terkait aplikasi bimbingan belajar BIMSI.

<https://drive.google.com/file/d/1H5McU3VA45qy2Q2bsZtIsrbCVfafMmBR/view>