

**LAPORAN DOKUMENTASI PROJECT AKHIR  
MOBILE APPLICATION DEVELOPMENT  
*KOSTHOST***



Disusun oleh:  
Alexander Bryan Wiratman (00000036371)  
Fiona Tanadi (00000034054)  
Juan Richard Harmel (00000035131)  
Marco Viriyatama (00000035986)  
Monica Cornelia (00000037164)

**PROGRAM STUDI SISTEM INFORMASI  
FAKULTAS TEKNIK DAN INFORMATIKA  
UNIVERSITAS MULTIMEDIA NUSANTARA  
TANGERANG  
2021**

## **DAFTAR ISI**

DAFTAR ISI.....	i
DAFTAR GAMBAR .....	ii
BAB I PENDAHULUAN.....	1
1.1    Latar Belakang .....	1
1.2    Tujuan Aplikasi .....	1
1.2.1    Untuk Pencari Kost .....	1
1.2.2    Untuk Pemilik Kost.....	2
1.3    Tujuan Aplikasi .....	2
1.3.1    Untuk Pencari Kost .....	2
1.3.2    Untuk Pemilik Kost.....	2
BAB II PROSES BISNIS DAN ERD DIAGRAM.....	3
2.1    Proses Bisnis .....	3
2.2    ERD Diagram.....	4
BAB III LAYOUT APLIKASI.....	5
3.1    Layout.....	5
3.2    Layout Pemilik Kost.....	8
3.3    Layout Pencari Kost .....	15
BAB IV CODING PADA APLIKASI .....	19
4.1    Coding PHP.....	19
4.2    Coding XML .....	38
4.3    Coding Java.....	107
4.4    Coding Android Manifest .....	172
4.5    Coding Build Gradle (Module: Kost_Host.app) .....	173
BAB V PERANAN.....	174

## DAFTAR GAMBAR

Diagram 1. ERD Diagram.....	4
Layout 2. Layout Log In .....	5
Layout 3. Layout Sign Up.....	6
Layout 4. Layout Sign Up Pemilik .....	6
Layout 5. Layout Sign Up Pencari.....	7
Layout 6. Layout Log In Pemilik.....	7
Layout 7. Layout Log In Pencari .....	8
Layout 8. Layout Dashboard Pemilik Kost.....	8
Layout 9. Layout Menu My Kost .....	9
Layout 10. Layout Tambah Kost .....	9
Layout 11. Layout Edit Kost.....	10
Layout 12. Layout Menu Booked .....	10
Layout 13. Layout Konfirmasi Penyewaan .....	11
Layout 14. Layout Booked Penyewaan .....	11
Layout 15. Layout Menu Penjaga.....	12
Layout 16. Layout Tambah Penjaga .....	12
Layout 17. Layout Dialog Edit dan Delete .....	13
Layout 19. Layout FAQ Pemilik .....	14
Layout 20. Layout Profil Pemilik .....	14
Layout 21. Layout Edit Profil Pemilik.....	15
Layout 22. Layout Dashboard Pencari Kost .....	15
Layout 23. Layout Search Kost Pencari .....	16
Layout 24. Layout Detail Kost.....	16
Layout 25. Layout Booked Kost Pencari .....	17
Layout 26. Layout FAQ Pencari .....	17
Layout 27. Layout Profil Pencari.....	18
Layout 28. Layout Edit Profil Pencari .....	18
Code 1. Code PHP Connect Database .....	19
Code 2. Code PHP Insert Data Kost .....	19

Code 3. Code PHP Insert Data Akun Pemilik .....	20
Code 4. Code PHP Insert Data Akun Pencari.....	21
Code 5. Code PHP Insert Penyewaan oleh Pencari .....	22
Code 6. Code PHP Edit untuk Mengurangi Jumlah Kamar Pemilik .....	23
Code 7. Code PHP Edit Update Kost.....	24
Code 8. Code PHP Edit Profil Pemilik .....	25
Code 9. Code PHP Edit Profil Pencari.....	26
Code 10. Code PHP Menambahkan Kamar Pemilik Kost Ketika Penyewaan dihapus .....	27
Code 11. Code PHP Edit Status Penyewaan Kost .....	28
Code 12. Code PHP Delete Kost sebagai Pemilik.....	28
Code 13. Code PHP Delete Booked Kost oleh Pemilik.....	29
Code 14. Code PHP Login sebagai Pemilik Kost.....	30
Code 15. Code PHP Login sebagai Pencari Kost .....	31
Code 16. Code PHP Select Data Booked Kost Pemilik Kost.....	32
Code 17. Code PHP Select Data Booked Kost Pencari Kost .....	33
Code 18. Code PHP Select Data Kost berdasarkan Kota .....	34
Code 19. Code PHP Select Data Kost Pemilik .....	35
Code 20. Code PHP Select Data Profil Pemilik Kost.....	36
Code 21. Code PHP Select Data Profil Pencari Kost .....	37
Code 22. Code XML Main .....	38
Code 23. Code XML Log In sebagai Pemilik Kost atau Pencari Kost .....	39
Code 24. Code XML Sign Up sebagai Pemilik Kost atau Pencari Kost .....	41
Code 25. Code XML Sign Up Pemilik Kost.....	43
Code 26. Code XML Sign Up Pencari Kost .....	46
Code 27. Code XML Log In sebagai Pemilik Kost .....	47
Code 28. Code XML Log In sebagai Pencari Kost .....	49
Code 29. Code XML Dashboard Pemilik Kost .....	53
Code 30. Code XML Kost Pemilik.....	54
Code 31. Code XML Tambah Kost.....	57
Code 32. Code XML Edit Kost.....	61
Code 33. Code XML List Kost.....	63

Code 34. Code XML Booked Kost Pemilik Kost.....	64
Code 35. Code XML Detail Booked Kost .....	66
Code 36. Code XML List Booked Pemilik.....	67
Code 37. Code XML Penjaga .....	68
Code 38. Code XML Tambah Penjaga .....	70
Code 39. Code XML Dialog Edit/Delete.....	71
Code 40. Code XML Edit Penjaga .....	73
Code 41. Code XML FAQ Pemilik .....	76
Code 42. Code XML Profil Pemilik .....	79
Code 43. Code XML Edit Profile Pemilik.....	83
Code 44. Code XML Dashboard Pencari Kost.....	86
Code 45. Code XML Menu Search.....	87
Code 46. Code XML Book Kost.....	91
Code 47. Code XML Menu Booked Kost Pencari Kost.....	92
Code 48. Code XML List Booked Pencari Kost.....	93
Code 49. Code XML FAQ Pencari.....	95
Code 50. Code XML Profil Pencari.....	99
Code 51. Code Edit Profile Pencari .....	104
Code 52. Code XML Toolbar .....	105
Code 53. Code XML Edit Toolbar .....	106
Code 54. Code Java Main Activity .....	107
Code 55. Code Java Login .....	108
Code 56. Code Java Sign Up .....	109
Code 57. Code Java Sign Up Pemilik.....	111
Code 58. Code Java Sign Up Pencari .....	112
Code 59. Code Java Login Pemilik.....	114
Code 60. Code Java Login Pencari .....	116
Code 61. Code Java Dashboard Pemilik Kost .....	118
Code 62. Code Java Kost Pemilik.....	122
Code 63. Code Java Tambah Kost.....	126
Code 64. Code Java Edit Kost .....	130

Code 65. Code Java Booked Kost Pemilik Kost.....	134
Code 66. Code Detail Booked Kost Pemilik .....	137
Code 67. Code Java Class Penjaga Kost.....	139
Code 68. Code Java Class MyDBHandler.....	141
Code 69. Code Java Penjaga.....	144
Code 70. Code Java Tambah Penjaga.....	146
Code 71. Code Java Edit Penjaga .....	147
Code 72. Code Java FAQ Pemilik .....	148
Code 73. Code Java Profile Pemilik .....	150
Code 74. Code Java Edit Profile Pemilik.....	154
Code 75. Code Java Dashboard Pencari Kost.....	156
Code 76. Code Java Menu Search .....	158
Code 77. Code Java Book Kost .....	160
Code 78. Code Java Booked Pencari .....	163
Code 79. Code Java FAQ Pencari.....	164
Code 80. Code Java Profile Pencari.....	167
Code 81. Code Java Edit Profile Pencari .....	170
Code 82. Code Android Manifest .....	172
Code 83. Code Build Gradle (Module:Kost_Host.app).....	173

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Persaingan dalam dunia kerja lebih berat dibandingkan sebelumnya, sehingga banyak orang yang memilih untuk menempuh pendidikan setinggi-tingginya, dengan harapan dapat memperoleh ilmu yang nantinya dapat membantu mereka dalam bersaing di dunia kerja. Dalam menempuh pendidikan tinggi, tidak jarang orang memilih untuk meninggalkan daerah asalnya lalu memilih untuk menuntut ilmu di kota-kota besar di Indonesia. Bagi mereka yang memilih meninggalkan kota asalnya, pasti membutuhkan tempat tinggal sementara, seperti kost yang berada di sekitar kampus, selama mereka menempuh pendidikan.

Kost dapat dikatakan adalah sebutan untuk kumpulan kamar-kamar yang disewakan secara bulanan atau tahunan. Kost biasanya berada di dekat lingkungan sekolah, universitas, dan perkantoran. Kost banyak dimanfaatkan oleh para perantau yang bekerja di wilayah lain untuk tinggal sementara karena jauh dari tempat tinggal aslinya. Kost dapat berisi berbagai perabotan seperti kasur, lemari, kursi, dan meja. Beberapa kost menyediakan fasilitas tambahan lain seperti pendingin ruangan, laundry, kamar mandi dalam, jaringan internet, dan sebagainya.

Pada umumnya, para pencari kost mengalami kendala dalam mencari kost, seperti perlu survey ke berbagai tempat untuk menemukan kost yang sesuai dengan kebutuhan. Para pencari kost juga harus “rebutan” untuk mendapatkan kamar kost yang diinginkan. Untuk mengetahui harga, mereka juga harus menanyakan ke satu tempat dan tempat lainnya. Oleh karena itu, kelompok kami membuat aplikasi untuk memudahkan para pencari kost agar dapat menemukan kost yang sesuai dengan harga dan lokasi yang mereka inginkan.

#### **1.2 Tujuan Aplikasi**

##### **1.2.1 Untuk Pencari Kost**

- Memudahkan para pencari kost untuk menemukan kost yang sesuai kebutuhan.
- Memudahkan para pencari kost untuk memesan kamar.
- Memudahkan para pencari kost untuk membandingkan beberapa kost sebelum mengambil keputusan untuk menyewa.
- Memudahkan para pencari kost untuk menghubungi pemilik kost.

### **1.2.2 Untuk Pemilik Kost**

- Memudahkan pemilik kost untuk mengetahui status kamar kost yang dimiliki.
- Memudahkan pemilik kost untuk mempromosikan kamar kostnya.
- Memudahkan pemilik kost untuk memberikan informasi kost kepada para pencari kost.

## **1.3 Tujuan Aplikasi**

### **1.3.1 Untuk Pencari Kost**

- Sign Up
- Sign In
- Cari (filter berdasarkan kota kost)
- Kost yang pernah di book(melihat kos yang sedang atau pernah disewa; isi dari fitur ini adalah nama kost dan alamat kost)
- FAQ
- Profil pengguna (mengubah identitas dan foto profil)

### **1.3.2 Untuk Pemilik Kost**

- Sign Up
- Sign In
- Kost saya (melihat unit kost yang sudah dimasukkan)
  - o Tambah kost (Nama, Alamat, Kota, Jenis, Foto, Jumlah Kamar, Fasilitas, dan Harga)
  - o Edit kost
  - o Hapus kost
- Kost yang di book oleh pencari (melihat kost yang di book oleh pencari kost yang nantinya dapat dikonfirmasi atau di hapus)
- FAQ
- Profil pengguna (mengubah identitas dan foto profil)

## **BAB II**

### **PROSES BISNIS DAN ERD DIAGRAM**

#### **2.1 Proses Bisnis**

Kost Host merupakan aplikasi yang berperan sebagai penghubung antara pemilik kost dengan pencari kost. Sehingga dapat dikatakan, terdapat 2 peran dalam kost host yaitu pemilik kost dan pencari kost. Awalnya, baik pemilik kost maupun pencari kost harus mendaftarkan diri untuk dapat mengakses layanan pada aplikasi dengan mengisi beberapa data. Setelah terdaftar, baik pemilik kost maupun pencari kost dapat log in ke dalam aplikasi dengan akun yang sudah didaftarkan oleh pengguna.

Pemilik kost dapat menambahkan kost yang dimilikinya dengan menginput data-data kost seperti nama kost, alamat kost, jenis kost, harga kost, foto kost, dan sebagainya. Kost yang didaftarkan oleh pemilik kost akan masuk ke database yang nantinya dapat diakses informasinya oleh pencari kost. Apabila terdapat kesalahan input yang dilakukan, pemilik kost juga dapat mengubah data kost, dan apabila pemilik kost sudah tidak ingin menyewakan unitnya lagi, pemilik kost dapat menghapus data kost. Selanjutnya, pemilik kost dapat memeriksa penyewaan yang dilakukan oleh pencari kost pada unit kost yang telah didaftarkan oleh pemilik kost. Pemilik dapat menghubungi kontak pencari kost yang ditampilkan pada menu booked. Apabila, pencari sudah mengkonfirmasi penyewaan kost, pemilik kost dapat mengkonfirmasi penyewaan pada menu booked.

Pencari kost dapat mencari kost yang diinginkannya menggunakan menu search dengan memasukkan kata kunci berupa kota kost. Dari hasil pencarian kost, pencari dapat melihat detail informasi kost yang dipilih. Apabila kost yang dilihat sudah sesuai dengan kebutuhan dan keinginan pencari kost, pencari dapat melakukan booking terhadap kamar kost tersebut. Kost yang sudah dibook oleh pencari akan tersimpan pada menu booked pencari sebagai history penyewaan dan tersimpan pada menu booked pemilik untuk nantinya dihubungi oleh pemilik.

## 2.2 ERD Diagram

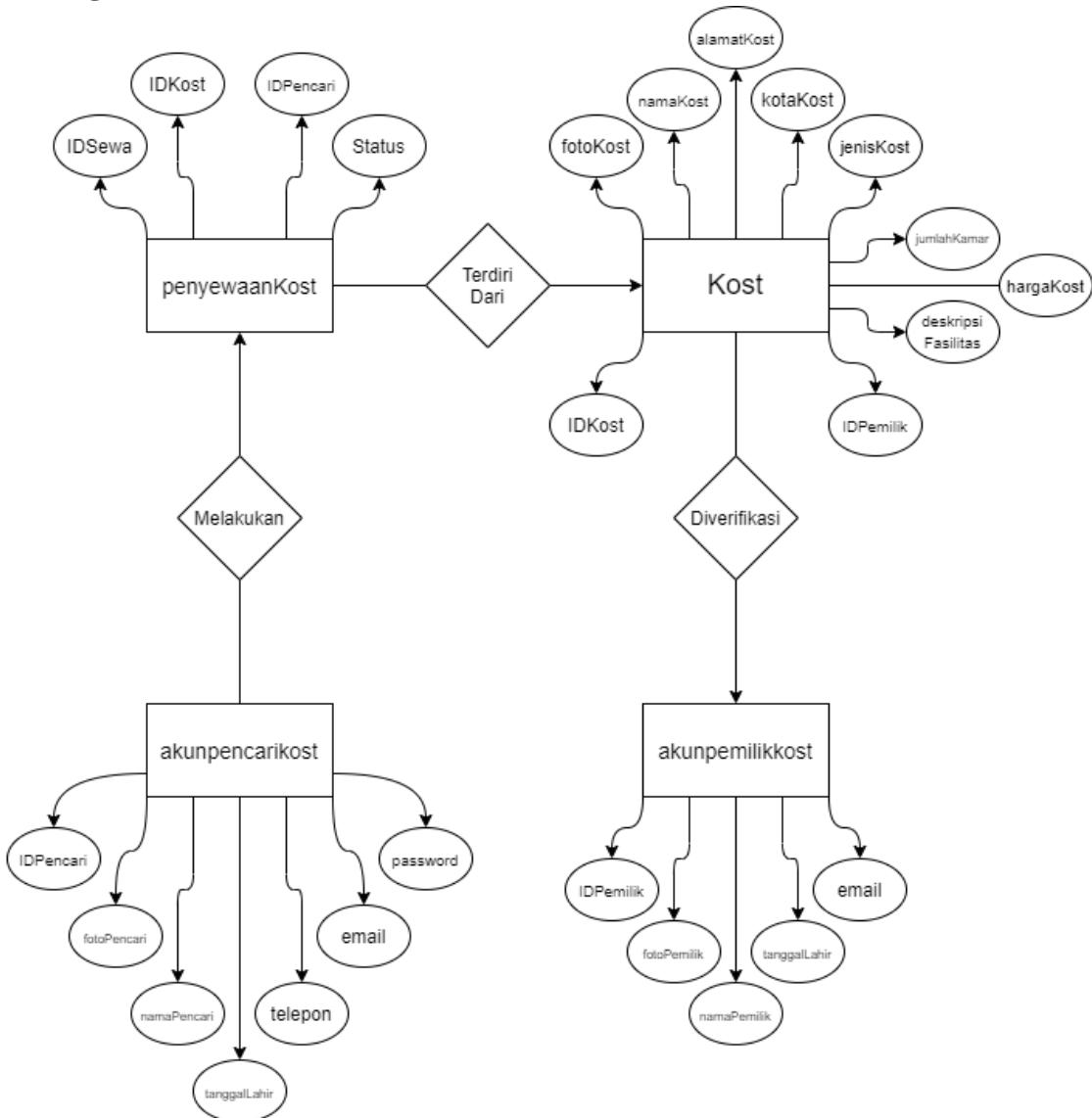


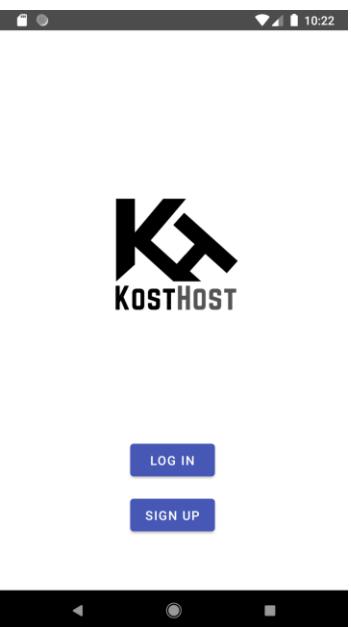
Diagram 1. ERD Diagram

## BAB III

### LAYOUT APLIKASI

#### 3.1 Layout

##### Layout Log In/Sign Up



The screenshot shows the KOSTHOST mobile application's login/signup screen. At the top, there is a large logo consisting of a stylized 'K' and 'H' followed by the word 'KOSTHOST'. Below the logo are two blue rectangular buttons with white text: 'LOG IN' and 'SIGN UP'. At the bottom of the screen is a black navigation bar with three white icons: a triangle pointing left, a circle, and a square.

Tampilan Log In/Sign Up seperti ditunjukkan pada gambar di samping ini adalah halaman Log In/Sign Up yang akan muncul ketika user menekan aplikasi kami. Terdapat dua pilihan yaitu Button Log In dan Button Sign Up.

Layout 1. Layout Log In/Sign Up

##### Layout Log In Pemilik/Pencari



The screenshot shows the KOSTHOST mobile application's log in screen for owners/advertisers. It features the same logo at the top. Below it, there is a text input field labeled 'Masuk Sebagai:' followed by two blue buttons: 'PENCARI KOST' and 'PEMILIK KOST'. At the bottom is a black navigation bar with three white icons: a triangle pointing left, a circle, and a square.

Tampilan Log In Pemilik/Pencari seperti ditunjukkan pada gambar di samping ini adalah halaman Log In yang akan muncul ketika user menekan button Log In. Pada halaman ini user dapat memilih akan log in sebagai Pencari Kost maupun Pemilik Kost.

Layout 2. Layout Log In

### Layout Sign Up Pemilik/Pencari

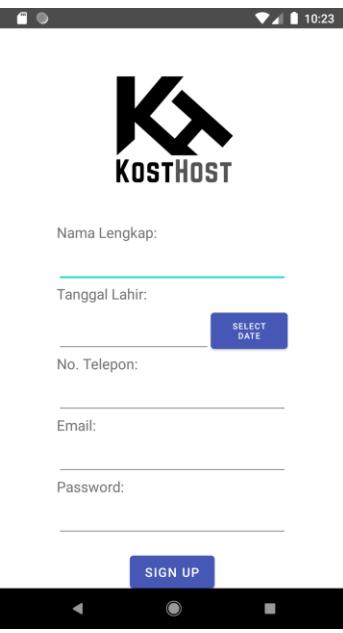


The screenshot shows the KostHost mobile application interface. At the top, there is a navigation bar with icons for back, forward, and search. Below the navigation bar is the KostHost logo, which consists of a stylized 'K' icon followed by the text 'KOSTHOST'. A large, dark rectangular area below the logo contains the text 'Daftar Sebagai:' in white. Underneath this text are two blue rectangular buttons with white text: 'PENCARI KOST' and 'PEMILIK KOST'. At the bottom of the screen is a black navigation bar with three small white icons: a left arrow, a circle, and a square.

**Layout 3. Layout Sign Up**

Tampilan Sign Up Pemilik/Pencari seperti ditunjukkan pada gambar di samping ini adalah halaman Sign Up yang akan muncul ketika user menekan button Sign Up. Pada halaman ini user dapat memilih akan sign up sebagai Pencari Kost maupun Pemilik Kost.

### Layout Sign Up Pemilik

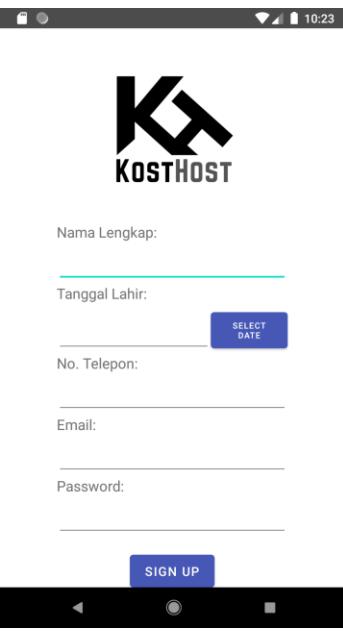


The screenshot shows the KostHost mobile application interface. At the top, there is a navigation bar with icons for back, forward, and search. Below the navigation bar is the KostHost logo. The main form area contains several input fields: 'Nama Lengkap:' with a text input field, 'Tanggal Lahir:' with a text input field and a blue 'SELECT DATE' button, 'No. Telepon:' with a text input field, 'Email:' with a text input field, and 'Password:' with a text input field. At the bottom of the form is a blue 'SIGN UP' button. Below the form is a black navigation bar with three small white icons: a left arrow, a circle, and a square.

**Layout 4. Layout Sign Up Pemilik**

Tampilan Sign Up Pemilik Kost seperti ditunjukkan pada gambar di samping ini adalah halaman Sign Up yang akan muncul ketika user menekan button Sign Up sebagai Pemilik Kost. User akan diminta untuk mengisi Nama Lengkap, Tanggal Lahir dengan button Select Date yang bertujuan memudahkan user mencari tanggal lahir user, Nomor Telepon, Email, serta Password.

### Layout Sign Up Pencari

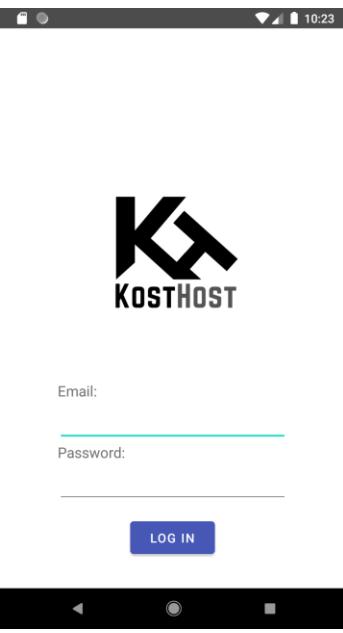


The screenshot shows the 'Sign Up' screen for users searching for accommodations. At the top is the KostHost logo. Below it are five input fields: 'Nama Lengkap', 'Tanggal Lahir' (with a 'SELECT DATE' button), 'No. Telepon', 'Email', and 'Password'. At the bottom is a blue 'SIGN UP' button.

**Layout 5. Layout Sign Up Pencari**

Tampilan Sign Up Pencari Kost seperti ditunjukkan pada gambar di samping ini adalah halaman Sign Up yang akan muncul ketika user menekan button Sign Up sebagai Pencari Kost. User akan diminta untuk mengisi Nama Lengkap, Tanggal Lahir dengan button Select Date yang bertujuan memudahkan user mencari tanggal lahir user, Nomor Telepon, Email, serta Password.

### Layout Log In Pemilik

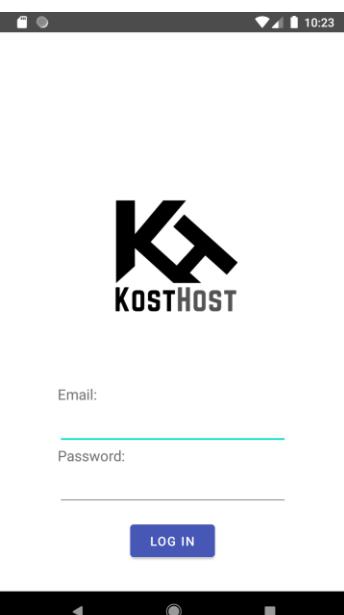


The screenshot shows the 'Log In' screen for property owners. At the top is the KostHost logo. Below it are two input fields: 'Email' and 'Password'. At the bottom is a blue 'LOG IN' button.

**Layout 6. Layout Log In Pemilik**

Tampilan Log In Pemilik Kost seperti ditunjukkan pada gambar di samping ini adalah halaman Log In yang akan muncul ketika user menekan button Log In sebagai Pemilik Kost. User akan diminta untuk mengisi Email dan Password yang sudah didaftarkan sebelumnya.

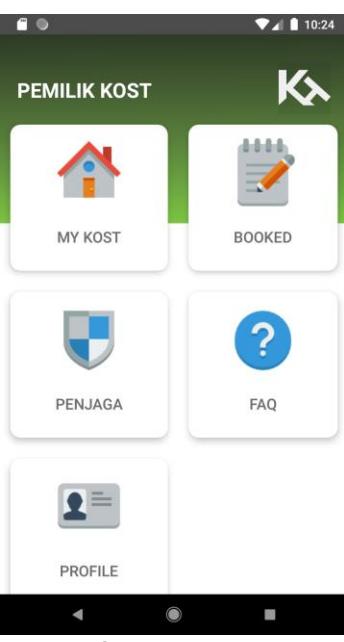
## Layout Log In Pencari



Layout 7. Layout Log In Pencari

Tampilan Log In Pencari seperti ditunjukkan pada gambar di samping ini adalah halaman Log In yang akan muncul ketika user menekan button Log In sebagai Pencari Kost. User akan diminta untuk mengisi Email dan Password yang sudah didaftarkan sebelumnya.

## 3.2 Layout Pemilik Kost Layout Dashboard



Layout 8. Layout Dashboard

Pemilik Kost

Tampilan Dashboard Pemilik Kost seperti ditunjukkan pada gambar di samping akan tampil setelah pemilik kost melakukan log in. Pada Dashboard Pemilik, terdapat 5 menu yang tersedia untuk pemilik kost yaitu:

1. Menu My Kost, yang berkaitan dengan aktivitas yang melibatkan kost pemilik, seperti menampilkan, menambahkan, mengubah, dan menghapus kost.
2. Menu Booked, yang berkaitan dengan aktivitas pengajuan sewa kost yang dimiliki pemilik oleh pencari kost.
3. Menu Penjaga, yang berkaitan dengan aktivitas yang melibatkan penjaga kost, seperti menampilkan, menambahkan, mengubah, dan menghapus data penjaga.
4. Menu FAQ, yang berkaitan dengan pertanyaan beserta jawabannya terkait layanan KostHost.
5. Menu Profile, yang berkaitan dengan profil pemilik dimana pemilik dapat melihat informasi dirinya yang sudah didaftarkan dan mengubah informasi tersebut.

Jika logo KH di kanan atas ditekan, pemilik kost akan diarahkan ke tampilan dashboard pemilik.

## Layout My Kost



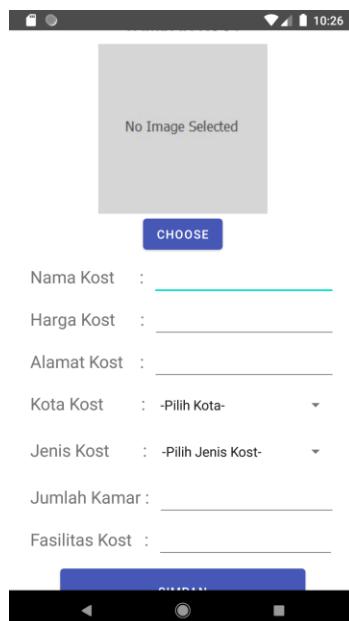
Layout 9. Layout Menu My Kost

Tampilan My Kost di samping tampil ketika menu My Kost dipilih dari dashboard pemilik kost. Menu My Kost disamping memanfaatkan database MySQL khususnya tabel Kost dalam menampilkan, menambahkan, mengubah, dan menghapus data kost.

Pada halaman My Kost terdapat list yang menampilkan informasi kost yang telah didaftarkan pemilik kost. Informasi yang ditampilkan yaitu nama kost, alamat kost, kota kost, jenis kost, ketersediaan kamar, fasilitas kost, dan harga sewa kost perbulan. Apabila kost pada list ditekan, pemilik akan diarahkan ke tampilan edit kost yang dipilih.

Untuk menambahkan kost, pemilik dapat menekan tombol + yang terdapat di bagian kanan bawah. Sedangkan, untuk kembali ke dashboard pemilik, dapat menekan logo KH di kanan atas.

## Layout My Kost - Tambah Kost



Layout 10. Layout Tambah Kost

Tampilan My Kost - Tambah Kost di samping tampil ketika tombol + ditekan pada halaman My Kost. Halaman ini berfungsi untuk menambahkan kost baru ke database (MySQL) tabel Kost.

Untuk menambahkan kost baru ke database, pemilik kost harus mengisi data nama kost, harga kost, alamat kost, kota kost, jenis kost, jumlah kamar, dan fasilitas kost, serta memilih foto kost dari gallery dengan menekan tombol choose. Setelah semua data terisi, pemilik kost dapat menyimpan data dengan menekan tombol simpan.

## Layout My Kost - Edit Kost



Layout 11. Layout Edit Kost

Tampilan My Kost - Edit Kost di samping tampil ketika suatu kost pada list di halaman My Kost dipilih. Halaman ini berfungsi untuk mengubah atau menghapus data kost yang tersimpan di database (MySQL) tabel Kost.

Untuk mengubah data kost yang tersimpan pada database, pemilik kost dapat langsung mengganti data yang berubah atau salah ketik, seperti nama kost, harga kost, alamat kost, kota kost, jenis kost, jumlah kamar, dan fasilitas kost, serta memperbarui foto kost dengan memilih gambar langsung dari gallery dengan menekan tombol choose. Setelah perubahan yang diinginkan sudah dilakukan, pemilik kost dapat menyimpan data dengan menekan tombol simpan.

Untuk menghapus data kost, pemilik kost dapat menekan tombol delete. Ketika menekan tombol delete, akan muncul alert dialog yang mengkonfirmasi aksi delete yang akan dilakukan. Jika sudah yakin akan menghapus data, pemilik kost dapat menekan tombol ya.

## Layout Booked



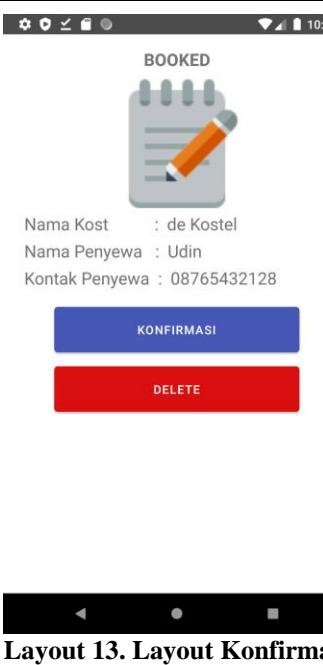
Layout 12. Layout Menu Booked

Tampilan Booked seperti ditunjukkan pada gambar di samping akan tampil ketika menu Booked dipilih dari dashboard pemilik kost. Menu Booked di samping memanfaatkan database MySQL khususnya tabel Penyewaan, AkunPencariKost, dan Kost dalam menampilkan data pengajuan sewa.

Terdapat informasi nama kost, nama penyewa, kontak penyewa, dan status penyewaan. Status penyewaan saat awal diajukan adalah "Available". Namun ketika pemilik kost sudah menghubungi pencari kost dan melakukan konfirmasi, status penyewaan akan berubah menjadi "Booked".

Untuk kembali ke tampilan dashboard pemilik, dapat menekan logo KH di kanan atas.

## Layout Booked - Available

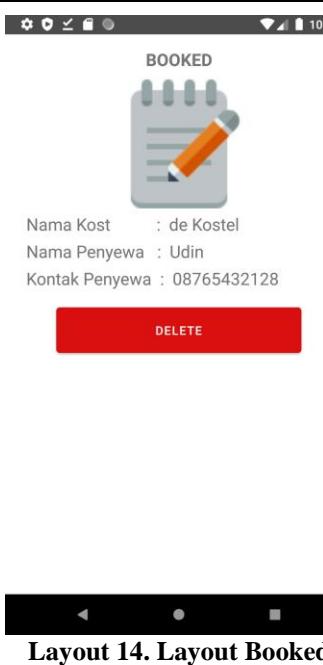


Layout 13. Layout Konfirmasi  
Penyewaan

Tampilan Booked - Available seperti ditunjukkan pada gambar di samping akan tampil ketika list view data kost pemilik yang telah dibook oleh pencari kost dengan status kamar kost masih available atau belum dikonfirmasi oleh pemilik kost ditekan. Tampilan di samping memanfaatkan database MySQL khususnya tabel Penyewaan, AkunPencariKost, dan Kost dalam menampilkan data pengajuan sewa.

Terdapat informasi nama kost, nama penyewa, dan juga kontak penyewa. Jika pemilik kost menyetujui kost miliknya untuk disewakan pada calon penyewa tersebut, pemilik kost dapat menekan tombol konfirmasi dengan data kamar kost kemudian akan secara otomatis berkurang 1, sedangkan jika pemilik kost tidak menyetujuinya, dapat menekan tombol delete agar data kost yang akan dibook tersebut dihapus.

## Layout Booked - Booked



Layout 14. Layout Booked  
Penyewaan

Tampilan Booked - Booked seperti ditunjukkan pada gambar di samping akan tampil ketika list view data kost pemilik yang telah dibook oleh pencari kost dengan status kamar kost sudah booked atau sudah dikonfirmasi oleh pemilik kost. Tampilan di samping memanfaatkan database MySQL khususnya tabel Penyewaan, AkunPencariKost, dan Kost dalam menampilkan data pengajuan sewa.

Terdapat informasi nama kost, nama penyewa, dan juga kontak penyewa. Pemilik kost dapat menekan tombol delete agar data kost yang telah dibook tersebut dihapus. Hal ini terjadi ketika kost sudah tidak lagi disewakan oleh penyewa kost tersebut. Data kamar kost kemudian akan secara otomatis bertambah 1.

## Layout Penjaga



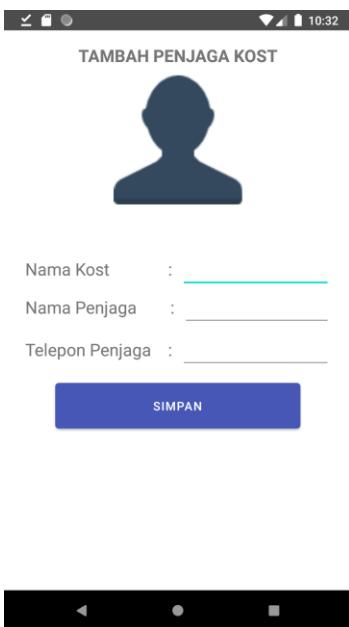
Layout 15. Layout Menu Penjaga

Tampilan Penjaga di samping tampil ketika menu Penjaga dipilih dari dashboard pemilik kost. Tampilan di samping memanfaatkan database SQLite yang bersifat internal dengan variabel ID, namaKost, namaPenjaga, dan teleponPenjaga.

Fitur penjaga kost di samping menggunakan database local yang bermanfaat untuk membantu pemilik kost dalam menyimpan data penjaga kost yang dimilikinya. Terdapat informasi nama kost, nama penjaga, dan telepon dari penjaga tersebut.

Tombol + di kanan bawah digunakan untuk menambah data penjaga kost. Jika logo KH di kanan atas ditekan, maka pemilik kost akan diarahkan kembali ke tampilan dashboard pemilik.

## Layout Penjaga - Tambah

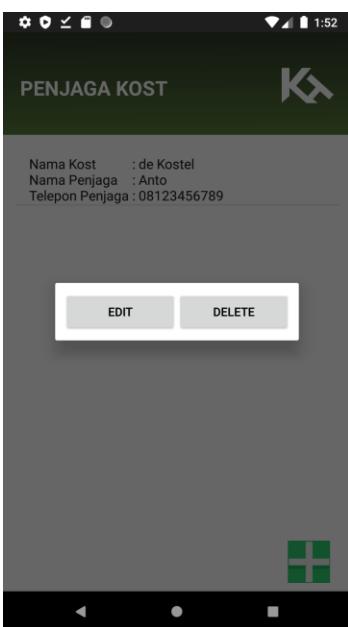


Layout 16. Layout Tambah Penjaga

Tampilan Tambah Penjaga Kost di samping akan tampil ketika tombol + sebelumnya sudah ditekan. Tampilan di samping memanfaatkan database SQLite yang bersifat internal dengan variabel ID, namaKost, namaPenjaga, dan teleponPenjaga.

Terdapat nama kost, nama penjaga, dan telepon penjaga yang dapat diisi. Setelah lengkap, pemilik kost dapat menekan tombol simpan yang kemudian akan diarahkan kembali ke tampilan penjaga kost beserta dengan data penjaga kost baru yang telah ditambahkan.

## Layout Penjaga - Dialog Edit/Delete

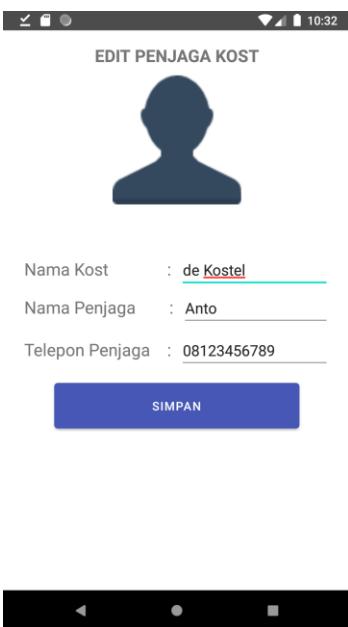


Layout 17. Layout Dialog Edit dan Delete

Tampilan Dialog Edit/Delete Penjaga Kost pada gambar di samping akan muncul ketika list view penjaga kost ditekan lama.

Terdapat 2 menu yang dapat dipilih yaitu edit dan delete. Edit digunakan untuk mengubah berbagai informasi penjaga kost yang telah ada sebelumnya dan menekan delete untuk menghapus data penjaga kost yang diinginkan.

## Layout Penjaga - Edit

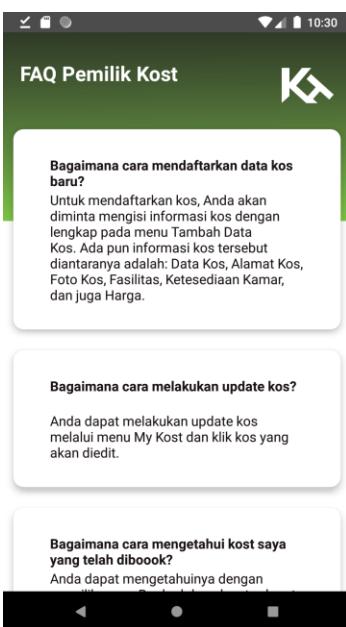


Layout 18. Layout Edit Penjaga

Tampilan Edit Penjaga Kost pada gambar di samping akan muncul ketika tombol edit dari dialog list view penjaga kost ditekan. Tampilan di samping memanfaatkan database SQLite yang bersifat internal dengan variabel ID, namaKost, namaPenjaga, dan teleponPenjaga.

Pada halaman ini kita dapat mengubah data nama kost, nama penjaga, dan telepon penjaga yang sudah ada sebelumnya, dan mengisi sesuai dengan data baru yang kita inginkan. Setelah lengkap, pemilik kost dapat menekan tombol simpan yang kemudian akan diarahkan kembali ke tampilan penjaga kost beserta dengan data penjaga kost baru yang telah diedit.

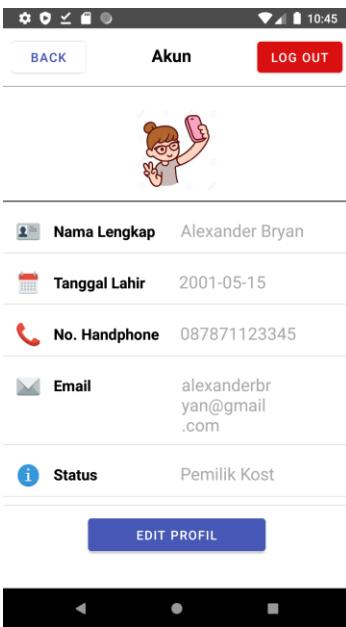
## Layout FAQ Pemilik



Layout 19. Layout FAQ Pemilik

Tampilan disamping adalah tampilan dari FAQ Pemilik. Layout ini akan muncul ketika user Log In sebagai pemilik dan menekan FAQ pada Dashboard. Menu Frequently Asked Question atau FAQ ini berisi tentang pertanyaan apa saja yang sering diterima oleh pemilik kost agar pemilik kost tidak kebingungan saat menggunakan aplikasi ini. Untuk kembali ke tampilan Dashboard pemilik dapat menekan logo KH pada kanan atas layout FAQ.

## Layout Profil Pemilik

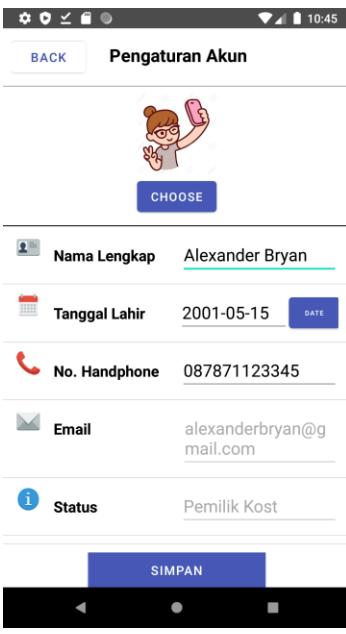


Layout 20. Layout Profil Pemilik

Tampilan disamping adalah tampilan dari Profil Pemilik. Layout ini akan muncul ketika user Log In sebagai pemilik dan menekan Profile pada Dashboard. Terdapat dua Button pada layout ini yaitu Button Back dan Button Log Out. Button Back digunakan untuk kembali ke halaman sebelumnya dan Button Log Out digunakan untuk kembali ke halaman Log In/Sign Up.

Lalu terdapat foto pemilik kost beserta data diri si pemilik kost, yaitu Nama Lengkap, Tanggal Lahir, Nomor Handphone, Email serta Status. Dibagian bawah layout terdapat tombol Edit Profil yang digunakan oleh pemilik kost untuk mengubah foto profil atau data diri.

### Layout Edit Profil Pemilik



Layout 21. Layout Edit Profil  
Pemilik

Tampilan disamping adalah tampilan dari Edit Profil Pemilik. Layout ini akan muncul ketika user Log In sebagai pemilik dan menekan Profile pada Dashboard. Terdapat Button pada kiri atas layout ini yaitu Button Back, dimana ketika ditekan maka akan kembali ke layout Profil Pemilik. Lalu terdapat Button Choose dibagian bawah foto profil yang bertujuan untuk mengubah foto profil pemilik. Lalu pemilik dapat mengubah 3 hal yaitu Nama Lengkap, Tanggal Lahir, serta Nomor Handphone. Setelah pemilik merasa sudah selesai dalam mengubah profil, maka terdapat Button Simpan yang digunakan untuk menyimpan data yang sudah di edit ke dalam database.

### 3.3 Layout Pencari Kost

#### Layout Pencari - Dashboard



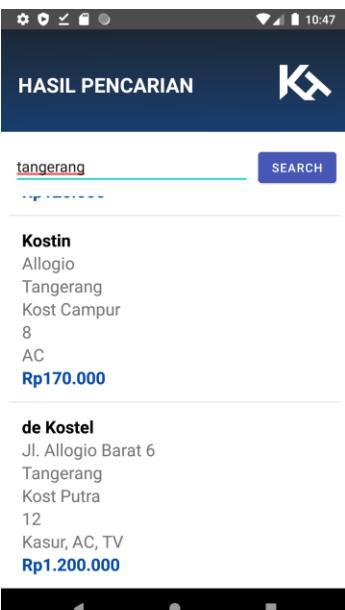
Layout 22. Layout Dashboard  
Pencari Kost

Tampilan Dashboard Pencari Kost seperti ditunjukkan pada gambar di samping akan tampil setelah pencari kost melakukan log in. Pada Dashboard Pencari, terdapat 4 menu yang tersedia untuk pencari kost yaitu:

1. Menu Search, yang berisi fitur search untuk mencari kost yang ingin dicari oleh pencari berdasarkan kota sebagai kata kuncinya.
2. Menu Booked, yang berkaitan dengan aktivitas pengajuan sewa oleh pencari kost terhadap kost yang dimiliki oleh pemilik kost.
3. Menu FAQ, yang berkaitan dengan FAQ (*Frequently Asked Questions*) yang berarti pertanyaan-pertanyaan yang sering ditanyakan beserta jawabannya terkait layanan KostHost.
4. Menu Profile, yang berkaitan dengan profil pencari dimana pencari dapat melihat informasi dirinya yang sudah didaftarkan dan mengubah informasi tersebut.

Jika logo KH di kanan atas ditekan, pencari kost akan diarahkan ke tampilan dashboard pencari.

## Layout Pencari - Search



Layout 23. Layout Search Kost  
Pencari

Tampilan Search oleh Pencari Kost seperti ditunjukkan pada gambar di samping akan tampil setelah pencari kost menekan tombol search yang ada sebelumnya pada Dashboard Pencari. Menu search kost disamping memanfaatkan database MySQL khususnya tabel Kost dalam menampilkan data kost yang ada berdasarkan kota kost.

Menu search ini berisi fitur search untuk mencari kost yang ingin dicari oleh pencari berdasarkan kota sebagai kata kuncinya. Pada gambar disamping kami mengetikkan kota "tangerang" sebagai kata kunci, lalu jika kita menekan tombol search, pada list view di bawah akan langsung muncul hasil pencarian berupa kost-kost yang berlokasi di kota Tangerang. Demikian juga berlaku pada hasil pencarian untuk kost yang berada di kota lain sesuai dengan nama kota yang kita ketik di kolom search.

Jika logo KH di pojok kanan atas ditekan, pencari kost akan diarahkan ke tampilan dashboard pencari.

## Layout Pencari – Search – Detail Kost



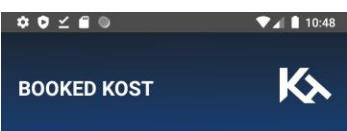
Layout 24. Layout Detail Kost

Tampilan Search Detail Kost oleh Pencari Kost seperti ditunjukkan pada gambar di samping akan tampil setelah pencari kost menekan salah satu kost dari list view kost pada halaman search kost.

Menu Book Kost disamping memanfaatkan database MySQL khususnya tabel Kost dalam menampilkan dan menambahkan data kost untuk tahap booking kost oleh pencari. Halaman ini berisi informasi detail mengenai kost, seperti nama kost, harga kost, alamat kost, kota kost, jenis kost, jumlah kamar dan fasilitas kost, seperti yang dapat dilihat pada gambar disamping.

Kemudian dibawah detail kost tersebut terdapat tombol book bagi pencari yang ingin melakukan booking terhadap kost yang diminatinya tersebut. Jika tombol book ditekan, pencari akan diarahkan ke sebuah halaman baru bernama Booked Kost yang berisi list kost yang sudah berhasil di book oleh pencari.

## Layout Pencari - Booked



Nama Kost: de Kostel  
Alamat Kost: Jl. Allegio Barat 6

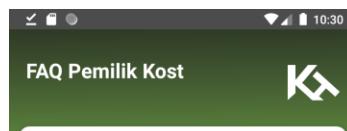
Tampilan Booked seperti ditunjukkan pada gambar di samping akan tampil ketika tombol Booked ditekan dari halaman detail kost. Menu Booked disamping memanfaatkan database MySQL dalam menampilkan kost mana saja yang sudah di booking oleh pencari kost.

Di halaman ini pencari kost dapat memantau daftar dari kost mana yang sudah di booking oleh dirinya. Terdapat informasi berupa nama kost dan alamat kost yang sudah di booking oleh pencari kost sebelumnya.

Untuk kembali ke tampilan dashboard pencari, dapat menekan logo KH di kanan atas atau menekan tombol back pada device.

Layout 25. Layout Booked Kost  
Pencari

## Layout Pencari - FAQ



Bagaimana cara mendaftarkan data kos baru?  
Untuk mendaftarkan kos, Anda akan diminta mengisi informasi kos dengan lengkap pada menu Tambah Data Kos. Ada pun informasi kos tersebut diantaranya adalah: Data Kos, Alamat Kos, Foto Kos, Fasilitas, Ketesedian Kamar, dan juga Harga.

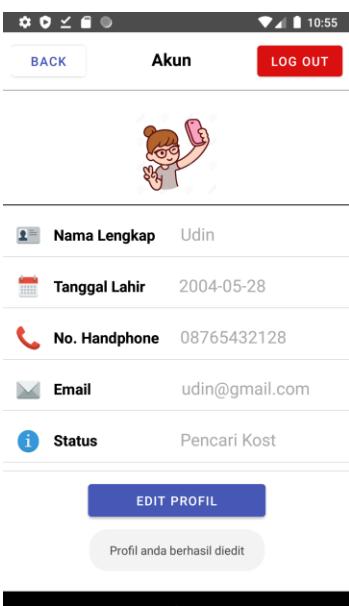
Bagaimana cara melakukan update kos?  
Anda dapat melakukan update kos melalui menu My Kost dan klik kos yang akan diedit.

Bagaimana cara mengetahui kost saya yang telah dibook?  
Anda dapat mengetahuinya dengan

Tampilan disamping adalah tampilan dari FAQ Pencari. Layout ini akan muncul ketika user Log In sebagai pencari dan menekan FAQ pada Dashboard. Menu Frequently Asked Question atau FAQ ini berisi tentang pertanyaan apa saja yang sering diajukan oleh pencari kost agar pencari kost dapat mengetahui apa yang diperlukan dengan mudah. Untuk kembali ke tampilan Dashboard Pencari Kost dapat menekan logo KH pada kanan atas layout FAQ.

Layout 26. Layout FAQ Pencari

## Layout Pencari - Profil

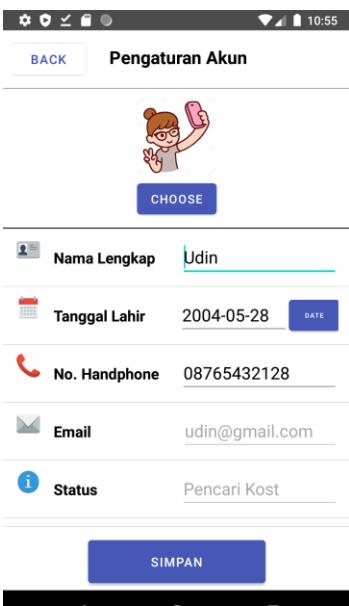


Layout 27. Layout Profil Pencari

Tampilan disamping adalah tampilan dari Profil Pencari. Layout ini akan muncul ketika user Log In sebagai pencari dan menekan Profile pada Dashboard. Terdapat dua Button pada layout ini yaitu Button Back dan Button Log Out. Button Back digunakan untuk kembali ke halaman sebelumnya dan Button Log Out digunakan untuk kembali ke halaman Log In/Sign Up.

Lalu terdapat foto pencari kost beserta data diri si pencari kost, yaitu Nama Lengkap, Tanggal Lahir, Nomor Handphone, Email serta Status. Dibagian bawah layout terdapat tombol Edit Profil yang digunakan oleh pemilik kost untuk mengubah foto profil atau data diri.

## Layout Pencari - Edit - Profil



Layout 28. Layout Edit Profil  
Pencari

Tampilan disamping adalah tampilan dari Edit Profil Pencari. Layout ini akan muncul ketika user Log In sebagai pencari dan menekan Profile pada Dashboard. Terdapat Button pada kiri atas layout ini yaitu Button Back, dimana ketika ditekan maka akan kembali ke layout Profil Pemilik. Lalu terdapat Button Choose dibagian bawah foto profil yang bertujuan untuk mengubah foto profil pencari. Lalu pencari dapat mengubah 3 hal yaitu Nama Lengkap, Tanggal Lahir, serta Nomor Handphone. Setelah pencari merasa sudah selesai dalam mengubah profil, maka terdapat Button Simpan yang digunakan untuk menyimpan data yang sudah di edit ke dalam database.

## BAB IV

# CODING PADA APLIKASI

### 4.1 Coding PHP

#### Coding Connect

```
connect.php > ...
1  <?php
2  //koneksi ke db
3  $servername = "localhost";
4  $username = "root";
5  $password = "";
6  $dbname = "kosthostmobile";
7
8  // Create connection
9  $connect = new mysqli($servername, $username, $password, $dbname);
10
11 // Check connection
12 if (!$connect) {
13     die("Connection failed: " . mysqli_connect_error());
14 }
15 ?>
```

Code 1. Code PHP Connect Database

Code di atas digunakan untuk membuat koneksi dengan database MySQL dengan nama server adalah localhost, nama username adalah root, password kosong, dan database bernama kosthostmobile. Query yang digunakan menggunakan mysqli dengan menggabungkan nama server, username, password, dan nama database yang sebelumnya sudah diinisialisasikan. Jika ternyata jaringan tidak terhubung dengan database, maka akan ditampilkan pesan koneksi gagal beserta dengan alasannya melalui fungsi die.

#### Coding Proses Insert Data

```
insert-kost.php > ...
1  <?php
2  header('Content-type:application/json;charset=utf-8');
3  include "connect.php";
4
5  if (isset($_POST['namaKost']) && isset($_POST['fotoKost']) && isset($_POST['alamatKost']) && isset($_POST['kotaKost']) && isset($_POST['jenisKost']) &&
6  && isset($_POST['jumlahKamar']) && isset($_POST['hargaKost']) && isset($_POST['deskripsiFasilitas']) && isset($_POST['IDPemilik'])) {
7      $namaKost = $_POST['namaKost'];
8      $alamatKost = $_POST['alamatKost'];
9      $kotaKost = $_POST['kotaKost'];
10     $jenisKost = $_POST['jenisKost'];
11     $jumlahKamar = $_POST['jumlahKamar'];
12     $hargaKost = $_POST['hargaKost'];
13     $deskripsiFasilitas = $_POST['deskripsiFasilitas'];
14     $IDPemilik = $_POST['IDPemilik'];
15     $fotoKost = $_POST['fotoKost'];
16
17     $target_dir = "uploads/kost";
18
19     if (!file_exists($target_dir)) {
20         mkdir($target_dir, 0777, true);
21     }
22
23     $target_dir = $target_dir."/". $namaKost.".".time().".jpeg";
24
25     if (file_put_contents($target_dir, base64_decode($fotoKost))) {
26         $q = mysqli_query($connect, "INSERT INTO kost (namakost, alamatKost, kotaKost, jenisKost, jumlahKamar, hargaKost, deskripsiFasilitas, IDPemilik, fotoKost)
27             VALUES ('$namaKost', '$alamatKost', '$kotaKost', '$jenisKost', '$jumlahKamar', '$hargaKost', '$deskripsiFasilitas', '$IDPemilik', '$target_dir')");
28         $response = array();
29
30         if ($q) {
31             $response["success"] = 1;
32             $response["message"] = "data berhasil ditambah";
33             echo json_encode($response);
34         } else {
35             $response["success"] = 0;
36             $response["message"] = "data gagal ditambah";
37             echo json_encode($response);
38         }
39     } else {
40         $response["success"] = 1;
41         $response["message"] = "data kosong";
42         echo json_encode($response);
43     }
44 }
45 ?>
```

Code 2. Code PHP Insert Data Kost

Code diatas digunakan untuk menambahkan data kost yang dimiliki oleh pemilik ke dalam database. Untuk melakukan koneksi ke database dipanggil fungsi yang sudah dibuat

pada file connect.php dengan menggunakan include seperti pada baris 3. Kemudian, dilakukan validasi untuk mengetahui apakah semua data yang diperlukan sudah dimasukkan dengan menggunakan fungsi isset seperti pada baris 5 dan 6. Fungsi isset akan mengembalikan nilai true jika data sudah diisi dan nilai false jika data belum diisi. Jika semua data yang diperlukan sudah diisi, setiap data dijadikan suatu variabel seperti pada baris 7 sampai baris 15, yaitu data namaKost pada variabel namaKost, data alamatKost pada variabel alamatKost, data kotaKost pada variabel kotaKost, data jenisKost pada variabel jenisKost, data jumlahKamar pada variabel jumlahKamar, data deskripsiFasilitas pada variabel deskripsiFasilitas, data IDPemilik pada variabel IDPemilik, dan data fotoKost pada variabel fotoKost. Hal ini dilakukan untuk memudahkan dalam melakukan coding selanjutnya. Variabel target\_dir pada baris 17 digunakan untuk menampung lokasi penyimpanan foto yang diunggah oleh pengguna. Jika file yang diunggah belum ada pada folder lokasi penyimpanan foto, dibuat direktori baru seperti pada baris 19 sampai 21. Untuk mengatur nama file yang dimasukkan dilakukan dengan menambahkan penamaan yang diinginkan digabungkan dengan lokasi yang sudah ditetapkan sebelumnya seperti pada baris 23. Jika gambar atau foto yang diunggah diletakkan pada lokasi yang ditentukan, dilakukan query insert data ke dalam database yang ditampung pada variabel q dengan menggunakan fungsi mysqli\_query dan terlebih dahulu memanggil variabel connect untuk menghubungkan ke database seperti pada baris 26 dan 27.

Kemudian respon yang diberikan ditampung ke dalam array dengan variabel response. Apabila respon pada kolom success bernilai 1 artinya data berhasil ditambahkan ke dalam database sehingga kolom message menampung pesan data berhasil ditambah, kedua nilai ini akan dicetak ke dalam bentuk JSON seperti pada baris 30 sampai 33. Sebaliknya, apabila respon pada kolom success bernilai 0 artinya data gagal ditambahkan ke dalam database sehingga kolom message menampung pesan data gagal ditambah, kedua nilai ini akan dicetak ke dalam bentuk JSON seperti pada baris 34 sampai 37. Apabila kolom success bernilai -1 artinya tidak ada data yang dimasukkan pada proses yang dijalankan sehingga kolom message menampung pesan data kosong, kedua data ini akan dicetak ke dalam bentuk JSON seperti pada baris 39 sampai 42.

```

insert-pemilik.php > ...
1  <?php
2   header('Content-type:application/json;charset=utf-8');
3   include "connect.php";
4
5   if (isset($_POST['namaPemilik']) && isset($_POST['tanggallahir']) && isset($_POST['telepon']) && isset($_POST['email']) && isset($_POST['password'])) {
6     $namaPemilik=$_POST['namaPemilik'];
7     $tanggallahir=$_POST['tanggallahir'];
8     $telepon=$_POST['telepon'];
9     $email=$_POST['email'];
10    $password=$_POST['password'];
11
12    $q=mysqli_query($connect,"INSERT INTO akunpemilikcost(namaPemilik, tanggallahir, telepon, email, password"
13    VALUES('$namaPemilik', '$tanggallahir', '$telepon', '$email', '$password')");
14    $response=array();
15
16    if($q){
17      $response["success"]=1;
18      $response["message"]="data berhasil ditambah";
19      echo json_encode($response);
20    } else {
21      $response["success"]=0;
22      $response["message"]="data gagal ditambah";
23      echo json_encode($response);
24    }
25  } else {
26    $response["success"]=-1;
27    $response["message"]="data kosong";
28    echo json_encode($response);
29  }
30 ?>
```

**Code 3. Code PHP Insert Data Akun Pemilik**

Code diatas digunakan untuk menambahkan data akun pemilik yang didaftarkan oleh pengguna ke dalam database. Untuk melakukan koneksi ke database dipanggil fungsi yang sudah dibuat pada file connect.php dengan menggunakan include seperti pada baris 3. Kemudian, dilakukan validasi untuk mengetahui apakah semua data yang diperlukan sudah dimasukkan dengan menggunakan fungsi isset seperti pada baris 5. Fungsi isset akan mengembalikan nilai true jika data sudah diisi dan nilai false jika data belum diisi. Jika semua data yang diperlukan sudah diisi, setiap data dijadikan suatu variabel seperti pada baris 6 sampai baris 10, yaitu data namaPemilik pada variabel namaPemilik, data tanggalLahir pada variabel tanggalLahir, data telepon pada variabel telepon, data email pada variabel email, dan data password pada variabel password. Hal ini dilakukan untuk memudahkan dalam melakukan coding selanjutnya. Kemudian dilakukan query insert data ke dalam database yang ditampung pada variabel q dengan menggunakan fungsi mysqli\_query dan terlebih dahulu memanggil variabel connect untuk menghubungkan ke database seperti pada baris 12 dan 13.

Kemudian respon yang diberikan ditampung ke dalam array dengan variabel response. Apabila respon pada kolom success bernilai 1 artinya data berhasil ditambahkan ke dalam database sehingga kolom message menampung pesan data berhasil ditambah, kedua nilai ini akan dicetak ke dalam bentuk JSON seperti pada baris 16 sampai 19. Sebaliknya, apabila respon pada kolom success bernilai 0 artinya data gagal ditambahkan ke dalam database sehingga kolom message menampung pesan data gagal ditambah, kedua nilai ini akan dicetak ke dalam bentuk JSON seperti pada baris 20 sampai 23. Apabila kolom success bernilai -1 artinya tidak ada data yang dimasukkan pada proses yang dijalankan sehingga kolom message menampung pesan data kosong, kedua data ini akan dicetak ke dalam bentuk JSON seperti pada baris 25 sampai 28.

```
insert-pencari.php > ...
1  <?php
2  header('Content-type:application/json;charset=utf-8');
3  include "connect.php";
4
5  if (isset($_POST['namaPencari']) && isset($_POST['tanggalLahir']) && isset($_POST['telepon']) && isset($_POST['email']) && isset($_POST['password'])) {
6      $namaPencari = $_POST['namaPencari'];
7      $tanggalLahir = $_POST['tanggalLahir'];
8      $telepon = $_POST['telepon'];
9      $email = $_POST['email'];
10     $password = $_POST['password'];
11
12     $q = mysqli_query($connect, "INSERT INTO akunpencarikost(namaPencari, tanggalLahir, email, password, telepon) VALUES('$namaPencari','$tanggalLahir',
13     '$email', '$password', '$telepon')");
14     $response = array();
15
16     if ($q) {
17         $response["success"] = 1;
18         $response["message"] = "data berhasil ditambah";
19         echo json_encode($response);
20     } else {
21         $response["success"] = 0;
22         $response["message"] = "data gagal ditambah";
23         echo json_encode($response);
24     }
25 } else {
26     $response["success"] = -1;
27     $response["message"] = "data kosong";
28     echo json_encode($response);
29 }
?>
```

#### Code 4. Code PHP Insert Data Akun Pencari

Code diatas digunakan untuk menambahkan data akun pencari yang didaftarkan oleh pengguna ke dalam database. Untuk melakukan koneksi ke database dipanggil fungsi yang sudah dibuat pada file connect.php dengan menggunakan include seperti pada baris 3. Kemudian, dilakukan validasi untuk mengetahui apakah semua data yang diperlukan sudah

dimasukkan dengan menggunakan fungsi isset seperti pada baris 5. Fungsi isset akan mengembalikan nilai true jika data sudah diisi dan nilai false jika data belum diisi. Jika semua data yang diperlukan sudah diisi, setiap data dijadikan suatu variabel seperti pada baris 6 sampai baris 10, yaitu data namaPencari pada variabel namaPencari, data tanggalLahir pada variabel tanggalLahir, data telepon pada variabel telepon, data email pada variabel email, dan data password pada variabel password. Hal ini dilakukan untuk memudahkan dalam melakukan coding selanjutnya. Kemudian dilakukan query insert data ke dalam database yang ditampung pada variabel q dengan menggunakan fungsi mysqli\_query dan terlebih dahulu memanggil variabel connect untuk menghubungkan ke database seperti pada baris 12 dan 13.

Kemudian respon yang diberikan ditampung ke dalam array dengan variabel response. Apabila respon pada kolom success bernilai 1 artinya data berhasil ditambahkan ke dalam database sehingga kolom message menampung pesan data berhasil ditambah, kedua nilai ini akan dicetak ke dalam bentuk JSON seperti pada baris 16 sampai 19. Sebaliknya, apabila respon pada kolom success bernilai 0 artinya data gagal ditambahkan ke dalam database sehingga kolom message menampung pesan data gagal ditambah, kedua nilai ini akan dicetak ke dalam bentuk JSON seperti pada baris 20 sampai 23. Apabila kolom success bernilai -1 artinya tidak ada data yang dimasukkan pada proses yang dijalankan sehingga kolom message menampung pesan data kosong, kedua data ini akan dicetak ke dalam bentuk JSON seperti pada baris 25 sampai 28.

```

1  <?php
2  header("content-type:application/json;charset=utf-8");
3  include "connect.php";
4
5  if (isset($_POST['IDKost']) && isset($_POST['IDPencari'])) {
6      $IDkost = $_POST['IDKost'];
7      $IDPencari = $_POST['IDPencari'];
8      $status = "Available";
9      $query = mysqli_query($connect, "INSERT INTO penyewaan(IDKost, IDPencari, status)
10      VALUES('$IDkost', '$IDPencari', '$status')");
11      $response = array();
12      if ($query) {
13          $response["success"] = 1;
14          $response["message"] = "Data Berhasil ditambah";
15          echo json_encode($response);
16      } else {
17          $response["success"] = 0;
18          $response["message"] = "Data Gagal ditambah";
19          echo json_encode($response);
20      }
21  } else {
22      $response["success"] = -1;
23      $response["message"] = "Data Kosong";
24      echo json_encode($response);
25  }

```

#### Code 5. Code PHP Insert Penyewaan oleh Pencari

Code diatas digunakan untuk menambahkan data penyewaan atas booking yang telah dilakukan oleh pencari terhadap kos yang diinginkannya ke dalam database. Langkah pertama yang dilakukan adalah membuka koneksi dengan menggunakan fungsi yang sudah dibuat sebelumnya pada file connect.php sehingga kita hanya perlu memanggil menggunakan include seperti pada baris ke 3. Kemudian pada baris selanjutnya kita memakai fungsi isset untuk memeriksa apakah variabel sudah diatur atau belum, disini kita akan memeriksa IDKost dan IDPencari. Fungsi isset ini akan mengembalikan nilai false data masih kosong. Jika semua data sudah diisi maka akan mengembalikan nilai true sehingga berikutnya kita akan

menjadikan data menjadi variabel seperti pada baris ke 6 dan ke 7, yaitu data IDKost pada variabel IDKost, dan data IDPencari pada variabel IDPencari. Pada baris ke 8, kita akan set status dengan nilai “Available” yang menyatakan bahwa kamar kos tersebut masih tersedia untuk di booking.

Variabel connect kemudian dipanggil untuk menghubungkan file php kita ini ke database. Setelahnya kita akan melakukan query insert yang ditampung dalam variable query dengan awalan fungsi mysqli\_query sebagai sarana untuk menampung perintah atau instruksi query yang hendak kita masukkan ke database. Lalu hasil respon dari pemasukan data akan ditampung dalam sebuah array dengan variabel-variabel response. Hasil dari response ini kemudian akan dicetak dalam bentuk response nilai -1, 0 ataupun 1 ke dalam format JSON seperti pada setiap nilai yang sukses diproses akan menampilkan message tersendiri. Variabel response dengan nilai 1 menyatakan bahwa data tersebut berhasil ditambahkan kedalam database, kemudian akan mencetak message “data berhasil ditambah”, sedangkan nilai 0 akan mencetak sebuah pesan bahwa “data gagal ditambahkan” dan jika responnya berupa nilai -1 artinya “data kosong” atau belum ada data yang dimasukkan pada proses, seperti yang dapat dilihat pada baris 12 sampai 24.

### Coding Proses Edit Data

```
1 <?php
2 header('Content-type:application/json;charset=utf-8');
3 include "connect.php";
4
5 if (isset($_POST['IDKost'])) {
6     $IDKost = $_POST['IDKost'];
7
8     $q = mysqli_query($connect,
9         "UPDATE kost SET jumlahKamar = jumlahKamar - 1 where IDKost='$IDKost'");
10    $response = array();
11    if ($q) {
12        $response["success"] = 1;
13        $response["message"] = "Data berhasil diupdate";
14        echo json_encode($response);
15    } else {
16        $response["success"] = 0;
17        $response["message"] = "Data gagal diupdate";
18        echo json_encode($response);
19    }
20 } else {
21     $response["success"] = -1;
22     $response["message"] = "Data kosong";
23     echo json_encode($response);
24 }
```

Code 6. Code PHP Edit untuk Mengurangi Jumlah Kamar Pemilik

Code diatas digunakan untuk mengurangi jumlah pada data kamar pemilik kos yang telah di booking oleh pencari ke dalam database. Langkah pertama yang dilakukan adalah membuka koneksi dengan menggunakan fungsi yang sudah dibuat sebelumnya pada file connect.php sehingga kita hanya perlu memanggilnya menggunakan include seperti pada baris ke 3. Kemudian pada baris selanjutnya kita memakai fungsi isset untuk memeriksa apakah variabel sudah diatur atau belum, disini kita akan memeriksa IDKost. Fungsi isset ini akan mengembalikan nilai false data masih kosong. Jika semua data sudah diisi maka akan mengembalikan nilai true sehingga berikutnya kita akan menjadikan data menjadi variabel seperti pada baris ke 6, yaitu data IDKost pada variabel IDKost. Variabel connect kemudian dipanggil untuk menghubungkan file php kita ini ke database.

Setelahnya kita akan melakukan query update jumlah kamar -1 yang ditampung dalam variabel q dengan awalan fungsi mysqli\_query sebagai sarana untuk menampung perintah query yang hendak kita masukkan ke database. Jumlah kamar disini di minus dengan angka satu dengan tujuan setiap kali ada satu kamar yang di book oleh pencari, jumlah total kamar yang berada di sisi pemilik akan berkurang satu juga jumlahnya sehingga data akan selalu terupdate. Lalu hasil respon dari pemasukan data akan ditampung dalam sebuah array dengan variabel-variabel response. Hasil dari response ini kemudian akan dicetak dalam bentuk response nilai -1, 0 ataupun 1 ke dalam format JSON seperti pada setiap nilai yang sukses diproses akan menampilkan message nya tersendiri. Variabel response dengan nilai 1 menyatakan bahwa data tersebut berhasil ditambahkan kedalam database, kemudian akan mencetak message “data berhasil diupdate”, sedangkan nilai 0 akan mencetak sebuah pesan bahwa “data gagal diupdate” dan jika responnya berupa nilai -1 artinya “data kosong” atau belum ada data yang dimasukkan pada proses, seperti yang dapat dilihat pada baris 11 sampai 23.

```

1 <?php
2 header("Content-type:application/json;charset=utf-8");
3 include "connect.php";
4
5 if (isset($_POST['namaKost']) && isset($_POST['fotoKost']) && isset($_POST['alamatKost']) && isset($_POST['kotaKost']) && isset($_POST['jenisKost']))
6 {
7     $IDKost = $_POST['IDKost'];
8     $namaKost = $_POST['namaKost'];
9     $alamatKost = $_POST['alamatKost'];
10    $kotaKost = $_POST['kotaKost'];
11    $jenisKost = $_POST['jenisKost'];
12    $jumlahKamar = $_POST['jumlahKamar'];
13    $hargaKost = $_POST['hargaKost'];
14    $deskripsiFasilitas = $_POST['deskripsiFasilitas'];
15    $fotoKost = $_POST['fotoKost'];
16
17    $target_dir = "uploads/kost";
18
19    if (!file_exists($target_dir)) {
20        mkdir($target_dir, 0777, true);
21    }
22
23    $target_dir = $target_dir."/". $namaKost."_" .time() . ".jpeg";
24
25    $target_dir = $target_dir."/". $namaKost."_" .time() . ".jpeg";
26
27    if (file_put_contents($target_dir, base64_decode($fotoKost))) {
28        $q = mysqli_query($connect, "UPDATE kost SET namaKost = '$namaKost', fotoKost = '$target_dir', alamatKost = '$alamatKost', kotaKost = '$kotaKost' WHERE IDKost = '$IDKost'");
29        $response = array();
30        if ($q) {
31            $response["success"] = 1;
32            $response["message"] = "Data berhasil diupdate";
33            echo json_encode($response);
34        } else {
35            $response["success"] = 0;
36            $response["message"] = "Data gagal diupdate";
37            echo json_encode($response);
38        }
39    } else {
40        $response["success"] = -1;
41        $response["message"] = "Data kosong";
42        echo json_encode($response);
43    }
44}

```

#### Code 7. Code PHP Edit Update Kost

Code diatas digunakan untuk mengedit data kost yang berada dalam database. Hal pertama yang dilakukan adalah melakukan koneksi dengan menggunakan fungsi yang telah dibuat sebelumnya pada file connect.php. Selanjutnya kita menggunakan fungsi isset untuk memeriksa apakah variabel yang digunakan sudah diatur atau belum, variabel yang digunakan disini adalah IDKost, namaKost, alamatKost, kotaKost, jenisKost, jumlahKamar, hargaKost, deskripsiFasilitas, dan fotoKost. Fungsi isset adalah mengembalikan nilai false pada data yang masih kosong. Jika data sudah diisi maka akan mengembalikan nilai true, selanjutnya kita

menjadikan data menjadi variabel yaitu data IDKost pada variabel IDKost, namaKost pada variabel namaKost, alamatKost pada variabel alamatKost, kotaKost pada variabel kotaKost, jenisKost pada variabel jenisKost, jumlahKamar pada variabel jumlahKamar, hargaKost pada variabel hargaKost, deskripsiFasilitas pada variabel deskripsiFasilitas, dan fotoKost pada variabel fotoKost. \$target\_dir = "uploads/" menentukan direktori tempat file akan ditempatkan, kali ini adalah file kost. Base64 mendekode data ke dalam format yang dapat dibaca manusia. Skema pengkodean Base64 umumnya digunakan ketika ada kebutuhan untuk menyandikan data biner, terutama ketika data tersebut perlu disimpan dan ditransfer melalui media yang dirancang untuk menangani teks.

Variable q dengan awalan fungsi mysqli\_query sebagai sarana untuk menampung perintah query yang hendak kita masukkan ke database. Data kost disini sesuai dengan data yang terdapat dalam database sehingga data yang diedit akan selalu terupdate di database. Lalu hasil respon dari penginputan data akan ditampung dalam sebuah array dengan variabel response. Hasil dari respon ini dicetak dalam bentuk nilai respon 1,0, atau -1 ke dalam format json. Nilai 1 menyatakan bahwa data berhasil dihapus kemudian mencetak pesan “Data berhasil diupdate”. Nilai 0 menyatakan bahwa data gagal dihapus kemudian mencetak pesan “Data gagal diupdate”. Apabila nilai -1 menyatakan bahwa data kosong, atau belum ada data yang dimasukkan.

```
<?php
header('content-type:application/json;charset=utf-8');
include 'connect.php';

if (isset($_POST['IDPemilik']) && isset($_POST['namaPemilik']) && isset($_POST['tanggalLahir']) && isset($_POST['telepon']) && isset($_POST['fotoPemilik'])) {
    $IDPemilik=$_POST['IDPemilik'];
    $namaPemilik=$_POST['namaPemilik'];
    $tanggalLahir=$_POST['tanggalLahir'];
    $telepon=$_POST['telepon'];
    $fotoPemilik = $_POST['fotoPemilik'];

    $target_dir = "uploads/profilpemilik";

    if (!file_exists($target_dir)) {
        mkdir($target_dir, 0777, true);
    }

    $target_dir = $target_dir . "/" . $namaPemilik . "_" . time() . ".jpeg";
    if (file_put_contents($target_dir, base64_decode($fotoPemilik))) {
        $query=mysqli_query($connect,"UPDATE akunpemilikcost SET namaPemilik='".$namaPemilik', tanggalLahir='".$tanggalLahir', telepon='".$telepon', fotoPemilik='".$target_dir"'");
        $response=array();
        if($query){
            $response["success"]=1;
            $response["message"]="Data Berhasil diubah";
            echo json_encode($response);
        } else {
            $response["success"]=0;
            $response["message"]="Data Gagal diubah";
            echo json_encode($response);
        }
    } else {
        $response["success"]=-1;
        $response["message"]="data kosong";
        echo json_encode($response);
    }
}
```

#### Code 8. Code PHP Edit Profil Pemilik

Code diatas digunakan untuk mengedit data profil pemilik yang berada dalam database. Hal pertama yang dilakukan adalah melakukan koneksi dengan menggunakan fungsi yang telah dibuat sebelumnya pada file connect.php. Selanjutnya kita menggunakan fungsi isset untuk memeriksa apakah variabel yang digunakan sudah diatur atau belum, variabel yang digunakan disini adalah IDPemilik, namaPemilik, tanggalLahir, telepon, dan fotoPemilik. Fungsi isset adalah mengembalikan nilai false pada data yang masih kosong. Jika data sudah

diisi maka akan mengembalikan nilai true, selanjutnya kita menjadikan data menjadi variabel yaitu data IDPemilik pada variabel IDPemilik, namaPemilik pada variabel namaPemilik, tanggalLahir pada variabel tanggalLahir, telepon pada variabel telepon , dan fotoPemilik pada variabel fotoPemilik. \$target\_dir = "uploads/" menentukan direktori tempat file akan ditempatkan, kali ini adalah file profilPemilik. Base64 mendekode data ke dalam format yang dapat dibaca manusia. Skema pengkodean Base64 umumnya digunakan ketika ada kebutuhan untuk menyandikan data biner, terutama ketika data tersebut perlu disimpan dan ditransfer melalui media yang dirancang untuk menangani teks.

Variable q dengan awalan fungsi mysqli\_query sebagai sarana untuk menampung perintah query yang hendak kita masukkan ke database. Data profil pemilik disini sesuai dengan data yang terdapat dalam database sehingga data yang diedit akan selalu terupdate di database. Lalu hasil respon dari penginputan data akan ditampung dalam sebuah array dengan variabel response. Hasil dari respon ini dicetak dalam bentuk nilai respon 1,0, atau -1 ke dalam format json. Nilai 1 menyatakan bahwa data berhasil dihapus kemudian mencetak pesan “Data berhasil diubah”. Nilai 0 menyatakan bahwa data gagal dihapus kemudian mencetak pesan “Data gagal diubah”. Apabila nilai -1 menyatakan bahwa data kosong, atau belum ada data yang dimasukkan.

```
?php
header('content-type:application/json;charset=utf-8');
include "connect.php";

if (isset($_POST['IDPencari']) && isset($_POST['namaPencari']) && isset($_POST['tanggalLahir']) && isset($_POST['telepon']) && isset($_POST['fotoPencari'])) {
    $IDPencari=$_POST['IDPencari'];
    $namaPencari=$_POST['namaPencari'];
    $tanggalLahir=$_POST['tanggalLahir'];
    $telepon=$_POST['telepon'];
    $fotoPencari = $_POST["fotoPencari"];
}

$target_dir = "uploads/profilPencari";

if (!file_exists($target_dir)) {
    mkdir($target_dir, 0777, true);
}

$target_dir = $target_dir."/". $namaPencari.".".time()."jpeg";
if (file_put_contents($target_dir, base64_decode($fotoPencari))) {
    $query=mysqli_query($connect,"UPDATE akunpencarikost SET namaPencari='".$namaPencari', tanggalLahir='".$tanggalLahir', telepon='".$telepon', fotoPencari='".$fotoPencari'");
    if($query){
        $response["success"]=1;
        $response["message"]="Data Berhasil diubah";
        echo json_encode($response);
    } else {
        $response["success"]=0;
        $response["message"]="Data Gagal diubah";
        echo json_encode($response);
    }
} else {
    $response["success"]=-1;
    $response["message"]="data kosong";
    echo json_encode($response);
}
}
```

**Code 9. Code PHP Edit Profil Pencari**

Code diatas digunakan untuk mengedit data profil pencari yang berada dalam database. Hal pertama yang dilakukan adalah melakukan koneksi dengan menggunakan fungsi yang telah dibuat sebelumnya pada file connect.php. Selanjutnya kita menggunakan fungsi isset untuk memeriksa apakah variabel yang digunakan sudah diatur atau belum, variabel yang digunakan disini adalah IDPencari, namaPencari, tanggalLahir, telepon, dan fotoPencari. Fungsi isset adalah mengembalikan nilai false pada data yang masih kosong. Jika data sudah diisi maka akan mengembalikan nilai true, selanjutnya kita menjadikan data menjadi variabel

yaitu data IDPencari pada variabel IDPencari, namaPencari pada variabel namaPencari, tanggalLahir pada variabel tanggalLahir, telepon pada variabel telepon , dan fotoPencari pada variabel fotoPencari. \$target\_dir = "uploads/" menentukan direktori tempat file akan ditempatkan, kali ini adalah file profilPencari. Base64 mendekode data ke dalam format yang dapat dibaca manusia. Skema pengkodean Base64 umumnya digunakan ketika ada kebutuhan untuk menyandikan data biner, terutama ketika data tersebut perlu disimpan dan ditransfer melalui media yang dirancang untuk menangani teks.

Variable q dengan awalan fungsi mysqli\_query sebagai sarana untuk menampung perintah query yang hendak kita masukkan ke database. Data profil pemilik disini sesuai dengan data yang terdapat dalam database sehingga data yang diedit akan selalu terupdate di database. Lalu hasil respon dari penginputan data akan ditampung dalam sebuah array dengan variabel response. Hasil dari respon ini dicetak dalam bentuk nilai respon 1,0, atau -1 ke dalam format json. Nilai 1 menyatakan bahwa data berhasil dihapus kemudian mencetak pesan “Data berhasil diubah”. Nilai 0 menyatakan bahwa data gagal dihapus kemudian mencetak pesan “Data gagal diubah”. Apabila nilai -1 menyatakan bahwa data kosong, atau belum ada data yang dimasukkan.

```
<?php
header('Content-type:application/json;charset=utf-8');
include "connect.php";

if (isset($_POST['IDKost'])) {
    $IDKost = $_POST['IDKost'];

    $q = mysqli_query($connect, "UPDATE kost SET jumlahKamar = jumlahKamar + 1 WHERE IDKost='$IDKost'");
    $response = array();
    if ($q) {
        $response["success"] = 1;
        $response["message"] = "Data berhasil diupdate";
        echo json_encode($response);
    } else {
        $response["success"] = 0;
        $response["message"] = "Data gagal diupdate";
        echo json_encode($response);
    }
} else {
    $response["success"] = -1;
    $response["message"] = "Data kosong";
    echo json_encode($response);
}
```

#### **Code 10. Code PHP Menambahkan Kamar Pemilik Kost Ketika Penyewaan dihapus**

Code diatas digunakan untuk menambah kamar kost yang berada dalam database. Hal pertama yang dilakukan adalah melakukan koneksi dengan menggunakan fungsi yang telah dibuat sebelumnya pada file connect.php. Selanjutnya kita menggunakan fungsi isset untuk memeriksa apakah variabel yang digunakan sudah diatur atau belum, variabel yang digunakan disini adalah IDKost, dan jumlahKamar. Fungsi isset adalah mengembalikan nilai false pada data yang masih kosong. Jika data sudah diganti maka akan mengembalikan nilai true, selanjutnya kita menjadikan data menjadi variabel yaitu data IDKost pada variabel IDKost, jumlahKamar pada variabel jumlahKamar.

```

<?php
header('Content-type:application/json;charset=utf-8');
include "connect.php";

if (isset($_POST['IDSewa'])) {
    $IDSewa = $_POST['IDSeva'];
    $status="Booked";
    $q = mysqli_query($connect, "UPDATE penyewaan SET status = '$status' where IDSeva='$IDSewa'");
    $response = array();
    if ($q) {
        $response["success"] = 1;
        $response["message"] = "Data berhasil diupdate";
        echo json_encode($response);
    } else {
        $response["success"] = 0;
        $response["message"] = "Data gagal diupdate";
        echo json_encode($response);
    }
} else {
    $response["success"] = -1;
    $response["message"] = "Data kosong";
    echo json_encode($response);
}

```

**Code 11. Code PHP Edit Status Penyewaan Kost**

Code diatas digunakan untuk mengubah status penyewaan kost yang berada dalam database. Hal pertama yang dilakukan adalah melakukan koneksi dengan menggunakan fungsi yang telah dibuat sebelumnya pada file connect.php. Selanjutnya kita menggunakan fungsi isset untuk memeriksa apakah variabel yang digunakan sudah diatur atau belum, variabel yang digunakan disini adalah IDKost, dan IDSeva. Fungsi isset adalah mengembalikan nilai false pada data yang masih kosong. Jika data sudah diganti maka akan mengembalikan nilai true, selanjutnya kita menjadikan data menjadi variabel yaitu data IDKost pada variabel IDKost, IDSeva pada variabel IDSeva.

### Coding Proses Delete Data

```

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

```

**Code 12. Code PHP Delete Kost sebagai Pemilik**

Code diatas digunakan untuk menghapus kost yang berada dalam database. Hal pertama yang dilakukan adalah melakukan koneksi dengan menggunakan fungsi yang telah dibuat sebelumnya pada file connect.php. Selanjutnya kita menggunakan fungsi isset untuk memeriksa apakah variabel yang digunakan sudah diatur atau belum, ID yang digunakan disini adalah IDKost. Fungsi isset adalah mengembalikan nilai false pada data yang masih kosong. Jika data sudah diisi maka akan mengembalikan nilai true, selanjutnya kita menjadikan data menjadi variabel yaitu data IDKost pada variabel IDKost. Kemudian variabel connect digunakan untuk menghubungkan file php ini ke database. Hasil respon dari pendeletan data ini kemudian ditampung dalam sebuah array dengan variabel respon. Hasil dari respon ini dicetak dalam bentuk nilai respon 1,0, atau -1 ke dalam format json. Nilai 1 menyatakan bahwa data berhasil dihapus kemudian mencetak pesan “Data berhasil dihapus”. Nilai 0 menyatakan bahwa data gagal dihapus kemudian mencetak pesan “Data gagal dihapus”. Apabila nilai -1 menyatakan bahwa data kosong, atau belum ada data yang dimasukkan.

```

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

#### **Code 13. Code PHP Delete Booked Kost oleh Pemilik**

Code diatas digunakan untuk menghapus kost yang sudah di booked oleh pencari ke dalam database. Hal pertama yang dilakukan adalah melakukan koneksi dengan menggunakan fungsi yang telah dibuat sebelumnya pada file connect.php. Selanjutnya kita menggunakan fungsi isset untuk memeriksa apakah variabel yang digunakan sudah diatur atau belum, ID yang digunakan disini adalah IDsewa. Fungsi isset adalah mengembalikan nilai false pada data yang masih kosong. Jika data sudah diisi maka akan mengembalikan nilai true, selanjutnya kita menjadikan data menjadi variabel yaitu data IDsewa pada variabel IDsewa. Kemudian variabel connect digunakan untuk menghubungkan file php ini ke database. Hasil respon dari pendeletan data ini kemudian ditampung dalam sebuah array dengan variabel respon. Hasil dari respon ini dicetak dalam bentuk nilai respon 1,0, atau -1 ke dalam format json. Nilai 1 menyatakan bahwa data berhasil dihapus kemudian mencetak pesan “Data

berhasil dihapus". Nilai 0 menyatakan bahwa data gagal dihapus kemudian mencetak pesan "Data gagal dihapus". Apabila nilai -1 menyatakan bahwa data kosong, atau belum ada data yang dimasukkan.

## Coding Proses Select Data

```
1 <?php
2 session_start();
3 include 'connect.php';
4
5 $email = $_POST['email'];
6 $password = $_POST['password'];
7
8 if(isset($email) && ($password)){
9     $query = mysqli_query($connect,
10     "SELECT * FROM akunpemilikcost WHERE email='".$email' and password='".$password'");
11     $response=array();
12     if (mysqli_num_rows($query)>0){
13         $response["data"]=[];
14         while($data=mysqli_fetch_array($query)){
15             $akun=[];
16             $akun["IDPemilik"]=$data["IDPemilik"];
17             array_push($response["data"],$akun);
18         }
19         $response["success"]=1;
20         $response["message"]=$akun["IDPemilik"];
21         echo json_encode($response);
22     } else {
23         $response["success"]=-1;
24         $response["message"]="Data Kosong";
25         echo json_encode($response);
26     }
27 }
28
29 ?>
```

Code 14. Code PHP Login sebagai Pemilik Kost

Code diatas digunakan untuk proses login sebagai pemilik kost. Langkah pertama yang akan kita lakukan adalah memulai eksekusi session pada server kemudian menyimpannya pada browser menggunakan fungsi session\_start();. Lalu, membuka koneksi ke database dengan memanggil fungsi yang sudah dibuat pada file connect.php dengan menggunakan include, seperti pada baris ke 3 pada gambar diatas. Selanjutnya pada baris ke 5 dan ke 6, kita akan menangkap data yang dikirim dari form login, disini data email akan menjadi variabel email, dan data password menjadi variabel password. Kemudian dilakukan validasi untuk mengetahui apakah semua data yang diperlukan sudah masuk menggunakan fungsi isset pada baris ke 8. Kemudian dilakukan query select data ke dalam database yang ditampung pada variabel query dengan menggunakan fungsi mysqli\_query dan terlebih dahulu memanggil variabel connect untuk menghubungkan ke database seperti pada baris ke 9. Setelahnya, respon yang diberikan ditampung ke dalam array dengan variabel response. Jika terdapat data yang sesuai dengan query yang diminta (disini kita akan meminta email dan password), data akan ditampung pada array kolom data. Data yang ditampung hanya IDPemilik. Apabila respon pada kolom success bernilai 1 artinya data berhasil diambil dari database sehingga kolom IDPemilik dan status akan dicetak ke dalam bentuk JSON seperti pada baris 19 sampai 21. Apabila kolom success bernilai -1, yang berarti tidak ada data yang dapat diambil dari database pada proses yang dijalankan sehingga kolom message menampung pesan data kosong, kedua data ini akan dicetak ke dalam bentuk JSON seperti pada baris 25 sampai dengan 27.

```

1 <?php
2 session_start();
3 include 'connect.php';
4
5 $email = $_POST['email'];
6 $password = $_POST['password'];
7
8 if(isset($email) && ($password)){
9     $query = mysqli_query($connect,
10     "SELECT * FROM akunpencari_kost WHERE email='$email' and password='$password'");
11     $response=array();
12     if (mysqli_num_rows($query)>0){
13         $response["data"]=[];
14         while($data=mysqli_fetch_array($query)){
15             $akun=[];
16             $akun["IDPencari"]=$data["IDPencari"];
17             array_push($response["data"],$akun);
18         }
19         $response["success"]=1;
20         $response["message"]=$akun["IDPencari"];
21         echo json_encode($response);
22     }
23     else
24     {
25         $response["success"]=-1;
26         $response["message"]="Data Kosong";
27         echo json_encode($response);
28     }
29 }
30 ?>
```

### Code 15. Code PHP Login sebagai Pencari Kost

Code diatas digunakan untuk proses login sebagai pencari kost. Langkah pertama yang akan kita lakukan adalah memulai eksekusi session pada server kemudian menyimpannya pada browser menggunakan fungsi session\_start();. Lalu, membuka koneksi ke database dengan memanggil fungsi yang sudah dibuat pada file connect.php dengan menggunakan include, seperti pada baris ke 3 pada gambar diatas. Selanjutnya pada baris ke 5 dan ke 6, kita akan menangkap data yang dikirim dari form login, disini data email akan menjadi variabel email, dan data password menjadi variabel password. Kemudian dilakukan validasi untuk mengetahui apakah semua data yang diperlukan sudah masuk menggunakan fungsi isset pada baris ke 8. Kemudian dilakukan query select data ke dalam database yang ditampung pada variabel query dengan menggunakan fungsi mysqli\_query dan terlebih dahulu memanggil variabel connect untuk menghubungkan ke database seperti pada baris ke 9. Setelahnya, respon yang diberikan ditampung ke dalam array dengan variabel response. Jika terdapat data yang sesuai dengan query yang diminta (disini kita akan meminta email dan password), data akan ditampung pada array kolom data. Data yang ditampung hanya IDPencari. Apabila respon pada kolom success bernilai 1 artinya data berhasil diambil dari database sehingga kolom IDPencari dan status akan dicetak ke dalam bentuk JSON seperti pada baris 19 sampai 21. Apabila kolom success bernilai -1, yang berarti tidak ada data yang dapat diambil dari database pada proses yang dijalankan sehingga kolom message menampung pesan data kosong, kedua data ini akan dicetak ke dalam bentuk JSON seperti pada baris 25 sampai dengan 27.

```

getbooked-pemilik.php > ...
1   <?php
2   header('content-type:application/json;charset=utf-8');
3   include "connect.php";
4
5   if(isset($_POST['IDPemilik'])){
6       $IDPemilik=$_POST['IDPemilik'];
7       $query=mysqli_query($connect,"SELECT penyewaan.*,
8       kost.namakost,
9       akunpencarikost.namaPencari,
10      akunpencarikost.telepon
11      FROM penyewaan
12      JOIN kost
13      ON penyewaan.IDKost = kost.IDKost
14      JOIN akunpencarikost
15      ON penyewaan.IDPencari = akunpencarikost.IDPencari
16      WHERE IDPemilik = '$IDPemilik'");
17      $response=array();
18      if (mysqli_num_rows($query)>0){
19          $response["data"]=[];
20          while($data=mysqli_fetch_array($query)){
21              $booked=array();
22              $booked["IDKost"]=$data["IDKost"];
23              $booked["IDSewa"]=$data["IDSewa"];
24              $booked["namakost"]=$data["namakost"];
25              $booked["namaPencari"]=$data["namaPencari"];
26              $booked["telepon"]=$data["telepon"];
27              $booked["status"]=$data["status"];
28              array_push($response["data"],$booked);
29          }
30          $response["success"]=1;
31          $response["IDKost"]=$booked["IDKost"];
32          $response["IDSewa"]=$booked["IDSewa"];
33          $response["namakost"]=$booked["namakost"];
34          $response["namaPencari"]=$booked["namaPencari"];
35          $response["telepon"]=$booked["telepon"];
36          $response["status"]=$booked["status"];
37          echo json_encode($response);
38      }  

39      else
40      {
41          $response["success"]=-1;
42          $response["message"]="Data Kosong";
43          echo json_encode($response);
44      }
}

```

**Code 16. Code PHP Select Data Booked Kost Pemilik Kost**

Code diatas digunakan untuk menampilkan data kost pemilik yang telah dibook oleh pemilik kost ke dalam database. Untuk melakukan koneksi ke database dipanggil fungsi yang sudah dibuat pada file connect.php dengan menggunakan include seperti pada baris 3. Kemudian, dilakukan validasi untuk mengetahui apakah data yang diperlukan yaitu IDPemilik sudah dimasukkan dengan menggunakan fungsi isset seperti pada baris 5. Fungsi isset akan mengembalikan nilai true jika data sudah diisi dan nilai false jika data belum diisi. Jika data yang diperlukan sudah diisi, data dijadikan suatu variabel seperti pada baris 6, yaitu data IDPemilik pada variabel IDPemilik. Hal ini dilakukan untuk memudahkan dalam melakukan coding selanjutnya. Kemudian dilakukan query select data ke dalam database yang ditampung pada variabel query dengan menggunakan fungsi mysqli\_query dan terlebih dahulu memanggil variabel connect untuk menghubungkan ke database seperti pada baris 7. Kemudian respon yang diberikan ditampung ke dalam array dengan variabel response. Jika terdapat data yang sesuai dengan query yang diminta, data akan ditampung pada array kolom data. Data yang ditampung adalah data IDKost, IDSewa, namakost, namaPencari, telepon, dan status. Untuk menampung semua data digunakan perulangan seperti pada baris 20 sampai 29. Apabila respon pada kolom success bernilai 1 artinya data berhasil diambil dari database sehingga kolom IDKost, IDSewa, namakost, namaPencari, telepon, dan status yang akan dicetak ke dalam bentuk JSON seperti pada baris 30 sampai 37. Apabila kolom success bernilai -1 artinya tidak ada data yang dapat diambil dari database pada proses yang dijalankan

sehingga kolom message menampung pesan data kosong, kedua data ini akan dicetak ke dalam bentuk JSON seperti pada baris 41 sampai 43.

```
getbooked-pencari.php > ...
1  <?php
2      header('content-type:application/json;charset=utf-8');
3      include "connect.php";
4
5      if(isset($_POST['IDPencari'])){
6          $IDPencari=$_POST['IDPencari'];
7          $query=mysqli_query($connect,"SELECT penyewaan.*,
8          kost.namaKost,
9          kost.alamatKost
10         FROM penyewaan
11        JOIN kost
12       ON penyewaan.IDKost = kost.IDKost
13      WHERE IDPencari = '$IDPencari'");
14      $response=array();
15      if (mysqli_num_rows($query)>0){
16          $response["data"]=[];
17          while($data=mysqli_fetch_array($query)){
18              $booked=[];
19              $booked["namaKost"]=$data["namaKost"];
20              $booked["alamatKost"]=$data["alamatKost"];
21              array_push($response["data"],$booked);
22          }
23          $response["success"]=1;
24          $response["namaKost"]=$booked["namaKost"];
25          $response["alamatKost"]=$booked["alamatKost"];
26          echo json_encode($response);
27      }
28      else
29      {
30          $response["success"]=-1;
31          $response["message"]="Data Kosong";
32          echo json_encode($response);
33      }
34  }
35 }
```

Code 17. Code PHP Select Data Booked Kost Pencari Kost

Code diatas digunakan untuk menampilkan data kost yang telah dibook oleh pencari kost. Untuk melakukan koneksi ke database dipanggil fungsi yang sudah dibuat pada file connect.php dengan menggunakan include seperti pada baris 3. Kemudian, dilakukan validasi untuk mengetahui apakah data yang diperlukan yaitu IDPencari sudah dimasukkan dengan menggunakan fungsi isset seperti pada baris 5. Fungsi isset akan mengembalikan nilai true jika data sudah diisi dan nilai false jika data belum diisi. Jika data yang diperlukan sudah diisi, data dijadikan suatu variabel seperti pada baris 6, yaitu data IDPencari pada variabel IDPencari . Hal ini dilakukan untuk memudahkan dalam melakukan coding selanjutnya. Kemudian dilakukan query select data ke dalam database yang ditampung pada variabel query dengan menggunakan fungsi mysqli\_query dan terlebih dahulu memanggil variabel connect untuk menghubungkan ke database seperti pada baris 7. Kemudian respon yang diberikan ditampung ke dalam array dengan variabel response. Jika terdapat data yang sesuai dengan query yang diminta, data akan ditampung pada array kolom data. Data yang ditampung adalah data IDKost, IDSewa, namaKost, namaPencari, telepon, dan status. Untuk menampung semua data digunakan perulangan seperti pada baris 17 sampai 22. Apabila respon pada kolom success bernilai 1 artinya data berhasil diambil dari database sehingga kolom namaKost dan alamatKost yang akan dicetak ke dalam bentuk JSON seperti pada baris 23 sampai 26. Apabila

kolom success bernilai -1 artinya tidak ada data yang dapat diambil dari database pada proses yang dijalankan sehingga kolom message menampung pesan data kosong, kedua data ini akan dicetak ke dalam bentuk JSON seperti pada baris 30 sampai 32.

```
getkostbykota-pencari.php > ...
1  ?php
2      header('content-type:application/json;charset=utf-8');
3      include "connect.php";
4
5      if(isset($_POST['kotaKost'])){
6          $kotaKost=$_POST['kotaKost'];
7          $query=mysqli_query($connect,"SELECT * FROM kost WHERE kotaKost = '$kotaKost'");
8          $response=array();
9          if (mysqli_num_rows($query)>0){
10              $response["data"]=[];
11              while($data=mysqli_fetch_array($query)){
12                  $kost=[];
13                  $kost["IDKost"]=$data["IDKost"];
14                  $kost["namakost"]=$data["namakost"];
15                  $kost["alamatKost"]=$data["alamatKost"];
16                  $kost["kotaKost"]=$data["kotaKost"];
17                  $kost["jenisKost"]=$data["jenisKost"];
18                  $kost["jumlahKamar"]=$data["jumlahKamar"];
19                  $kost["hargaKost"]=$data["hargaKost"];
20                  $kost["deskripsiFasilitas"]=$data["deskripsiFasilitas"];
21                  $kost["IDPemilik"]=$data["IDPemilik"];
22                  array_push($response["data"],$kost);
23              }
24              $response["success"]=1;
25              $response["message"]="Data Berhasil diambil";
26              echo json_encode($response);
27          }
28      } else {
29          $response["success"]=-1;
30          $response["message"]="Data Kosong";
31          echo json_encode($response);
32      }
33  }
34
35
```

Code 18. Code PHP Select Data Kost berdasarkan Kota

Code diatas digunakan untuk menampilkan data kost berdasarkan nama kota yang diisi oleh pencari kost. Untuk melakukan koneksi ke database dipanggil fungsi yang sudah dibuat pada file connect.php dengan menggunakan include seperti pada baris 3. Kemudian, dilakukan validasi untuk mengetahui apakah data yang diperlukan yaitu kotaKost sudah dimasukkan dengan menggunakan fungsi isset seperti pada baris 5. Fungsi isset akan mengembalikan nilai true jika data sudah diisi dan nilai false jika data belum diisi. Jika data yang diperlukan sudah diisi, data dijadikan suatu variabel seperti pada baris 6, yaitu data kotaKost pada variabel kotaKost . Hal ini dilakukan untuk memudahkan dalam melakukan coding selanjutnya. Kemudian dilakukan query select data ke dalam database yang ditampung pada variabel query dengan menggunakan fungsi mysqli\_query dan terlebih dahulu memanggil variabel connect untuk menghubungkan ke database seperti pada baris 7. Kemudian respon yang diberikan ditampung ke dalam array dengan variabel response. Jika terdapat data yang sesuai dengan query yang diminta, data akan ditampung pada array kolom data. Data yang ditampung adalah data IDKost, namakost, alamatKost, kotaKost, jenisKost, jumlahKamar, hargaKost, deskripsiFasilitas dan IDPemilik. Untuk menampung semua data digunakan perulangan seperti pada baris 11 sampai 23. Apabila respon pada kolom success bernilai 1 artinya data berhasil diambil dari database sehingga kolom message yang akan dicetak ke dalam bentuk JSON seperti pada baris 24 sampai 26. Apabila kolom success bernilai -1 artinya tidak ada data yang dapat diambil dari database pada proses yang dijalankan sehingga kolom message

menampung pesan data kosong, kedua data ini akan dicetak ke dalam bentuk JSON seperti pada baris 30 sampai 32.

```
getmykost-pemilik.php > ...
1  <?php
2  header('content-type:application/json;charset=utf-8');
3  include "connect.php";
4
5  if(isset($_POST['IDPemilik'])){
6      $IDPemilik=$_POST['IDPemilik'];
7      $query=mysqli_query($connect,"SELECT * FROM kost WHERE IDPemilik = $IDPemilik");
8      $response=array();
9      if (mysqli_num_rows($query)>0){
10         $response["data"]=[];
11         while($data=mysqli_fetch_array($query)){
12             $kost=[];
13             $kost["IDKost"]=$data["IDKost"];
14             $kost["namakost"]=$data["namakost"];
15             $kost["alamatkost"]=$data["alamatkost"];
16             $kost["kotakost"]=$data["kotakost"];
17             $kost["jeniskost"]=$data["jeniskost"];
18             $kost["jumlahkamar"]=$data["jumlahkamar"];
19             $kost["hargaKost"]=$data["hargaKost"];
20             $kost["deskripsiFasilitas"]=$data["deskripsiFasilitas"];
21             $kost["IDPemilik"]=$data["IDPemilik"];
22             $kost["fotoKost"]=$data["fotoKost"];
23             array_push($response["data"],$kost);
24         }
25         $response["success"]=1;
26         $response["message"]="Data Berhasil diambil";
27         echo json_encode($response);
28     }
29     else
30     {
31         $response["success"]=-1;
32         $response["message"]="Data Kosong";
33     }
34 }
35 ?>
```

**Code 19. Code PHP Select Data Kost Pemilik**

Code diatas digunakan untuk menampilkan data kost yang telah ditambahkan oleh pemilik kost ke dalam database. Untuk melakukan koneksi ke database dipanggil fungsi yang sudah dibuat pada file connect.php dengan menggunakan include seperti pada baris 3. Kemudian, dilakukan validasi untuk mengetahui apakah data yang diperlukan yaitu IDPemilik sudah dimasukkan dengan menggunakan fungsi isset seperti pada baris 5. Fungsi isset akan mengembalikan nilai true jika data sudah diisi dan nilai false jika data belum diisi. Jika data yang diperlukan sudah diisi, data dijadikan suatu variabel seperti pada baris 6, yaitu data IDPemilik pada variabel IDPemilik. Hal ini dilakukan untuk memudahkan dalam melakukan coding selanjutnya. Kemudian dilakukan query select data ke dalam database yang ditampung pada variabel query dengan menggunakan fungsi mysqli\_query dan terlebih dahulu memanggil variabel connect untuk menghubungkan ke database seperti pada baris 7. Kemudian respon yang diberikan ditampung ke dalam array dengan variabel response. Jika terdapat data yang sesuai dengan query yang diminta, data akan ditampung pada array kolom data. Data yang ditampung adalah data IDKost, namaKost, alamatKost, kotaKost, jenisKost, jumlahKamar, hargaKost, deskripsiFasilitas, IDPemilik, dan fotoKost. Untuk menampung semua data digunakan perulangan seperti pada baris 11 sampai 24. Apabila respon pada kolom success bernilai 1 artinya data berhasil diambil dari database sehingga kolom message menampung pesan data berhasil diambil, kedua nilai ini akan dicetak ke dalam bentuk JSON

seperti pada baris 25 sampai 27. Apabila kolom success bernilai -1 artinya tidak ada data yang dapat diambil dari database pada proses yang dijalankan sehingga kolom message menampung pesan data kosong, kedua data ini akan dicetak ke dalam bentuk JSON seperti pada baris 31 sampai 33.

```
<?php
header('content-type:application/json;charset=utf-8');
include "connect.php";

if(isset($_POST['IDPemilik'])){
    $IDPemilik=$_POST['IDPemilik'];
    $query=mysqli_query($connect,"SELECT * FROM akunpemilikcost WHERE IDPemilik = '$IDPemilik'");
    $response=array();
    if (mysql_num_rows($query)>0){
        $response["data"]=[];
        while($data=mysqli_fetch_array($query)){
            $akunpemilikcost=array();
            $akunpemilikcost["IDPemilik"]=$data["IDPemilik"];
            $akunpemilikcost["namaPemilik"]=$data["namaPemilik"];
            $akunpemilikcost["tanggalLahir"]=$data["tanggalLahir"];
            $akunpemilikcost["telepon"]=$data["telepon"];
            $akunpemilikcost["email"]=$data["email"];
            $akunpemilikcost["password"]=$data["password"];
            $akunpemilikcost["fotoPemilik"]=$data["fotoPemilik"];
            array_push($response["data"],$akunpemilikcost);
        }
        $response["success"]=1;
        $response["namaPemilik"]=$akunpemilikcost["namaPemilik"];
        $response["tanggalLahir"]=$akunpemilikcost["tanggalLahir"];
        $response["telepon"]=$akunpemilikcost["telepon"];
        $response["email"]=$akunpemilikcost["email"];
        $response["fotoPemilik"]=$akunpemilikcost["fotoPemilik"];
        echo json_encode($response);
    }
    else
    {
        $response["success"]=-1;
        $response["message"]="Data Kosong";
        echo json_encode($response);
    }
}
?>
```

**Code 20. Code PHP Select Data Profil Pemilik Kost**

Code diatas digunakan untuk mengisi data profil pemilik kost yang nantinya akan dimasukkan kedalam database. Hal pertama yang dilakukan adalah melakukan koneksi dengan menggunakan fungsi yang telah dibuat sebelumnya pada file connect.php. Selanjutnya jika data sudah diisi maka akan memberikan respon “success”, selanjutnya kita menjadikan data menjadi variabel yaitu data IDPemilik pada variabel IDPemilik, namaPemilik pada variabel namaPemilik, tanggalLahir pada variabel tanggalLahir, telepon pada variabel telepon , email pada variabel email, password pada variabel password, dan fotoPemilik pada variabel fotoPemilik.

```

<?php
header('content-type:application/json;charset=utf-8');
include "connect.php";

if(isset($_POST['IDPencari'])){
    $IDPencari=$_POST['IDPencari'];
    $query=mysqli_query($connect,"SELECT * FROM akunpencarikost WHERE IDPencari = '$IDPencari'");
    $response=array();
    if (mysqli_num_rows($query)>0){
        $response["data"]=array();
        while($data=mysqli_fetch_array($query)){
            $akunpencarikost=array();
            $akunpencarikost["IDPencari"]=$data["IDPencari"];
            $akunpencarikost["namaPencari"]=$data["namaPencari"];
            $akunpencarikost["tanggalLahir"]=$data["tanggalLahir"];
            $akunpencarikost["telepon"]=$data["telepon"];
            $akunpencarikost["email"]=$data["email"];
            $akunpencarikost["password"]=$data["password"];
            $akunpencarikost["fotoPencari"]=$data["fotoPencari"];
            array_push($response["data"],$akunpencarikost);
        }
        $response["success"]=1;
        $response["namaPencari"]=$akunpencarikost["namaPencari"];
        $response["tanggalLahir"]=$akunpencarikost["tanggalLahir"];
        $response["telepon"]=$akunpencarikost["telepon"];
        $response["email"]=$akunpencarikost["email"];
        $response["fotoPencari"]=$akunpencarikost["fotoPencari"];
        echo json_encode($response);
    }
    else
    {
        $response["success"]=-1;
        $response["message"]="Data Kosong";
        echo json_encode($response);
    }
}
?>

```

**Code 21. Code PHP Select Data Profil Pencari Kost**

Code diatas digunakan untuk mengisi data profil pemilik kost yang nantinya akan dimasukkan kedalam database. Hal pertama yang dilakukan adalah melakukan koneksi dengan menggunakan fungsi yang telah dibuat sebelumnya pada file connect.php. Selanjutnya jika data sudah diisi maka akan memberikan respon “success”, selanjutnya kita menjadikan data menjadi variabel yaitu data IDPencari pada variabel IDPencari, namaPencari pada variabel namaPencari, tanggalLahir pada variabel tanggalLahir, telepon pada variabel telepon, email pada variabel email, password pada variabel password, dan fotoPencari pada variabel fotoPencari.

## 4.2 Coding XML

### Coding XML pada Aplikasi Bagian Log In dan Sign Up

#### Coding XML Main Activity

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#FFFFFF"
    android:backgroundTintMode="screen"
    android:foregroundTint="#FFE8B8"
    tools:context=".MainActivity">
    <ImageView
        android:id="@+id/imageView"
        android:layout_width="209dp"
        android:layout_height="324dp"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true"
        android:layout_alignParentEnd="true"
        android:layout_marginStart="103dp"
        android:layout_marginTop="103dp"
        android:layout_marginEnd="99dp"
        android:src="@drawable/kosthost1" />
    <LinearLayout
        android:layout_width="332dp"
        android:layout_height="274dp"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginStart="38dp"
        android:layout_marginTop="437dp"
        android:layout_marginEnd="41dp"
        android:layout_marginBottom="19dp"
        android:gravity="center"
        android:orientation="vertical">
        <Button
            android:id="@+id/btnLogin"
            android:layout_width="100dp"
            android:layout_height="50dp"
            android:background="#FFFFFF"
            android:backgroundTint="#4657B5"
            android:text="Log in"
            android:textColor="@color/white"
            android:textAlignment="center"
            android:textSize="15dp" />
        <Button
            android:id="@+id/btnSignup"
            android:layout_marginTop="15dp"
            android:layout_width="100dp"
            android:layout_height="50dp"
            android:background="#FFFFFF"
            android:backgroundTint="#4657B5"
            android:text="Sign up"
            android:textColor="@color/white"
            android:textAlignment="center"
            android:textSize="15dp" />
    </LinearLayout>
</RelativeLayout>
```

Code 22. Code XML Main

Pada coding XML diatas adalah coding yang digunakan untuk tampilan awal ketika user membuka aplikasi KostHost. Pada halaman ini terdapat beberapa elemen yang digunakan seperti layout dan widget. Layout yang digunakan adalah relative layout dan linear layout. Relative layout membuat berbagai elemen penyusunan tampilan sehingga dapat dengan mudah diatur secara relatif terhadap layout tersebut. Linear layout digunakan untuk mengatur posisi berbagai elemen di dalamnya secara linear baik dalam bentuk vertikal maupun horizontal. Selain itu juga ada widget yang digunakan, yaitu image view dan button. Button berupa tombol yang dapat digunakan untuk memulai aktivitas lain. Disini digunakan button yang bertuliskan “Log In” dan dibawahnya terdapat button yang

bertuliskan “Sign Up”. Kedua button tersebut akan mengarahkan user ke dalam halaman Log In jika sudah memiliki account atau Sign Up jika belum memiliki account, sesuai button yang ditekan oleh user. Kemudian ada image view yang fungsinya untuk menampilkan foto di dalam aplikasi android, yaitu logo “KostHost”.

### Coding XML Log In Pemilik/Pencari

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#FFFFFF"
    android:backgroundTintMode="screen"
    android:foregroundTint="#FFEB3B"
    tools:context=".Login">
    <ImageView
        android:id="@+id/imageView"
        android:layout_width="209dp"
        android:layout_height="324dp"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true"
        android:layout_alignParentEnd="true"
        android:layout_marginStart="103dp"
        android:layout_marginTop="103dp"
        android:layout_marginEnd="99dp"
        android:src="@drawable/kosthost1" />
    <LinearLayout
        android:layout_width="332dp"
        android:layout_height="274dp"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginStart="38dp"
        android:layout_marginTop="437dp"
        android:layout_marginEnd="41dp"
        android:layout_marginBottom="19dp"
        android:gravity="center"
        android:orientation="vertical">
        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Masuk Sebagai:"
            android:layout_marginBottom="20dp"
            android:textAlignment="center"
            android:textSize="17dp"/>
        <Button
            android:id="@+id/btnLoginPencari"
            android:layout_width="200dp"
            android:layout_height="50dp"
            android:background="#FFFFFF"
            android:backgroundTint="#44657B5"
            android:fontFamily="sans-serif-medium"
            android:text="PENCARI KOST"
            android:textAlignment="center"
            android:textColor="@color/white"
            android:gravity="center"
            android:textSize="15dp" />
        <Button
            android:id="@+id/btnLoginPemilik"
            android:layout_marginTop="15dp"
            android:layout_width="200dp"
            android:layout_height="50dp"
            android:background="#FFFFFF"
            android:backgroundTint="#44657B5"
            android:text="PEMILIK KOST"
            android:fontFamily="sans-serif-medium"
            android:textAlignment="center"
            android:textColor="@color/white"
            android:gravity="center"
            android:textSize="15dp" />
    </LinearLayout>
</RelativeLayout>
```

Code 23. Code XML Log In sebagai Pemilik Kost atau Pencari Kost

Pada coding XML diatas adalah coding yang digunakan untuk tampilan Log In sebagai Pencari ataupun Login sebagai Pemilik. Pada halaman ini terdapat beberapa elemen yang digunakan seperti layout dan widget. Layout yang digunakan adalah relative layout dan linear layout. Relative layout membuat berbagai elemen penyusunan tampilan sehingga dapat dengan mudah diatur secara relatif terhadap layout tersebut. Linear layout digunakan untuk mengatur posisi berbagai elemen di dalamnya secara linear baik dalam bentuk vertikal maupun horizontal. Selain itu juga ada widget yang digunakan, yaitu text view, image view dan button. Text view digunakan untuk menampilkan suatu text di dalam aplikasi, disini text view berisi kalimat petunjuk untuk “masuk sebagai:” yaitu untuk login sebagai pencari maupun sebagai pemilik. Sedangkan button berupa tombol yang dapat digunakan untuk memulai aktivitas lain. Disini digunakan button yang bertuliskan “Pencari kost” dan dibawahnya terdapat button yang bertuliskan “Pemilik kost”. Kedua button tersebut akan mengarahkan user ke dalam halaman dashboard Login sebagai pemilik atau Login sebagai pencari, sesuai button yang ditekan oleh user. Kemudian ada image view yang fungsinya untuk menampilkan foto di dalam aplikasi android, yaitu logo “KostHost”.

### Coding XML Sign Up Pemilik/Pencari

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#FFFFFF"
    android:backgroundTintMode="screen"
    android:foregroundTint="#FFEB3B"
    tools:context=".Signup">
    <ImageView
        android:id="@+id/imageView"
        android:layout_width="209dp"
        android:layout_height="324dp"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true"
        android:layout_alignParentEnd="true"
        android:layout_marginStart="103dp"
        android:layout_marginTop="103dp"
        android:layout_marginEnd="99dp"
        android:src="@drawable/kosthost1" />
    <LinearLayout
        android:layout_width="332dp"
        android:layout_height="274dp"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginStart="38dp"
        android:layout_marginTop="437dp"
        android:layout_marginEnd="41dp"
        android:layout_marginBottom="19dp"
        android:gravity="center"
        android:orientation="vertical">
        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Daftar Sebagai:"
            android:layout_marginBottom="20dp"
            android:textAlignment="center"
            android:textSize="17dp"/>
```



```
<Button  
    android:id="@+id/btnSignupPencari"  
    android:layout_width="200dp"  
    android:layout_height="50dp"  
    android:background="#FFFFFF"  
    android:backgroundTint="#4657B5"  
    android:fontFamily="sans-serif-medium"  
    android:text="PENCARI KOST"  
    android:textAlignment="center"  
    android:textColor="@color/white"  
    android:gravity="center"  
    android:textSize="15dp" />  
<Button  
    android:id="@+id/btnSignupPemilik"  
    android:layout_marginTop="15dp"  
    android:layout_width="200dp"  
    android:layout_height="50dp"  
    android:background="#FFFFFF"  
    android:backgroundTint="#4657B5"  
    android:text="PEMILIK KOST"  
    android:fontFamily="sans-serif-medium"  
    android:textAlignment="center"  
    android:textColor="@color/white"  
    android:gravity="center"  
    android:textSize="15dp" />  
</LinearLayout>  
</RelativeLayout>
```

**Code 24. Code XML Sign Up sebagai Pemilik Kost atau Pencari Kost**

Pada coding XML diatas adalah coding yang digunakan untuk tampilan Sign Up sebagai Pencari ataupun Sign Up sebagai Pemilik. Pada halaman ini terdapat beberapa elemen yang digunakan seperti layout dan widget. Layout yang digunakan adalah Relative Layout dan Linear Layout. Relative layout membuat berbagai elemen penyusunan tampilan sehingga dapat dengan mudah diatur secara relatif terhadap layout tersebut. Linear layout digunakan untuk mengatur posisi berbagai elemen di dalamnya secara linear baik dalam bentuk vertikal maupun horizontal. Selain itu juga ada widget yang digunakan, yaitu Image View, Text View, dan Button. Image View digunakan untuk menampilkan gambar di dalam aplikasi android, pada halaman ini, gambar yang ditampilkan adalah logo kost host. Text View digunakan untuk menampilkan suatu text di dalam aplikasi, disini Text View berisi petunjuk “Daftar sebagai: ” yaitu untuk mendaftar sebagai pencari kost maupun sebagai pemilik kost. Sedangkan Button berupa tombol yang dapat digunakan untuk memulai aktivitas lain. Disini digunakan button yang bertuliskan “Pencari kost” dan dibawahnya terdapat Button yang bertuliskan “Pemilik kost”. Kedua button tersebut akan mengarahkan user ke dalam halaman Sign Up sebagai pencari kost atau Sign Up sebagai pemilik kost, sesuai button yang ditekan oleh pengguna.

## Coding XML Sign Up Pemilik

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#FFFFFF"
    android:backgroundTintMode="screen"
    android:foregroundTint="#FFE83B"
    tools:context=".SignupPemilik">
    <ImageView
        android:id="@+id/imageView"
        android:layout_width="wrap_content"
        android:layout_height="200dp"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true"
        android:layout_alignParentEnd="true"
        android:layout_marginStart="100dp"
        android:layout_marginTop="20dp"
        android:layout_marginEnd="101dp"
        android:src="@drawable/kosthost1" />
    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="481dp"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginStart="38dp"
        android:layout_marginTop="230dp"
        android:layout_marginEnd="41dp"
        android:layout_marginBottom="0dp"
        android:gravity="center"
        android:orientation="vertical"
        android:paddingHorizontal="30dp">
        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginBottom="5dp"
            android:text="Nama Lengkap:"
            android:textSize="17dp" />
        <EditText
            android:id="@+id/edtNamaPemilik"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:inputType="textPersonName"
            android:textSize="17dp" />
        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginBottom="5dp"
            android:text="Tanggal Lahir:"
            android:textSize="17dp" />
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal">
            <EditText
                android:id="@+id/edtTanggallahirPemilik"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:layout_weight="1"
                android:inputType="date"
                android:textSize="17dp" />
            <Button
                android:id="@+id/btnTimePicker"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:layout_weight="2"
                android:background="#4657B5"
                android:gravity="center"
                android:text="SELECT DATE"
                android:textColor="@color/white"
                android:textSize="10dp"
                android:backgroundTint="#4657B5"
                android:layout_toRightOf="@+id/edtTanggallahirPemilik"/>
        </LinearLayout>
    </LinearLayout>
</RelativeLayout>
```

```

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="5dp"
    android:text="No. Telepon:"
    android:textSize="17dp" />
<EditText
    android:id="@+id edtTeleponPemilik"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:inputType="phone"
    android:textSize="17dp" />
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="5dp"
    android:text="Email:"
    android:textSize="17dp" />
<EditText
    android:id="@+id edtEmailPemilik"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:inputType="textEmailAddress"
    android:textSize="17dp" />
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="5dp"
    android:text="Password:"
    android:textSize="17dp" />
<EditText
    android:id="@+id edtPasswordPemilik"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:inputType="textPassword"
    android:textSize="17dp" />
<Button
    android:id="@+id btnSignUp"
    android:layout_width="100dp"
    android:layout_height="50dp"
    android:layout_marginTop="15dp"
    android:background="#FFFFFF"
    android:backgroundTint="#4657B5"
    android:fontFamily="sans-serif-medium"
    android:gravity="center"
    android:text="SIGN UP"
    android:textAlignment="center"
    android:textColor="@color/white"
    android:textSize="15dp" />
</LinearLayout>
</RelativeLayout>

```

**Code 25. Code XML Sign Up Pemilik Kost**

Pada coding XML diatas adalah coding yang digunakan untuk tampilan Sign Up sebagai Pemilik Kost. Pada halaman ini terdapat beberapa elemen yang digunakan seperti layout dan widget. Layout yang digunakan adalah Relative Layout dan Linear Layout. Relative Layout membuat berbagai elemen penyusunan tampilan sehingga dapat dengan mudah diatur secara relatif terhadap layout tersebut. Linear Layout digunakan untuk mengatur posisi berbagai elemen di dalamnya secara linear baik dalam bentuk vertikal maupun horizontal. Selain itu juga ada widget yang digunakan, yaitu Image View, Text View, Edit Text, dan Button. Image View digunakan untuk menampilkan gambar pada aplikasi android, pada tampilan ini, gambar yang ditampilkan yaitu logo Kost Host. Text View digunakan untuk menampilkan suatu text di dalam aplikasi, disini Text View berisi petunjuk untuk memasukkan data seperti “Nama Lengkap: ”, “Tanggal Lahir: ”, “No. Telepon: ”, “Email: ”, dan “Password: ”. Edit Text digunakan untuk memasukkan data

berupa text pada aplikasi android, pada halaman ini data yang dimasukkan akan diproses untuk dimasukkan ke dalam database. Data tersebut adalah nama lengkap, tanggal lahir yang diperoleh dari memilih tanggal dari datetimepicker yang dapat diakses melalui Button Select Date, nomor telepon, email, dan password. Sedangkan Button adalah tombol yang dapat digunakan untuk memulai suatu aktivitas atau proses. Pada tampilan ini, digunakan Button yang bertuliskan “Sign Up” yang akan melakukan proses insert data ke dalam database tabel AkunPemilikKost dan mengarahkan user ke dalam halaman Log In Pemilik.

## Coding XML Sign Up Pencari

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#FFFFFF"
    android:backgroundTintMode="screen"
    android:foregroundTint="#FFEB3B"
    tools:context=".SignupPencari">
    <ImageView
        android:id="@+id/imageView"
        android:layout_width="wrap_content"
        android:layout_height="200dp"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true"
        android:layout_alignParentEnd="true"
        android:layout_marginStart="100dp"
        android:layout_marginTop="20dp"
        android:layout_marginEnd="101dp"
        android:src="@drawable/kosthost1" />
    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="50dp"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginStart="37dp"
        android:layout_marginTop="232dp"
        android:layout_marginEnd="41dp"
        android:layout_marginBottom="-2dp"
        android:gravity="center"
        android:orientation="vertical"
        android:paddingHorizontal="30dp">
        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginBottom="5dp"
            android:text="Nama Lengkap:"
            android:textSize="17dp" />
```

```
<EditText  
    android:id="@+id/edtNamaPencari"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:inputType="textPersonName"  
    android:textSize="17dp" />  
<TextView  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_marginBottom="5dp"  
    android:text="Tanggal Lahir:"  
    android:textSize="17dp" />  
<LinearLayout  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:orientation="horizontal">  
    <EditText  
        android:id="@+id/edtTanggallahirPencari"  
        android:layout_width="match_parent"  
        android:layout_height="match_parent"  
        android:layout_weight="1"  
        android:inputType="date"  
        android:textSize="17dp" />  
    <Button  
        android:id="@+id	btnTimePicker"  
        android:layout_width="match_parent"  
        android:layout_height="match_parent"  
        android:layout_weight="2"  
        android:background="#4657B5"  
        android:gravity="center"  
        android:text="SELECT DATE"  
        android:backgroundTint="#4657B5"  
        android:textColor="@color/white"  
        android:textSize="10dp"  
        android:layout_toRightOf="@+id/edtTanggallahirPencari"/>  
    </LinearLayout>  
    <TextView  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"  
        android:layout_marginBottom="5dp"  
        android:text="No. Telepon:"  
        android:textSize="17dp" />  
    <EditText  
        android:id="@+id/edtTeleponPencari"  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"  
        android:inputType="phone"  
        android:textSize="17dp" />  
    <TextView  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"  
        android:layout_marginBottom="5dp"  
        android:text="Email:"  
        android:textSize="17dp" />  
    <EditText  
        android:id="@+id/edtEmailPencari"  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"  
        android:inputType="textEmailAddress"  
        android:textSize="17dp" />  
    <TextView  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"  
        android:layout_marginBottom="5dp"  
        android:text="Password:"  
        android:textSize="17dp" />  
    <EditText  
        android:id="@+id/edtPasswordPencari"  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"  
        android:inputType="textPassword"  
        android:textSize="17dp" />
```

```
<Button  
    android:id="@+id/btnSignUp"  
    android:layout_width="100dp"  
    android:layout_height="50dp"  
    android:layout_marginTop="15dp"  
    android:background="#FFFFFF"  
    android:backgroundTint="#4657B5"  
    android:fontFamily="sans-serif-medium"  
    android:gravity="center"  
    android:text="SIGN UP"  
    android:textAlignment="center"  
    android:textColor="@color/white"  
    android:textSize="15dp" />  
</LinearLayout>  
</RelativeLayout>
```

Code 26. Code XML Sign Up Pencari Kost

Pada coding XML diatas adalah coding yang digunakan untuk tampilan Sign Up sebagai Pencari Kost. Pada halaman ini terdapat beberapa elemen yang digunakan seperti layout dan widget. Layout yang digunakan adalah relative layout dan linear layout. Relative layout membuat berbagai elemen penyusunan tampilan sehingga dapat dengan mudah diatur secara relatif terhadap layout tersebut. Linear layout digunakan untuk mengatur posisi berbagai elemen di dalamnya secara linear baik dalam bentuk vertikal maupun horizontal. Selain itu juga ada widget yang digunakan, yaitu Image View, Text View, Edit Text, dan Button. Image View digunakan untuk menampilkan gambar pada aplikasi android, pada tampilan ini, gambar yang ditampilkan yaitu logo Kost Host. Text view digunakan untuk menampilkan suatu text di dalam aplikasi, disini text view berisi petunjuk untuk memasukkan data seperti “Nama Lengkap: ”, “Tanggal Lahir: ”, “No. Telepon: ”, “Email: ”, dan “Password: ”. Edit Text digunakan untuk memasukkan data berupa text pada aplikasi android, pada halaman ini data yang dimasukkan akan diproses untuk dimasukkan ke dalam database. Data tersebut adalah nama lengkap, tanggal lahir yang diperoleh dari memilih tanggal dari datetimepicker yang dapat diakses melalui button Select Date, nomor telepon, email, dan password. Sedangkan Button adalah tombol yang dapat digunakan untuk memulai suatu aktivitas atau proses. Pada tampilan ini, digunakan button yang bertuliskan “Sign Up” yang akan melakukan proses insert data ke dalam database tabel AkunPencariKost dan mengarahkan user ke dalam halaman Log In Pencari.

## Coding XML Log In Pemilik

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#FFFFFF"
    android:backgroundTintMode="screen"
    android:foregroundTint="#FFE83B"
    tools:context=".LoginPemilik">
    <ImageView
        android:id="@+id/imageView"
        android:layout_width="209dp"
        android:layout_height="324dp"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true"
        android:layout_alignParentEnd="true"
        android:layout_marginStart="103dp"
        android:layout_marginTop="103dp"
        android:layout_marginEnd="99dp"
        android:src="@drawable/kosthost1" />
    <LinearLayout
        android:layout_width="332dp"
        android:layout_height="274dp"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginStart="38dp"
        android:layout_marginTop="40dp"
        android:layout_marginEnd="41dp"
        android:layout_marginBottom="19dp"
        android:gravity="center"
        android:orientation="vertical"
        android:paddingHorizontal="30dp">
        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Email:"
            android:layout_marginBottom="5dp"
            android:textSize="17dp"/>
        <EditText
            android:id="@+id/edtEmailPemilik"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:inputType="textEmailAddress"
            android:textSize="17dp"/>
        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Password:"
            android:layout_marginBottom="5dp"
            android:textSize="17dp"/>
        <EditText
            android:id="@+id/edtPasswordPemilik"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:inputType="textPassword"
            android:textSize="17dp"/>
        <Button
            android:id="@+id/btnLogin"
            android:layout_marginTop="15dp"
            android:layout_width="100dp"
            android:layout_height="50dp"
            android:background="#FFFFFF"
            android:backgroundTint="#4657B5"
            android:text="LOG IN"
            android:fontFamily="sans-serif-medium"
            android:textAlignment="center"
            android:textColor="@color/white"
            android:gravity="center"
            android:textSize="15dp" />
    </LinearLayout>
</RelativeLayout>
```

Code 27. Code XML Log In sebagai Pemilik Kost

Pada coding XML diatas adalah coding yang digunakan untuk tampilan Log In sebagai Pemilik. Pada halaman ini terdapat beberapa elemen yang digunakan seperti layout dan widget. Layout yang digunakan adalah relative layout dan linear layout. Relative layout

membuat berbagai elemen penyusunan tampilan sehingga dapat dengan mudah diatur secara relatif terhadap layout tersebut. Linear layout digunakan untuk mengatur posisi berbagai elemen di dalamnya secara linear baik dalam bentuk vertikal maupun horizontal. Selain itu juga ada widget yang digunakan, yaitu edit text, text view dan image view. Edit Text merupakan sebuah field untuk memasukkan text di aplikasi android, text yang dimasukkan didalam edit text akan memiliki value yang nantinya juga dapat di get untuk dimasukkan ke dalam database. Text view digunakan untuk menampilkan suatu text di dalam aplikasi, disini text view berisi kalimat petunjuk untuk memasukkan “Email: ” dan “Password: ”. Sedangkan button adalah tombol yang dapat digunakan untuk memulai aktivitas lain. Disini digunakan button yang bertuliskan “Login” yang akan mengarahkan user ke dalam halaman dashboard pemilik, jika email dan password yang dimasukkan benar dan akun sudah pernah dibuat sebelumnya. Jika email atau password salah, ataupun akun belum pernah dibuat sebelumnya maka user tidak dapat masuk ke dashboard pemilik. Kemudian ada image view yang fungsinya untuk menampilkan foto di dalam aplikasi android, yaitu logo “KostHost”.

### Coding XML Log In Pencari

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#FFFFFF"
    android:backgroundTintMode="screen"
    android:foregroundTint="#FEEB3B"
    tools:context=".LoginPencari">
    <ImageView
        android:id="@+id/imageView"
        android:layout_width="209dp"
        android:layout_height="324dp"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true"
        android:layout_alignParentEnd="true"
        android:layout_marginStart="103dp"
        android:layout_marginTop="103dp"
        android:layout_marginEnd="99dp"
        android:src="@drawable/kosthost1" />
    <LinearLayout
        android:layout_width="332dp"
        android:layout_height="274dp"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:background="#D9E1F2"
        android:orientation="vertical"
        android:padding="10dp">
        <EditText
            android:id="@+id/editText"
            android:layout_width="300dp"
            android:layout_height="40dp"
            android:layout_gravity="center"
            android:layout_marginBottom="10dp"
            android:background="#FFFFFF"
            android:fontFamily="sans-serif"
            android:hint="Email"
            android:imeOptions="actionNext" />
        <EditText
            android:id="@+id/editText2"
            android:layout_width="300dp"
            android:layout_height="40dp"
            android:layout_gravity="center"
            android:layout_marginBottom="10dp"
            android:background="#FFFFFF"
            android:fontFamily="sans-serif"
            android:hint="Password" />
        <Button
            android:id="@+id/button"
            android:layout_width="150dp"
            android:layout_height="40dp"
            android:layout_gravity="center"
            android:background="#FEEB3B"
            android:fontFamily="sans-serif"
            android:text="Login" />
    

```

```

        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginStart="38dp"
        android:layout_marginTop="40dp"
        android:layout_marginEnd="41dp"
        android:layout_marginBottom="19dp"
        android:gravity="center"
        android:orientation="vertical"
        android:paddingHorizontal="30dp">
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Email:"
        android:layout_marginBottom="5dp"
        android:textSize="17dp"/>
    <EditText
        android:id="@+id/edtEmailPencari"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="textEmailAddress"
        android:textSize="17dp"/>
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Password:"
        android:layout_marginBottom="5dp"
        android:textSize="17dp"/>
    <EditText
        android:id="@+id/edtPasswordPencari"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="textPassword"
        android:textSize="17dp"/>
    <Button
        android:id="@+id/btnLogin"
        android:layout_marginTop="15dp"
        android:layout_width="100dp"
        android:layout_height="50dp"
        android:background="#FFFFFF"
        android:backgroundTint="#4657B5"
        android:text="LOG IN"
        android:fontFamily="sans-serif-medium"
        android:textAlignment="center"
        android:textColor="@color/white"
        android:gravity="center"
        android:textSize="15dp" />
</LinearLayout>
</RelativeLayout>

```

**Code 28. Code XML Log In sebagai Pencari Kost**

Pada coding XML diatas adalah coding yang digunakan untuk tampilan Log In sebagai Pencari. Pada halaman ini terdapat beberapa elemen yang digunakan seperti layout dan widget. Layout yang digunakan adalah relative layout dan linear layout. Relative layout membuat berbagai elemen penyusunan tampilan sehingga dapat dengan mudah diatur secara relatif terhadap layout tersebut. Linear layout digunakan untuk mengatur posisi berbagai elemen di dalamnya secara linear baik dalam bentuk vertikal maupun horizontal. Selain itu juga ada widget yang digunakan, yaitu edit text, text view dan image view. Edit Text merupakan sebuah field untuk memasukkan text di aplikasi android, text yang dimasukkan didalam edit text akan memiliki value yang nantinya juga dapat di get untuk dimasukkan ke dalam database. Text view digunakan untuk menampilkan suatu text di dalam aplikasi, disini text view berisi kalimat petunjuk untuk memasukkan “Email: ” dan “Password: ”. Sedangkan button adalah tombol yang dapat digunakan untuk memulai aktivitas lain. Disini digunakan button yang bertuliskan “Login” yang akan mengarahkan

user ke dalam halaman dashboard pencari, jika email dan password yang dimasukkan benar dan akun sudah pernah dibuat sebelumnya. Jika email atau password salah, ataupun akun belum pernah dibuat sebelumnya maka user tidak dapat masuk ke dashboard pencari. Kemudian ada image view yang fungsinya untuk menampilkan foto di dalam aplikasi android, yaitu logo “KostHost”.

## Coding XML pada Aplikasi Bagian Pemilik Kost

### Coding XML Dashboard

```
<?xml version="1.0" encoding="UTF-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/bghijau"
    tools:context=".DashboardPemilik">

    <LinearLayout
        android:orientation="vertical"
        android:layout_width="match_parent"
        android:layout_height="match_parent">
        <RelativeLayout
            android:layout_marginTop="32dp"
            android:layout_marginLeft="20dp"
            android:layout_marginRight="20dp"
            android:layout_width="match_parent"
            android:layout_height="wrap_content">

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginTop="17dp"
                android:textSize="24sp"
                android:textStyle="bold"
                android:textColor="#FFFFFF"
                android:text="PEMILIK KOST"/>

            <ImageButton
                android:id="@+id/btnKH"
                android:layout_alignParentRight="true"
                android:background="@drawable/logoputih"
                android:layout_width="62dp"
                android:layout_height="62dp"/>
        </RelativeLayout>
        <GridLayout
            android:columnCount="2"
            android:rowCount="3"
            android:alignmentMode="alignMargins"
            android:columnOrderPreserved="false"
            android:layout_width="match_parent"
            android:layout_height="wrap_content">
            <androidx.cardview.widget.CardView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_columnWeight="1"
                android:layout_rowWeight="1"
                app:cardElevation="6dp"
                app:cardCornerRadius="12dp"
                android:layout_margin="12dp">
```

```
<LinearLayout
    android:orientation="vertical"
    android:padding="16dp"
    android:gravity="center"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <ImageView
        android:src="@drawable/house"
        android:layout_width="80dp"
        android:layout_height="80dp"/>
    <Button
        android:id="@+id/btnKostSaya"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="12dp"
        android:backgroundTint="#00000000"
        style="?android:attr/borderlessButtonStyle"
        android:text="My Kost"
        android:textColor="#6f6f6f"
        android:textSize="18sp"/>
</LinearLayout>
<androidx.cardview.widget.CardView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_columnWeight="1"
    android:layout_rowWeight="1"
    app:cardElevation="6dp"
    app:cardCornerRadius="12dp"
    android:layout_margin="12dp">
    <LinearLayout
        android:orientation="vertical"
        android:padding="16dp"
        android:gravity="center"
        android:layout_width="match_parent"
        android:layout_height="match_parent">
        <ImageView
            android:src="@drawable/notepad"
            android:layout_width="80dp"
            android:layout_height="80dp"/>
        <Button
            android:id="@+id/btnBookedKost"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="12dp"
            android:backgroundTint="#00000000"
            style="?android:attr/borderlessButtonStyle"
            android:text="Booked"
            android:textColor="#6f6f6f"
            android:textSize="18sp"/>
    </LinearLayout>
</androidx.cardview.widget.CardView>
<androidx.cardview.widget.CardView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_columnWeight="1"
    android:layout_rowWeight="1"
    app:cardElevation="6dp"
    app:cardCornerRadius="12dp"
    android:layout_margin="12dp">
    <LinearLayout
        android:orientation="vertical"
        android:padding="16dp"
        android:gravity="center"
        android:layout_width="match_parent"
        android:layout_height="match_parent">
```

```
<ImageView  
    android:src="@drawable/shield"  
    android:layout_width="80dp"  
    android:layout_height="80dp"/>  
<Button  
    android:id="@+id(btnPenjagaKost"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_marginTop="12dp"  
    android:backgroundTint="#00FFFFFF"  
    style="?android:attr/borderlessButtonStyle"  
    android:text="Penjaga"  
    android:textColor="#6f6f6f"  
    android:textSize="18sp"/>  
</LinearLayout>  
</androidx.cardview.widget.CardView>  
<androidx.cardview.widget.CardView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_columnWeight="1"  
    android:layout_rowWeight="1"  
    app:cardElevation="6dp"  
    app:cardCornerRadius="12dp"  
    android:layout_margin="12dp">  
<LinearLayout  
    android:orientation="vertical"  
    android:padding="16dp"  
    android:gravity="center"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent">  
<ImageView  
    android:src="@drawable/signquestion"  
    android:layout_width="80dp"  
    android:layout_height="80dp"/>  
<Button  
    android:id="@+id	btnFAQPemilik"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_marginTop="12dp"  
    android:backgroundTint="#00FFFFFF"  
    style="?android:attr/borderlessButtonStyle"  
    android:text="FAQ"  
    android:textColor="#6f6f6f"  
    android:textSize="18sp"/>  
</LinearLayout>  
</androidx.cardview.widget.CardView>  
<androidx.cardview.widget.CardView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_columnWeight="1"  
    android:layout_rowWeight="1"  
    app:cardElevation="6dp"  
    app:cardCornerRadius="12dp"  
    android:layout_margin="12dp">  
<LinearLayout  
    android:orientation="vertical"  
    android:padding="16dp"  
    android:gravity="center"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent">  
<ImageView  
    android:src="@drawable/user"  
    android:layout_width="80dp"  
    android:layout_height="80dp"/>
```

```
        }
    }
}
<Button
    android:id="@+id/btnProfile"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="12dp"
    android:backgroundTint="#00FFFFFF"
    style="?android:attr/borderlessButtonStyle"
    android:text="Profile"
    android:textColor="#6f6f6f"
    android:textSize="18sp"/>
</LinearLayout>
</androidx.cardview.widget.CardView>
</GridLayout>
</LinearLayout>
</ScrollView>
```

**Code 29. Code XML Dashboard Pemilik Kost**

Pada coding XML yang digunakan untuk tampilan Dashboard terdapat beberapa elemen yang digunakan seperti layout dan widget. Selain itu terdapat scroll view untuk menampung berbagai komponen di dalamnya. Scroll view digunakan untuk membuat tampilan android dapat discroll ke atas atau ke bawah sehingga mampu menyimpan banyak komponen di dalamnya. Layout yang digunakan adalah relative layout, linear layout, dan grid layout. Relative layout bertujuan untuk membuat berbagai elemen penyusun tampilan dapat dengan mudah diatur posisinya sedemikian rupa secara relatif terhadap layout tersebut. Linear layout digunakan untuk mengatur posisi berbagai elemen di dalamnya secara linear baik dalam bentuk vertikal maupun horizontal. Grid layout bertujuan untuk membuat tampilan dengan komposisi baris dan kolom yang jumlahnya sudah ditentukan, seperti: `android:columnCount="2"` dan `android:rowCount="3"`.

Lalu widget yang digunakan adalah text view, image view, card view, dan button. Text view digunakan untuk menampilkan suatu text di dalam aplikasi sedangkan image view digunakan untuk menampilkan gambar di dalam aplikasi. Gambar yang digunakan pada dashboard diperoleh dari drawable. Card view digunakan untuk membuat tampilan seperti grid, namun dalam bentuk yang lebih menarik yaitu seperti kumpulan kartu dalam bentuk segi empat yang tersusun rapi. Button digunakan sebagai suatu konfirmasi yang jika ditekan akan menjalankan perintah tertentu atau mengarahkan ke activity baru.

## Coding XML My Kost

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/bghijaupendek"
    tools:context=".KostPemilik">
    <LinearLayout
        android:orientation="vertical"
        android:layout_width="match_parent"
        android:layout_height="match_parent">
        <RelativeLayout
            android:layout_marginTop="32dp"
            android:layout_marginLeft="20dp"
            android:layout_marginRight="20dp"
            android:layout_width="match_parent"
            android:layout_height="wrap_content">
            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginTop="17dp"
                android:textSize="24sp"
                android:textColor="#FFFFFF"
                android:text="MY KOST"
                android:textStyle="bold"/>
            <ImageButton
                android:id="@+id/btnKH"
                android:layout_alignParentRight="true"
                android:background="@drawable/logoputih"
                android:layout_width="62dp"
                android:layout_height="62dp"/>
        </RelativeLayout>
        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_marginTop="50dp"
            android:layout_marginLeft="15dp"
            android:layout_marginRight="15dp">
            <ListView
                android:id="@+id/listKost"
                android:layout_width="match_parent"
                android:layout_height="match_parent">
            </ListView>
            <ImageButton
                android:id="@+id/btnTambahKost"
                android:layout_width="50dp"
                android:layout_height="50dp"
                android:layout_alignParentRight="true"
                android:layout_alignParentBottom="true"
                android:layout_marginRight="10dp"
                android:layout_marginBottom="10dp"
                android:src="@drawable/add"/>
        </RelativeLayout>
    </LinearLayout>
</RelativeLayout>
```

Code 30. Code XML Kost Pemilik

Pada coding XML yang digunakan untuk tampilan My Kost terdapat beberapa elemen yang digunakan seperti layout dan widget. Selain itu terdapat scroll view untuk menampung berbagai komponen di dalamnya. Layout yang digunakan adalah linear layout dan relative layout. Linear layout digunakan untuk mengatur posisi berbagai elemen di dalamnya secara linear baik dalam bentuk vertikal maupun horizontal. Relative Layout merupakan layout yang berfungsi untuk mengatur tata letak komponen atau widget aplikasi android dengan cara relative tidak hanya vertikal atau horisontal saja seperti pada Linear Layout.

Kemudian widget yang digunakan adalah text view, list view, dan button. Text View digunakan untuk menampilkan teks dalam aplikasi, dan tampilan gambar digunakan untuk menampilkan gambar dalam aplikasi. ListView sering digunakan untuk menampilkan daftar. ImageButton biasanya digunakan untuk menampilkan gambar atau icon bukannya teks namun berfungsi sebagai button. Button di layout ini digunakan untuk konfirmasi, dan jika ditekan, perintah akan dijalankan.

### Coding XML Tambah Kost

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".TambahKost">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="16dp"
            android:text="TAMBAH KOST"
            android:textStyle="bold"
            android:textAlignment="center"
            android:textSize="20sp"
            android:layout_gravity="center"/>
        <ImageView
            android:id="@+id/imageKost"
            android:layout_width="200dp"
            android:layout_height="200dp"
            android:layout_gravity="center"
            android:layout_marginTop="10dp"
            app:srcCompat="@drawable/noimage" />
        <Button
            android:id="@+id/btnChoose"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Choose"
            android:layout_gravity="center"
            app:backgroundTint="#4657B5"/>
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="vertical"
            android:layout_marginTop="5dp"
            android:layout_marginBottom="10dp"
            android:layout_marginLeft="25dp"
            android:layout_marginRight="25dp">
            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:orientation="horizontal">
                <TextView
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:text="Nama Kost :"
                    android:textSize="20sp" />
                <EditText
                    android:id="@+id/edtNama"
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:ems="10"
                    android:inputType="textPersonName"
                    android:layout_marginLeft="10dp"/>
            </LinearLayout>
            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:orientation="horizontal"
                android:layout_marginTop="5dp">
                <TextView
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:text="Harga Kost :"
                    android:textSize="20sp" />
```

```
<EditText
    android:id="@+id/edtHarga"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:inputType="number"
    android:layout_marginLeft="10dp"/>
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:layout_marginTop="5dp">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Alamat Kost :"
        android:textSize="20sp" />
    <EditText
        android:id="@+id/edtAlamat"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:layout_marginLeft="10dp"/>
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:layout_marginTop="5dp">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Kota Kost :"
        android:textSize="20sp" />
    <Spinner
        android:id="@+id/spinKotaKost"
        android:layout_width="match_parent"
        android:layout_height="45dp"
        android:layout_marginLeft="10dp"
        android:layout_marginBottom="5dp"/>
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:layout_marginTop="5dp">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Jenis Kost :"
        android:textSize="20sp" />
    <Spinner
        android:id="@+id/spinJenisKost"
        android:layout_width="match_parent"
        android:layout_height="45dp"
        android:layout_marginLeft="10dp"
        android:layout_marginBottom="5dp"/>
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:layout_marginTop="5dp">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Jumlah Kamar :"
        android:textSize="20sp" />
```

```

<EditText
    android:id="@+id/edtJumlahKamarKost"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:inputType="number"
    android:layout_marginLeft="10dp"/>
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:layout_marginTop="5dp">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Fasilitas Kost :"
        android:textSize="20sp" />
    <EditText
        android:id="@+id/edtFasilitasKost"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="textLongMessage"
        android:layout_marginLeft="10dp"/>
</LinearLayout>
<EditText
    android:id="@+id/edtIDPemilikKost"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:visibility="gone"/>
<Button
    android:id="@+id/btnTambahKost"
    android:layout_width="290dp"
    android:layout_height="65dp"
    android:text="Simpan"
    android:layout_marginTop="5dp"
    android:layout_gravity="center"
    app:backgroundTint="#4657B5" />
</LinearLayout>
</LinearLayout>
</ScrollView>

```

**Code 31. Code XML Tambah Kost**

Pada coding XML diatas adalah coding yang digunakan untuk tampilan Tambah Kost. Pada halaman ini terdapat beberapa elemen yang digunakan seperti layout dan widget. Layout yang digunakan adalah Scroll View dan Linear Layout. Scroll View digunakan untuk membuat tampilan aplikasi dapat di scroll. Sedangkan, Linear Layout digunakan untuk mengatur posisi berbagai elemen di dalamnya secara linear baik dalam bentuk vertikal maupun horizontal. Selain itu juga ada widget yang digunakan, yaitu Text View, Image View, Button, Edit Text, dan Spinner. Text View digunakan untuk menampilkan suatu text di dalam aplikasi, disini Text View berisi tulisan judul halaman yaitu Tambah Kost, dan keterangan untuk data yang harus dimasukkan yaitu nama kost, harga kost, alamat kost, kota kost, jenis kost, jumlah kamar kost, dan fasilitas kost. Kemudian, Image View merupakan salah satu widget yang digunakan untuk menampilkan sebuah gambar. Pada tampilan ini Image View digunakan untuk menampilkan foto kost yang dipilih pemilik kost dari Gallery. Button merupakan tombol yang dapat digunakan untuk memulai suatu aktivitas atau proses. Pada tampilan ini, button ada 2, yaitu button choose untuk memilih gambar dari gallery dan button simpan yang akan digunakan untuk memasukkan data kost yang sudah diinput ke database. Edit Text digunakan untuk memasukkan data berupa text pada aplikasi android, disini edit text meminta pengguna untuk memasukkan data kost seperti nama kost, harga kost, alamat kost, jumlah kamar kost, dan fasilitas kost. Sedangkan Spinner merupakan widget yang dapat menampilkan beberapa pilihan dalam bentuk drop-down. Dengan menggunakan Spinner, pilihan data

yang dapat dimasukkan oleh pengguna menjadi terbatas namun lebih mudah karena pengguna dapat memasukkan data hanya dengan memilih pilihan yang sudah disediakan. Pada tampilan ini, Spinner digunakan dalam pemilihan Kota Kost dan Jenis Kost.

### Coding XML Edit Kost

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".EditKost">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="16dp"
            android:text="EDIT KOST"
            android:textStyle="bold"
            android:textAlignment="center"
            android:textSize="20sp"
            android:layout_gravity="center"/>
        <ImageView
            android:id="@+id/imageKost"
            android:layout_width="200dp"
            android:layout_height="200dp"
            android:layout_gravity="center"
            android:layout_marginTop="10dp"
            app:srcCompat="@drawable/noimage" />
        <Button
            android:id="@+id/btnChoose"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Choose"
            android:layout_gravity="center"
            app:backgroundTint="#4657B5"/>
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="vertical"
            android:layout_marginTop="5dp"
            android:layout_marginBottom="10dp"
            android:layout_marginLeft="25dp"
            android:layout_marginRight="25dp">
            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:orientation="horizontal">
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:layout_marginTop="5dp">
    <EditText
        android:id="@+id/edtNama"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="textPersonName"
        android:layout_marginLeft="10dp"/>
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:layout_marginTop="5dp">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Nama Kost :"
        android:textSize="20sp" />
    <EditText
        android:id="@+id/edtNama"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="textPersonName"
        android:layout_marginLeft="10dp"/>
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:layout_marginTop="5dp">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Harga Kost :"
        android:textSize="20sp" />
    <EditText
        android:id="@+id/edtHarga"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="number"
        android:layout_marginLeft="10dp"/>
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:layout_marginTop="5dp">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Alamat Kost :"
        android:textSize="20sp" />
    <EditText
        android:id="@+id/edtAlamat"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:layout_marginLeft="10dp"/>
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:layout_marginTop="5dp">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Kota Kost :"
        android:textSize="20sp" />
    <Spinner
        android:id="@+id/spinKotaKost"
        android:layout_width="match_parent"
        android:layout_height="45dp"
        android:layout_marginLeft="10dp"
        android:layout_marginBottom="5dp"/>
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:layout_marginTop="5dp">
```

```
<Textview
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Jenis Kost :"
    android:textSize="20sp" />
<Spinner
    android:id="@+id/spinJenisKost"
    android:layout_width="match_parent"
    android:layout_height="45dp"
    android:layout_marginLeft="10dp"
    android:layout_marginBottom="5dp"/>
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:layout_marginTop="5dp">
    <Textview
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Jumlah Kamar :"
        android:textSize="20sp" />
    <EditText
        android:id="@+id/edtJumlahKamarKost"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="number"
        android:layout_marginLeft="10dp"/>
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:layout_marginTop="5dp">
    <Textview
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Fasilitas Kost :"
        android:textSize="20sp" />
    <EditText
        android:id="@+id/edtFasilitasKost"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="textLongMessage"
        android:layout_marginLeft="10dp"/>
</LinearLayout>
```

```

        |         <Button
        |             android:id="@+id/btnEditKost"
        |             android:layout_width="290dp"
        |             android:layout_height="65dp"
        |             android:text="Simpan"
        |             android:layout_marginTop="5dp"
        |             android:layout_gravity="center"
        |             app:backgroundTint="#4657B5" />
        |         <Button
        |             android:id="@+id/btnDeleteKost"
        |             android:layout_width="290dp"
        |             android:layout_height="65dp"
        |             android:text="Delete"
        |             android:layout_marginTop="5dp"
        |             android:layout_gravity="center"
        |             app:backgroundTint="#DA1010" />
        |         <EditText
        |             android:id="@+id/edtIDKost"
        |             android:layout_width="wrap_content"
        |             android:layout_height="wrap_content"
        |             android:ems="10"
        |             android:visibility="gone"/>
        |     </LinearLayout>
        | </LinearLayout>
    </ScrollView>

```

**Code 32. Code XML Edit Kost**

Pada coding XML yang digunakan untuk tampilan Edit Kost terdapat beberapa elemen yang digunakan seperti layout dan widget. Selain itu terdapat scroll view untuk menampung berbagai komponen di dalamnya. Scroll view digunakan untuk membuat tampilan android dapat discroll ke atas atau ke bawah sehingga mampu menyimpan banyak komponen di dalamnya. Layout yang digunakan adalah linear layout. Linear layout digunakan untuk mengatur posisi berbagai elemen di dalamnya secara linear baik dalam bentuk vertikal maupun horizontal.

Lalu widget yang digunakan adalah text view, image view, edit text, spinner, dan button. Text view digunakan untuk menampilkan suatu text di dalam aplikasi sedangkan image view digunakan untuk menampilkan gambar di dalam aplikasi. Gambar yang digunakan pada dashboard diperoleh dari drawable. Edit text memiliki tujuan yang mirip dengan text view, namun edit text memiliki kemampuan dapat menerima input dari pengguna. Spinner digunakan untuk menampilkan beberapa opsi yang dapat dipilih pengguna. Pada Edit Kost, spinner digunakan untuk memilih kota dan jenis kost. Button digunakan sebagai suatu konfirmasi yang jika ditekan akan menjalankan perintah tertentu atau mengarahkan ke activity baru.

## Coding XML List Kost

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="15dp"
        android:orientation="vertical">

        <TextView
            android:id="@+id/txtKost"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Kost"
            android:textColor="#000000"
            android:textSize="20dp"
            android:textStyle="bold" />

        <TextView
            android:id="@+id/txtAlamat"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Alamat"
            android:textSize="20dp" />

        <TextView
            android:id="@+id/txtKota"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Kota"
            android:textSize="20dp" />

        <TextView
            android:id="@+id/txtJenis"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Jenis Kost"
            android:textSize="20dp" />

        <TextView
            android:id="@+id/txtJumlahKamar"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Jumlah Kamar Kost"
            android:textSize="20dp" />

        <TextView
            android:id="@+id/txtFasilitas"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Fasilitas Kost"
            android:textSize="20dp" />
```

```

        ]
        <TextView
            android:id="@+id/txtHarga"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Harga"
            android:textColor="#004aad"
            android:textSize="20dp"
            android:textStyle="bold" />

        ]
        <TextView
            android:id="@+id/txtIDKost"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:visibility="gone" />

        ]
        <TextView
            android:id="@+id/txtHargaAsli"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:visibility="gone" />

        ]
        <TextView
            android:id="@+id/txtfotoKost"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:visibility="gone" />
    
```

**Code 33. Code XML List Kost**

Pada coding XML yang digunakan untuk tampilan List Booked Pencari terdapat beberapa elemen yang digunakan seperti layout dan widget. Layout yang digunakan adalah linear layout dan relative layout. Linear layout digunakan untuk mengatur posisi berbagai elemen di dalamnya secara linear baik dalam bentuk vertikal maupun horizontal. Relative Layout merupakan layout yang berfungsi untuk mengatur tata letak komponen atau widget aplikasi android dengan cara relative tidak hanya vertikal atau horisontal saja seperti pada Linear Layout. Lalu widget yang digunakan adalah text view. Text View disini bertujuan menampilkan data yang sudah diinput seperti Nama Kost, Alamat, Kota, dan lain-lain.

## Coding XML Menu Booked

### Coding XML Booked

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/bghijaupendek"
    tools:context=".BookedPemilik">
    
```

```

        <LinearLayout
            android:orientation="vertical"
            android:layout_width="match_parent"
            android:layout_height="match_parent">
```

```

        <RelativeLayout
            android:layout_marginTop="32dp"
            android:layout_marginLeft="20dp"
            android:layout_marginRight="20dp"
            android:layout_width="match_parent"
            android:layout_height="wrap_content">
```

```

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginTop="17dp"
                android:text="BOOKED KOST"
                android:textColor="#FFFFFF"
                android:textSize="24sp"
                android:textStyle="bold" />
```

```

        <ImageButton
            android:id="@+id/btnKH"
            android:layout_alignParentRight="true"
            android:background="@drawable/logoputih"
            android:layout_width="62dp"
            android:layout_height="62dp"/>
    </RelativeLayout>
    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginTop="50dp"
        android:layout_marginLeft="15dp"
        android:layout_marginRight="15dp">
        <ListView
            android:id="@+id/listBooked"
            android:layout_width="match_parent"
            android:layout_height="match_parent">
        </ListView>
    </RelativeLayout>
</LinearLayout>
</RelativeLayout>

```

**Code 34. Code XML Booked Kost Pemilik Kost**

Pada coding XML yang digunakan untuk tampilan Booked terdapat beberapa elemen yang digunakan seperti layout dan widget. Layout yang digunakan adalah relative layout dan linear layout. Relative layout bertujuan untuk membuat berbagai elemen penyusun tampilan dapat dengan mudah diatur posisinya sedemikian rupa secara relatif terhadap layout tersebut. Linear layout digunakan untuk mengatur posisi berbagai elemen di dalamnya secara linear baik dalam bentuk vertikal maupun horizontal. Lalu widget yang digunakan adalah text view dan image button. Text view digunakan untuk menampilkan suatu text di dalam aplikasi sedangkan image button merupakan suatu gambar yang dapat digunakan pula sebagai tombol yang jika ditekan akan memberikan aksi tertentu seperti menjalankan suatu perintah atau membuka activity baru.

### Coding XML Detail Booked

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".DetailbookedPemilik">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="16dp"
            android:text="BOOKED"
            android:textStyle="bold"
            android:textAlignment="center"
            android:textSize="20sp"
            android:layout_gravity="center"/>
        <ImageView
            android:layout_width="156dp"
            android:layout_height="152dp"
            android:layout_gravity="center"
            android:layout_marginTop="10dp"
            app:srcCompat="@drawable/notepad" />
    </LinearLayout>
</RelativeLayout>

```

```
<LinearLayout  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:orientation="vertical"  
    android:layout_marginTop="5dp"  
    android:layout_marginBottom="10dp"  
    android:layout_marginLeft="25dp"  
    android:layout_marginRight="25dp">  
    <LinearLayout  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"  
        android:orientation="horizontal">  
            <TextView  
                android:layout_width="wrap_content"  
                android:layout_height="wrap_content"  
                android:text="Nama Kost :"  
                android:textSize="20sp" />  
            <TextView  
                android:id="@+id/txtNamaKost"  
                android:layout_width="match_parent"  
                android:layout_height="wrap_content"  
                android:textSize="20sp"  
                android:layout_marginLeft="10dp"/>  
        </LinearLayout>  
        <LinearLayout  
            android:layout_width="match_parent"  
            android:layout_height="wrap_content"  
            android:orientation="horizontal"  
            android:layout_marginTop="5dp">  
            <TextView  
                android:layout_width="wrap_content"  
                android:layout_height="wrap_content"  
                android:text="Nama Penyewa :"  
                android:textSize="20sp" />  
            <TextView  
                android:id="@+id/txtNamaPenyewa"  
                android:layout_width="match_parent"  
                android:layout_height="wrap_content"  
                android:textSize="20sp"  
                android:layout_marginLeft="10dp"/>  
        </LinearLayout>  
        <LinearLayout  
            android:layout_width="match_parent"  
            android:layout_height="wrap_content"  
            android:orientation="horizontal"  
            android:layout_marginTop="5dp">  
            <TextView  
                android:layout_width="wrap_content"  
                android:layout_height="wrap_content"  
                android:text="Kontak Penyewa :"  
                android:textSize="20sp" />  
            <TextView  
                android:id="@+id/txtKontakPenyewa"  
                android:layout_width="match_parent"  
                android:layout_height="wrap_content"  
                android:textSize="20sp"  
                android:layout_marginLeft="10dp"/>  
        </LinearLayout>  
    </LinearLayout>  
    <Button  
        android:id="@+id/btnConfirmBooked"  
        android:layout_width="290dp"  
        android:layout_height="65dp"  
        android:text="Konfirmasi"  
        android:layout_marginTop="5dp"  
        android:layout_gravity="center"  
        app:backgroundTint="#4657B5" />  
    <Button  
        android:id="@+id/btnDeleteBooked"  
        android:layout_width="290dp"  
        android:layout_height="65dp"  
        android:text="Delete"  
        android:layout_marginTop="5dp"  
        android:layout_gravity="center"  
        app:backgroundTint="#DA1010" />
```

```

        <TextView
            android:id="@+id/txtIDSewa"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:ems="10"
            android:visibility="gone"/>
    </LinearLayout>
</RelativeLayout>

```

**Code 35. Code XML Detail Booked Kost**

Pada coding XML yang digunakan untuk tampilan Detail Booked terdapat beberapa elemen yang digunakan seperti layout dan widget. Layout yang digunakan adalah relative layout dan linear layout. Relative layout bertujuan untuk membuat berbagai elemen penyusun tampilan dapat dengan mudah diatur posisinya sedemikian rupa secara relatif terhadap layout tersebut. Linear layout digunakan untuk mengatur posisi berbagai elemen di dalamnya secara linear baik dalam bentuk vertikal maupun horizontal. Lalu widget yang digunakan adalah text view, image view, dan button. Text view digunakan untuk menampilkan suatu text di dalam aplikasi sedangkan image view digunakan untuk menampilkan suatu gambar di dalam aplikasi. Button digunakan sebagai suatu konfirmasi yang jika ditekan akan menjalankan perintah tertentu atau mengarahkan ke activity baru.

### Coding XML List Booked Pemilik

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:layout_margin="10dp">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical">

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:orientation="horizontal">

                <TextView
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:text="Nama Kost: " />

```

```

<TextView
    android:id="@+id/txtNamaKost"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:gravity="left"
    android:text="Nama Kost" />
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Nama Penyewa: " />
<TextView
    android:id="@+id/txtNamaPencari"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:gravity="left"
    android:text="Nama Penyewa" />
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Kontak Penyewa: " />
<TextView
    android:id="@+id/txtTelepon"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:gravity="left"
    android:text="Kontak Penyewa" />
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Status: " />
<TextView
    android:id="@+id/txtStatus"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:gravity="left" />
</LinearLayout>
</LinearLayout>
<TextView
    android:id="@+id/txtIDSewa"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:visibility="gone" />
<TextView
    android:id="@+id/txtIDKost"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:visibility="gone" />
</RelativeLayout>

```

#### Code 36. Code XML List Booked Pemilik

Pada coding XML yang digunakan untuk tampilan List Booked Pemilik terdapat beberapa elemen yang digunakan seperti layout dan widget. Layout yang digunakan adalah linear layout dan relative layout. Linear layout digunakan untuk mengatur posisi berbagai elemen di dalamnya secara linear baik dalam bentuk vertikal maupun horizontal. Relative Layout merupakan layout yang berfungsi untuk mengatur tata letak komponen atau widget

aplikasi android dengan cara relative tidak hanya vertikal atau horisontal saja seperti pada Linear Layout. Lalu widget yang digunakan adalah text view. Text View disini bertujuan menampilkan data yang sudah diinput seperti Nama Penyewa, Kontak Penyewa, dan Status.

## Coding XML Menu Penjaga

### Coding XML Menu Penjaga



```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/bghijaupendek"
    tools:context=".Penjaga">
    <LinearLayout
        android:orientation="vertical"
        android:layout_width="match_parent"
        android:layout_height="match_parent">
        <RelativeLayout
            android:layout_marginTop="32dp"
            android:layout_marginLeft="20dp"
            android:layout_marginRight="20dp"
            android:layout_width="match_parent"
            android:layout_height="wrap_content">
            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginTop="17dp"
                android:textSize="24sp"
                android:textColor="#FFFFFF"
                android:text="PENJAGA KOST"
                android:textStyle="bold"/>
            <ImageButton
                android:id="@+id/btnKH"
                android:layout_alignParentRight="true"
                android:background="@drawable/logoputih"
                android:layout_width="62dp"
                android:layout_height="62dp"/>
        </RelativeLayout>
        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_marginTop="50dp"
            android:layout_marginLeft="15dp"
            android:layout_marginRight="15dp">
            <ListView
                android:id="@+android:id/list"
                android:layout_width="match_parent"
                android:layout_height="match_parent">
            </ListView>
            <ImageButton
                android:id="@+id/btnTambahPenjaga"
                android:layout_width="50dp"
                android:layout_height="50dp"
                android:layout_alignParentRight="true"
                android:layout_alignParentBottom="true"
                android:layout_marginRight="10dp"
                android:layout_marginBottom="10dp"
                android:src="@drawable/add"/>
        </RelativeLayout>
    </LinearLayout>
</RelativeLayout>
```

Code 37. Code XML Penjaga

Pada coding XML diatas adalah coding yang digunakan untuk tampilan list view Penjaga kost. Pada halaman ini terdapat beberapa elemen yang digunakan seperti layout dan widget. Layout yang digunakan adalah relative layout dan linear layout. Relative layout membuat berbagai elemen penyusunan tampilan sehingga dapat dengan mudah diatur secara relatif terhadap layout tersebut. Linear layout digunakan untuk mengatur posisi berbagai elemen di dalamnya secara linear baik dalam bentuk vertikal maupun horizontal.

Selain itu juga ada widget yang digunakan, yaitu text view, list view dan image button. Text view digunakan untuk menampilkan suatu text di dalam aplikasi, disini text view berisi tulisan judul untuk halaman, yaitu “Penjaga Kost”. List view adalah sebuah widget yang dipakai untuk menampilkan nilai-nilai dan data dalam bentuk list atau daftar. Pada halaman ini, list view akan menampilkan daftar penjaga kost yang sudah di tambahkan sebelumnya. Sedangkan image button adalah tombol yang dapat digunakan untuk memulai aktivitas lain, namun untuk gambar button ini bisa di custom sesuai keinginan kita, hanya tinggal memilih dari drawable atau bisa juga meng-import gambar external. Disini digunakan image button berupa tanda add atau tambah, sehingga ketika di klik akan diarahkan ke menu tambah penjaga. Kemudian ada image button dengan logo inisial “KH” sebagai representasi nama aplikasi KostHost yang bisa di klik dan akan mengarahkan user kembali ke dashboard pemilik kost.

## Coding XML Tambah Penjaga

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".TambahPenjaga">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="16dp"
            android:text="TAMBAH PENJAGA KOST"
            android:textStyle="bold"
            android:textAlignment="center"
            android:textSize="20sp"
            android:layout_gravity="center"/>
        <ImageView
            android:layout_width="156dp"
            android:layout_height="152dp"
            android:layout_gravity="center"
            android:layout_marginTop="10dp"
            app:srcCompat="@drawable/profile" />
        <LinearLayout
            android:paddingTop="50dp"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="vertical"
            android:layout_marginTop="5dp"
            android:layout_marginBottom="10dp"
            android:layout_marginLeft="25dp"
            android:layout_marginRight="25dp">
            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:orientation="horizontal">
```

```

<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal">
                <TextView
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:text="Nama Kost :"
                    android:textSize="20sp" />
                <EditText
                    android:id="@+id/edtNamaKost"
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:ems="10"
                    android:inputType="text"
                    android:layout_marginLeft="10dp"/>
            </LinearLayout>
            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:orientation="horizontal">
                    <TextView
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:text="Nama Penjaga :"
                        android:textSize="20sp" />
                    <EditText
                        android:id="@+id/edtNamaPenjaga"
                        android:layout_width="match_parent"
                        android:layout_height="wrap_content"
                        android:ems="10"
                        android:inputType="textPersonName"
                        android:layout_marginLeft="10dp"/>
                </LinearLayout>
                <LinearLayout
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:orientation="horizontal"
                    android:layout_marginTop="5dp">
                        <TextView
                            android:layout_width="wrap_content"
                            android:layout_height="wrap_content"
                            android:text="Telepon Penjaga :"
                            android:textSize="20sp" />
                        <EditText
                            android:id="@+id/edtTeleponPenjaga"
                            android:layout_width="match_parent"
                            android:layout_height="wrap_content"
                            android:ems="10"
                            android:inputType="number"
                            android:layout_marginLeft="10dp"/>
                    </LinearLayout>
                    <Button
                        android:id="@+id/btnSimpan"
                        android:layout_width="290dp"
                        android:layout_height="65dp"
                        android:text="Simpan"
                        android:textColor="#FFFFFF"
                        android:layout_marginTop="10dp"
                        android:layout_gravity="center"
                        app:backgroundTint="#465785" />
                </LinearLayout>
            </LinearLayout>
        </RelativeLayout>

```

**Code 38. Code XML Tambah Penjaga**

Pada coding XML diatas adalah coding yang digunakan untuk tampilan Tambah Penjaga. Pada halaman ini terdapat beberapa elemen yang digunakan seperti layout dan widget. Layout yang digunakan adalah Relative Layout dan Linear Layout. Relative layout bertujuan untuk membuat berbagai elemen penyusun tampilan dapat dengan mudah diatur posisinya sedemikian rupa secara relatif terhadap layout tersebut. Sedangkan, Linear Layout digunakan untuk mengatur posisi berbagai elemen di dalamnya secara linear baik dalam bentuk vertikal maupun horizontal. Selain itu juga ada widget yang digunakan, yaitu Text View, Image View, Edit Text, dan Button. Text View digunakan untuk menampilkan suatu text di dalam aplikasi, disini Text View berisi tulisan judul halaman yaitu Tambah Penjaga Kost, dan keterangan untuk data yang harus dimasukkan yaitu nama kost, nama penjaga, dan kontak penjaga. Kemudian, Image View merupakan salah satu widget yang digunakan untuk menampilkan sebuah gambar. Pada tampilan ini Image View digunakan untuk menampilkan gambar siluet orang untuk merepresentasikan penjaga kost. Edit Text

digunakan untuk memasukkan data berupa text pada aplikasi android, disini edit text meminta pengguna untuk memasukkan data kost seperti nama kost, nama penjaga, dan kontak penjaga. Button merupakan tombol yang dapat digunakan untuk memulai suatu aktivitas atau proses. Pada tampilan ini, button yang digunakan adalah button simpan untuk memasukkan data penjaga yang sudah diinput ke database local.

### Coding XML Dialog Edit/Delete pada Menu Penjaga

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:app="http://schemas.android.com/apk/res-auto">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:layout_margin="10dp">
        <Button
            android:id="@+id/btnEditPenjaga"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="Edit"
            android:textColor="@color/black"
            android:enabled="true"/>
        <Button
            android:id="@+id/btnDeletePenjaga"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="Delete"
            android:textColor="@color/black"
            android:enabled="true"/>
    </LinearLayout>
</RelativeLayout>
```

Code 39. Code XML Dialog Edit/Delete

Pada coding XML yang digunakan untuk tampilan Dialog Penjaga terdapat beberapa elemen yang digunakan seperti layout dan widget. Layout yang digunakan adalah linear layout dan relative layout. Linear layout digunakan untuk mengatur posisi berbagai elemen di dalamnya secara linear baik dalam bentuk vertikal maupun horizontal. Relative Layout merupakan layout yang berfungsi untuk mengatur tata letak komponen atau widget aplikasi android dengan cara relative tidak hanya vertikal atau horizontal saja seperti pada Linear Layout. Lalu widget yang digunakan adalah button. Terdapat dua Button di halaman ini yaitu Edit Penjaga dan Delete Penjaga. Edit Penjaga digunakan untuk mengedit data diri penjaga, sedangkan Delete Penjaga digunakan untuk menghapus data diri penjaga. Button pada layout ini digunakan sebagai suatu konfirmasi yang jika ditekan akan menjalankan perintah.

## Coding XML Edit Penjaga

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".EditPenjaga">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="16dp"
            android:text="EDIT PENJAGA KOST"
            android:textStyle="bold"
            android:textAlignment="center"
            android:textSize="20sp"
            android:layout_gravity="center"/>
        <ImageView
            android:layout_width="156dp"
            android:layout_height="152dp"
            android:layout_gravity="center"
            android:layout_marginTop="10dp"
            app:srcCompat="@drawable/profile" />
        <LinearLayout
            android:paddingTop="50dp"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="vertical"
            android:layout_marginTop="5dp"
            android:layout_marginBottom="10dp"
            android:layout_marginLeft="25dp"
            android:layout_marginRight="25dp">
            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:orientation="horizontal">
                <TextView
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:text="Nama Kost :"
                    android:textSize="20sp" />
                <EditText
                    android:id="@+id/editNamaKost"
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:ems="10"
                    android:inputType="text"
                    android:layout_marginLeft="10dp"/>
            </LinearLayout>
            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:orientation="horizontal">
                <TextView
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:text="Nama Penjaga :"
                    android:textSize="20sp" />
                <EditText
                    android:id="@+id/editNamaPenjaga"
                    android:layout_width="match_parent"
```

```

        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="textPersonName"
        android:layout_marginLeft="10dp"/>
    
```

```

</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:layout_marginTop="5dp">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Telepon Penjaga \:"/>
        android:textSize="20sp" />
    <EditText
        android:id="@+id/edtTeleponPenjaga"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="number"
        android:layout_marginLeft="10dp"/>
    
```

```

</LinearLayout>
<Button
    android:id="@+id/btnSimpan"
    android:layout_width="290dp"
    android:layout_height="65dp"
    android:text="Simpan"
    android:textColor="#FFFFFF"
    android:layout_marginTop="10dp"
    android:layout_gravity="center"
    app:backgroundTint="#465785" />

```

#### Code 40. Code XML Edit Penjaga

Pada coding XML yang digunakan untuk tampilan Edit Penjaga terdapat beberapa elemen yang digunakan seperti layout dan widget. Layout yang digunakan adalah linear layout dan relative layout. Linear layout digunakan untuk mengatur posisi berbagai elemen di dalamnya secara linear baik dalam bentuk vertikal maupun horizontal. Relative Layout merupakan layout yang berfungsi untuk mengatur tata letak komponen atau widget aplikasi android dengan cara relative tidak hanya vertikal atau horisontal saja seperti pada Linear Layout. Lalu widget yang digunakan adalah text view, image view, edit text, dan button. Text view digunakan untuk menampilkan suatu text di dalam aplikasi sedangkan image view digunakan untuk menampilkan gambar di dalam aplikasi. Gambar yang digunakan pada dashboard diperoleh dari drawable. Edit text memiliki tujuan yang mirip dengan text view, namun edit text memiliki kemampuan dapat menerima input dari pengguna. Button pada layout ini digunakan sebagai suatu konfirmasi yang jika ditekan akan menjalankan perintah.

## Coding XML Menu FAQ

### Coding XML FAQ Pemilik

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/bghijau"
    tools:context=".faqPemilik">

    <LinearLayout
        android:orientation="vertical"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <RelativeLayout
            android:layout_marginTop="32dp"
            android:layout_marginRight="20dp"
            android:layout_marginLeft="20dp"
            android:layout_width="match_parent"
            android:layout_height="wrap_content">
            <TextView
                android:text="FAQ Pemilik Kost"
                android:textSize="24sp"
                android:textStyle="bold"
                android:textColor="#FFFFFF"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"/>
        </RelativeLayout>

        <GridLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:alignmentMode="alignMargins"
            android:columnCount="1"
            android:columnOrderPreserved="false"
            android:rowCount="4">

            <androidx.cardview.widget.CardView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_rowWeight="1"
                android:layout_columnWeight="1"
                android:layout_margin="12dp"
                android:layout_marginTop="20dp"
                app:cardCornerRadius="12dp"
                app:cardElevation="6dp">

                <LinearLayout
                    android:layout_width="match_parent"
                    android:layout_height="match_parent"
                    android:gravity="center"
                    android:orientation="vertical"
                    android:padding="30dp">
```

```

        <TextView
            android:layout_width="300dp"
            android:layout_height="40dp"
            android:layout_marginTop="1dp"
            android:text="Bagaimana cara mendaftarkan data kos baru?"
            android:textColor="#1B1313"
            android:textSize="16sp"
            android:textStyle="bold" />

        <TextView
            android:layout_width="300dp"
            android:layout_height="wrap_content"
            android:layout_marginTop="1dp"
            android:text="Untuk mendaftarkan kos, Anda akan diminta mengisi informasi kos"
            android:textColor="#1B1313"
            android:textSize="16sp" />
    </LinearLayout>
</androidx.cardview.widget.CardView>

<androidx.cardview.widget.CardView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_rowWeight="1"
    android:layout_columnWeight="1"
    android:layout_margin="12dp"
    app:cardCornerRadius="12dp"
    app:cardElevation="6dp">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:gravity="center"
        android:orientation="vertical"
        android:padding="30dp">

        <TextView
            android:layout_width="300dp"
            android:layout_height="40dp"
            android:layout_marginTop="1dp"
            android:text="Bagaimana cara melakukan update kos?"
            android:textColor="#1B1313"
            android:textSize="16sp"
            android:textStyle="bold" />

        <TextView
            android:layout_width="300dp"
            android:layout_height="wrap_content"
            android:layout_marginTop="1dp"
            android:text="Anda dapat melakukan update kos melalui menu My Kost dan klik kos yang akan diedit."
            android:textColor="#1B1313"
            android:textSize="16sp" />
    </LinearLayout>
</androidx.cardview.widget.CardView>

<androidx.cardview.widget.CardView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_rowWeight="1"
    android:layout_columnWeight="1"
    android:layout_margin="12dp"
    app:cardCornerRadius="12dp"
    app:cardElevation="6dp">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:gravity="center"
        android:orientation="vertical"
        android:padding="30dp">

        <TextView
            android:layout_width="300dp"
            android:layout_height="40dp"
            android:layout_marginTop="1dp"
            android:text="Bagaimana cara mengetahui kost saya yang telah dibook?"
            android:textColor="#1B1313"
            android:textSize="16sp"
            android:textStyle="bold" />

```

```

<TextView
    android:layout_width="300dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="1dp"
    android:text="Anda dapat mengetahuinya dengan memilih menu Booked dan akan terdapat kumpulan data"
    android:textColor="#1B1313"
    android:textSize="16sp" />
</LinearLayout>
</androidx.cardview.widget.CardView>
</GridLayout>
</LinearLayout>
</ScrollView>

```

#### **Code 41. Code XML FAQ Pemilik**

Dalam coding XML yang digunakan untuk tampilan Edit FAQ Pemilik, beberapa elemen digunakan, seperti tata letak dan widget. Selain itu terdapat Scroll View dan Card View. Scroll View digunakan untuk membuat tampilan layar bisa scroll ke atas dan ke bawah. CardView adalah sebuah view atau widget yang memberikan tampilan seperti kertas yang mengambang pada layar. Tampilan ini biasanya digunakan untuk menampilkan informasi berupa teks atau gambar di dalamnya.

Layout yang digunakan adalah Linear Layout, Grid Layout, dan Relative Layout. Linear Layout digunakan untuk mengatur posisi berbagai elemen secara linier, yang dapat vertikal atau horizontal. Grid layout sering digunakan untuk membuat desain website dengan konten lebih tertata rapi. Relative Layout merupakan layout yang berfungsi untuk mengatur tata letak komponen atau widget aplikasi android dengan cara relative tidak hanya vertikal atau horizontal saja seperti pada Linear Layout. Kemudian widget yang digunakan adalah text view dan button. Text View digunakan untuk menampilkan teks dalam aplikasi, dan tampilan gambar digunakan untuk menampilkan gambar dalam aplikasi. ImageButton biasanya digunakan untuk menampilkan gambar atau icon bukannya teks namun berfungsi sebagai button. Button di layout ini digunakan untuk konfirmasi, dan jika ditekan, perintah akan dijalankan.

### **Coding XML Menu Profile**

#### **Coding XML Profile Pemilik**

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".ProfilePemilik">
    <include layout="@layout/toolbar"/>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:paddingTop="15dp"
        android:orientation="vertical">
        <ImageView
            android:id="@+id/imagePemilik"
            android:layout_width="100dp"
            android:layout_height="100dp"
            android:layout_gravity="center"
            android:layout_marginTop="10dp"
            app:srcCompat="@drawable/noimage" />
        <View
            android:layout_width="match_parent"
            android:layout_height="1dp"
            android:background="@color/black"
            android:layout_marginTop="10dp"/>
    </LinearLayout>

```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:paddingLeft="15dp"
    android:paddingRight="15dp"
    android:layout_marginTop="20dp"
    android:layout_gravity="center_vertical">
<ImageView
    android:layout_width="30dp"
    android:layout_height="30dp"
    android:src="@drawable/user"/>
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Nama Lengkap"
    android:textStyle="bold"
    android:textColor="@color/black"
    android:textSize="18dp"
    android:layout_marginLeft="15dp" />
<TextView
    android:id="@+id/txtNamaPemilik"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="30dp"
    android:textSize="20dp"
    android:textColor="@android:color/darker_gray"/>
</LinearLayout>
<View
    android:layout_width="match_parent"
    android:layout_height="0.5dp"
    android:background="@android:color/darker_gray"
    android:layout_marginTop="10dp"
    android:paddingLeft="15dp"
    android:paddingRight="15dp" />
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:paddingLeft="15dp"
    android:paddingRight="15dp"
    android:layout_marginTop="20dp"
    android:layout_gravity="center_vertical">
<ImageView
    android:layout_width="30dp"
    android:layout_height="30dp"
    android:src="@drawable/calendar"/>
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Tanggal Lahir"
    android:textStyle="bold"
    android:textColor="@color/black"
    android:textSize="18dp"
    android:layout_marginLeft="15dp" />
<TextView
    android:id="@+id/txtTanggalLahir"
    android:layout_width="174dp"
    android:layout_height="wrap_content"
    android:layout_marginLeft="40dp"
    android:textColor="@android:color/darker_gray"
    android:textSize="20dp" />
</LinearLayout>
<View
    android:layout_width="match_parent"
    android:layout_height="0.5dp"
    android:background="@android:color/darker_gray"
    android:layout_marginTop="10dp"
    android:paddingLeft="15dp"
    android:paddingRight="15dp" />
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:paddingLeft="15dp"
    android:paddingRight="15dp"
    android:layout_marginTop="20dp"
    android:layout_gravity="center_vertical">
<ImageView
    android:layout_width="30dp"
    android:layout_height="30dp"
    android:src="@drawable/call"/>
```

```
<TextView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="No. Handphone"  
    android:textStyle="bold"  
    android:textColor="@color/black"  
    android:textSize="18dp"  
    android:layout_marginLeft="15dp" />  
<TextView  
    android:id="@+id/txtTelepon"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_marginLeft="25dp"  
    android:textSize="20dp"  
    android:textColor="@android:color/darker_gray"/>  
</LinearLayout>  
<View  
    android:layout_width="match_parent"  
    android:layout_height="0.5dp"  
    android:background="@android:color/darker_gray"  
    android:layout_marginTop="10dp"  
    android:paddingLeft="15dp"  
    android:paddingRight="15dp" />  
<LinearLayout  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:orientation="horizontal"  
    android:paddingLeft="15dp"  
    android:paddingRight="15dp"  
    android:layout_marginTop="20dp"  
    android:layout_gravity="center_vertical">  
    <ImageView  
        android:layout_width="30dp"  
        android:layout_height="30dp"  
        android:src="@drawable/envelope"/>  
    <TextView  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:text="Email"  
        android:textStyle="bold"  
        android:textColor="@color/black"  
        android:textSize="18dp"  
        android:layout_marginLeft="15dp" />  
    <TextView  
        android:id="@+id/txtEmail"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:layout_marginLeft="107dp"  
        android:textSize="20dp"  
        android:textColor="@android:color/darker_gray"/>  
</LinearLayout>  
<View  
    android:layout_width="match_parent"  
    android:layout_height="0.5dp"  
    android:background="@android:color/darker_gray"  
    android:layout_marginTop="10dp"  
    android:paddingLeft="15dp"  
    android:paddingRight="15dp" />  
<LinearLayout  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:orientation="horizontal"  
    android:paddingLeft="15dp"  
    android:paddingRight="15dp"  
    android:layout_marginTop="20dp"  
    android:layout_gravity="center_vertical">  
    <ImageView  
        android:layout_width="30dp"  
        android:layout_height="30dp"  
        android:src="@drawable/info"/>  
    <TextView  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:text="Status"  
        android:textStyle="bold"  
        android:textColor="@color/black"  
        android:textSize="18dp"  
        android:layout_marginLeft="15dp" />
```



```

<TextView
    android:id="@+id/txtStatus"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="98dp"
    android:textSize="20dp"
    android:textColor="@android:color/darker_gray"/>
</LinearLayout>
<View
    android:layout_width="match_parent"
    android:layout_height="0.5dp"
    android:background="@android:color/darker_gray"
    android:layout_marginTop="10dp"
    android:paddingLeft="15dp"
    android:paddingRight="15dp" />
<View
    android:layout_width="match_parent"
    android:layout_height="0.5dp"
    android:background="@android:color/darker_gray"
    android:layout_marginTop="10dp"
    android:paddingLeft="15dp"
    android:paddingRight="15dp" />
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp"
    android:gravity="center"
    android:orientation="vertical">
    <Button
        android:id="@+id/btnEditProfil"
        android:layout_width="210dp"
        android:layout_height="50dp"
        android:backgroundTint="#465785"
        android:text="Edit Profil"
        android:gravity="center"
        android:textColor="@color/white"
        android:textSize="15dp" />
</LinearLayout>
</LinearLayout>

```

**Code 42. Code XML Profil Pemilik**

Pada coding XML diatas adalah coding yang digunakan untuk tampilan Profil Pemilik kost. Pada halaman ini terdapat beberapa elemen yang digunakan seperti layout dan widget. Layout yang digunakan adalah linear layout. Linear layout digunakan untuk mengatur posisi berbagai elemen di dalamnya secara linear baik dalam bentuk vertikal maupun horizontal. Selain itu juga ada widget yang digunakan, yaitu image toolbar, view, image view, text view, dan button. Toolbar merupakan sebuah tools berupa action back, logout, dan sebagainya yang dikategorikan ke dalam view group dan dapat ditempatkan dimanapun dalam layout kita. Image view merupakan salah satu widget yang digunakan untuk menampilkan sebuah gambar. View merupakan sebuah obyek yang menggambar komponen tampilan ke layar yang mana pengguna dapat melihat dan berinteraksi langsung. Text view digunakan untuk menampilkan suatu text di dalam aplikasi, disini text view berisi tulisan judul untuk masing-masing detail data profil, yaitu nama, tanggal lahir, nomor handphone, email, dan status. Sedangkan button adalah tombol yang dapat digunakan untuk memulai aktivitas lain. Button yang digunakan pada layout ini adalah button “Edit profil” yang ketika klik akan diarahkan ke menu edit profile pemilik.

## Coding XML Edit Profile Pemilik

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".EditProfilePemilik">

    <include layout="@layout/toolbar_edit"/>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:orientation="vertical">

        <ImageView
            android:id="@+id/imagePemilik"
            android:layout_width="180dp"
            android:layout_height="180dp"
            android:layout_gravity="center"
            android:layout_marginTop="6dp"
            app:srcCompat="@drawable/noimage" />

        <Button
            android:id="@+id/btnChoose"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Choose"
            android:layout_gravity="center"
            app:backgroundTint="#46657B5" />

        <View
            android:layout_width="match_parent"
            android:layout_height="1dp"
            android:background="@color/black"
            android:layout_marginTop="10dp" />

    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:paddingLeft="15dp"
        android:negativeRight="15dp"

        android:paddingRight="15dp"
        android:layout_marginTop="10dp"
        android:layout_gravity="center_vertical">

        <ImageView
            android:layout_width="30dp"
            android:layout_height="30dp"
            android:src="@drawable/user" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Nama Lengkap"
            android:textStyle="bold"
            android:textColor="@color/black"
            android:textSize="18dp"
            android:layout_marginLeft="15dp" />

        <EditText
            android:id="@+id/txtNamaPemilik"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginLeft="30dp"
            android:textSize="20dp"
            android:textColor="@android:color/black" />

    </LinearLayout>

    <View
        android:layout_width="match_parent"
        android:layout_height="0.5dp"
        android:background="@android:color/darker_gray"
        android:layout_marginTop="10dp"
        android:paddingLeft="15dp"
        android:paddingRight="15dp" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:paddingLeft="15dp"
        android:paddingRight="15dp"
        android:layout_marginTop="10dp"
        android:layout_gravity="center_vertical">
```

```

<ImageView
    android:layout_width="30dp"
    android:layout_height="30dp"
    android:src="@drawable/calendar"/>

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Tanggal Lahir"
    android:textStyle="bold"
    android:textColor="@color/black"
    android:textSize="18dp"
    android:layout_marginLeft="15dp" />

<EditText
    android:id="@+id/txtTanggalLahir"
    android:layout_width="130dp"
    android:layout_height="wrap_content"
    android:layout_marginLeft="40dp"
    android:textColor="@android:color/black"
    android:textSize="20dp" />

<Button
    android:id="@+id	btnTimePicker"
    android:layout_width="60dp"
    android:layout_height="match_parent"
    android:background="#465785"
    android:gravity="center"
    android:text="DATE"
    android:textColor="@color/white"
    android:textSize="8dp"
    android:backgroundTint="#465785"
    android:layout_toRightOf="@+id/txtTanggalLahir"/>

</LinearLayout>
<View
    android:layout_width="match_parent"
    android:layout_height="0.5dp"
    android:background="@android:color/darker_gray"
    android:layout_marginTop="10dp"
    android:paddingLeft="15dp"
    android:paddingRight="15dp" />

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:paddingLeft="15dp"
    android:paddingRight="15dp"
    android:layout_marginTop="10dp"
    android:layout_gravity="center_vertical">

    <ImageView
        android:layout_width="30dp"
        android:layout_height="30dp"
        android:src="@drawable/call"/>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="No. Handphone"
        android:textStyle="bold"
        android:textColor="@color/black"
        android:textSize="18dp"
        android:layout_marginLeft="15dp" />

    <EditText
        android:id="@+id/txtTelepon"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="25dp"
        android:textSize="20dp"
        android:textColor="@android:color/black"/>

</LinearLayout>

<View
    android:layout_width="match_parent"
    android:layout_height="0.5dp"
    android:background="@android:color/darker_gray"
    android:layout_marginTop="10dp"
    android:paddingLeft="15dp"
    android:paddingRight="15dp" />

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:paddingLeft="15dp"

```

```

        android:paddingRight="15dp"
        android:layout_marginTop="10dp"
        android:layout_gravity="center_vertical">

    <ImageView
        android:layout_width="30dp"
        android:layout_height="30dp"
        android:src="@drawable/envelope"/>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Email"
        android:textStyle="bold"
        android:textColor="@color/black"
        android:textSize="18dp"
        android:layout_marginLeft="15dp" />

    <EditText
        android:id="@+id/txtEmail"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="107dp"
        android:textSize="20dp"
        android:textColor="@android:color/darker_gray"
        android:enabled="false"/>

</LinearLayout>

<View
    android:layout_width="match_parent"
    android:layout_height="0.5dp"
    android:background="@android:color/darker_gray"
    android:layout_marginTop="10dp"
    android:paddingLeft="15dp"
    android:paddingRight="15dp" />

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:paddingLeft="15dp"
    android:paddingRight="15dp"
    android:layout_marginTop="10dp"
    android:layout_gravity="center_vertical">

    <ImageView
        android:layout_width="30dp"
        android:layout_height="30dp"
        android:src="@drawable/info"/>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Status"
        android:textStyle="bold"
        android:textColor="@color/black"
        android:textSize="18dp"
        android:layout_marginLeft="15dp" />

    <EditText
        android:id="@+id/txtStatus"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="98dp"
        android:textSize="20dp"
        android:enabled="false"
        android:textColor="@android:color/darker_gray"/>

</LinearLayout>

<View
    android:layout_width="match_parent"
    android:layout_height="0.5dp"
    android:background="@android:color/darker_gray"
    android:layout_marginTop="10dp"
    android:paddingLeft="15dp"
    android:paddingRight="15dp" />

<View
    android:layout_width="match_parent"
    android:layout_height="0.5dp"
    android:background="@android:color/darker_gray"
    android:layout_marginTop="10dp"
    android:paddingLeft="15dp"
    android:paddingRight="15dp" />

```

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp"
    android:gravity="center"
    android:orientation="vertical">

    <Button
        android:id="@+id	btnSimpan"
        android:layout_width="210dp"
        android:layout_height="62dp"
        android:background="#44657B5"
        android:gravity="center"
        android:text="Simpan"
        android:textColor="@color/white"
        android:textSize="15dp" />

</LinearLayout>

```

**Code 43. Code XML Edit Profile Pemilik**

Dalam coding XML yang digunakan untuk tampilan Edit Profile Pemilik, beberapa elemen digunakan, seperti tata letak dan widget. Layout yang digunakan adalah Linear Layout. Linear Layout digunakan untuk mengatur posisi berbagai elemen secara linier, yang dapat vertikal atau horizontal. Kemudian widget yang digunakan adalah text view, image view, edit text dan button. Text View digunakan untuk menampilkan teks dalam aplikasi, dan tampilan gambar digunakan untuk menampilkan gambar dalam aplikasi. Image View yang digunakan pada dashboard diperoleh dari drawable. Edit Text memiliki tujuan yang mirip dengan tampilan teks, tetapi mengedit teks memiliki kemampuan untuk menerima input pengguna. Button di layout ini digunakan untuk konfirmasi, dan jika ditekan, perintah akan dijalankan.

## Coding XML pada Aplikasi Bagian Pencari Kost

### Coding XML Dashboard

```

<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/bgbiru"
    tools:context=".DashboardPencari">

    <LinearLayout
        android:orientation="vertical"
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginRight="20dp"
            android:layout_marginLeft="20dp"
            android:layout_marginTop="80dp">

```

```

        <TextView
            android:id="@+id/txtPencariKost"
            android:text="PENCARI KOST"
            android:textSize="24dp"
            android:textStyle="bold"
            android:layout_marginTop="15dp"
            android:textColor="#FFFFFF"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"/>

        <ImageButton
            android:id="@+id/btnKH"
            android:layout_alignParentRight="true"
            android:background="@drawable/logoputih"
            android:layout_width="62dp"
            android:layout_height="62dp"/>

    </RelativeLayout>

    <GridLayout
        android:columnCount="2"
        android:rowCount="2"
        android:alignmentMode="alignMargins"
        android:columnOrderPreserved="false"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="50dp">

        <androidx.cardview.widget.CardView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_columnWeight="1"
            android:layout_rowWeight="1"
            android:elevation="6dp"
            app:cardElevation="6dp"
            app:cardCornerRadius="12dp"
            android:layout_margin="12dp"
            >

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:gravity="center"
                android:orientation="vertical"
                android:padding="16dp">

                <ImageView
                    android:layout_width="80dp"
                    android:layout_height="80dp"
                    android:src="@drawable/search" />

                <Button
                    android:id="@+id/btnSearch"
                    style="?android:attr:borderlessButtonStyle"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:layout_marginTop="12dp"
                    android:backgroundTint="#00FFFFFF"
                    android:text="Search"
                    android:textColor="#6f6f6f"
                    android:textSize="18sp" />

            </LinearLayout>

        </androidx.cardview.widget.CardView>

        <androidx.cardview.widget.CardView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_columnWeight="1"
            android:layout_rowWeight="1"
            android:elevation="6dp"
            app:cardElevation="6dp"
            app:cardCornerRadius="12dp"
            android:layout_margin="12dp"
            >

```

```
<LinearLayout
    android:orientation="vertical"
    android:padding="16dp"
    android:gravity="center"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <ImageView
        android:src="@drawable/notepad"
        android:layout_width="80dp"
        android:layout_height="80dp"/>

    <Button
        android:id="@+id	btnBooked"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="12dp"
        android:backgroundTint="#00FFFFFF"
        android:text="Booked"
        style="?android:attr/borderlessButtonStyle"
        android:textColor="#66f6f6f"
        android:textSize="18sp" />

</LinearLayout>
</androidx.cardview.widget.CardView>

<androidx.cardview.widget.CardView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_columnWidth="1"
    android:layout_rowWeight="1"
    android:elevation="6dp"
    app:cardElevation="6dp"
    app:cardCornerRadius="12dp"
    android:layout_margin="12dp">

    <LinearLayout
        android:orientation="vertical"
        android:padding="16dp"
        android:gravity="center"
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <ImageView
            android:src="@drawable/signquestion"
            android:layout_width="80dp"
            android:layout_height="80dp"/>

        <Button
            android:id="@+id	btnFAQ"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="12dp"
            android:backgroundTint="#00FFFFFF"
            android:text="FAQ"
            style="?android:attr/borderlessButtonStyle"
            android:textColor="#66f6f6f"
            android:textSize="18sp" />

    </LinearLayout>
</androidx.cardview.widget.CardView>
<androidx.cardview.widget.CardView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_columnWidth="1"
    android:layout_rowWeight="1"
    android:elevation="6dp"
    app:cardElevation="6dp"
    app:cardCornerRadius="12dp"
    android:layout_margin="12dp">
```

```
<LinearLayout  
    android:orientation="vertical"  
    android:padding="16dp"  
    android:gravity="center"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent">  
  
    <ImageView  
        android:src="@drawable/user"  
        android:layout_width="80dp"  
        android:layout_height="80dp"/>  
  
    <Button  
        android:id="@+id/btnProfile"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:layout_marginTop="12dp"  
        android:backgroundTint="#0000FFFF"  
        android:text="Profile"  
        style="?android:attr/borderlessButtonStyle"  
        android:textColor="#666666"  
        android:textSize="18sp" />  
  
    </LinearLayout>  
    </androidx.cardview.widget.CardView>  
    </GridLayout>  
</LinearLayout>  
</ScrollView>
```

**Code 44. Code XML Dashboard Pencari Kost**

Pada coding XML yang digunakan untuk tampilan Dashboard terdapat beberapa elemen yang digunakan seperti layout dan widget. Selain itu terdapat scroll view untuk menampung berbagai komponen di dalamnya. Scroll view digunakan untuk membuat tampilan android dapat discroll ke atas atau ke bawah sehingga mampu menyimpan banyak komponen di dalamnya. Layout yang digunakan adalah relative layout, linear layout, dan grid layout. Relative layout bertujuan untuk membuat berbagai elemen penyusun tampilan dapat dengan mudah diatur posisinya sedemikian rupa secara relatif terhadap layout tersebut. Linear layout digunakan untuk mengatur posisi berbagai elemen di dalamnya secara linear baik dalam bentuk vertikal maupun horizontal. Grid layout bertujuan untuk membuat tampilan dengan komposisi baris dan kolom yang jumlahnya sudah ditentukan, seperti: android:columnCount="2" dan android:rowCount="2".

Lalu widget yang digunakan adalah text view, image view, card view, dan button. Text view digunakan untuk menampilkan suatu text di dalam aplikasi sedangkan image view digunakan untuk menampilkan gambar di dalam aplikasi. Gambar yang digunakan pada dashboard diperoleh dari drawable. Card view digunakan untuk membuat tampilan seperti grid, namun dalam bentuk yang lebih menarik yaitu seperti kumpulan kartu dalam bentuk segi empat yang tersusun rapi. Button digunakan sebagai suatu konfirmasi yang jika ditekan akan menjalankan perintah tertentu atau mengarahkan ke activity baru.

## Coding XML Menu Search

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/bgbirupendek"
    tools:context=".SearchPencari">

    <LinearLayout
        android:orientation="vertical"
        android:layout_width="match_parent"
        android:layout_height="match_parent">
        <RelativeLayout
            android:layout_marginTop="32dp"
            android:layout_marginLeft="20dp"
            android:layout_marginRight="20dp"
            android:layout_width="match_parent"
            android:layout_height="wrap_content">
            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginTop="17dp"
                android:textSize="24sp"
                android:textColor="#FFFFFF"
                android:textStyle="bold"
                android:text="HASIL PENCARIAN"/>
            <ImageButton
                android:id="@+id/btnKH"
                android:layout_alignParentRight="true"
                android:background="@drawable/logoputih"
                android:layout_width="62dp"
                android:layout_height="62dp"/>
        </RelativeLayout>
        <LinearLayout
            android:orientation="horizontal"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="45dp"
            android:layout_marginLeft="15dp"
            android:layout_marginRight="15dp"
            android:layout_marginBottom="10dp">
            <EditText
                android:id="@+id/edtSearch"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_weight="1"
                android:hint="Masukkan Kota Kost"/>
            <Button
                android:id="@+id/btnSearch"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginLeft="10dp"
                android:text="Search"
                android:backgroundTint="#4657B5"/>
        </LinearLayout>
        <LinearLayout
            android:orientation="vertical"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_marginLeft="10dp"
            android:layout_marginRight="10dp">
            <ListView
                android:id="@+id/listHasilCariKost"
                android:layout_width="match_parent"
                android:layout_height="match_parent">
            </ListView>
        </LinearLayout>
    </LinearLayout>
</RelativeLayout>
```

Code 45. Code XML Menu Search

Pada coding XML yang digunakan untuk tampilan Menu Search. terdapat beberapa elemen yang digunakan seperti layout dan widget. Layout yang digunakan adalah Relative Layout dan Linear Layout. Relative Layout bertujuan untuk membuat berbagai elemen penyusun tampilan dapat dengan mudah diatur posisinya sedemikian rupa secara relatif terhadap layout tersebut. Linear Layout digunakan untuk mengatur posisi berbagai elemen di dalamnya secara linear baik dalam bentuk vertikal maupun horizontal. Lalu widget yang digunakan adalah Text View, Image Button, Edit Text, Button, dan List View. Text view

digunakan untuk menampilkan suatu text di dalam aplikasi, disini text view berisi tulisan “Hasil Pencarian” untuk memberi keterangan halaman yang sedang dibuka. Image Button merupakan tombol dengan gambar yang digunakan untuk mempermudah tampilan namun dapat digunakan untuk memulai aktivitas atau proses. Pada tampilan ini, Image Button digunakan agar pencari kost dapat kembali ke halaman dashboard pencari kost. Edit Text digunakan untuk memasukkan data berupa text pada aplikasi android, disini Edit Text meminta pengguna untuk memasukkan kata kunci pencarian berupa nama kota. Button merupakan tombol yang dapat digunakan untuk memulai aktivitas atau suatu proses. Pada tampilan ini, button Search digunakan untuk melakukan proses pencarian yang hasilnya akan ditampilkan dalam bentuk List pada List View.

## Coding XML Book Kost

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".BookKost">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="16dp"
            android:text="BOOK KOST"
            android:textStyle="bold"
            android:textAlignment="center"
            android:textSize="20sp"
            android:layout_gravity="center"/>
        <ImageView
            android:id="@+id/imageKost"
            android:layout_width="156dp"
            android:layout_height="152dp"
            android:layout_gravity="center"
            android:layout_marginTop="10dp"
            app:srcCompat="@drawable/noimage" />
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="vertical"
            android:layout_marginTop="5dp"
            android:layout_marginBottom="10dp"
            android:layout_marginLeft="25dp"
            android:layout_marginRight="25dp">
            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:orientation="horizontal">
                <TextView
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:text="Nama Kost :"
                    android:textSize="20sp" />
```

```
<TextView
    android:id="@+id/txtNamaKost"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textSize="20sp"
    android:layout_marginLeft="10dp"/>
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:layout_marginTop="5dp">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Harga Kost :"
        android:textSize="20sp" />
    <TextView
        android:id="@+id/txtHargaKost"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:textSize="20sp"
        android:layout_marginLeft="10dp"/>
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:layout_marginTop="5dp">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Alamat Kost :"
        android:textSize="20sp" />
    <TextView
        android:id="@+id/txtAlamatKost"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:textSize="20sp"
        android:layout_marginLeft="10dp"/>
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:layout_marginTop="5dp">
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Kota Kost :"
            android:textSize="20sp" />
        <TextView
            android:id="@+id/txtKotaKost"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:ems="10"
            android:textSize="20sp"
            android:layout_marginLeft="10dp"/>
    
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:layout_marginTop="5dp">
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Jenis Kost :"
            android:textSize="20sp" />
        <TextView
            android:id="@+id/txtJenisKost"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:ems="10"
            android:textSize="20sp"
            android:layout_marginLeft="10dp"/>
    
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:layout_marginTop="5dp">
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Jumlah Kamar :"
            android:textSize="20sp" />
        <TextView
            android:id="@+id/txtJumlahKamar"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:ems="10"
            android:textSize="20sp"
            android:layout_marginLeft="10dp"/>
    

```

```
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:layout_marginTop="5dp">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Fasilitas Kost :"
        android:textSize="20sp" />
    <TextView
        android:id="@+id/txtFasilitas"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:textSize="20sp"
        android:layout_marginLeft="10dp"/>
</LinearLayout>
<Button
    android:id="@+id	btnBookKost"
    android:layout_width="290dp"
    android:layout_height="65dp"
    android:text="Book"
    android:layout_marginTop="5dp"
    android:layout_gravity="center"
    app:backgroundTint="#465785" />
<TextView
    android:id="@+id/txtIDKost"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:textSize="20sp"
    android:visibility="gone"
    android:layout_marginLeft="10dp"/>
</LinearLayout>
</LinearLayout>
</RelativeLayout>
```

Code 46. Code XML Book Kost

Pada coding XML yang digunakan untuk tampilan Book Kost. terdapat beberapa elemen yang digunakan seperti layout dan widget. Layout yang digunakan adalah relative layout dan linear layout. Relative layout bertujuan untuk membuat berbagai elemen penyusun tampilan dapat dengan mudah diatur posisinya sedemikian rupa secara relatif terhadap layout tersebut. Linear layout digunakan untuk mengatur posisi berbagai elemen di dalamnya secara linear baik dalam bentuk vertikal maupun horizontal. Lalu widget yang digunakan adalah text view dan button. Text view digunakan untuk menampilkan suatu text di dalam aplikasi sedangkan button merupakan tombol yang dapat digunakan untuk memulai aktivitas yang lain, misalnya membuka suatu activity baru.

## Coding XML Menu Booked

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/bgbirupendek"
    tools:context=".BookedPencari">
    <LinearLayout
        android:orientation="vertical"
        android:layout_width="match_parent"
        android:layout_height="match_parent">
        <RelativeLayout
            android:layout_marginTop="32dp"
            android:layout_marginLeft="20dp"
            android:layout_marginRight="20dp"
            android:layout_width="match_parent"
            android:layout_height="wrap_content">
            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginTop="17dp"
                android:text="BOOKED KOST"
                android:textColor="#FFFFFF"
                android:textSize="24sp"
                android:textStyle="bold" />
            <ImageButton
                android:id="@+id/btnKH"
                android:layout_alignParentRight="true"
                android:background="@drawable/logoputih"
                android:layout_width="62dp"
                android:layout_height="62dp"/>
        </RelativeLayout>
        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_marginTop="50dp"
            android:layout_marginLeft="15dp"
            android:layout_marginRight="15dp">
            <ListView
                android:id="@+id/listBooked"
                android:layout_width="match_parent"
                android:layout_height="match_parent">
            </ListView>
        </RelativeLayout>
    </LinearLayout>
</RelativeLayout>
```

Code 47. Code XML Menu Booked Kost Pencari Kost

Pada coding XML yang digunakan untuk tampilan Booked. terdapat beberapa elemen yang digunakan seperti layout dan widget. Layout yang digunakan adalah relative layout dan linear layout. Relative layout bertujuan untuk membuat berbagai elemen penyusun tampilan dapat dengan mudah diatur posisinya sedemikian rupa secara relatif terhadap layout tersebut. Linear layout digunakan untuk mengatur posisi berbagai elemen di dalamnya secara linear baik dalam bentuk vertikal maupun horizontal. Lalu widget yang digunakan adalah text view dan image button. Text view digunakan untuk menampilkan suatu text di dalam aplikasi sedangkan image button merupakan suatu gambar yang dapat digunakan pula sebagai tombol yang jika ditekan akan memberikan aksi tertentu seperti menjalankan suatu perintah atau membuka activity baru.

## Coding XML List Booked Pencari

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:layout_margin="10dp">
        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="2"
            android:orientation="vertical">
            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:orientation="horizontal">
                <TextView
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:text="Nama Kost: "/>
                <TextView
                    android:id="@+id/txtNamaKost"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:gravity="left"
                    android:text="Nama Kost"/>
            </LinearLayout>
            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:orientation="horizontal">
                <TextView
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:text="Alamat Kost: "/>
                <TextView
                    android:id="@+id/txtAlamatKost"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:gravity="left"
                    android:text="Alamat Kost"/>
            </LinearLayout>
        </LinearLayout>
    </LinearLayout>
</RelativeLayout>
```

Code 48. Code XML List Booked Pencari Kost

Pada coding XML yang digunakan untuk tampilan List Booked Pencari terdapat beberapa elemen yang digunakan seperti layout dan widget. Layout yang digunakan adalah linear layout dan relative layout. Linear layout digunakan untuk mengatur posisi berbagai elemen di dalamnya secara linear baik dalam bentuk vertikal maupun horizontal. Relative Layout merupakan layout yang berfungsi untuk mengatur tata letak komponen atau widget aplikasi android dengan cara relatif tidak hanya vertikal atau horizontal saja seperti pada Linear Layout. Lalu widget yang digunakan adalah text view. Text View disini bertujuan menampilkan data yang sudah diinput seperti Nama Kost dan Alamat Kost.

## Coding XML Menu FAQ

### Coding XML FAQ Pencari

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/bgbirupendek"
    tools:context=".faqPencari">

    <LinearLayout
        android:orientation="vertical"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <RelativeLayout
            android:layout_marginTop="32dp"
            android:layout_marginRight="20dp"
            android:layout_marginLeft="20dp"
            android:layout_width="match_parent"
            android:layout_height="wrap_content">
            <TextView
                android:text="FAQ Pencari Kost"
                android:textSize="24sp"
                android:textStyle="bold"
                android:textColor="#FFFFFF"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"/>
            <ImageButton
                android:id="@+id/btnKH"
                android:layout_alignParentRight="true"
                android:background="@drawable/logoputih"
                android:layout_width="62dp"
                android:layout_height="62dp"/>
        </RelativeLayout>
        <GridLayout
            android:columnCount="1"
            android:rowCount="4"
            android:alignmentMode="alignMargins"
            android:columnOrderPreserved="false"
            android:layout_width="match_parent"
            android:layout_height="wrap_content">
            <androidx.cardview.widget.CardView
                android:layout_marginTop="20dp"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_columnWeight="1"
                android:layout_rowWeight="1"
                app:cardElevation="6dp"
                app:cardCornerRadius="12dp"
                android:layout_margin="12dp">
```

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:orientation="vertical"
    android:padding="40dp">

    <TextView
        android:layout_width="300dp"
        android:layout_height="40dp"
        android:layout_marginTop="1dp"
        android:text="Bagaimana cara mencari kos di KostHost?"
        android:textColor="#1B1313"
        android:textSize="16sp"
        android:textStyle="bold" />

    <TextView
        android:layout_width="300dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="1dp"
        android:text="Pencarian anda dimudahkan dengan mencari kos berdasarkan kota yang anda inginkan."
        android:textColor="#1B1313"
        android:textSize="16sp" />
</LinearLayout>
</androidx.cardview.widget.CardView>

<androidx.cardview.widget.CardView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_columnWeight="1"
    android:layout_rowWeight="1"
    app:cardElevation="6dp"
    app:cardCornerRadius="12dp"
    android:layout_margin="12dp">
<LinearLayout
    android:orientation="vertical"
    android:padding="40dp"
    android:gravity="center"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <TextView
        android:layout_width="300dp"
        android:layout_height="40dp"
        android:layout_marginTop="1dp"
        android:text="Bagaimana jika saya ingin survey kos yang akan disewa?"
        android:textColor="#1B1313"
        android:textSize="16sp"
        android:textStyle="bold"/>
    <TextView
        android:layout_width="300dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="1dp"
        android:text="Anda dapat melakukan survey terhadap kos yang ingin Anda sewa dengan melakukan"
        android:textColor="#1B1313"
        android:textSize="16sp" />
</LinearLayout>
</androidx.cardview.widget.CardView>

<androidx.cardview.widget.CardView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_columnWeight="1"
    android:layout_rowWeight="1"
    app:cardElevation="6dp"
    app:cardCornerRadius="12dp"
    android:layout_margin="12dp">
<LinearLayout
    android:orientation="vertical"
    android:padding="40dp"
    android:gravity="center"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <TextView
        android:layout_width="300dp"
        android:layout_height="40dp"
        android:layout_marginTop="1dp"
        android:text="Bagaimana cara mengetahui kos yang telah saya book?"
        android:textColor="#1B1313"
        android:textSize="16sp"
        android:textStyle="bold"/>
    <TextView
        android:layout_width="300dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="1dp"
        android:text="Anda dapat memilih menu booked untuk melihat daftar kos yang telah anda book."
        android:textColor="#1B1313"
        android:textSize="16sp" />
</LinearLayout>

```

**Code 49. Code XML FAQ Pencari**

Dalam coding XML yang digunakan untuk tampilan Edit FAQ Pencari, beberapa elemen digunakan, seperti tata letak dan widget. Selain itu terdapat Scroll View dan Card View. Scroll View digunakan untuk membuat tampilan layar bisa di scroll ke atas dan ke bawah. CardView adalah sebuah view atau widget yang memberikan tampilan seperti kertas yang mengambang pada layar. Tampilan ini biasanya digunakan untuk menampilkan informasi berupa teks atau gambar di dalamnya.

Layout yang digunakan adalah Linear Layout, Grid Layout, dan Relative Layout. Linear Layout digunakan untuk mengatur posisi berbagai elemen secara linier, yang dapat vertikal atau horizontal. Grid layout sering digunakan untuk membuat desain website dengan konten lebih tertata rapi. Relative Layout merupakan layout yang berfungsi untuk mengatur tata letak komponen atau widget aplikasi android dengan cara relative tidak hanya vertikal atau horizontal saja seperti pada Linear Layout.

Kemudian widget yang digunakan adalah text view dan button. Text View digunakan untuk menampilkan teks dalam aplikasi, dan tampilan gambar digunakan untuk menampilkan gambar dalam aplikasi. ImageButton biasanya digunakan untuk menampilkan gambar atau icon bukannya teks namun berfungsi sebagai button. Button di layout ini digunakan untuk konfirmasi, dan jika ditekan, perintah akan dijalankan.

## Coding XML Menu Profile

### Coding XML Profile Pencari

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".ProfilePencari">
    <include layout="@layout/toolbar"/>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:paddingTop="15dp"
        android:orientation="vertical">
        <ImageView
            android:id="@+id/imagePencari"
            android:layout_width="100dp"
            android:layout_height="100dp"
            android:layout_gravity="center"
            android:layout_marginTop="10dp"
            app:srcCompat="@drawable/noimage" />
        <View
            android:layout_width="match_parent"
            android:layout_height="1dp"
            android:background="@color/black"
            android:layout_marginTop="10dp"/>
    </LinearLayout>
```

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:paddingLeft="15dp"
    android:paddingRight="15dp"
    android:layout_marginTop="20dp"
    android:layout_gravity="center_vertical">
    <ImageView
        android:layout_width="30dp"
        android:layout_height="30dp"
        android:src="@drawable/user"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Nama Lengkap"
        android:textStyle="bold"
        android:textColor="@color/black"
        android:textSize="18dp"
        android:layout_marginLeft="15dp" />
    <TextView
        android:id="@+id/txtNamaPencari"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="30dp"
        android:textSize="20dp"
        android:textColor="@android:color/darker_gray"/>
</LinearLayout>
...
<View
    android:layout_width="match_parent"
    android:layout_height="0.5dp"
    android:background="@android:color/darker_gray"
    android:layout_marginTop="10dp"
    android:paddingLeft="15dp"
    android:paddingRight="15dp" />
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:paddingLeft="15dp"
    android:paddingRight="15dp"
    android:layout_marginTop="20dp"
    android:layout_gravity="center_vertical">
    <ImageView
        android:layout_width="30dp"
        android:layout_height="30dp"
        android:src="@drawable/calendar"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Tanggal Lahir"
        android:textStyle="bold"
        android:textColor="@color/black"
        android:textSize="18dp"
        android:layout_marginLeft="15dp" />
    <TextView
        android:id="@+id/txtTanggallahir"
        android:layout_width="174dp"
        android:layout_height="wrap_content"
        android:layout_marginLeft="40dp"
        android:textColor="@android:color/darker_gray"
        android:textSize="20dp" />
</LinearLayout>
<View
    android:layout_width="match_parent"
    android:layout_height="0.5dp"
    android:background="@android:color/darker_gray"
    android:layout_marginTop="10dp"
    android:paddingLeft="15dp"
    android:paddingRight="15dp" />
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:paddingLeft="15dp"
    android:paddingRight="15dp"
    android:layout_marginTop="20dp"
    android:layout_gravity="center_vertical">
    <ImageView
        android:layout_width="30dp"
        android:layout_height="30dp"
        android:src="@drawable/call"/>

```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="No. Handphone"
    android:textStyle="bold"
    android:textColor="@color/black"
    android:textSize="18dp"
    android:layout_marginLeft="15dp" />
<TextView
    android:id="@+id/txtTelepon"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="25dp"
    android:textSize="20dp"
    android:textColor="@android:color/darker_gray"/>
</LinearLayout>
<View
    android:layout_width="match_parent"
    android:layout_height="0.5dp"
    android:background="@android:color/darker_gray"
    android:layout_marginTop="10dp"
    android:paddingLeft="15dp"
    android:paddingRight="15dp" />
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:paddingLeft="15dp"
    android:paddingRight="15dp"
    android:layout_marginTop="20dp"
    android:layout_gravity="center_vertical">
    <ImageView
        android:layout_width="30dp"
        android:layout_height="30dp"
        android:src="@drawable/envelope"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Email"
        android:textStyle="bold"
        android:textColor="@color/black"
        android:textSize="18dp"
        android:layout_marginLeft="15dp" />
    <TextView
        android:id="@+id/txtEmail"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="107dp"
        android:textSize="20dp"
        android:textColor="@android:color/darker_gray"/>
</LinearLayout>
<View
    android:layout_width="match_parent"
    android:layout_height="0.5dp"
    android:background="@android:color/darker_gray"
    android:layout_marginTop="10dp"
    android:paddingLeft="15dp"
    android:paddingRight="15dp" />
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:paddingLeft="15dp"
    android:paddingRight="15dp"
    android:layout_marginTop="20dp"
    android:layout_gravity="center_vertical">
    <ImageView
        android:layout_width="30dp"
        android:layout_height="30dp"
        android:src="@drawable/info"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Status"
        android:textStyle="bold"
        android:textColor="@color/black"
        android:textSize="18dp"
        android:layout_marginLeft="15dp" />
```

```

        <TextView
            android:id="@+id/txtStatus"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginLeft="98dp"
            android:textSize="20dp"
            android:textColor="@android:color/darker_gray"/>
    </LinearLayout>
<View
        android:layout_width="match_parent"
        android:layout_height="0.5dp"
        android:background="@android:color/darker_gray"
        android:layout_marginTop="10dp"
        android:paddingLeft="15dp"
        android:paddingRight="15dp" />
<View
        android:layout_width="match_parent"
        android:layout_height="0.5dp"
        android:background="@android:color/darker_gray"
        android:layout_marginTop="10dp"
        android:paddingLeft="15dp"
        android:paddingRight="15dp" />
<LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:gravity="center"
        android:orientation="vertical">
    <Button
        android:id="@+id/btnEditProfil"
        android:layout_width="210dp"
        android:layout_height="50dp"
        android:backgroundTint="#4657B5"
        android:text="Edit Profil"
        android:gravity="center"
        android:textColor="@color/white"
        android:textSize="15dp" />
    </LinearLayout>
</LinearLayout>

```

#### Code 50. Code XML Profil Pencari

Pada coding XML diatas adalah coding yang digunakan untuk tampilan Profil Pencari kost. Pada halaman ini terdapat beberapa elemen yang digunakan seperti layout dan widget. Layout yang digunakan adalah linear layout. Linear layout digunakan untuk mengatur posisi berbagai elemen di dalamnya secara linear baik dalam bentuk vertikal maupun horizontal. Selain itu juga ada widget yang digunakan, yaitu image toolbar, view, image view, text view, dan button. Toolbar merupakan sebuah tools berupa action back, logout, dan sebagainya yang dikategorikan ke dalam view group dan dapat ditempatkan dimanapun dalam layout kita. Image view merupakan salah satu widget yang digunakan untuk menampilkan sebuah gambar. View merupakan sebuah obyek yang menggambarkan komponen tampilan ke layar yang mana pengguna dapat melihat dan berinteraksi langsung. Text view digunakan untuk menampilkan suatu text di dalam aplikasi, disini text view berisi tulisan judul untuk masing-masing detail data profil, yaitu nama, tanggal lahir, nomor handphone, email, dan status. Sedangkan button adalah tombol yang dapat digunakan untuk memulai aktivitas lain. Button yang digunakan pada layout ini adalah button “Edit profil” yang ketika klik akan diarahkan ke menu edit profile pencari.

## Coding XML Edit Profile Pencari

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".EditProfilePencari">

    <include layout="@layout/toolbar_edit"/>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:orientation="vertical">

        <ImageView
            android:id="@+id/imagePencari"
            android:layout_width="100dp"
            android:layout_height="100dp"
            android:layout_gravity="center"
            android:layout_marginTop="6dp"
            app:srcCompat="@drawable/noimage" />
        <Button
            android:id="@+id/btnChoose"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Choose"
            android:layout_gravity="center"
            app:backgroundTint="#4657B5"/>
        <View
            android:layout_width="match_parent"
            android:layout_height="1dp"
            android:background="@color/black"
            android:layout_marginTop="10dp"/>

    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:paddingLeft="15dp"
        android:paddingRight="15dp"
        android:layout_marginTop="10dp"
        android:layout_gravity="center_vertical">

        <ImageView
            android:layout_width="30dp"
            android:layout_height="30dp"
            android:src="@drawable/user"/>

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Nama Lengkap"
            android:textStyle="bold"
            android:textColor="@color/black"
            android:textSize="18dp"
            android:layout_marginLeft="15dp" />

    
```

```

<EditText
    android:id="@+id/txtNamaPencari"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="30dp"
    android:textSize="20dp"
    android:textColor="@android:color/black"/>

</LinearLayout>

<View
    android:layout_width="match_parent"
    android:layout_height="0.5dp"
    android:background="@android:color/darker_gray"
    android:layout_marginTop="10dp"
    android:paddingLeft="15dp"
    android:paddingRight="15dp" />

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:paddingLeft="15dp"
    android:paddingRight="15dp"
    android:layout_marginTop="10dp"
    android:layout_gravity="center_vertical">

    <ImageView
        android:layout_width="30dp"
        android:layout_height="30dp"
        android:src="@drawable/calendar"/>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Tanggal Lahir"
        android:textStyle="bold"
        android:textColor="@color/black"
        android:textSize="18dp"
        android:layout_marginLeft="15dp" />

    <EditText
        android:id="@+id/txtTanggalLahir"
        android:layout_width="130dp"
        android:layout_height="wrap_content"
        android:layout_marginLeft="40dp"
        android:textColor="@android:color/black"
        android:textSize="20dp" />

    <Button
        android:id="@+id/btnTimePicker"
        android:layout_width="60dp"
        android:layout_height="match_parent"
        android:background="#4657B5"
        android:gravity="center"
        android:text="DATE"
        android:textColor="@color/white"
        android:textSize="8dp"
        android:backgroundTint="#4657B5"
        android:layout_toRightOf="@+id/txtTanggalLahir"/>

```

</LinearLayout>

```

<View
    android:layout_width="match_parent"
    android:layout_height="0.5dp"
    android:background="@android:color/darker_gray"
    android:layout_marginTop="10dp"
    android:paddingLeft="15dp"
    android:paddingRight="15dp" />

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:paddingLeft="15dp"
    android:paddingRight="15dp"
    android:layout_marginTop="10dp"
    android:layout_gravity="center_vertical">

```

```

<ImageView
    android:layout_width="30dp"
    android:layout_height="30dp"
    android:src="@drawable/call"/>

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="No. Handphone"
    android:textStyle="bold"
    android:textColor="@color/black"
    android:textSize="18dp"
    android:layout_marginLeft="15dp" />

<EditText
    android:id="@+id/txTelepon"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="25dp"
    android:textSize="20dp"
    android:textColor="@android:color/black"/>

</LinearLayout>

<View
    android:layout_width="match_parent"
    android:layout_height="0.5dp"
    android:background="@android:color/darker_gray"
    android:layout_marginTop="10dp"
    android:paddingLeft="15dp"
    android:paddingRight="15dp" />

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:paddingLeft="15dp"
    android:paddingRight="15dp"
    android:layout_marginTop="10dp"
    android:layout_gravity="center_vertical">

    <ImageView
        android:layout_width="30dp"
        android:layout_height="30dp"
        android:src="@drawable/envelope"/>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Email"
        android:textStyle="bold"
        android:textColor="@color/black"
        android:textSize="18dp"
        android:layout_marginLeft="15dp" />

```

```
<EditText  
    android:id="@+id/txtEmail"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_marginLeft="107dp"  
    android:textSize="20dp"  
    android:textColor="@android:color/darker_gray"  
    android:enabled="false"/>  
  
</LinearLayout>  
  
<View  
    android:layout_width="match_parent"  
    android:layout_height="0.5dp"  
    android:background="@android:color/darker_gray"  
    android:layout_marginTop="10dp"  
    android:paddingLeft="15dp"  
    android:paddingRight="15dp" />  
  
<LinearLayout  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:orientation="horizontal"  
    android:paddingLeft="15dp"  
    android:paddingRight="15dp"  
    android:layout_marginTop="10dp"  
    android:layout_gravity="center_vertical">  
  
<ImageView  
    android:layout_width="30dp"  
    android:layout_height="30dp"  
    android:src="@drawable/info"/>  
  
<TextView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Status"  
    android:textStyle="bold"  
    android:textColor="@color/black"  
    android:textSize="18dp"  
    android:layout_marginLeft="15dp" />  
  
<EditText  
    android:id="@+id/txtStatus"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_marginLeft="98dp"  
    android:textSize="20dp"  
    android:enabled="false"  
    android:textColor="@android:color/darker_gray"/>  
  
</LinearLayout>
```

```

<View
    android:layout_width="match_parent"
    android:layout_height="0.5dp"
    android:background="@android:color/darker_gray"
    android:layout_marginTop="10dp"
    android:paddingLeft="15dp"
    android:paddingRight="15dp" />

<View
    android:layout_width="match_parent"
    android:layout_height="0.5dp"
    android:background="@android:color/darker_gray"
    android:layout_marginTop="10dp"
    android:paddingLeft="15dp"
    android:paddingRight="15dp" />

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp"
    android:gravity="center"
    android:orientation="vertical">

    <Button
        android:id="@+id/btnSimpan"
        android:layout_width="210dp"
        android:layout_height="62dp"
        android:backgroundTint="#4657B5"
        android:gravity="center"
        android:text="Simpan"
        android:textColor="@color/white"
        android:textSize="15dp" />

    </LinearLayout>
</LinearLayout>

```

**Code 51. Code Edit Profile Pencari**

Dalam coding XML yang digunakan untuk tampilan Edit Profile Pencari, beberapa elemen digunakan, seperti tata letak dan widget. Layout yang digunakan adalah Linear Layout. Linear Layout digunakan untuk mengatur posisi berbagai elemen secara linier, yang dapat vertikal atau horizontal. Kemudian widget yang digunakan adalah text view, image view, edit text dan button. Text View digunakan untuk menampilkan teks dalam aplikasi, dan tampilan gambar digunakan untuk menampilkan gambar dalam aplikasi. Image View yang digunakan pada dashboard diperoleh dari drawable. Edit Text memiliki tujuan yang mirip dengan tampilan teks, tetapi mengedit teks memiliki kemampuan untuk menerima input pengguna. Button di layout ini digunakan untuk konfirmasi, dan jika ditekan, perintah akan dijalankan.

## Code XML Toolbar

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:padding="10dp"
        android:id="@+id/linear_ll"
        android:orientation="horizontal">

        <Button
            android:id="@+id/btnKembali"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="BACK"
            android:backgroundTint="@color/white"
            android:textColor="#465785"
            android:layout_alignParentRight="true"
            android:textStyle="normal"
            android:textSize="15dp"
            />

        <TextView
            android:layout_weight="1"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:text="Akun"
            android:textSize="20dp"
            android:gravity="center"
            android:textStyle="bold"
            android:textColor="@android:color/black"
            />

        <Button
            android:id="@+id/btnKeluar"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentRight="true"
            android:backgroundTint="#DA1010"
            android:text="LOG OUT"
            android:textColor="@android:color/white"
            android:textSize="15dp"
            android:textStyle="normal" />

    </LinearLayout>
    <View
        android:layout_width="match_parent"
        android:layout_height="0.5dp"
        android:background="@color/black"
        android:layout_below="@+id/linear_ll"
        android:paddingRight="8dp"
        android:paddingLeft="8dp"/>
</RelativeLayout>
```

**Code 52. Code XML Toolbar**

Pada coding XML yang digunakan untuk tampilan Toolbar terdapat beberapa elemen yang digunakan seperti layout dan widget. Layout yang digunakan adalah linear layout dan relative layout. Linear layout digunakan untuk mengatur posisi berbagai elemen di dalamnya secara linear baik dalam bentuk vertikal maupun horizontal. Relative Layout merupakan layout yang berfungsi untuk mengatur tata letak komponen atau widget aplikasi android dengan cara relative tidak hanya vertikal atau horisontal saja seperti pada Linear

Layout. Lalu widget yang digunakan adalah text view dan Button. Text View disini digunakan untuk menampilkan tulisan “Akun”. Button yang digunakan pada halaman ini ada dua yaitu Back dan Log Out, Button Back digunakan untuk keluar dari halaman yang sedang digunakan, sedangkan Log Out digunakan untuk kembali ke halaman Login/SignUp.

### Code XML Edit Toolbar

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:padding="10dp"
        android:id="@+id/linear_ll"
        android:orientation="horizontal">

        <Button
            android:id="@+id/btnKembali"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="BACK"
            android:backgroundTint="@color/white"
            android:textColor="#4657B5"
            android:layout_alignParentRight="true"
            android:textStyle="normal"
            android:textSize="15dp"
        />

        <TextView
            android:layout_weight="1"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:text="Pengaturan Akun"
            android:textSize="21dp"
            android:gravity="center"
            android:textStyle="bold"
            android:layout_marginRight="85dp"
            android:textColor="@android:color/black"
        />
    </LinearLayout>

    <View
        android:layout_width="match_parent"
        android:layout_height="0.5dp"
        android:background="@color/black"
        android:layout_below="@+id/linear_ll"
        android:paddingRight="8dp"
        android:paddingLeft="8dp"/>
</RelativeLayout>
```

Code 53. Code XML Edit Toolbar

Pada coding XML yang digunakan untuk tampilan Toolbar terdapat beberapa elemen yang digunakan seperti layout dan widget. Layout yang digunakan adalah linear layout dan relative layout. Linear layout digunakan untuk mengatur posisi berbagai elemen di dalamnya secara linear baik dalam bentuk vertikal maupun horizontal. Relative Layout merupakan layout yang berfungsi untuk mengatur tata letak komponen atau widget aplikasi android dengan cara relative tidak hanya vertikal atau horisontal saja seperti pada Linear Layout. Lalu widget yang digunakan adalah text view dan Button. Text View disini

digunakan untuk menampilkan tulisan “Pengaturan Akun”. Button yang digunakan pada halaman ini ada hanya satu yaitu Button Back. Button Back digunakan untuk keluar dari halaman yang sedang digunakan.

### 4.3 Coding Java

#### Coding Java pada Aplikasi Bagian Log In dan Sign Up

##### Coding Java Main Activity

```
package com.projectK1.kosthost;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class MainActivity extends AppCompatActivity {
    private Button btnLogin, btnSignup;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        btnLogin = findViewById(R.id.btnLogin);
        btnSignup = findViewById(R.id.btnSignup);

        btnLogin.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new Intent(getApplicationContext(), Login.class);
                startActivity(i);
            }
        });
        btnSignup.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new Intent(getApplicationContext(), Signup.class);
                startActivity(i);
            }
        });
    }
}
```

Code 54. Code Java Main Activity

Pada coding Java diatas adalah coding yang digunakan untuk tampilan awal aplikasi yaitu MainActivity yang berisi Log In atau Sign Up bagi Pemilik Kost atau Pencari Kost. Hal pertama yang dilakukan adalah mendeklarasikan button dengan variabel btnLogin dan btnSignup. Setelahnya kita membuat fungsi @Override dengan method OnCreate yang berfungsi untuk melakukan inisialisasi pada saat kondisi awal Activity baru diciptakan, sehingga setContentView secara otomatis akan memperlihatkan halaman activity Main. Kemudian memanggil dua button dari tampilan yang sebelumnya sudah dideklarasikan dengan menggunakan method findViewById dan memasukkan ID button yang telah diinisialisasikan pada file XML. Kedua button ini akan kita set sehingga ketika di-klik akan merujuk ke aktivitas tertentu menggunakan method setOnClickListener. Pada activity ini, button btnLogin dibuat untuk mengarahkan pengguna ke activity Login sehingga diperlukan penggunaan intent yang dipanggil dan disimpan dalam variabel i dan getApplicationContext serta rujukan activity class selanjutnya yaitu Login. startActivity(i) berfungsi untuk memulai activity yang dituju. Sedangkan btnSignup dibuat untuk

mengarahkan pengguna ke activity Signup dengan menggunakan intent yang dipanggil dan disimpan dalam variabel i dan getApplicationContext serta rujukan activity class selanjutnya yaitu Signup. startActivity(i) berfungsi untuk memulai activity yang dituju.

### Coding Java Log In Pemilik/Pencari

```
package com.projectK1.kosthost;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class Login extends AppCompatActivity {
    private Button btnLoginPencari, btnLoginPemilik;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
        btnLoginPencari = findViewById(R.id.btnLoginPencari);
        btnLoginPemilik = findViewById(R.id.btnLoginPemilik);

        btnLoginPencari.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new Intent(getApplicationContext(), LoginPencari.class);
                startActivity(i);
            }
        });

        btnLoginPemilik.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new Intent(getApplicationContext(), LoginPemilik.class);
                startActivity(i);
            }
        });
    }
}
```

Code 55. Code Java Login

Pada coding Java diatas adalah coding yang digunakan untuk tampilan Log In sebagai Pemilik Kost atau Pencari Kost. Hal pertama yang dilakukan adalah mendeklarasikan button dengan variabel btnLoginPencari dan btnLoginPemilik. Selanjutnya, kita membuat fungsi @Override dengan method OnCreate yang berfungsi untuk menginisialisasi kondisi awal Activity baru diciptakan, sehingga setContentView secara otomatis akan memperlihatkan halaman activity Login. Kemudian kita memanggil dua button dari tampilan yang sebelumnya sudah dideklarasikan yaitu btnLoginPemilik dan btnLoginPencari, dengan menggunakan method findViewById dan memasukkan ID button yang telah diinisialisasikan pada file XML sebelumnya. Kedua button ini akan kita set sehingga ketika di-klik akan merujuk ke aktivitas tertentu menggunakan method setOnClickListener. Pada activity ini, button btnLoginPencari dibuat untuk mengarahkan pengguna ke activity Login untuk sisi Pencari sehingga diperlukan penggunaan intent yang dipanggil dan disimpan dalam variabel i dan getApplicationContext yang berfungsi untuk merujuk ke activity class selanjutnya yaitu menuju halaman Login Pencari. startActivity(i) berfungsi untuk memulai activity yang dituju. Sedangkan btnLoginPemilik dibuat untuk mengarahkan pengguna ke activity Login untuk sisi Pemilik sehingga diperlukan penggunaan intent yang dipanggil dan disimpan dalam variabel i dan

getApplicationContext yang berfungsi untuk merujuk ke activity class selanjutnya yaitu menuju halaman Login Pemilik. startActivity(i) berfungsi untuk memulai activity yang dituju.

### Coding Java Sign Up Pemilik/Pencari

```
package com.projectK1.kosthost;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class Signup extends AppCompatActivity {
    private Button btnSignupPencari, btnSignupPemilik;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_signup);
        btnSignupPencari = findViewById(R.id.btnSignupPencari);
        btnSignupPemilik = findViewById(R.id.btnSignupPemilik);

        btnSignupPencari.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new Intent(getApplicationContext(), SignupPencari.class);
                startActivity(i);
            }
        });

        btnSignupPemilik.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new Intent(getApplicationContext(), SignupPemilik.class);
                startActivity(i);
            }
        });
    }
}
```

Code 56. Code Java Sign Up

Pada coding Java diatas adalah coding yang digunakan untuk tampilan Signup sebagai Pemilik Kost maupun Pencari Kost. Hal pertama yang dilakukan adalah mendeklarasikan button dengan variabel btnSignupPencari dan btnSignupPemilik. Kedua button ini akan di-set sehingga ketika di-klik akan merujuk ke aktivitas tertentu menggunakan method setOnClickListener. Pada activity ini, button btnSignupPencari dibuat untuk mengarahkan pengguna ke activity Signup untuk Pencari, dan button btnSignupPemilik dibuat untuk mengarahkan pengguna ke activity Signup untuk Pemilik.

## Coding Java Sign Up Pemilik

```
import ...  
  
public class SignupPemilik extends AppCompatActivity {  
    private EditText edtNamaPemilik, edtTanggallahirPemilik, edtTeleponPemilik, edtEmailPemilik, edtPasswordPemilik;  
    private Button btnDatePicker, btnSignUp;  
    int year, month, dayOfMonth;  
    String namaPemilik, tanggallahir, telepon, email, password;  
    Calendar calendar;  
    DatePickerDialog datePickerDialog;  
    String URL_INSERT PEMILIK = "http://192.168.1.7/kosthostmobile/insert-pemilik.php";  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_signup_pemilik);  
        edtNamaPemilik = findViewById(R.id.edtNamaPemilik);  
        edtTanggallahirPemilik = findViewById(R.id.edtTanggallahirPemilik);  
        edtTeleponPemilik = findViewById(R.id.edtTeleponPemilik);  
        edtEmailPemilik = findViewById(R.id.edtEmailPemilik);  
        edtPasswordPemilik = findViewById(R.id.edtPasswordPemilik);  
        btnDatePicker = findViewById(R.id.btnTimePicker);  
        btnSignUp = findViewById(R.id.btnSignUp);  
  
        btnDatePicker.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View v) {  
                calendar = Calendar.getInstance();  
                year = calendar.get(Calendar.YEAR);  
                month = calendar.get(Calendar.MONTH);  
                dayOfMonth = calendar.get(Calendar.DAY_OF_MONTH);  
                datePickerDialog = new DatePickerDialog(context: SignupPemilik.this,  
                new DatePickerDialog.OnDateSetListener() {  
                    @Override  
                    public void onDateSet(DatePicker datePicker, int year, int month, int day) {  
                        edtTanggallahirPemilik.setText(year + "-" + (month + 1) + "-" + day);  
                    }  
                }, year, month, dayOfMonth);  
                datePickerDialog.show();  
            }  
        });  
  
        btnSignUp.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View v) {  
                public void onClick(View v) {  
                    namaPemilik = edtNamaPemilik.getText().toString();  
                    tanggallahir = edtTanggallahirPemilik.getText().toString();  
                    telepon = edtTeleponPemilik.getText().toString();  
                    email = edtEmailPemilik.getText().toString();  
                    password = edtPasswordPemilik.getText().toString();  
                    insertPemilik(namaPemilik, tanggallahir, telepon, email, password);  
                }  
            }  
        });  
  
        private void insertPemilik(String namaPemilik, String tanggalLahir, String telepon, String email, String password) {  
            RequestQueue queue = Volley.newRequestQueue(context: SignupPemilik.this);  
            StringRequest request = new StringRequest(Request.Method.POST, URL_INSERT PEMILIK, new Response.Listener<String>() {  
                @Override  
                public void onResponse(String response) {  
                    try {  
                        JSONObject jsonObject = new JSONObject(response);  
                        int success = jsonObject.getInt("success");  
                        if (success == 1) {  
                            Toast.makeText(context: SignupPemilik.this, text: "Akun berhasil ditambahkan", Toast.LENGTH_SHORT).show();  
                            Intent i = new Intent(getApplicationContext(), LoginPemilik.class);  
                            startActivity(i);  
                        } else {  
                            Toast.makeText(context: SignupPemilik.this, text: "Akun gagal ditambahkan", Toast.LENGTH_SHORT).show();  
                            edtEmailPemilik.setText("");  
                            edtPasswordPemilik.setText("");  
                        }  
                    } catch (JSONException e) {  
                        Toast.makeText(context: SignupPemilik.this, e.getMessage(), Toast.LENGTH_SHORT).show();  
                    }  
                }  
            }, new Response.ErrorListener() {  
                @Override  
                public void onErrorResponse(VolleyError error) {  
                    String message = null;  
                    if (error instanceof NetworkError) {  
                        message = "Cannot connect to Internet...Please check your connection!";  
                    } else if (error instanceof ServerError) {  
                        message = "The server could not be found. Please try again after some time!!";  
                    } else if (error instanceof AuthFailureError) {  
                        message = "Cannot connect to Internet...Please check your connection!";  
                    } else if (error instanceof ParseError) {  
                        message = "Parsing error! Please try again after some time!!";  
                    } else if (error instanceof NoConnectionError) {  
                        message = "Cannot connect to Internet...Please check your connection!";  
                    }  
                }  
            }  
        }  
    }  
}
```

```

        } else if (error instanceof NoConnectionError) {
            message = "Cannot connect to Internet..Please check your connection!";
        } else if (error instanceof TimeoutError) {
            message = "Connection TimeOut! Please check your internet connection.";
        }
        Toast.makeText( context: SignupPemilik.this, message, Toast.LENGTH_SHORT).show();
    }
}
@Nullable
@Override
protected Map<String, String> getParams() throws AuthFailureError {
    Map<String, String> params = new HashMap<>();
    params.put("namaPemilik", namaPemilik);
    params.put("tanggalLahir", tanggalLahir);
    params.put("telepon", telepon);
    params.put("email", email);
    params.put("password", password);
    return params;
}
};

queue.getCache().clear();
queue.add(request);
}
}

```

**Code 57. Code Java Sign Up Pemilik**

Pada coding Java diatas adalah coding yang digunakan untuk tampilan Sign Up sebagai Pemilik Kost. Hal pertama yang dilakukan adalah mendeklarasikan edit text pemilik dengan variabel edtNamaPemilik, edtTanggalLahirPemilik, edtTeleponPemilik, edtEmailPemilik, edtPasswordPemilik, dan ada btnDatePicker. Setelah selesai diisi, Pemilik kost dapat menekan button dengan variabel btnLogin.

### Coding Java Sign Up Pencari

```

package com.projectK1.kosthost;

import ...

public class SignupPencari extends AppCompatActivity {
    private EditText edtNamaPencari, edtTanggalLahirPencari, edtTeleponPencari, edtEmailPencari, edtPasswordPencari;
    private Button btnDatePicker, btnSignUp;
    int year, month, dayOfMonth;
    String namaPencari, tanggalLahir, telepon, email, password;
    Calendar calendar;
    DatePickerDialog datePickerDialog;
    String URL_INSERT_PENCARI = "http://192.168.1.7/kosthostmobile/insert-pencari.php";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_signup_pencari);
        edtNamaPencari = findViewById(R.id.edtNamaPencari);
        edtTanggalLahirPencari = findViewById(R.id.edtTanggalLahirPencari);
        edtTeleponPencari = findViewById(R.id.edtTeleponPencari);
        edtEmailPencari = findViewById(R.id.edtEmailPencari);
        edtPasswordPencari = findViewById(R.id.edtPasswordPencari);
        btnDatePicker = findViewById(R.id.btnDatePicker);
        btnSignUp = findViewById(R.id.btnSignUp);
    }

    btnDatePicker.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            calendar = Calendar.getInstance();
            year = calendar.get(Calendar.YEAR);
            month = calendar.get(Calendar.MONTH);
            dayOfMonth = calendar.get(Calendar.DAY_OF_MONTH);
            datePickerDialog = new DatePickerDialog( context: SignupPencari.this,
                new DatePickerDialog.OnDateSetListener() {
                    @Override
                    public void onDateSet(DatePicker datePicker, int year, int month, int day) {
                        edtTanggalLahirPencari.setText(year + "-" + (month + 1) + "-" + day);
                    }
                }, year, month, dayOfMonth);
            datePickerDialog.show();
        }
    });

    btnSignUp.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {

```

```

        public void onClick(View v) {
            namaPencari = edtNamaPencari.getText().toString();
            tanggalLahir = edtTanggalLahirPencari.getText().toString();
            telepon = edtTeleponPencari.getText().toString();
            email = edtEmailPencari.getText().toString();
            password = edtPasswordPencari.getText().toString();
            insertPencari(namaPencari, tanggalLahir, telepon, email, password);
        });
    });

    private void insertPencari(String namaPencari, String tglPencari, String telpPencari, String emailPencari, String passwordPencari) {
        RequestQueue queue = Volley.newRequestQueue( context: SignupPencari.this );
        StringRequest request = new StringRequest(Request.Method.POST, URL_INSERT_PENCARI, 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: SignupPencari.this, text: "Akun berhasil ditambahkan", Toast.LENGTH_SHORT).show();
                        Intent i = new Intent(getApplicationContext(), LoginPencari.class);
                        startActivity(i);
                    } else {
                        Toast.makeText( context: SignupPencari.this, text: "Akun gagal ditambahkan", Toast.LENGTH_SHORT).show();
                        edtEmailPencari.setText("");
                        edtPasswordPencari.setText("");
                    }
                } catch (JSONException e) {
                    Toast.makeText( context: SignupPencari.this, e.getMessage(), Toast.LENGTH_SHORT).show();
                }
            }
        }, new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError error) {
                String message = null;
                if (error instanceof NetworkError) {
                    message = "Cannot connect to Internet...Please check your connection!";
                } else if (error instanceof ServerError) {
                    message = "The server could not be found. Please try again after some time!!";
                } else if (error instanceof AuthFailureError) {
                    message = "Cannot connect to Internet...Please check your connection!";
                } else if (error instanceof ParseError) {
                    message = "Parsing error! Please try again after some time!!";
                } else if (error instanceof NoConnectionError) {
                    message = "Cannot connect to Internet...Please check your connection!";
                } else if (error instanceof TimeoutError) {
                    message = "Connection TimeOut! Please check your internet connection.";
                }
                Toast.makeText( context: SignupPencari.this, message, Toast.LENGTH_SHORT).show();
            }
        });
    }

    @Nullable
    @Override
    protected Map<String, String> getParams() throws AuthFailureError {
        Map<String, String> params = new HashMap<>();
        params.put("namaPencari", namaPencari);
        params.put("tanggalahir", tanggalLahir);
        params.put("telepon", telepon);
        params.put("email", email);
        params.put("password", password);
        return params;
    }
};

queue.getCache().clear();
queue.add(request);
}
}

```

#### Code 58. Code Java Sign Up Pencari

Pada coding Java diatas adalah coding yang digunakan untuk tampilan Sign Up sebagai Pencari Kost. Hal pertama yang dilakukan adalah mendeklarasikan edit text pemilik dengan variabel edtNamaPencari, edtTanggalLahirPencari, edtTeleponPencari, edtEmailPencari, edtPasswordPencari, dan ada btnDatePicker. Setelah selesai diisi, Pencari kost dapat menekan button dengan variabel btnLogin.

## Coding Java Log In Pemilik

```
package com.projectK1.kosthost;

import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ListAdapter;
import android.widget.ListView;
import android.widget.SimpleAdapter;
import android.widget.TextView;
import android.widget.Toast;
import com.android.volley.AuthFailureError;
import com.android.volley.NetworkError;
import com.android.volley.NoConnectionError;
import com.android.volley.ParseError;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.ServerError;
import com.android.volley.TimeoutError;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

public class LoginPemilik extends AppCompatActivity {
    private EditText edtEmailPemilik, edtPasswordPemilik;
    private Button btnLogin;
    String URL_LOGIN_PEMILIK = "http://192.168.1.7/kosthostmobile/login-pemilik.php";
    String email, password;
    String IDPemilik;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login_pemilik);
        edtEmailPemilik = findViewById(R.id.edtEmailPemilik);
        edtPasswordPemilik = findViewById(R.id.edtPasswordPemilik);
        btnLogin = findViewById(R.id.btnLogin);

        btnLogin.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                email = edtEmailPemilik.getText().toString();
                password = edtPasswordPemilik.getText().toString();
                login(email, password);
            }
        });
    }

    private void login(String email, String password) {
        RequestQueue queue = Volley.newRequestQueue(this);
        StringRequest request = new StringRequest(Request.Method.POST, URL_LOGIN_PEMILIK,
            new Response.Listener<String>() {
                @Override
                public void onResponse(String response) {
                    try {
                        JSONObject jsonObject = new JSONObject(response);
                        int success = jsonObject.getInt("success");
                        if (success == 1) {
                            Toast.makeText(LoginPemilik.this, "Login berhasil",
                                Toast.LENGTH_SHORT).show();
                            Intent i = new Intent(getApplicationContext(), DashboardPemilik.class);
                            IDPemilik = jsonObject.getString("message");
                            //Toast.makeText(LoginPemilik.this, IDPemilik, Toast.LENGTH_SHORT).show();
                            i.putExtra("IDPemilik", IDPemilik);
                            startActivity(i);
                        } else {
                            Toast.makeText(LoginPemilik.this, "Login gagal, " +
                                "username atau password salah", Toast.LENGTH_SHORT).show();
                        }
                    } catch (JSONException e) {
                        Toast.makeText(LoginPemilik.this, e.getMessage(),
                            Toast.LENGTH_SHORT).show();
                    }
                }
            }
        );
    }
}
```

```
        }, new Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError error) {
        String message = null;
        if (error instanceof NetworkError) {
            message = "Cannot connect to Internet...Please check your connection!";
        } else if (error instanceof ServerError) {
            message = "The server could not be found. Please try again after some time!!";
        } else if (error instanceof AuthFailureError) {
            message = "Cannot connect to Internet...Please check your connection!";
        } else if (error instanceof ParseError) {
            message = "Parsing error! Please try again after some time!!";
        } else if (error instanceof NoConnectionError) {
            message = "Cannot connect to Internet...Please check your connection!";
        } else if (error instanceof TimeoutError) {
            message = "Connection TimeOut! Please check your internet connection..";
        }
        Toast.makeText(context, LoginPemilik.this, message, Toast.LENGTH_SHORT).show();
    }
}) {
    @Nullable
    @Override
    protected Map<String, String> getParams() throws AuthFailureError {
        Map<String, String> params = new HashMap<>();
        params.put("email", email);
        params.put("password", password);
        return params;
    }
};
queue.getCache().clear();
queue.add(request);
}
}
```

## **Code 59. Code Java Login Pemilik**

Pada coding Java diatas adalah coding yang digunakan untuk tampilan Log In sebagai Pemilik Kost. Hal pertama yang dilakukan adalah mendeklarasikan edit text pemilik dengan variabel edtEmailPemilik dan edtPasswordPemilik, kemudian button dengan variabel btnLogin. Lalu URL\_LOGIN\_PEMILIK sebagai lokasi file php yang akan digunakan untuk melakukan CRUD dari android ke dalam database. Berbagai String lainnya juga diinisialisasikan seperti email, password dan IDPemilik. Selanjutnya, kita membuat fungsi @Override dengan method OnCreate yang berfungsi untuk menginisialisasi kondisi awal Activity baru diciptakan, sehingga setContentView secara otomatis akan memperlihatkan halaman activity Login untuk sisi Pemilik. Kemudian kita memanggil dua edit text dari tampilan yang sebelumnya sudah dideklarasikan yaitu edtEmailPemilik dan edtPasswordPemilik, serta button dengan menggunakan method findViewById dan memasukkan ID edit text dan ID button yang telah diinisialisasikan pada file XML sebelumnya. Kedua edit text ini merupakan sebuah field dimana kita dapat mengetikkan text dan fungsi edit text akan menangkap value text yang sudah kita input untuk mengecheck data, sedangkan button btnLogin akan kita set sehingga ketika diklik akan merujuk ke aktivitas tertentu menggunakan method setOnClickListener. Pada activity ini, button btnLogin dibuat dengan method onClick pada view v sehingga ketika button Login diklik, value email dan password akan di dapatkan, seperti yang dapat kita lihat pada baris selanjutnya kita menggunakan get.Text().toString(); untuk menyimpan nilai text yang sudah kita input ke dalam bentuk string.

Lalu kita menginisialisasi untuk kelas login menggunakan private void. RequestQueue dan StringRequest digunakan untuk menghubungkan coding yang kita buat dengan URL lokasi file php agar dapat terkoneksi dengan database MySQL. OnResponse digunakan sebagai penampung dari hasil proses di dalamnya yang meliputi try dan catch untuk mendapatkan data dari variabel yang dikirim dari file php seperti IDPemilik, email

dan password. Jika variabel memang cocok dengan yang sudah ada di database, maka login akan berhasil. Namun jika data tidak cocok atau belum ada sebelumnya, maka login akan gagal. Penyebab gagalnya login ini bisa karena salah memasukkan email atau password juga.

Dibawahnya kita juga membuat JSONException e untuk menampilkan pesan berupa toast singkat sehingga kita dapat mengetahui jika ada error yang muncul pada saat proses dijalankan. onErrorResponse bermanfaat untuk memberikan respon tertentu seperti toast yang muncul pada halaman login ini terhadap error yang didapatkan. Kita menggunakan fitur VolleyError untuk membantu menampung sementara jika ada error. Selanjutnya, getParams() digunakan untuk mengirimkan variabel yang diperlukan dalam file php untuk melakukan query. Didalamnya kita menggunakan kelas Hashmap yang akan menangkap nilai params String. Dalam file ini diperlukan email dan password agar query MySQL pada file php dapat berjalan dengan baik yaitu mendapatkan data pemilik berdasarkan email dan password yang dimasukkan ketika ingin login. Bagian akhir, harus ditambahkan queue.getCache().clear(); dan queue.add(request); agar program dapat berjalan dan terupdate tampilannya.

### Coding Java Log In Pencari

```
package com.projectK1.kosthost;

import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ListAdapter;
import android.widget.ListView;
import android.widget.SimpleAdapter;
import android.widget.TextView;
import android.widget.Toast;
import com.android.volley.AuthFailureError;
import com.android.volley.NetworkError;
import com.android.volley.NoConnectionError;
import com.android.volley.ParseError;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.ServerError;
import com.android.volley.TimeoutError;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
```

```

public class LoginPencari extends AppCompatActivity {
    private EditText edtEmailPencari, edtPasswordPencari;
    private Button btnLogin;
    String URL_LOGIN_PENCARI = "http://192.168.1.7/kosthostmobile/login-pencari.php";
    String email, password;
    String IDPencari;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login_pencari);
        edtEmailPencari = findViewById(R.id.edtEmailPencari);
        edtPasswordPencari = findViewById(R.id.edtPasswordPencari);
        btnLogin = findViewById(R.id.btnLogin);

        btnLogin.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                email = edtEmailPencari.getText().toString();
                password = edtPasswordPencari.getText().toString();
                login(email, password);
            }
        });
    }

    private void login(String email, String password) {
        RequestQueue queue = Volley.newRequestQueue(this);
        StringRequest request = new StringRequest(Request.Method.POST, URL_LOGIN_PENCARI,
            new Response.Listener<String>() {
                @Override
                public void onResponse(String response) {
                    try {
                        JSONObject jsonObject = new JSONObject(response);
                        int success = jsonObject.getInt("success");
                        if (success == 1) {
                            Toast.makeText(LoginPencari.this, "Login berhasil",
                                Toast.LENGTH_SHORT).show();
                            Intent i = new Intent(getApplicationContext(), DashboardPencari.class);
                            IDPencari = jsonObject.getString("message");
                            //Toast.makeText(LoginPencari.this, IDPencari, Toast.LENGTH_SHORT).show();
                            i.putExtra("IDPencari", IDPencari);
                            startActivity(i);
                        } else {
                            Toast.makeText(LoginPencari.this, "Login gagal," +
                                "username atau password salah", Toast.LENGTH_SHORT).show();
                        }
                    } catch (JSONException e) {
                        Toast.makeText(LoginPencari.this, e.getMessage(), Toast.LENGTH_SHORT).show();
                    }
                }
            }, new Response.ErrorListener() {
                @Override
                public void onErrorResponse(VolleyError error) {
                    String message = null;
                    if (error instanceof NetworkError) {
                        message = "Cannot connect to Internet...Please check your connection!";
                    } else if (error instanceof ServerError) {
                        message = "The server could not be found. Please try again after some time!!";
                    } else if (error instanceof AuthFailureError) {
                        message = "Cannot connect to Internet...Please check your connection!";
                    } else if (error instanceof ParseError) {
                        message = "Parsing error! Please try again after some time!!";
                    } else if (error instanceof NoConnectionError) {
                        message = "Cannot connect to Internet...Please check your connection!";
                    } else if (error instanceof TimeoutError) {
                        message = "Connection TimeOut! Please check your internet connection..";
                    }
                    Toast.makeText(LoginPencari.this, message, Toast.LENGTH_SHORT).show();
                }
            });
        queue.getCache().clear();
        queue.add(request);
    }
}

```

#### Code 60. Code Java Login Pencari

Pada coding Java diatas adalah coding yang digunakan untuk tampilan Log In sebagai Pencari Kost. Hal pertama yang dilakukan adalah mendeklarasikan edit text pemilik dengan variabel edtEmailPencari dan edtPasswordPencari, kemudian button

dengan variabel btnLogin. Lalu URL\_LOGIN\_PENCARI sebagai lokasi file php yang akan digunakan untuk melakukan CRUD dari android ke dalam database. Berbagai String lainnya juga diinisialisasikan seperti email, password dan IDPencari. Selanjutnya, kita membuat fungsi @Override dengan method OnCreate yang berfungsi untuk menginisialisasi kondisi awal Activity baru diciptakan, sehingga setContentView secara otomatis akan memperlihatkan halaman activity Login untuk sisi Pemilik. Kemudian kita memanggil dua edit text dari tampilan yang sebelumnya sudah dideklarasikan yaitu edtEmailPencari dan edtPasswordPencari, serta button dengan menggunakan method findViewById dan memasukkan ID edit text dan ID button yang telah diinisialisasikan pada file XML sebelumnya. Kedua edit text ini merupakan sebuah field dimana kita dapat mengetikkan text dan fungsi edit text akan menangkap value text yang sudah kita input untuk mengecheck data, sedangkan button btnLogin akan kita set sehingga ketika di-klik akan merujuk ke aktivitas tertentu menggunakan method setOnClickListener. Pada activity ini, button btnLogin dibuat dengan method onClick pada view v sehingga ketika button Login diklik, value email dan password akan di dapatkan, seperti yang dapat kita lihat pada baris selanjutnya kita menggunakan get.Text().toString(); untuk menyimpan nilai text yang sudah kita input ke dalam bentuk string.

Lalu kita menginisialisasi untuk kelas login menggunakan private void. RequestQueue dan StringRequest digunakan untuk menghubungkan coding yang kita buat dengan URL lokasi file php agar dapat terkoneksi dengan database MySQL. OnResponse digunakan sebagai penampung dari hasil proses di dalamnya yang meliputi try dan catch untuk mendapatkan data dari variabel yang dikirim dari file php seperti IDPencari, email dan password. Jika variabel memang cocok dengan yang sudah ada di database, maka login akan berhasil. Namun jika data tidak cocok atau belum ada sebelumnya, maka login akan gagal. Penyebab gagalnya login ini bisa karena salah memasukkan email atau password juga.

Dibawahnya kita juga membuat JSONException e untuk menampilkan pesan berupa toast singkat sehingga kita dapat mengetahui jika ada error yang muncul pada saat proses dijalankan. onErrorResponse bermanfaat untuk memberikan respon tertentu seperti toast yang muncul pada halaman login ini terhadap error yang didapatkan. Kita menggunakan fitur VolleyError untuk membantu menampung sementara jika ada error. Selanjutnya, getParams() digunakan untuk mengirimkan variabel yang diperlukan dalam file php untuk melakukan query. Didalamnya kita menggunakan kelas Hashmap yang akan menangkap nilai params String. Dalam file ini diperlukan email dan password agar query MySQL pada file php dapat berjalan dengan baik yaitu mendapatkan data pemilik berdasarkan email dan password yang dimasukkan ketika ingin login. Bagian akhir, harus ditambahkan queue.getCache().clear(); dan queue.add(request); agar program dapat berjalan dan terupdate tampilannya.

## Coding Java pada Aplikasi Bagian Pemilik Kost

### Coding Java Dashboard

```
package com.projectK1.kosthost;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageButton;

public class DashboardPemilik extends AppCompatActivity {
    private Button btnKostSaya, btnBookedKost, btnPenjagaKost, btnFAQPemilik, btnProfile;
    private ImageButton btnKH;
    String IDPemilik;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_dashboard_pemilik);

        btnKostSaya = findViewById(R.id.btnKostSaya);
        btnBookedKost = findViewById(R.id.btnBookedKost);
        btnPenjagaKost = findViewById(R.id.btnPenjagaKost);
        btnFAQPemilik = findViewById(R.id.btnFAQPemilik);
        btnProfile = findViewById(R.id.btnProfile);
        Intent i = getIntent();
        IDPemilik = i.getStringExtra("IDPemilik");

        btnKostSaya.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new Intent(getApplicationContext(), KostPemilik.class);
                i.putExtra("IDPemilik", IDPemilik);
                startActivity(i);
            }
        });
        btnBookedKost.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new Intent(getApplicationContext(), BookedPemilik.class);
                i.putExtra("IDPemilik", IDPemilik);
                startActivity(i);
            }
        });
        btnPenjagaKost.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new Intent(getApplicationContext(), Penjaga.class);
                i.putExtra("IDPemilik", IDPemilik);
                startActivity(i);
            }
        });
        btnFAQPemilik.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new Intent(getApplicationContext(), faqPemilik.class);
                i.putExtra("IDPemilik", IDPemilik);
                startActivity(i);
            }
        });
        btnProfile.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new Intent(getApplicationContext(), ProfilePemilik.class);
                i.putExtra("IDPemilik", IDPemilik);
                startActivity(i);
            }
        });

        btnKH = findViewById(R.id.btnKH);
        btnKH.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new Intent(getApplicationContext(), DashboardPemilik.class);
                i.putExtra("IDPemilik", IDPemilik);
                startActivity(i);
            }
        });
    }
}
```

**Code 61. Code Java Dashboard Pemilik Kost**

Pada coding Java diatas adalah coding yang digunakan untuk tampilan Dashboard Pemilik. Hal pertama yang dilakukan adalah mendeklarasikan Button btnKostSaya, btnBookedKost, btnPenjagaKost, btnFAQPemilik, btnProfile, dan ImageButton btnKH.

Setelahnya kita membuat fungsi @Override dengan method OnCreate yang berfungsi untuk melakukan inisialisasi pada saat kondisi awal Activity baru diciptakan, sehingga setContentView secara otomatis akan memperlihatkan halaman activity DashboardPemilik. Get Intent digunakan untuk mengambil beberapa data dari variabel dari activity sebelumnya yang dikirim. Variabel yang diterima di activity ini adalah IDPemilik yang didapatkan dari LoginPemilik. Find view by id digunakan untuk melakukan inisialisasi variabel yang sudah dideklarasikan di awal agar dapat terkoneksi dengan komponen-komponen dalam activity ini.

Setiap button yang sudah dideklarasikan dibuatkan setOnClickListener agar dapat menjalankan perintah tertentu. btnKostSaya jika ditekan akan menampilkan activity KostPemilik. btnBookedKost jika ditekan akan menampilkan activity BookedPemilik. btnPenjagaKost jika ditekan akan menampilkan activity Penjaga. btnFAQPemilik jika ditekan akan menampilkan activity faqpemilik. btnProfile jika ditekan akan menampilkan activity ProfilePemilik. btnKH jika ditekan akan menampilkan activity DashboardPencari kembali. Setiap menjalankan activity baru, intent IDPemilik dikirimkan agar data di setiap activity sesuai dengan IDPemilik yang telah log in.

## Coding Java Menu My Kost

### Coding Java My Kost

```
package com.projectK1.kosthost;

import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ImageButton;
import android.widget.ListAdapter;
import android.widget.ListView;
import android.widget.SimpleAdapter;
import android.widget.TextView;
import android.widget.Toast;

import com.android.volley.AuthFailureError;
import com.android.volley.NetworkError;
import com.android.volley.NoConnectionError;
import com.android.volley.ParseError;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.ServerError;
import com.android.volley.TimeoutError;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
```

```

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.text.NumberFormat;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Locale;
import java.util.Map;

public class KostPemilik extends AppCompatActivity {
    private ListView listKost;
    private ImageButton btnTambahKost, btnKost;
    ArrayList<HashMap<String, String>> listkost;
    String URL_GET_KOST = "http://192.168.1.7/kosthostmobile/getmykost-pemilik.php";
    String IDKost, namaKost, alamatKost, kotaKost, jenisKost, jumlahKamar, hargaKost,
        deskripsiFasilitas, fotoKost;
    String IDPemilik;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_kost_pemilik);
        Intent i = getIntent();
        IDPemilik = i.getStringExtra( name: "IDPemilik");

        btnTambahKost = findViewById(R.id.btnAddKost);
        btnTambahKost.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new Intent(getApplicationContext(), TambahKost.class);
                i.putExtra( name: "IDPemilik", IDPemilik);
                startActivity(i);
            }
        });
    }

    listKost = findViewById(R.id.listKost);
    listkost = new ArrayList<>();
    RequestQueue queue = Volley.newRequestQueue( context: KostPemilik.this);

    StringRequest request = new StringRequest(Request.Method.POST, URL_GET_KOST,
        new Response.Listener<String>() {
            @Override
            public void onResponse(String response) {
                try {
                    JSONObject jsonObject = new JSONObject(response);
                    JSONArray jsonArray = jsonObject.getJSONArray( name: "data");

                    for (int i = 0; i < jsonArray.length(); i++) {
                        JSONObject kost = jsonArray.getJSONObject(i);
                        IDKost = kost.getString( name: "IDkost");
                        namaKost = kost.getString( name: "namaKost");
                        alamatKost = kost.getString( name: "alamatKost");
                        kotaKost = kost.getString( name: "kotakost");
                        jenisKost = kost.getString( name: "jenisKost");
                        jumlahKamar = kost.getString( name: "jumlahKamar");
                        hargaKost = kost.getString( name: "hargaKost");
                        fotoKost = kost.getString( name: "fotoKost");

                        double dhargaKost = Double.parseDouble(hargaKost);
                        Locale localeID = new Locale( language: "in", country: "ID");
                        NumberFormat formatRupiah = NumberFormat.getCurrencyInstance(localeID);
                        String hargaKostFormat = formatRupiah.format(dhargaKost);
                        deskripsiFasilitas = kost.getString( name: "deskripsiFasilitas");
                    }
                } catch (JSONException e) {
                    e.printStackTrace();
                }
            }
        });
}

```

```

        HashMap<String, String> map = new HashMap<>();
        map.put("IDKost", IDKost);
        map.put("namaKost", namaKost);
        map.put("alamatKost", alamatKost);
        map.put("kotaKost", kotaKost);
        map.put("jenisKost", jenisKost);
        map.put("jumlahKamar", jumlahKamar);
        map.put("hargaKost", hargaKost);
        map.put("hargaKostFormat", hargaKostFormat);
        map.put("deskripsiFasilitas", deskripsiFasilitas);
        map.put("fotoKost", fotoKost);
        listKost.add(map);

        String[] from = {IDKost, namaKost, alamatKost, kotaKost, jenisKost, jumlahKamar,
                        hargaKost, hargaKostFormat, deskripsiFasilitas, fotoKost};
        int[] to = {R.id.txtIDKost, R.id.txtNamaKost, R.id.txtAlamat, R.id.txtKota, R.id.txtJenis, R.id.txtJumlahKamar,
                   R.id.txtHargaAsli, R.id.txtHarga, R.id.txtFasilitas, R.id.txtFotoKost};

       ListAdapter adapter = new SimpleAdapter( context: KostPemilik.this, listKost, R.layout.list_kost, from, to);
        listKost.setAdapter(adapter);
    }
} catch (JSONException e) {
    Toast.makeText( context: KostPemilik.this, e.getMessage(), Toast.LENGTH_SHORT).show();
}
}

}, new Response.ErrorListener() {
@Override
public void onErrorResponse(VolleyError error) {
    String message = null;
    if (error instanceof NetworkError) {
        message = "Cannot connect to Internet...Please check your connection!";
    } else if (error instanceof ServerError) {
        message = "The server could not be found. Please try again after some time!!";
    } else if (error instanceof AuthFailureError) {
        message = "Cannot connect to Internet...Please check your connection!";
    } else if (error instanceof ParseError) {
        message = "Parsing error! Please try again after some time!!!";
    } else if (error instanceof NoConnectionError) {
        message = "Cannot connect to Internet...Please check your connection!";
    } else if (error instanceof TimeoutError) {
        message = "Connection TimeOut! Please check your internet connection.";
    }
    Toast.makeText( context: KostPemilik.this, message, Toast.LENGTH_SHORT).show();
}
}

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

```

```

        queue.getCache().clear();
        queue.add(request);
    }

    listKost.setOnItemClickListener(new AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
            String get_nama = ((TextView) view.findViewById(R.id.txtKost)).getText().toString();
            String get_alamat = ((TextView) view.findViewById(R.id.txtAlamat)).getText().toString();
            String get_kota = ((TextView) view.findViewById(R.id.txtKota)).getText().toString();
            String get_jenis = ((TextView) view.findViewById(R.id.txtJenis)).getText().toString();
            String get_jumlahKamar = ((TextView) view.findViewById(R.id.txtJumlahKamar)).getText().toString();
            String get_fasilitas = ((TextView) view.findViewById(R.id.txtFasilitas)).getText().toString();
            String get_harga = ((TextView) view.findViewById(R.id.txtHargaAsli)).getText().toString();
            String get_idkost = ((TextView) view.findViewById(R.id.txtIDKost)).getText().toString();
            String get_fotoKost = ((TextView) view.findViewById(R.id.txtfotoKost)).getText().toString();

            Intent i = new Intent( mContext, EditKost.class);
            i.putExtra( name: "namaKost", get_nama);
            i.putExtra( name: "alamatKost", get_alamat);
            i.putExtra( name: "kotaKost", get_kota);
            i.putExtra( name: "jenisKost", get_jenis);
            i.putExtra( name: "jumlahKamar", get_jumlahKamar);
            i.putExtra( name: "hargaKost", get_harga);
            i.putExtra( name: "deskripsiFasilitas", get_fasilitas);
            i.putExtra( name: "IDKost", get_idkost);
            i.putExtra( name: "IDPemilik", IDPemilik);
            i.putExtra( name: "fotoKost", get_fotoKost);
            startActivity(i);
        }
    });

    btnKH = findViewById(R.id.btnKH);
    btnKH.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent i = new Intent(getApplicationContext(), DashboardPemilik.class);
            i.putExtra( name: "IDPemilik", IDPemilik);
            startActivity(i);
            KostPemilik.this.finish();
        }
    });
}

```

**Code 62. Code Java Kost Pemilik**

Pada coding Java diatas adalah coding yang digunakan untuk tampilan My Kost pada sisi pemilik kost. Hal pertama yang dilakukan adalah mendeklarasikan ListView dengan variabel listKost, ImageButton dengan variabel btnKH dan variabel btnTambahKost. Lalu inisialisasi ArrayList<HashMap<String, String>> dengan nama listkost yang bertujuan untuk tempat penyimpanan berbagai data String yang nantinya akan ditampilkan ke dalam listkost. Terdapat URL\_GET\_KOST sebagai lokasi file php yang akan digunakan untuk melakukan CRUD dari android ke dalam database. Berbagai String lainnya juga diinisialisasikan seperti IDKost, IDPemilik, namaKost, alamatKost, kotaKost, jenisKost, jumlahKamar, hargaKost, deskripsiFasilitas, dan fotoKost. Setelahnya kita membuat fungsi @Override dengan method OnCreate yang berfungsi untuk melakukan inisialisasi pada saat kondisi awal Activity baru diciptakan, sehingga setContentView secara otomatis akan memperlihatkan halaman activity kost pemilik. Get Intent digunakan untuk mengambil beberapa data dari variabel dari activity sebelumnya yang dikirim. Variabel string yang diterima di activity ini adalah IDPemilik yang didapatkan dari Dashboard Pemilik. Find view by id digunakan untuk melakukan insialisasi variabel yang sudah dideklarasikan di awal agar dapat terkoneksi dengan komponen dalam activity ini. Kali ini kami memanggil button btnTambahKost dengan findViewById sesuai dengan id

yang sudah di set dalam XML sebelumnya. btnTambahKost kemudian kita setOnTouchListener sehingga ketika di klik (OnClick) intent nya akan mengarahkan ke activity tambah kost. Lalu memanggil string IDPemilik dengan i.putExtra. Intent kemudian diakhiri dengan startActivity(i) agar intent dapat berjalan.

Berikutnya kita akan menginisialisasi listKost dengan findViewById, array list listkost juga diinisialisasikan. RequestQueue dan StringRequest digunakan untuk menghubungkan coding yang kita buat dengan URL lokasi file php agar dapat terkoneksi dengan database MySQL. OnResponse digunakan sebagai penampung dari hasil proses di dalamnya. Proses di dalamnya meliputi try dan catch untuk get String yang bertujuan mendapatkan semua variabel data yang dikirimkan dari file php, seperti IDKost, namaKost, alamatKost, kotaKost, jenisKost, jumlahKamar, hargaKost dan fotoKost. Untuk hargaKost, kita melakukan inisialisasi tambahan berupa Double.parseDouble untuk mengembalikan nilai yang direpresentasikan oleh hargaKost. Lalu menggunakan kelas Locale yaitu untuk mengubah format angka pada harga kost sesuai lokasi yang diinput yaitu Indonesia kemudian getCurrencyInstance agar dapat berubah menjadi format Indonesia rupiah.

Setelahnya kita akan getString deskripsiFasilitas, setelah didapatkan dalam suatu JSONObject, dengan menggunakan HashMap lalu dilakukan put ke dalam variabel-variabel yang akan disimpan ke dalam array listkost pada activity ini hashmap berisi IDKost, namaKost, alamatKost, kotaKost, jenisKost, jumlahKamar, hargaKost, hargaKostFormat, deskripsiFasilitas, dan fotoKost. Berbagai data tersebut disimpan ke dalam suatu adapter untuk ditampilkan di listkost. String[] from dan int[] to harus saling berhubungan untuk menandakan dari mana data yang diambil dan ke mana data tersebut akan ditampilkan. Data akan ditampilkan ke dalam layout list\_kost. List\_kost berisi berbagai komponen yang akan ditampilkan ke dalam listkost dengan adapter.

Dibawahnya kita juga membuat JSONException e untuk menampilkan pesan berupa toast singkat sehingga kita dapat mengetahui jika ada error yang muncul pada saat proses dijalankan. onErrorResponse bermanfaat untuk memberikan respon spesifik seperti toast terhadap error yang didapatkan. Sedangkan di bawahnya, getParams() digunakan untuk mengirimkan variabel yang diperlukan dalam file php untuk melakukan query. Dalam activity ini diperlukan IDPemilik agar query MySQL pada file php dapat berjalan dengan baik yaitu mendapatkan data kost berdasarkan IDPemilik yang login. Bagian akhir, harus ditambahkan queue.getCache().clear(); dan queue.add(request); agar program dapat berjalan dan selalu terupdate tampilannya.

Selanjutnya adalah menetapkan setOnTouchListener untuk listKost, jika listKost ditekan maka akan melakukan intent ke activity EditKost dengan mengirimkan berbagai variabel yang dikirimkan pada string get, contohnya nama, alamat, kota, jenis, jumlahKamar, fasilitas, harga, idkost, dan fotoKost. Terdapat setOnTouchListener juga untuk btnKH sehingga ketika logo KH ditekan (OnClick), pengguna akan diarahkan untuk kembali ke dashboard pemilik. Semua intent yang digunakan harus diakhiri dengan

startActivity(i) agar intent dapat berjalan. Pada akhir java tertulis KostPemilik.this.finish(); yang akan terpanggil ketika kita menekan tombol sehingga activity yang sekarang akan di destroy dan kembali ke activity sebelumnya.

## Coding Java Tambah Kost

```

package com.projectk1.kosthost;

import ...

public class TambahKost extends AppCompatActivity {
    private EditText edtNama, edtHarga, edtAlamat, edtJumlahKamarKost, edtFasilitasKost, edtIDPemilikKost;
    private Spinner spinKotaKost, spinJenisKost;
    private Button btnTambahKost, btnChoose;
    private ImageView imageKost;
    Bitmap bitmap;
    final int CODE_GALLERY_REQUEST = 999;
    private String[] kota = {"-Pilih Kota-", "Jakarta", "Tangerang", "Bogor"};
    private String[] tipeKost = {"-Pilih Jenis Kost-", "Kost Putra", "Kost Putri", "Kost Campur"};
    private String namakost, hargakost, alamatKost, kotakost, jenisKost, jumlahKamar, deskripsiFasilitas, IDPemilik, fotoKost;
    String URL_INSERT_KOST = "http://192.168.1.7/kosthostmobile/insert-kost.php";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_tambah_kost);

        Intent i = getIntent();
        IDPemilik = i.getStringExtra("IDPemilik");
        edtIDPemilikKost = findViewById(R.id.edtIDPemilikKost);
        edtIDPemilikKost.setText(IDPemilik);
        btnChoose = findViewById(R.id.btnChoose);
        imageKost = findViewById(R.id.imageKost);

        edtNama = findViewById(R.id.edtNama);
        edtHarga = findViewById(R.id.edtHarga);
        edtAlamat = findViewById(R.id.edtAlamat);
        edtJumlahKamarKost = findViewById(R.id.edtJumlahKamarKost);
        edtFasilitasKost = findViewById(R.id.edtFasilitasKost);

        btnChoose.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                ActivityCompat.requestPermissions(
                    TambahKost.this,
                    new String[]{Manifest.permission.READ_EXTERNAL_STORAGE},
                    CODE_GALLERY_REQUEST
                );
            }
        });
    });

    spinKotaKost = findViewById(R.id.spinKotaKost);
    ArrayAdapter adapterKota = new ArrayAdapter<(context: this, android.R.layout.simple_spinner_item, kota);
    spinKotaKost.setAdapter(adapterKota);

    spinKotaKost.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
        @Override
        public void onItemSelected(AdapterView<> parent, View view, int position, long id) {
            String value = parent.getItemAtPosition(position).toString();
            kotakost = value;
        }
    });

    @Override
    public void onNothingSelected(AdapterView<> parent) {

    });
};

    spinJenisKost = findViewById(R.id.spinJenisKost);
    ArrayAdapter adapterJenisKost = new ArrayAdapter<(context: this, android.R.layout.simple_spinner_item, tipeKost);
    spinJenisKost.setAdapter(adapterJenisKost);

    spinJenisKost.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
        @Override
        public void onItemSelected(AdapterView<> parent, View view, int position, long id) {
            String value = parent.getItemAtPosition(position).toString();
            jenisKost = value;
        }
    });

    @Override
    public void onNothingSelected(AdapterView<> parent) {
}
}

```

```

btnTambahKost = findViewById(R.id.btnAddKost);
btnTambahKost.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        namakost = editNama.getText().toString();
        hargakost = editHarga.getText().toString();
        alamatkost = editAlamat.getText().toString();
        jumlahkamar = editJumlahKamarKost.getText().toString();
        deskripsiFasilitas = editFasilitaskost.getText().toString();
        IDPenilik = editIDPenilikKost.getText().toString();

        tambahKost(namakost, alamatkost, kotaKost, jeniskost, jumlahkamar, hargakost, deskripsiFasilitas, IDPenilik);
    }
});

@Override
public void onRequestPermissionsResult(int requestCode, @NotNull String[] permissions, @NotNull int[] grantResults) {
    if (requestCode == CODE_GALLERY_REQUEST) {
        if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
            Intent intent = new Intent(Intent.ACTION_PICK);
            intent.setType("image/*");
            startActivityForResult(intent.createChooser(intent, "Select Image"), CODE_GALLERY_REQUEST);
        } else {
            Toast.makeText(getApplicationContext(), "Tidak ada akses ke gallery", Toast.LENGTH_LONG).show();
        }
        return;
    }
    super.onRequestPermissionsResult(requestCode, permissions, grantResults);
}

@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
    if (requestCode == CODE_GALLERY_REQUEST && resultCode == RESULT_OK && data != null) {
        Uri filePath = data.getData();

        try {
            InputStream inputStream = getContentResolver().openInputStream(filePath);
            bitmap = BitmapFactory.decodeStream(inputStream);
            imageKost.setImageBitmap(bitmap);
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        }
    }
}

super.onActivityResult(requestCode, resultCode, data);
}

private void tambahKost(final String namakost, final String alamatkost, final String kotaKost, final String jeniskost, final String jumlahkamar, final String hargakost, final String deskripsiFasilitas, final String IDPenilik) {
    RequestQueue queue = Volley.newRequestQueue(TambahKost.this);
    StringRequest request = new StringRequest(Request.Method.POST, URL_INSERT_KOST, new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            try {
                JSONObject jsonObject = new JSONObject(response);
                int success = jsonObject.getInt("success");
                if (success == 1) {
                    Toast.makeText(TambahKost.this, "Kost berhasil ditambahkan", Toast.LENGTH_SHORT).show();
                    Intent i = new Intent(getApplicationContext(), KostPenilik.class);
                    i.putExtra("IDpenilik", IDPenilik);
                    startActivity(i);
                } else {
                    Toast.makeText(TambahKost.this, "Kost gagal ditambahkan", Toast.LENGTH_SHORT).show();
                }
            } catch (JSONException e) {
                Toast.makeText(TambahKost.this, e.getMessage(), Toast.LENGTH_SHORT).show();
            }
        }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            String message = null;
        }
    });
}

```

```

    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            String message = null;
            if (error instanceof NetworkError) {
                message = "Cannot connect to Internet...Please check your connection!";
            } else if (error instanceof ServerError) {
                message = "The server could not be found. Please try again after some time!";
            } else if (error instanceof AuthFailureError) {
                message = "Cannot connect to Internet...Please check your connection!";
            } else if (error instanceof ParseError) {
                message = "Parsing error! Please try again after some time!";
            } else if (error instanceof NoConnectionError) {
                message = "Cannot connect to Internet...Please check your connection!";
            } else if (error instanceof TimeoutError) {
                message = "Connection TimeOut! Please check your internet connection.";
            }
            Toast.makeText(context, TambahKost.this, message, Toast.LENGTH_SHORT).show();
        }
    });
}
@Nullable
@Override
protected Map<String, String> getParams() throws AuthFailureError {
    Map<String, String> params = new HashMap<>();
    fotoKost = imageToString(bitmap);
    params.put("namakost", namakost);
    params.put("alamatkost", alamatkost);
    params.put("kotakost", kotakost);
    params.put("jeniskost", jeniskost);
    params.put("jumlahKamar", jumlahKamar);
    params.put("hargakost", hargakost);
    params.put("deskripsiFasilitas", deskripsiFasilitas);
    params.put("IDPemilik", IDPemilik);
    params.put("fotoKost", fotoKost);
    return params;
}
queue.getCache().clear();
queue.add(request);
}

private String imageToString(Bitmap bitmap) {
    ByteArrayOutputStream outputStream = new ByteArrayOutputStream();
    bitmap.compress(Bitmap.CompressFormat.JPEG, quality, outputStream);
    byte[] imageBytes = outputStream.toByteArray();

    String encodedImage = Base64.encodeToString(imageBytes, android.util.Base64.DEFAULT);
    return encodedImage;
}
}

```

### Code 63. Code Java Tambah Kost

Code diatas merupakan code yang digunakan untuk pemilik menambah kost mereka. Hal pertama adalah mendeklarasikan EditText dengan id edtNama, edtHarga, edtAlamat, edtJumlahKamarKost, edtFasilitasKost, edtIDKost. Selanjutnya mendeklarasikan Spinner dengan id spinKotaKost, spinJenisKost. Lalu mendeklarasikan button dengan id btnEditKost, btnDeleteKost, btnChoose. Kemudian mendeklarasikan ImageView dengan id imageKost. Lalu terdapat Bitmap dengan id bitmap yang berhubungan dengan imageKost. Kemudian final int Code\_Gallery\_Request yang digunakan untuk membaca data external. Lalu terdapat 2 String, yang pertama ada String Kota yang terdapat dalam spinKotaKost yang digunakan untuk memilih kota. Yang kedua adalah String Jenis yang terdapat dalam spinJenisKost yang digunakan untuk memilih jenis kost. Terdapat 3 URL, pertama adalah URL\_FOTO\_KOST sebagai lokasi file php yang akan digunakan untuk mendapatkan foto kost dari database. URL\_EDIT\_KOST sebagai lokasi file php yang akan digunakan untuk melakukan EDIT dari android ke dalam database. URL\_DELETE\_KOST sebagai lokasi file hp yang akan digunakan untuk melakukan DELETE dari android ke dalam database. Berbagai String juga dipanggil

seperti IDKost, namaKost, hargaKost, alamatKost, kotaKost, jumlahKamar, deskripsiFasilitas, jenisKost, fotoKost. Kemudian menginisialisasikan String IDPemilik.

## Coding Java Edit Kost

```
package com.projectk1.kosthost;

import ...

public class EditKost extends AppCompatActivity {
    private EditText edtNama, edtHarga, edtAlamat, edtJumlahKamarKost, edtFasilitasKost, edtIDKost;
    private Spinner spinKotaKost, spinJeniskost;
    private Button btnEditKost, btnDeleteKost, btnChoose;
    private ImageView imageKost;
    Bitmap bitmap;
    final int CODE_GALLERY_REQUEST = 999;
    private String[] kota = {"Pilih Kota-", "Jakarta", "Tangerang", "Bogor"};
    private String[] jenis = {"Pilih Jenis Kost-", "Kost Putra", "Kost Putri", "Kost Campur"};
    private String IDkost, namakost, hargakost, alamatkost, kotaKost, jumlahKamar, deskripsiFasilitas, jenisKost, fotoKost;
    String IDPemilik;
    String URL_FOTO_KOST;

    String URL_EDIT_KOST = "http://192.168.1.7/kosthostmobile/edit-kost.php";
    String URL_DELETE_KOST = "http://192.168.1.7/kosthostmobile/delete-kost.php";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_edit_kost);

        edtIDKost = findViewById(R.id.edtIDKost);
        edtNama = findViewById(R.id.edtNama);
        edtHarga = findViewById(R.id.edtHarga);
        edtAlamat = findViewById(R.id.edtAlamat);
        edtJumlahKamarKost = findViewById(R.id.edtJumlahKamarKost);
        edtFasilitasKost = findViewById(R.id.edtFasilitasKost);
        btnEditKost = findViewById(R.id.btnEditKost);
        btnDeleteKost = findViewById(R.id.btnDeleteKost);
        btnChoose = findViewById(R.id.btnChoose);
        imageKost = findViewById(R.id.imageKost);

        Intent i = getIntent();
        IDPemilik = i.getStringExtra("IDPemilik");
        String namakost = i.getStringExtra("namakost");
        edtNama.setText(namakost);
        String hargakost = i.getStringExtra("hargaKost");
        edtHarga.setText(hargakost);
        String alamatkost = i.getStringExtra("alamatKost");
        edtAlamat.setText(alamatkost);
        String kotakost = i.getStringExtra("kotaKost");
        String jenisKost = i.getStringExtra("jeniskost");
        String jumlahkamar = i.getStringExtra("jumlahKamar");
        edtJumlahKamarKost.setText(jumlahkamar);
        String fasilitaskost = i.getStringExtra("deskripsiFasilitas");
        edtFasilitasKost.setText(fasilitaskost);
        String idkost = i.getStringExtra("IDkost");
        edtIDKost.setText(idkost);
        String fotoKost = i.getStringExtra("fotoKost");
        URL_FOTO_KOST = "http://192.168.1.7/kosthostmobile/" + fotoKost;
        Picasso.get().load(URL_FOTO_KOST).into(imageKost);
```

```

btnChoose.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        ActivityCompat.requestPermissions(
            activity, EditKost.this,
            new String[]{Manifest.permission.READ_EXTERNAL_STORAGE},
            CODE_GALLERY_REQUEST
        );
    }
});
spinKotaKost = findViewById(R.id.spinKotaKost);
ArrayAdapter adapterKota = new ArrayAdapter( context: this, android.R.layout.simple_spinner_item, kota);
spinKotaKost.setAdapter(adapterKota);
spinKotaKost.setSelection(getIndexKota(spinKotaKost, kotakost));

spinKotaKost.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
    @Override
    public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {
        String kota = parent.getItemAtPosition(position).toString();
        kotakost = kota;
    }

    @Override
    public void onNothingSelected(AdapterView<?> parent) {

    }
});

spinJenisKost = findViewById(R.id.spinJenisKost);
ArrayAdapter adapterJenisKost = new ArrayAdapter( context: this, android.R.layout.simple_spinner_item, jenis);
spinJenisKost.setAdapter(adapterJenisKost);
spinJenisKost.setSelection(getIndexJenis(spinJenisKost, jeniskost));

spinJenisKost.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
    @Override
    public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {
        String jenis = parent.getItemAtPosition(position).toString();
        jeniskost = jenis;
    }

    @Override
    public void onNothingSelected(AdapterView<?> parent) {

    }
});

btnEditKost.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        IDKost = edtIDKost.getText().toString();
        namakost = edtNama.getText().toString();
        hargakost = edtharga.getText().toString();
        alamatKost = edtAlamat.getText().toString();
        jumlahKamar = edtJumlahKamarKost.getText().toString();
        deskripsiFasilitas = edtfasilitaskost.getText().toString();
        editkost(IDKost, namakost, hargakost, alamatKost, kotaKost, jeniskost, jumlahKamar, deskripsiFasilitas);
    }
});

btnDeleteKost.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        AlertDialog.Builder builder = new AlertDialog.Builder( context: EditKost.this);
        builder.setTitle("Konfirmasi Hapus");
        builder.setMessage("Apakah anda yakin ingin menghapus kost ?");

        builder.setPositiveButton( text: "Ya", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                IDKost = edtIDKost.getText().toString();
                deletekost(IDKost);
            }
        });

        builder.setNegativeButton( text: "Tidak", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {

            }
        });
        builder.show();
    }
});

private int getIndexKota(Spinner spinKotaKost, String kotakost) {
    int indexKota = 0;
    for (int i = 0; i < spinKotaKost.getCount(); i++) {
        if (spinKotaKost.getItemAtPosition(i).toString().equalsIgnoreCase(kotakost)) {
            indexKota = i;
            break;
        }
    }
    return indexKota;
}

private int getIndexJenis(Spinner spinJenisKost, String jeniskost) {
    int indexJenis = 0;
    for (int i = 0; i < spinJenisKost.getCount(); i++) {
        if (spinJenisKost.getItemAtPosition(i).toString().equalsIgnoreCase(jeniskost)) {
            indexJenis = i;
            break;
        }
    }
    return indexJenis;
}

```

```

private void deletekost(String IDKost) {
    RequestQueue queue = Volley.newRequestQueue( context: EditKost.this);
    StringRequest request = new StringRequest(Request.Method.POST, URL_DELETE_KOST, 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: EditKost.this, text: "Data Kost berhasil dihapus", Toast.LENGTH_SHORT).show();
                    Intent i = new Intent(getApplicationContext(), KostPemilik.class);
                    i.putExtra( name: "IDPemilik", IDPemilik);
                    startActivity(i);
                    EditKost.this.finish();
                } else {
                    Toast.makeText( context: EditKost.this, text: "Data Kost gagal dihapus", Toast.LENGTH_SHORT).show();
                }
            } catch (JSONException e) {
                Toast.makeText( context: EditKost.this, e.getMessage(), Toast.LENGTH_SHORT).show();
            }
        }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            String message = null;
            if (error instanceof NetworkError) {
                message = "Cannot connect to Internet...Please check your connection!";
            } else if (error instanceof ServerError) {
                message = "The server could not be found. Please try again after some time!!";
            } else if (error instanceof AuthFailureError) {
                message = "Cannot connect to Internet...Please check your connection!";
            } else if (error instanceof ParseError) {
                message = "Parsing error! Please try again after some time!!";
            } else if (error instanceof NoConnectionError) {
                message = "Cannot connect to Internet...Please check your connection!";
            } else if (error instanceof TimeoutError) {
                message = "Connection TimeOut! Please check your internet connection.";
            }
            Toast.makeText( context: EditKost.this, message, Toast.LENGTH_SHORT).show();
        }
    });
    @Nullable
    @Override
    protected Map<String, String> getParams() throws AuthFailureError {
        Map<String, String> params = new HashMap<>();
        params.put("IDKost", IDKost);
        return params;
    }
    queue.getCache().clear();
    queue.add(request);
}

private void editkost(String IDKost, String namakost, String hargakost, String alamatKost, String kotaKost, String jenisKost, String jumlahKamar, String deskripsiFasilitas) {
    RequestQueue queue = Volley.newRequestQueue( context: EditKost.this);
    StringRequest request = new StringRequest(Request.Method.POST, URL_EDIT_KOST, 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: EditKost.this, text: "Data Kost berhasil diedit", Toast.LENGTH_SHORT).show();
                    Intent i = new Intent(getApplicationContext(), KostPemilik.class);
                    i.putExtra( name: "IDPemilik", IDPemilik);
                    startActivity(i);
                    EditKost.this.finish();
                } else {
                    Toast.makeText( context: EditKost.this, text: "Data Kost gagal diedit", Toast.LENGTH_SHORT).show();
                }
            } catch (JSONException e) {
                Toast.makeText( context: EditKost.this, e.getMessage(), Toast.LENGTH_SHORT).show();
            }
        }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            String message = null;
            if (error instanceof NetworkError) {
                message = "Cannot connect to Internet...Please check your connection!";
            } else if (error instanceof ServerError) {
                message = "The server could not be found. Please try again after some time!!";
            } else if (error instanceof AuthFailureError) {
                message = "Cannot connect to Internet...Please check your connection!";
            } else if (error instanceof ParseError) {
                message = "Parsing error! Please try again after some time!!";
            } else if (error instanceof NoConnectionError) {
                message = "Cannot connect to Internet...Please check your connection!";
            } else if (error instanceof TimeoutError) {
                message = "Connection TimeOut! Please check your internet connection.";
            }
            Toast.makeText( context: EditKost.this, message, Toast.LENGTH_SHORT).show();
        }
    });
}

```

```

    });
    @Nullable
    @Override
    protected Map<String, String> getParams() throws AuthFailureError {
        Map<String, String> params = new HashMap<>();
        fotoKost = imageToString(bitmap);
        params.put("IDKost", IDKost);
        params.put("namakost", namakost);
        params.put("hargakost", hargakost);
        params.put("alamatKost", alamatKost);
        params.put("kotakost", kotakost);
        params.put("jeniskost", jeniskost);
        params.put("jumlahKamar", jumlahKamar);
        params.put("deskripsifasilitas", deskripsiFasilitas);
        params.put("fotokost", fotokost);
        return params;
    }
};

queue.getCache().clear();
queue.add(request);
}

@Override
public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {
    if (requestCode == CODE_GALLERY_REQUEST) {
        if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
            Intent intent = new Intent(Intent.ACTION_PICK);
            intent.setType("image/*");
            startActivityForResult(intent.createChooser(intent, title: "Select Image"), CODE_GALLERY_REQUEST);
        } else {
            Toast.makeText(getApplicationContext(), text: "Tidak ada akses ke gallery", Toast.LENGTH_LONG).show();
        }
        return;
    }
    super.onRequestPermissionsResult(requestCode, permissions, grantResults);
}

@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
    if (requestCode == CODE_GALLERY_REQUEST && resultCode == RESULT_OK && data != null) {
        Uri filePath = data.getData();

        try {
            InputStream inputStream = getContentResolver().openInputStream(filePath);
            bitmap = BitmapFactory.decodeStream(inputStream);
            imageKost.setImageBitmap(bitmap);

        } catch (FileNotFoundException e) {
            e.printStackTrace();
        }
    }

    super.onActivityResult(requestCode, resultCode, data);
}

private String imageToString(Bitmap bitmap) {
    ByteArrayOutputStream outputStream = new ByteArrayOutputStream();
    bitmap.compress(Bitmap.CompressFormat.JPEG, quality: 100, outputStream);
    byte[] imageBytes = outputStream.toByteArray();

    String encodedImage = Base64.encodeToString(imageBytes, android.util.Base64.DEFAULT);
    return encodedImage;
}
}

```

#### Code 64. Code Java Edit Kost

Code diatas merupakan code yang digunakan untuk pemilik mengedit kost mereka. Hal pertama adalah mendeklarasikan EditText dengan id edtNama, edtHarga, edtAlamat, edtJumlahKamarKost, edtFasilitasKost, edtIDKost. Selanjutnya mendeklarasikan Spinner dengan id spinKotaKost, spinJenisKost. Lalu mendeklarasikan button dengan id btnEditKost, btnDeleteKost, btnChoose. Kemudian mendeklarasikan ImageView dengan id imageKost. Lalu terdapat Bitmap dengan id bitmap yang berhubungan dengan imageKost. Kemudian final int Code\_Gallery\_Request yang digunakan untuk membaca data external. Lalu terdapat 2 String, yang pertama ada String Kota yang terdapat dalam spinKotaKost yang digunakan untuk memilih kota. Yang kedua adalah String Jenis yang terdapat dalam spinJenisKost yang digunakan untuk memilih jenis kost. Terdapat 3 URL, pertama adalah URL\_FOTO\_KOST sebagai lokasi file php yang akan digunakan untuk mendapatkan foto kost dari database. URL\_EDIT\_KOST sebagai lokasi file php yang akan digunakan untuk melakukan EDIT dari android ke dalam database. URL\_DELETE\_KOST

sebagai lokasi file hp yang akan digunakan untuk melakukan DELETE dari android ke dalam database. Berbagai String juga dipanggil seperti IDKost, namaKost, hargaKost, alamatKost, kotaKost, jumlahKamar, deskripsiFasilitas, jenisKost, fotoKost. Kemudian menginisialisasikan String IDPemilik.

Setelahnya kita membuat fungsi @Override dengan method OnCreate yang berfungsi untuk melakukan inisialisasi pada saat kondisi awal Activity baru diciptakan, sehingga setContentView secara otomatis akan memperlihatkan halaman activity EditKost. Get Intent digunakan untuk mengambil beberapa data dari variabel dari activity sebelumnya yang dikirim. Variabel yang diterima di activity ini adalah IDPemilik. Find view by id digunakan untuk melakukan inisialisasi variabel yang sudah dideklarasikan di awal agar dapat terkoneksi dengan komponen-komponen dalam activity ini. URL\_FOTO\_KOST = "http://192.168.1.7/kosthostmobile/" + fotoKost; digunakan untuk membuat suatu URL yang tepat untuk mengambil foto kost dari database. Untuk menetapkan foto tersebut dari URL ke dalam ImageView dapat menggunakan Picasso, yaitu dengan Picasso.get().load (URL\_FOTO\_KOST).into(imageKost); Untuk menggunakan metode Picasso, harus menambahkan implementation 'com.squareup.picasso:picasso:2.71828' pada build.gradle.

setOnClickListener pada button digunakan untuk mendeklarasikan aksi yang akan dijalankan setelah tombol ditekan, biasanya digunakan untuk menampilkan activity lain. RequestQueue dan StringRequest digunakan untuk menghubungkan coding yang kita buat dengan URL lokasi file php agar dapat terkoneksi dengan database MySQL. OnResponse digunakan sebagai penampung dari hasil proses di dalamnya. Proses di dalamnya meliputi try dan catch untuk menjalankan perintah sesuai dengan file php. Setelah didapatkan response dengan membuat suatu JSONObject, nilai dari success = 1 menandakan bahwa proses berjalan dengan lancar sedangkan selain 1 berarti proses tidak berjalan dengan baik dan terdapat error. Jika success = 1, akan muncul toast bahwa kost berhasil di book dan melakukan intent ke activity EditKost dengan mengirimkan IDPemilik.

Spinner yang digunakan dalam class ini menggunakan Array Adapter. Mendeklarasikan Array Adapter sesuai dengan id yang sudah dideklarasikan sebelumnya. Menginisialisasi Index agar sesuai dengan apa yang dipilih. OnErrorResponse digunakan untuk memberikan respon tertentu seperti toast Error. Sedangkan di bawahnya, getParams() digunakan untuk mengirimkan variabel yang diperlukan dalam file php untuk melakukan query. Dalam file ini diperlukan IDKost dan IDPencari agar query MySQL pada file php dapat berjalan dengan baik yaitu mendapatkan data IDKost yang akan dibook berdasarkan IDPencari yang login. Bagian akhir, harus ditambahkan queue.getCache().clear(); dan queue.add(request); agar program dapat berjalan dan terupdate tampilannya. OnRequestPermissionsResult digunakan jika aplikasi perlu menggunakan sumber daya atau informasi di luar kotak pasirnya sendiri, kita dapat mendeklarasikan izin dan menyiapkan permintaan izin yang menyediakan akses ini. Ini adalah bagian dari alur kerja untuk menggunakan izin. OnActivityResult digunakan untuk

memulai aktivitas lain, baik dalam aplikasi maupun dari aplikasi lain, tidak harus berupa operasi satu arah. kita juga dapat memulai aktivitas lain dan menerima hasilnya kembali. aplikasi dapat menerima foto yang diambil sebagai hasilnya. `imageToString` digunakan untuk mengubah file gambar menjadi String.

## Coding Java Menu Booked

### Coding Java Booked Pemilik

```
package com.projectx1.kosthost;

import android.annotation.Nullable;
import android.appCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ImageButton;
import android.widget.ListAdapter;
import android.widget.ListView;
import android.widget.SimpleAdapter;
import android.widget.TextView;
import android.widget.Toast;

import com.android.volley.AuthFailureError;
import com.android.volley.NetworkError;
import com.android.volley.NoConnectionError;
import com.android.volley.ParseError;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.ServerError;
import com.android.volley.TimeoutError;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;

public class BookedPemilik extends AppCompatActivity {
    private ListView listBooked;
    private ImageButton btnBH;
    ArrayList<HashMap<String, String>> listbooked;
    String URL_GET_BOOKEDHOST = "http://192.168.1.7/kosthostmobile/getbooked-pemilik.php";
    String IDkost, IDsewa, namaKost, namaPencari, telepon, status;
    String IDPemilik;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_booked_pemilik);
        Intent i = getIntent();
    }
}
```

```

IDPenilik = i.getStringExtra( name: "IDPenilik");
listBooked = findViewById(R.id.listBooked);
listBooked = new ArrayList();
RequestQueue queue = Volley.newRequestQueue( context: BookedPenilik.this);
StringRequest request = new StringRequest(Request.Method.POST, URL_GET_BOOKEDKOST,
new Response.Listener<String>() {
    @Override
    public void onResponse(String response) {
        try {
            JSONObject jsonObject = new JSONObject(response);
            JSONArray jsonArray = jsonObject.getJSONArray( name: "data");

            for (int i = 0; i < jsonArray.length(); i++) {
                JSONObject kost = jsonArray.getJSONObject(i);
                IDkost = kost.getString( name: "IDkost");
                IDsewa = kost.getString( name: "IDsewa");
                namakost = kost.getString( name: "namakost");
                namapencari = kost.getString( name: "namapencari");
                telepon = kost.getString( name: "telepon");
                status = kost.getString( name: "status");

                HashMap<String, String> map = new HashMap();
                map.put("IDkost", IDkost);
                map.put("IDsewa", IDsewa);
                map.put("namakost", namakost);
                map.put("namapencari", namapencari);
                map.put("telepon", telepon);
                map.put("status", status);

                listBooked.add(map);
            }

            String[] from = {"IDkost", "IDsewa", "namakost", "namapencari", "telepon", "status"};
            int[] to = {R.id.txtIDkost, R.id.txtIDsewa, R.id.txtNamakost, R.id.txtNamapencari, R.id.txtTelepon, R.id.txtStatus};
           ListAdapter adapter = new SimpleAdapter( context: BookedPenilik.this, listBooked, R.layout.item_bookedpenilik, from, to);
            listBooked.setAdapter(adapter);
        } catch (JSONException e) {
            Toast.makeText( context: BookedPenilik.this, e.getMessage(), Toast.LENGTH_SHORT).show();
        }
    }
}, new Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError error) {
        String message = null;
        if (error instanceof NetworkError) {
            message = "Cannot connect to Internet...Please check your connection!";
        } else if (error instanceof ServerError) {
            message = "The server could not be found. Please try again after some time!!";
        } else if (error instanceof AuthFailureError) {
            message = "The server requires you to be authenticated";
        } else if (error instanceof ParseError) {
            message = "Parsing error! Please try again after some time!!";
        } else if (error instanceof NoConnectionError) {
            message = "Cannot connect to Internet...Please check your connection!";
        } else if (error instanceof TimeoutError) {
            message = "Connection TimeOut! Please check your internet connection.";
        }
        Toast.makeText( context: BookedPenilik.this, message, Toast.LENGTH_SHORT).show();
    }
});

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

queue.getCache().clear();
queue.add(request);
listBooked.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
        String get_idkost = ((TextView) view.findViewById(R.id.txtIDkost)).getText().toString();
        String get_idsawa = ((TextView) view.findViewById(R.id.txtIDsewa)).getText().toString();
        String get_namakost = ((TextView) view.findViewById(R.id.txtNamakost)).getText().toString();
        String get_namapencari = ((TextView) view.findViewById(R.id.txtNamapencari)).getText().toString();
        String get_telepon = ((TextView) view.findViewById(R.id.txtTelepon)).getText().toString();
        String get_status = ((TextView) view.findViewById(R.id.txtStatus)).getText().toString();

        Intent i = new Intent(getApplicationContext(), DetailbookedPenilik.class);
        i.putExtra( name: "IDkost", get_idkost);
        i.putExtra( name: "IDsewa", get_idsawa);
        i.putExtra( name: "namakost", get_namakost);
        i.putExtra( name: "namapencari", get_namapencari);
        i.putExtra( name: "telepon", get_telepon);
        i.putExtra( name: "IDPenilik", IDPenilik);
        i.putExtra( name: "status", get_status);
        startActivity(i);
    }
});
btnKH = findViewById(R.id.btnKH);
btnKH.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent i = new Intent(getApplicationContext(), DashboardPenilik.class);
    }
});

```

```
i.putExtra( name: "IDPemilik", IDPemilik);
startActivity(i);
BookedPemilik.this.finish();
```

Code 65. Code Java Booked Kost Pemilik Kost

Pada coding Java diatas adalah coding yang digunakan untuk tampilan Booked Pemilik. Hal pertama yang dilakukan adalah mendeklarasikan ListView dengan variabel listBooked, ImageButton dengan variabel btnKH. Lalu inisialisasi ArrayList<HashMap<String, String>> dengan nama listBooked yang bertujuan untuk tempat penyimpanan berbagai data yang nantinya akan ditampilkan ke dalam listBooked. Terdapat URL\_GET\_BOOKEDKOST sebagai lokasi file php yang akan digunakan untuk melakukan CRUD dari android ke dalam database. Berbagai String lainnya juga diinisialisasikan seperti IDKost, IDSewa, namaKost, namaPencari, telepon, status, dan IDPemilik. Setelahnya kita membuat fungsi @Override dengan method OnCreate yang berfungsi untuk melakukan inisialisasi pada saat kondisi awal Activity baru diciptakan, sehingga setContentView secara otomatis akan memperlihatkan halaman activity BookedPemilik. Get Intent digunakan untuk mengambil beberapa data dari variabel dari activity sebelumnya yang dikirim. Variabel yang diterima di activity ini adalah IDPemilik yang didapatkan dari Dashboard Pemilik. Find view by id digunakan untuk melakukan inisialisasi variabel yang sudah dideklarasikan di awal agar dapat terkoneksi dengan komponen-komponen dalam activity ini. Setelah listBooked diinisialisasi dengan findViewById, array list listbooked juga diinisialisasikan. RequestQueue dan StringRequest digunakan untuk menghubungkan coding yang kita buat dengan URL lokasi file php agar dapat terkoneksi dengan database MySQL.

OnResponse digunakan sebagai penampung dari hasil proses di dalamnya. Proses di dalamnya meliputi try dan catch untuk mendapatkan data variabel yang dikirimkan dari file php, seperti IDKost, IDSewa, namaKost, namaPencari, telepon, dan status. Setelah didapatkan dengan membuat suatu JSONObject, berbagai data tersebut disimpan ke dalam suatu adapter untuk ditampilkan di listBooked. Langkahnya adalah dengan menggunakan HashMap lalu dilakukan put ke dalam variabel-variabel yang akan disimpan ke dalam array listbooked. String[] from dan int[] to harus saling berhubungan untuk menandakan dari mana data yang diambil dan ke mana data tersebut akan ditampilkan. Data akan ditampilkan ke dalam layout list \_bookedpemilik. list \_bookedpemilik berisi berbagai komponen yang akan ditampilkan ke dalam listBooked dengan adapter.

onErrorResponse bermanfaat untuk memberikan respon tertentu seperti toast akan error yang didapatkan. Sedangkan di bawahnya, getParams() digunakan untuk mengirimkan variabel yang diperlukan dalam file php untuk melakukan query. Dalam file ini diperlukan IDPemilik agar query MySQL pada file php dapat berjalan dengan baik yaitu mendapatkan data booked berdasarkan IDPemilik yang login. Bagian akhir, harus ditambahkan queue.getCache().clear(); dan queue.add(request); agar program dapat berjalan dan terupdate tampilannya.

Selanjutnya adalah menetapkan setOnItemClickListener untuk listBooked, jika listBooked ditekan maka akan melakukan intent ke activity DetailBookedPemilik dengan mengirimkan berbagai variabel yang dikirimkan. Terdapat setOnItemClickListener pula untuk btnKH yaitu ketiga logo KH ditekan, pengguna akan diarahkan untuk kembali ke dashboard. Semua intent yang digunakan harus diakhiri dengan startActivity(i) agar intent dapat berjalan.

## Coding Java Detail Booked Pemilik

```

package com.projectK1.kosthost;

import ...

public class DetailbookedPemilik extends AppCompatActivity {
    TextView txtNamaKost, txtNamaPenyewa, txtKontakPenyewa, txtIDSewa;
    Button btnDeleteBooked, btnConfirmBooked;
    String IDKost, IDSewa, namakost, namaPencari, telepon, IDPemilik, status;

    String URL_DELETE_BOOKEDKOST = "http://192.168.1.7/kosthostmobile/deletebooked-pemilik.php";
    String URL_CONFIRM_BOOKEDKOST = "http://192.168.1.7/kosthostmobile/minuskamar-pemilik.php";
    String URL_UPDATE_STATUS = "http://192.168.1.7/kosthostmobile/updatetestatus-penyeaan.php";
    String URL_TAMBAH_KAMAR = "http://192.168.1.7/kosthostmobile/tambahkamar-pemilik.php";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_detailbooked_pemilik);
        txtNamaKost = findViewById(R.id.txtNamaKost);
        txtNamaPenyewa = findViewById(R.id.txtNamaPenyewa);
        txtKontakPenyewa = findViewById(R.id.txtKontakPenyewa);
        txtIDSewa = findViewById(R.id.txtIDSewa);
        btnDeleteBooked = findViewById(R.id.btnDeleteBooked);
        btnConfirmBooked = findViewById(R.id.btnConfirmBooked);
        Intent i = getIntent();
        IDPemilik = i.getStringExtra("name: "IDPemilik");
        IDKost = i.getStringExtra("name: "IDKost");
        IDSewa = i.getStringExtra("name: "IDSewa");
        namakost = i.getStringExtra("name: "namakost");
        namaPencari = i.getStringExtra("name: "namaPencari");
        telepon = i.getStringExtra("name: "telepon");
        txtNamaKost.setText(namakost);
        txtNamaPenyewa.setText(namaPencari);
        txtKontakPenyewa.setText(telepon);
        txtIDSewa.setText(IDSewa);

        status = i.getStringExtra("name: "status");
        if (status.equalsIgnoreCase("Booked")) {
            btnConfirmBooked.setVisibility(View.GONE);
        } else {
            btnConfirmBooked.setEnabled(true);
        }

        btnConfirmBooked.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                minusKamar(IDKost);
                updateStatus(IDSewa);
            }
        });

        btnDeleteBooked.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                tambahKamar(IDKost);
                deleteBooked(IDSewa);
            }
        });
    }
}

```

```

private void tambahKamar(String IDKost) {
    RequestQueue queue = Volley.newRequestQueue( context: DetailbookedPemilik.this);
    StringRequest request = new StringRequest(Request.Method.POST, URL_TAMBAH_KAMAR, 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) {
                    return;
                } else {
                    Toast.makeText( context: DetailbookedPemilik.this, text: "Data penyewaan gagal dihapus", Toast.LENGTH_SHORT).show();
                }
            } catch (JSONException e) {
                Toast.makeText( context: DetailbookedPemilik.this, e.getMessage(), Toast.LENGTH_SHORT).show();
            }
        }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            String message = null;
            if (error instanceof NetworkError) {
                message = "Cannot connect to Internet...Please check your connection!";
            } else if (error instanceof ServerError) {
                message = "The server could not be found. Please try again after some time!!";
            } else if (error instanceof AuthFailureError) {
                message = "Cannot connect to Internet...Please check your connection!";
            } else if (error instanceof ParseError) {
                message = "Parsing error! Please try again after some time!!";
            } else if (error instanceof NoConnectionError) {
                message = "Cannot connect to Internet...Please check your connection!";
            } else if (error instanceof TimeoutError) {
                message = "Connection Timeout! Please check your internet connection.";
            }
            Toast.makeText( context: DetailbookedPemilik.this, message, Toast.LENGTH_SHORT).show();
        }
    }) {
        @Nullable
        @Override
        protected Map<String, String> getParams() throws AuthFailureError {
            Map<String, String> params = new HashMap<>();
            params.put("IDKost", IDKost);
            return params;
        }
    };
    queue.getCache().clear();
    queue.add(request);
}

private void deleteBooked(String IDSewa) {
    RequestQueue queue = Volley.newRequestQueue( context: DetailbookedPemilik.this);
    StringRequest request = new StringRequest(Request.Method.POST, URL_DELETE_BOOKEDKOST, 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) {
                    Intent i = new Intent(getApplicationContext(), BookedPemilik.class);
                    i.putExtra( name: "IDPemilik", IDPemilik);
                    startActivity(i);
                    DetailbookedPemilik.this.finish();
                } else {
                    Toast.makeText( context: DetailbookedPemilik.this, text: "Data penyewaan gagal dihapus", Toast.LENGTH_SHORT).show();
                }
            } catch (JSONException e) {
                Toast.makeText( context: DetailbookedPemilik.this, e.getMessage(), Toast.LENGTH_SHORT).show();
            }
        }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            String message = null;
            if (error instanceof NetworkError) {
                message = "Cannot connect to Internet...Please check your connection!";
            } else if (error instanceof ServerError) {
                message = "The server could not be found. Please try again after some time!!";
            } else if (error instanceof AuthFailureError) {
                message = "Cannot connect to Internet...Please check your connection!";
            } else if (error instanceof ParseError) {
                message = "Parsing error! Please try again after some time!!";
            } else if (error instanceof NoConnectionError) {
                message = "Cannot connect to Internet...Please check your connection!";
            } else if (error instanceof TimeoutError) {
                message = "Connection Timeout! Please check your internet connection.";
            }
            Toast.makeText( context: DetailbookedPemilik.this, message, Toast.LENGTH_SHORT).show();
        }
    }) {
        @Nullable
        @Override
        protected Map<String, String> getParams() throws AuthFailureError {
            Map<String, String> params = new HashMap<>();
            params.put("IDSewa", IDSewa);
            return params;
        }
    };
    queue.getCache().clear();
    queue.add(request);
}

```

```

private void minusKamar(String IDkost) {
    RequestQueue queue = Volley.newRequestQueue( context: DetailbookedPemilik.this);
    StringRequest request = new StringRequest(Request.Method.POST, URL_CONFIRM_BOOKEDKOST, 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) {
                    return;
                } else {
                    Toast.makeText( context: DetailbookedPemilik.this, text: "Data penyewaan gagal dikonfirmasi", Toast.LENGTH_SHORT).show();
                }
            } catch (JSONException e) {
                Toast.makeText( context: DetailbookedPemilik.this, e.getMessage(), Toast.LENGTH_SHORT).show();
            }
        }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            String message = null;
            if (error instanceof NetworkError) {
                message = "Cannot connect to Internet..Please check your connection!";
            } else if (error instanceof ServerError) {
                message = "The server could not be found. Please try again after some time!";
            } else if (error instanceof AuthFailureError) {
                message = "Cannot connect to Internet..Please check your connection!";
            } else if (error instanceof ParseError) {
                message = "Parsing error! Please try again after some time!";
            } else if (error instanceof NoConnectionError) {
                message = "Cannot connect to Internet..Please check your connection!";
            } else if (error instanceof TimeoutError) {
                message = "Connection TimeOut! Please check your internet connection.";
            }
            Toast.makeText( context: DetailbookedPemilik.this, message, Toast.LENGTH_SHORT).show();
        }
    });
    @Nullable
    @Override
    protected Map<String, String> getParams() throws AuthFailureError {
        Map<String, String> params = new HashMap<>();
        params.put("IDkost", IDkost);
        return params;
    }
};

queue.getCache().clear();
queue.add(request);
}

private void updateStatus(String IDsewa) {
    RequestQueue queue = Volley.newRequestQueue( context: DetailbookedPemilik.this);
    StringRequest request = new StringRequest(Request.Method.POST, URL_UPDATE_STATUS, 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) {
                    Intent i = new Intent(getApplicationContext(), BookedPemilik.class);
                    i.putExtra( name: "IDPemilik", IDPemilik);
                    startActivity(i);
                    DetailbookedPemilik.this.finish();

                } else {
                    Toast.makeText( context: DetailbookedPemilik.this, text: "Data penyewaan berhasil dikonfirmasi", Toast.LENGTH_SHORT).show();
                }
            } catch (JSONException e) {
                Toast.makeText( context: DetailbookedPemilik.this, e.getMessage(), Toast.LENGTH_SHORT).show();
            }
        }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            String message = null;
            if (error instanceof NetworkError) {
                message = "Cannot connect to Internet..Please check your connection!";
            } else if (error instanceof ServerError) {
                message = "The server could not be found. Please try again after some time!";
            } else if (error instanceof AuthFailureError) {
                message = "Cannot connect to Internet..Please check your connection!";
            } else if (error instanceof ParseError) {
                message = "Parsing error! Please try again after some time!";
            } else if (error instanceof NoConnectionError) {
                message = "Cannot connect to Internet..Please check your connection!";
            } else if (error instanceof TimeoutError) {
                message = "Connection TimeOut! Please check your internet connection.";
            }
            Toast.makeText( context: DetailbookedPemilik.this, message, Toast.LENGTH_SHORT).show();
        }
    });
    @Nullable
    @Override
    protected Map<String, String> getParams() throws AuthFailureError {
        Map<String, String> params = new HashMap<>();
        params.put("IDsewa", IDsewa);
        return params;
    }
};

queue.getCache().clear();
queue.add(request);
}
}

```

#### Code 66. Code Detail Booked Kost Pemilik

Codingan diatas merupakan codingan Java yang digunakan untuk tampilan detail booked pemilik. Hal pertama yang dilakukan adalah menginisialisasi Button

btnDeleteBooked, btnConfirmBooked dan String IDKost, IDSewa, namaKost, namaPencari, telepon, IDPemilik, dan status. Kemudian mendeklarasikan TextView seperti txtNamaKost, txtNamaPenyewa, txtKontakPenyewa, dan txtIDSewa. Terdapat 4 URL, URL\_DELETE\_BOOKEDKOST sebagai lokasi file php yang akan digunakan untuk melakukan DELETE dari android ke dalam database, URL\_CONFIRM\_BOOKEDKOST sebagai lokasi file php yang akan digunakan untuk READ data dari android ke dalam database, URL\_UPDATE\_STATUS sebagai lokasi file php yang akan digunakan untuk melakukan UPDATE dari android ke dalam database, URL\_TAMBAH\_KAMAR sebagai lokasi file php yang akan digunakan untuk melakukan UPDATE dari android ke dalam database.

Setelahnya kita membuat fungsi @Override dengan method OnCreate yang berfungsi untuk melakukan inisialisasi pada saat kondisi awal Activity baru diciptakan, sehingga setContentView secara otomatis akan memperlihatkan halaman activity Detail Booked Pemilik. setOnClickListener pada button digunakan untuk mendeklarasikan aksi yang akan dijalankan setelah tombol ditekan, biasanya digunakan untuk menampilkan activity lain. RequestQueue dan StringRequest digunakan untuk menghubungkan coding yang kita buat dengan URL lokasi file php agar dapat terkoneksi dengan database MySQL. OnErrorResponse digunakan untuk memberikan respon tertentu seperti toast Error. Sedangkan di bawahnya, getParams() digunakan untuk mengirimkan variabel yang diperlukan dalam file php untuk melakukan query. Bagian akhir, harus ditambahkan queue.getCache().clear(); dan queue.add(request); agar program dapat berjalan dan terupdate tampilannya. Dalam file ini diperlukan IDKost agar query MySQL pada file php dapat berjalan dengan baik. OnResponse digunakan sebagai penampung dari hasil proses di dalamnya. Proses di dalamnya meliputi try dan catch untuk mendapatkan data variabel yang dikirimkan dari file php seperti IDKost.

## **Coding Java Menu Penjaga**

## **Coding Java Class Penjaga Kost**

## **Code 67. Code Java Class Penjaga Kost**

Pada coding Java diatas adalah coding yang digunakan pada class Penjaga Kost. Awalnya data kolom yang digunakan akan dideklarasikan terlebih dahulu beserta dengan tipe datanya. Pada Class ini, kolom id tipe data yang digunakan adalah long, sedangkan kolom namakost, namapenjaga, dan teleponpenjaga tipe data yang digunakan adalah String. Kemudian dibuatkan Constructor untuk class PenjagaKost dengan mengetikkan public PenjagaKost(){ }. Kemudian dibuat method-method untuk get dan set kolom-kolom yang digunakan dan sudah dideklarasikan sebelumnya. Method untuk set terlihat seperti pada method setID, setNamaKost, setNamaPenjaga, dan setTeleponPenjaga, yaitu dengan mengetikkan public void namaMethod() {this.\_namakolom = namakolom;}. Kemudian untuk method get terlihat seperti pada method getID, getNamaKost, getNamaPenjaga, dan getTeleponPenjaga, yaitu dengan mengetikkan public tipedata namaMethod() {return this.\_namakolom;}. Selanjutnya dibuat method untuk mengubah objek menjadi String yaitu pada method toString(). Fungsinya adalah untuk mencetak data yang dimasukkan ke database.

## Coding Java Class MyDBHandler

```
package com.projectK1.kosthost;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.SQLException;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import java.util.ArrayList;

public class MyDBHandler extends SQLiteOpenHelper {
    //Deklarasi variabel konstanta untuk pembuatan database, tabel dan kolom yang diperlukan
    private static final int DATABASE_VERSION = 1;
    private static final String DATABASE_NAME = "datapenjaga.db";
    private static final String TABLE_NAME = "penjaga";
    private static final String COLUMN_ID = "_id";
    private static final String COLUMN_NAMAKOST = "namakost";
    private static final String COLUMN_NAMAPENJAGA = "namapenjaga";
    private static final String COLUMN_TELEPONPENJAGA = "teleponpenjaga";

    //Constructor untuk Class MyDBHandler
    public MyDBHandler(Context context) { super(context, DATABASE_NAME, null, DATABASE_VERSION); }

    @Override
    public void onCreate(SQLiteDatabase sqLiteDatabase) {
        String CREATE_TABLE_PENJAGA = "CREATE TABLE " + TABLE_NAME + "(" +
            COLUMN_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, " +
            COLUMN_NAMAKOST + " VARCHAR(50) NOT NULL, " +
            COLUMN_NAMAPENJAGA + " VARCHAR(50) NOT NULL, " +
            COLUMN_TELEPONPENJAGA + " VARCHAR(15) NOT NULL)";
        sqLiteDatabase.execSQL(CREATE_TABLE_PENJAGA);
    }

    //Method yang digunakan untuk Upgrade Tabel
    @Override
    public void onUpgrade(SQLiteDatabase sqLiteDatabase, int i, int ii) {
        sqLiteDatabase.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME);
        onCreate(sqLiteDatabase);
    }

    /**-- Insert, Select, Update, Delete ----*/
    private SQLiteDatabase database;

    //Method untuk Open Database Connection
    public void open() throws SQLException {
        database = this.getWritableDatabase();
    }

    //Inisialisasi Semua Kolom di Tabel Database
    private String[] allColumns =
        {COLUMN_ID, COLUMN_NAMAKOST, COLUMN_NAMAPENJAGA, COLUMN_TELEPONPENJAGA};

    //Method untuk Memindahkan Isi Cursor ke Objek Penjaga
    private PenjagaKost cursorToPenjagaKost(Cursor cursor) {
        PenjagaKost penjagaKost = new PenjagaKost();
        penjagaKost.setID(cursor.getLong( columnIndex: 0));
        penjagaKost.setNamaKost(cursor.getString( columnIndex: 1));
        penjagaKost.setNamaPenjaga(cursor.getString( columnIndex: 2));
        penjagaKost.setTeleponPenjaga(cursor.getString( columnIndex: 3));
        return penjagaKost;
    }

    //Method untuk Menambahkan Penjaga Kost Baru
    public void createPenjagaKost(String namakost, String namaPenjaga, String teleponPenjaga) {
        ContentValues values = new ContentValues();
        values.put(COLUMN_NAMAKOST, namakost);
        values.put(COLUMN_NAMAPENJAGA, namaPenjaga);
        values.put(COLUMN_TELEPONPENJAGA, teleponPenjaga);
        database.insert(TABLE_NAME, null, values);
    }

    //Method untuk Mendapatkan Detail per Penjaga Kost
    public PenjagaKost getPenjagaKost(long id) {
        PenjagaKost penjagaKost = new PenjagaKost();
        Cursor cursor = database.query(TABLE_NAME, allColumns, selection: "_id=" + id, selectionArgs: null, groupBy: null, having: null, orderBy: null);
        cursor.moveToFirst();
        penjagaKost = cursorToPenjagaKost(cursor);
        cursor.close();
        return penjagaKost;
    }

    //Method untuk Mendapatkan Data Semua Penjaga Kost di Tabel Penjaga
    public ArrayList<PenjagaKost> getAllPenjagaKost() {
        ArrayList<PenjagaKost> daftarPenjagaKost = new ArrayList<>();
        Cursor cursor = database.query(TABLE_NAME, allColumns, selection: null, selectionArgs: null, groupBy: null, having: null, orderBy: null);
        cursor.moveToFirst();
        while (!cursor.isAfterLast()) {
            PenjagaKost penjagaKost = cursorToPenjagaKost(cursor);
            daftarPenjagaKost.add(penjagaKost);
            cursor.moveToNext();
        }
        cursor.close();
        return daftarPenjagaKost;
    }
}
```

```

//Method untuk Memperbarui Data Penjaga Kost
public void updatePenjagaKost(PenjagaKost penjagaKost) {
    String filter = "_id=" + penjagaKost.getID();
    ContentValues args = new ContentValues();
    args.put(COLUMN_NAMAKOST, penjagaKost.getNamaKost());
    args.put(COLUMN_NAMAPENJAGA, penjagaKost.getNamaPenjaga());
    args.put(COLUMN_TELEPONPENJAGA, penjagaKost.getTeleponPenjaga());
    database.update(TABLE_NAME, args, filter, whereArgs: null);
}

//Method untuk Menghapus Data Penjaga Kost
public void deletePenjagaKost(long id) {
    String filter = "_id=" + id;
    database.delete(TABLE_NAME, filter, whereArgs: null);
}
}

```

**Code 68. Code Java Class MyDBHandler**

Pada coding Java diatas adalah coding yang digunakan pada class MyDBHandler yang digunakan untuk melakukan proses CRUD ke database lokal yaitu SQLite. Untuk melakukannya diperlukan extends SQLiteOpenHelper. Kemudian melakukan deklarasi variabel untuk pembuatan database, tabel, dan kolom yang diperlukan. Pada aplikasi ini, dibuat database dengan nama datapenjaga.db, dengan tabel bernama penjaga yang memiliki kolom id, namakost, namapenjaga, dan teleponpenjaga. Lalu dibuat constructor untuk class MyDBHandler yang membuat database yang sudah dideklarasikan. Kemudian ketika aplikasi dijalankan (method onCreate), terdapat String yang menampung query pembuatan tabel yaitu CREATE TABLE dimana didalamnya memuat nama tabel, kolom dan keterangan kolom seperti tipe data, ukuran, dan keterangan lainnya. Kemudian query tersebut akan dieksekusi dengan fungsi execSQL. Untuk mengupgrade tabel digunakan method onUpgrade yang isinya menghapus tabel apabila tabel dengan nama yang dideklarasikan sudah ada dengan cara “DROP TABLE IF EXIST” disertai nama tabel dan dieksekusi menggunakan fungsi execSQL. Kemudian memanggil method onCreate untuk membuat ulang tabel.

Untuk melakukan CRUD ke database digunakan SQLiteDatabase dengan variabel yang dideklarasikan. Pada class ini digunakan variabel database. Selanjutnya dilakukan koneksi ke database menggunakan method open() yang isinya memanfaatkan fungsi getWritableDatabase. Kemudian melakukan inisialisasi kolom yang terdapat pada tabel dalam database. Inisialisasi dilakukan menggunakan array, pada class ini bernama allColumns yang isinya kolom id, namakost, namapenjaga, dan teleponpenjaga. Lalu, dibuat method yang dapat memindahkan isi cursor ke objek (cursorToPenjagaKost), dimana isi method ini memanfaatkan class PenjagaKost dan method set yang sudah dibuat pada class PenjagaKost. Untuk melakukan insert data atau menambahkan data, dibuat method createPenjagaKost dengan parameter namakost, namapenjaga, dan teleponpenjaga dan memanfaatkan class ContentValues dan fungsi put untuk memasukkan data ke masing-masing kolom, serta fungsi insert pada database. Untuk memperoleh detil data penjaga kost, dibuat method getPenjagaKost yang memanfaatkan class PenjagaKost yang sudah dibuat sebelumnya, fungsi cursor yang diinisialisasikan menjadi query untuk mendapatkan data dengan id tertentu dan menggunakan fungsi moveToFirst untuk memindahkan cursor ke baris data atas, lalu memindahkan isi cursor ke objek penjaga menggunakan method

cursorToPenjagaKost yang sudah dibuat sebelumnya, kemudian cursor ditutup. Sedangkan untuk mendapatkan semua data penjaga kost yang tersimpan di database, dibuat method getAllPenjagaKost dengan memanfaatkan ArrayList, fungsi cursor yang diinisialisasikan menjadi query untuk mendapatkan semua data dalam tabel dan menggunakan fungsi moveToFirst untuk memindahkan cursor ke baris data atas kemudian memanfaatkan fungsi perulangan while untuk memperoleh data selanjutnya dengan memanfaatkan method cursorToPenjagaKost, fungsi add untuk memasukkan data ke array list, dan fungsi moveToNext untuk memindahkan cursor ke data selanjutnya. Kemudian setelah semua data diambil, cursor ditutup. Untuk mengubah data pada database dibuat method updatePenjagaKost dengan memanfaatkan class PenjagaKost dan method get yang sudah dibuat pada class tersebut. Kemudian setelah mendapatkan datanya dengan memanggil method get dan memasukkan data kedalam kolom menggunakan class ContentValues, digunakan fungsi update pada database. Selanjutnya, untuk menghapus data dari database dibuat method deletePenjagaKost dengan parameter id yang ditampung pada variabel String kemudian memanfaatkan fungsi delete pada database.

## Coding Java Menu Penjaga

```
package com.projectK1.kosthost;

import android.app.AlertDialog;
import android.app.Dialog;
import android.app.ListActivity;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.database.SQLException;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.ImageButton;
import android.widget.ImageView;
import android.widget.ListView;
import android.widget.Toast;
import java.util.ArrayList;

public class Penjaga extends ListActivity implements AdapterView.OnItemLongClickListener {
    private MyDBHandler dbHandler;
    private ArrayList<PenjagaKost> values;
    private ListView list;
    private ImageButton btnTambahPenjaga, btnKH;
    private Button btnEditPenjaga, btnDeletePenjaga;
    Context context = this;
    String IDPemilik;
```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_penjaga);
    dbHandler = new MyDBHandler( context: this);
    Intent i = getIntent();
    IDPemilik = i.getStringExtra( name: "IDPemilik");
    //Membuka koneksi database
    try {
        dbHandler.open();
    } catch (SQLException e) {
        e.printStackTrace();
    }

    values = dbHandler.getAllPenjagaKost();

    ArrayAdapter<PenjagaKost> adapter = new ArrayAdapter<>( context: this,
    | android.R.layout.simple_list_item_1, values);
    setListAdapter(adapter);

    list = (ListView) findViewById(android.R.id.list);
    list.setOnItemLongClickListener(this);

    btnTambahPenjaga = findViewById(R.id.btnAddPenjaga);
    btnTambahPenjaga.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent i = new Intent(getApplicationContext(), TambahPenjaga.class);
            i.putExtra( name: "IDPemilik", IDPemilik);
            startActivity(i);
        }
    });

    btnKH = findViewById(R.id.btnKH);
    btnKH.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent i = new Intent(getApplicationContext(), DashboardPemilik.class);
            i.putExtra( name: "IDPemilik", IDPemilik);
            startActivity(i);
            Penjaga.this.finish();
        }
    });
}

@Override
public boolean onItemLongClick(AdapterView<?> parent, View view, int i, long l) {
    final Dialog dialog = new Dialog( context: this);
    dialog.setContentView(R.layout.dialog_penjaga);
    dialog.setTitle("Pilih Aksi!");
    dialog.show();

    final PenjagaKost penjagaKost = (PenjagaKost) getListAdapter().getItem(i);
    final long id = penjagaKost.getID();

    btnEditPenjaga = dialog.findViewById(R.id.btnAddPenjaga);
    btnDeletePenjaga = dialog.findViewById(R.id.btnDeletePenjaga);

    btnEditPenjaga.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            PenjagaKost penjagaKost = dbHandler.getPenjagaKost(id);
            Intent i = new Intent(getApplicationContext(), EditPenjaga.class);
            Bundle bundle = new Bundle();
            bundle.putLong("id", penjagaKost.getID());
            bundle.putString("namakost", penjagaKost.getNamaKost());
            bundle.putString("namapenjaga", penjagaKost.getNamaPenjaga());
            bundle.putString("teleponpenjaga", penjagaKost.getTeleponPenjaga());
            i.putExtras(bundle);
            i.putExtra( name: "IDPemilik", IDPemilik);
            startActivity(i);
            dialog.dismiss();
        }
    });
}

```



```
btnDeletePenjaga.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        AlertDialog.Builder konfirmasi = new AlertDialog.Builder(context);
        konfirmasi.setTitle("Hapus Data Penjaga Kost");
        konfirmasi.setMessage("Apakah Anda yakin akan menghapus data ini?");
        konfirmasi.setPositiveButton("Ya", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                dbHandler.deletePenjagaKost(id);

                finish();
                startActivity(getIntent());

                Toast.makeText(context, "Data Penjaga berhasil dihapus", Toast.LENGTH_SHORT).show();
            }
        });
        konfirmasi.setNegativeButton("Tidak", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
            }
        });
        konfirmasi.show();
        dialog.dismiss();
    }
});
return true;
}
```

Code 69. Code Java Penjaga

Pada coding Java diatas adalah coding yang digunakan untuk tampilan Menu Penjaga dengan memanfaatkan extends ListActivity dan implements AdapterView.OnItemLongClickListener. Hal pertama yang dilakukan adalah mendeklarasikan widget yang digunakan seperti Image Button dengan variabel btnTambahPenjaga dan btnKH, List View dengan variabel list, Button dengan variabel btnEditPenjaga dan btnDeletePenjaga. Kemudian mendeklarasikan class MyDBHandler dengan variabel dbHandler, ArrayList<PenjagaKost> dengan variabel values, Context dengan variabel context yang diinisialisasi berisi this atau artinya activity Penjaga, dan String IDPemilik. Setelahnya kita membuat fungsi @Override dengan method OnCreate yang berfungsi untuk melakukan inisialisasi pada saat kondisi awal Activity baru diciptakan, sehingga setContentView secara otomatis akan memperlihatkan halaman activity Penjaga. Kemudian kita menggunakan intent sebagai jembatan penghubung interaksi antar activity pada aplikasi android. Lalu, kita akan mendapatkan data tambahan berupa IDPemilik yang direpresentasikan sebagai string sehingga digunakan fungsi i.getStringExtra. Untuk menghubungkan activity Penjaga dengan database SQLite, pertama perlu memanggil class MyDBHandler yang sudah dibuat sebelumnya kemudian membuka koneksi ke database menggunakan fungsi open(). Kemudian menampung semua data yang tersimpan pada tabel di database ke dalam array list dengan menginisialisasi isi dari variabel array list yaitu values dengan memanggil method getAllPenjagaKost dari class MyDBHandler. Selanjutnya, digunakan ArrayAdapter untuk menampilkan nilai array yang terdapat pada variabel values ke dalam layout simple\_list\_item\_1.

Kemudian dilakukan pemanggilan image button dari tampilan berdasarkan ID menggunakan method findViewById. Dua image button yang dipanggil adalah btnTambahPenjaga dan btnKH. Kedua button ini akan kita set sehingga ketika di-klik akan merujuk ke aktivitas tertentu menggunakan method setOnClickListener. Pada activity ini, button btnTambahPenjaga dibuat untuk mengarahkan pengguna ke activity Tambah

Penjaga sehingga diperlukan penggunaan intent yang dipanggil dan disimpan dalam variabel i dan getApplicationContext serta rujukan activity class selanjutnya yaitu TambahPenjaga. Kemudian dilakukan parsing data IDPemilik yang sebelumnya diterima oleh activity dan ditampung pada variabel String IDPemilik dengan menggunakan fungsi putExtra. startActivity(i) berfungsi untuk memulai activity yang dituju. Sedangkan, button btnKH dibuat untuk mengarahkan pengguna ke activity Dashboard Pemilik Kost sehingga diperlukan penggunaan intent yang dipanggil dan disimpan dalam variabel i dan getApplicationContext serta rujukan activity class selanjutnya yaitu DashboardPemilik. Kemudian dilakukan parsing data IDPemilik yang sebelumnya diterima oleh activity dan ditampung pada variabel String IDPemilik dengan menggunakan fungsi putExtra. startActivity(i) berfungsi untuk memulai activity yang dituju. Sedangkan, Penjaga.this.finish(); yang akan terpanggil ketika kita menekan tombol sehingga activity yang sekarang akan di destroy dan kembali ke activity sebelumnya.

Untuk menampilkan data dalam bentuk list diperlukan list view yang dipanggil dari tampilan menggunakan method findViewById. List View ini berfungsi menampilkan data yang telah ditampung pada array list. Kemudian memudahkan proses selanjutnya ketika suatu item pada list ditekan, yaitu dilanjutkan ke proses edit atau delete. Untuk melakukan hal ini, dapat menggunakan method setOnItemClickListener. Pada activity ini memanfaatkan implementasi AdapterView.setOnItemClickListener sehingga dibuat method boolean onItemClick yang membuat dialog yang menampilkan layout dialog\_penjaga yang sudah dibuat. Untuk memberi petunjuk, pada dialog diberi judul “Pilih Aksi” kemudian dialog dimunculkan menggunakan fungsi show. Kemudian memanggil class PenjagaKost yang sudah dibuat sebelumnya yang diinisialisasikan berisi item yang dipilih, kemudian memperoleh id item tersebut dengan menggunakan method getId yang dibuat pada class PenjagaKost kemudian ditampung pada variabel id. Selanjutnya, dilakukan pemanggilan button dari tampilan berdasarkan ID menggunakan method findViewById. Dua button yang dipanggil adalah btnEditPenjaga dan btnDeletePenjaga. Kedua button ini akan kita set sehingga ketika di-klik akan merujuk ke aktivitas tertentu menggunakan method setOnClickListener. Pada activity ini, button btnEditPenjaga dibuat untuk mengarahkan pengguna ke activity Edit Penjaga sehingga diperlukan penggunaan intent yang dipanggil dan disimpan dalam variabel i dan getApplicationContext serta rujukan activity class selanjutnya yaitu EditPenjaga.

Kemudian dilakukan parsing data id, namakost, namapenjaga, dan telepon penjaga dengan memanfaatkan fungsi putLong dan putString dari class Bundle, dimana isi datanya diperoleh dari mengambil data dari method get yang dibuat pada class PenjagaKost. Semua data yang sudah diambil di bundle kemudian bundle di intent beserta dengan IDPemilik ke activity EditPenjaga menggunakan fungsi putExtra dan putExtras. startActivity(i) berfungsi untuk memulai activity yang dituju. dialog.dismiss() digunakan untuk menutup dialog yang dimunculkan. Sedangkan pada button btnDeletePenjaga dibuat untuk melakukan proses delete data dari database. Untuk mengkonfirmasi penghapusan data,

dimunculkan alert dialog dengan nama konfirmasi, yang apabila user menekan “Ya”, akan dipanggil method deletePenjagaKost dengan parameter id dari MyDBHandler lalu proses ini akan diselesaikan menggunakan fungsi finish(), kemudian melakukan intent ke halaman penjaga kost sehingga tampilan pada halaman kost seperti di refresh, kemudian memunculkan Toast yang menunjukkan keberhasilan penghapusan data. Namun, apabila user menekan “Tidak”, proses delete tidak akan dilakukan. konfirmasi.show() digunakan untuk memunculkan alert dialog, sedangkan dialog.dismiss() digunakan untuk menutup dialog aksi. Pada bagian akhir code terdapat code return true yaitu mengembalikan nilai true untuk method boolean onItemLongClick artinya item tersebut ditekan lama oleh pengguna.

## Coding Java Tambah Penjaga

```
public class TambahPenjaga extends AppCompatActivity {
    String IDPemilik;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_tambah_penjaga);
        final EditText edtNamaKost = (EditText) findViewById(R.id.edtNamaKost);
        final EditText edtNamaPenjaga = (EditText) findViewById(R.id.edtNamaPenjaga);
        final EditText edtTeleponPenjaga = (EditText) findViewById(R.id.edtTeleponPenjaga);
        Button btnSimpan = (Button) findViewById(R.id.btnSimpan);
        Intent i = getIntent();
        IDPemilik = i.getStringExtra( name: "IDPemilik");

        //Buat objek untuk Class MyDBHandler
        final MyDBHandler dbHandler = new MyDBHandler( context: this);
        //Membuka Koneksi Database
        try {
            dbHandler.open();
        } catch (SQLException e) {
            e.printStackTrace();
        }

        btnSimpan.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                PenjagaKost penjagaKost = new PenjagaKost();
                String namakost = edtNamaKost.getText().toString();
                String namaPenjaga = edtNamaPenjaga.getText().toString();
                String teleponPenjaga = edtTeleponPenjaga.getText().toString();
                dbHandler.createPenjagaKost(namakost, namaPenjaga, teleponPenjaga);
                Toast.makeText( context: TambahPenjaga.this, text: "Data berhasil ditambahkan", Toast.LENGTH_SHORT).show();
                edtNamaKost.setText("");
                edtNamaPenjaga.setText("");
                edtTeleponPenjaga.setText("");
                edtNamaKost.requestFocus();
                Intent i = new Intent( packageContext: TambahPenjaga.this, Penjaga.class);
                i.putExtra( name: "IDPemilik", IDPemilik);
                startActivity(i);
                TambahPenjaga.this.finish();
                dbHandler.close();
            }
        });
    }
}
```

**Code 70. Code Java Tambah Penjaga**

Pada coding Java diatas adalah coding yang digunakan untuk tampilan Tambah Penjaga. Hal pertama yang dilakukan adalah mendeklarasikan String namaKost, namaPenjaga, nomorPenjaga dan long ID. String lainnya juga diinisialisasikan seperti IDPemilik. Setelahnya kita membuat fungsi @Override dengan method OnCreate yang berfungsi untuk melakukan inisialisasi pada saat kondisi awal Activity baru diciptakan, sehingga setContentView secara otomatis akan memperlihatkan halaman activity EditPenjaga. Get Intent digunakan untuk mengambil beberapa data dari variabel dari activity sebelumnya yang dikirim. myDBHandler digunakan untuk mengambil data dari SQLite dengan database lokal yaitu datapenjaga.db. dengan tabel bernama penjaga yang

memiliki kolom id, namakost, namapenjaga, dan teleponpenjaga. Lalu dibuat constructor untuk class MyDBHandler yang membuat database yang sudah dideklarasikan. setOnItemClickListener pada button digunakan untuk mendeklarasikan aksi yang akan dijalankan setelah tombol ditekan, biasanya digunakan untuk menampilkan activity lain. kita dapat menggunakan onClick menggunakan mendapatkan id dari setiap tombol atau tampilan. Toast digunakan untuk memberikan respon tertentu seperti Toast akan error yang didapatkan.

## Coding Java Edit Penjaga

```

package com.projectK1.kosthost;

import ...

public class EditPenjaga extends AppCompatActivity {
    private long id;
    private String namaKost;
    private String namaPenjaga;
    private String teleponPenjaga;
    String IDPemilik;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_edit_penjaga);
        Intent i = getIntent();
        IDPemilik = i.getStringExtra("IDPemilik");
        final EditText edtNamaKost = (EditText) findViewById(R.id.edtNamaKost);
        final EditText edtNamaPenjaga = (EditText) findViewById(R.id.edtNamaPenjaga);
        final EditText edtTeleponPenjaga = (EditText) findViewById(R.id.edtTeleponPenjaga);
        Button btnSimpan = (Button) findViewById(R.id.btnSimpan);

        //Buat objek untuk Class MyDBHandler
        final MyDBHandler dbHandler = new MyDBHandler(context, this);
        //Membuka Koneksi Database
        try {
            dbHandler.open();
        } catch (SQLException e) {
            e.printStackTrace();
        }

        Bundle bundle = this.getIntent().getExtras();
        id = bundle.getLong("id");
        namaKost = bundle.getString("namaKost");
        namaPenjaga = bundle.getString("namaPenjaga");
        teleponPenjaga = bundle.getString("teleponPenjaga");
        this.setTitle("Edit Data dengan ID: " + id);
        edtNamaKost.setText(namaKost);
        edtNamaPenjaga.setText(namaPenjaga);
        edtTeleponPenjaga.setText(teleponPenjaga);

        btnSimpan.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                PenjagaKost penjagaKost = new PenjagaKost();
                penjagaKost.setId(id);
                penjagaKost.setNamaKost(edtNamaKost.getText().toString());
                penjagaKost.setNamaPenjaga(edtNamaPenjaga.getText().toString());
                penjagaKost.setTeleponPenjaga(edtTeleponPenjaga.getText().toString());
                dbHandler.updatePenjagaKost(penjagaKost);

                Toast.makeText(context, EditPenjaga.this, text: "Data Penjaga berhasil diedit", Toast.LENGTH_SHORT).show();
                Intent i = new Intent(getApplicationContext(), EditPenjaga.this, Penjaga.class);
                i.putExtra("IDPemilik", IDPemilik);
                startActivity(i);
                EditPenjaga.this.finish();
                dbHandler.close();
            }
        });
    }
}

```

**Code 71. Code Java Edit Penjaga**

Pada coding Java diatas adalah coding yang digunakan untuk tampilan Edit Penjaga. Hal pertama yang dilakukan adalah mendeklarasikan String namaKost, namaPenjaga, nomorPenjaga dan long ID. String lainnya juga diinisialisasikan seperti IDPemilik. Setelahnya kita membuat fungsi @Override dengan method OnCreate yang berfungsi untuk melakukan inisialisasi pada saat kondisi awal Activity baru diciptakan, sehingga setContentView secara otomatis akan memperlihatkan halaman activity EditPenjaga. Get Intent digunakan untuk mengambil beberapa data dari variabel dari activity sebelumnya yang dikirim. myDBHandler digunakan untuk mengambil data dari SQLite dengan database lokal yaitu datapenjaga.db. dengan tabel bernama penjaga yang memiliki kolom id, namakost, namapenjaga, dan teleponpenjaga. Lalu dibuat constructor

untuk class MyDBHandler yang membuat database yang sudah dideklarasikan. setOnItemClickListener pada button digunakan untuk mendeklarasikan aksi yang akan dijalankan setelah tombol ditekan, biasanya digunakan untuk menampilkan activity lain. kita dapat menggunakan onClick menggunakan mendapatkan id dari setiap tombol atau tampilan. Toast digunakan untuk memberikan respon tertentu seperti Toast akan error yang didapatkan.

## Coding Java Menu FAQ

```
package com.projectK1.kosthost;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.ImageButton;

public class faqPemilik extends AppCompatActivity {
    private ImageButton btnKH;
    String IDPemilik;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_faq_pemilik);
        Intent i = getIntent();
        IDPemilik = i.getStringExtra("IDPemilik");
        btnKH = findViewById(R.id.btnKH);
        btnKH.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new Intent(getApplicationContext(), DashboardPemilik.class);
                i.putExtra("IDPemilik", IDPemilik);
                startActivity(i);
                faqPemilik.this.finish();
            }
        });
    }
}
```

Code 72. Code Java FAQ Pemilik

Pada coding Java diatas adalah coding yang digunakan untuk tampilan FAQ pemilik Kost. Hal pertama yang dilakukan adalah mendeklarasikan image button dengan id btnKH, dan String IDPemilik. Setelahnya kita membuat fungsi @Override dengan method OnCreate yang berfungsi untuk melakukan inisialisasi pada saat kondisi awal Activity baru diciptakan, sehingga setContentView secara otomatis akan memperlihatkan halaman activity FAQ pemilik. Kemudian kita menggunakan intent sebagai jembatan penghubung interaksi antar activity pada aplikasi android. Lalu, kita akan mendapatkan data tambahan berupa IDPemilik yang direpresentasikan sebagai string sehingga digunakan fungsi i.getStringExtra. Kemudian memanggil button dari activity berdasarkan IDnya dengan method findViewById. Button KH ini akan kita set sehingga ketika di-klik akan merujuk ke aktivitas tertentu sehingga digunakan method setOnClickListener. Pada activity ini, button KH dibuat untuk mengarahkan user kembali ke dashboard pemilik sehingga diperlukan penggunaan intent yang dipanggil dan disimpan dalam variabel i dan getApplicationContext yang merujuk ke activity class Dashboard Pemilik. startActivity(i) berfungsi untuk memulai activity yang dituju. Pada akhir java ditulis Faqpemilik.this.finish(); yang akan terpanggil ketika kita menekan tombol sehingga activity yang sekarang akan di destroy dan kembali ke activity sebelumnya.

## Coding Java Menu Profile

```
package com.projectK1.kosthost;

import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;
import com.android.volley.AuthFailureError;
import com.android.volley.NetworkError;
import com.android.volley.NoConnectionError;
import com.android.volley.ParseError;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.ServerError;
import com.android.volley.TimeoutError;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import com.squareup.picasso.Picasso;
import org.json.JSONException;
import org.json.JSONObject;
import java.util.HashMap;
import java.util.Map;

public class ProfilePemilik extends AppCompatActivity {
    private Button btnKembali, btnKeluar, btnEditProfil;
    String URL_PROFILE_PEMILIK = "http://192.168.1.7/kosthostmobile/profil-pemilik.php";
    String namaPemilik, tanggallahir, telepon, email, IDPemilik, fotoPemilik;
    private TextView txtNamaPemilik, txtTanggallahir, txtTelepon, txtEmail, txtStatus;
    String URL_FOTO_PEMILIK;
    private ImageView imagePemilik;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_profile_pemilik);
        btnKembali = findViewById(R.id.btnKembali);
        btnKeluar = findViewById(R.id.btnKeluar);
        btnEditProfil = findViewById(R.id.btnEditProfil);
        txtNamaPemilik = findViewById(R.id.txtNamaPemilik);
        txtTanggallahir = findViewById(R.id.txtTanggallahir);
        txtTelepon = findViewById(R.id.txtTelepon);
        txtEmail = findViewById(R.id.txtEmail);
        txtStatus = findViewById(R.id.txtStatus);
        imagePemilik = findViewById(R.id.imagePemilik);
        Intent i = getIntent();
        IDPemilik = i.getStringExtra("IDPemilik");
        profilePemilik(IDPemilik);

        btnKembali.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new Intent(getApplicationContext(), DashboardPemilik.class);
                i.putExtra("IDPemilik", IDPemilik);
                startActivity(i);
            }
        });
        btnKeluar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new Intent(getApplicationContext(), MainActivity.class);
                startActivity(i);
            }
        });
    }
}
```

```

        btnEditProfil.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new Intent(getApplicationContext(), EditProfilePemilik.class);
                i.putExtra( name: "IDPemilik", IDPemilik);
                i.putExtra( name: "namaPemilik", namaPemilik);
                i.putExtra( name: "tanggallahir", tanggallahir);
                i.putExtra( name: "telepon", telepon);
                i.putExtra( name: "email", email);
                i.putExtra( name: "status", txtStatus.getText().toString());
                i.putExtra( name: "fotoPemilik", fotoPemilik);
                startActivity(i);
            }
        });

    private void profilePemilik(String IDPemilik) {
        RequestQueue queue = Volley.newRequestQueue( context: ProfilePemilik.this);
        StringRequest request = new StringRequest(Request.Method.POST, URL_PROFILE_PEMILIK,
            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) {
                            namaPemilik = jsonObject.getString( name: "namaPemilik");
                            tanggallahir = jsonObject.getString( name: "tanggallahir");
                            telepon = jsonObject.getString( name: "telepon");
                            email = jsonObject.getString( name: "email");
                            fotoPemilik = jsonObject.getString( name: "fotoPemilik");
                            URL_FOTO_PEMILIK = "http://192.168.1.7/kosthostmobile/" + fotoPemilik;
                            Picasso.get().load(URL_FOTO_PEMILIK).into(imagePemilik);

                            txtNamaPemilik.setText(namaPemilik);
                            txtTanggallahir.setText(tanggallahir);
                            txtTelepon.setText(telepon);
                            txtEmail.setText(email);
                            txtStatus.setText("Pemilik Kost");
                        } else {
                            Toast.makeText( context: ProfilePemilik.this, text: "Data profil gagal diambil", Toast.LENGTH_SHORT).show();
                        }
                    } catch (JSONException e) {
                        Toast.makeText( context: ProfilePemilik.this, e.getMessage(), Toast.LENGTH_SHORT).show();
                    }
                }
            }, new Response.ErrorListener() {
                @Override
                public void onErrorResponse(VolleyError error) {
                    String message = null;
                    if (error instanceof NetworkError) {
                        message = "Cannot connect to Internet...Please check your connection!";
                    } else if (error instanceof ServerError) {
                        message = "The server could not be found. Please try again after some time!";
                    } else if (error instanceof AuthFailureError) {
                        message = "Cannot connect to Internet...Please check your connection!";
                    } else if (error instanceof ParseError) {
                        message = "Parsing error! Please try again after some time!";
                    } else if (error instanceof NoConnectionError) {
                        message = "Cannot connect to Internet...Please check your connection!";
                    } else if (error instanceof TimeoutError) {
                        message = "Connection TimeOut! Please check your internet connection.";
                    }
                    Toast.makeText( context: ProfilePemilik.this, message, Toast.LENGTH_SHORT).show();
                }
            });
        });

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

queue.getCache().clear();
queue.add(request);
}

```

### Code 73. Code Java Profile Pemilik

Pada coding Java diatas adalah coding yang digunakan untuk tampilan Profile Pemilik. Hal pertama yang dilakukan adalah mendeklarasikan Button yang digunakan pada activity yaitu btnKembali, btnKeluar, dan btnEditprofil, Text View yang digunakan pada activity yaitu txtNamaPemilik, txtTanggallahir, txtTelepon, txtEmail, dan txtStatus, serta Image View yang digunakan pada activity yaitu imagePemilik. Terdapat URL\_PROFILE\_PEMILIK sebagai lokasi file php yang akan digunakan untuk melakukan CRUD dari android ke dalam database dan String URL\_FOTO\_PEMILIK yang akan digunakan untuk menampung url foto yang tersimpan pada database. Berbagai String

lainnya juga dideklarasikan seperti namaPemilik, tanggalLahir, telepon, email, IDPemilik, dan fotoPemilik. Setelahnya kita membuat fungsi @Override dengan method OnCreate yang berfungsi untuk melakukan inisialisasi pada saat kondisi awal Activity baru diciptakan, sehingga setContentView secara otomatis akan memperlihatkan halaman activity ProfilePemilik. Kemudian dilakukan pemanggilan button, text view, dan image view yang digunakan pada activity dari tampilan berdasarkan ID menggunakan method findViewById. Get Intent digunakan untuk mengambil data dari activity sebelumnya yang dikirim. Variabel yang diterima di activity ini adalah IDPemilik yang didapatkan dari Dashboard Pemilik. Kemudian IDPemilik dijadikan parameter pada pemanggilan method profilPemilik yaitu method yang digunakan untuk menampilkan informasi akun pemilik yang sedang log in dari database. Terdapat 3 button pada activity ini yang akan kita set sehingga ketika diklik akan merujuk ke aktivitas tertentu menggunakan method setOnClickListener. Button pertama adalah btnKembali yang berfungsi mengarahkan user ke activity Dashboard Pemilik sehingga diperlukan penggunaan intent yang dipanggil dan disimpan dalam variabel i dan getApplicationContext serta rujukan activity class selanjutnya yaitu DashboardPemilik. Kemudian dilakukan parsing data IDPemilik yang sebelumnya diterima oleh activity dan ditampung pada variabel String IDPemilik dengan menggunakan fungsi putExtra. startActivity(i) berfungsi untuk memulai activity yang dituju. Button berikutnya adalah btnKeluar yang digunakan untuk Log Out dari aplikasi dan kembali ke halaman utama aplikasi yaitu MainActivity sehingga diperlukan penggunaan intent yang dipanggil dan disimpan dalam variabel i dan getApplicationContext serta rujukan activity class selanjutnya yaitu MainActivity. startActivity(i) berfungsi untuk memulai activity yang dituju. Button terakhir adalah btnEditProfil yang berfungsi mengarahkan user ke activity Edit Profil Pemilik sehingga diperlukan penggunaan intent yang dipanggil dan disimpan dalam variabel i dan getApplicationContext serta rujukan activity class selanjutnya yaitu EditProfilePemilik. Kemudian dilakukan parsing data IDPemilik, namaPemilik, tanggalLahir, telepon, email, status, dan fotoPemilik yang terdapat pada activity dengan fungsi putExtra. startActivity(i) berfungsi untuk memulai activity yang dituju.

Method profilePemilik yang akan dipanggil merupakan method yang digunakan untuk menampilkan data akun pemilik dengan yang tersimpan pada database tabel AkunPemilikKost. Pada method digunakan RequestQueue dan StringRequest yang digunakan untuk menghubungkan coding yang kita buat dengan URL lokasi file php agar dapat terkoneksi dengan database MySQL. OnResponse digunakan sebagai penampung dari hasil proses di dalamnya. Proses di dalamnya meliputi try dan catch untuk mendapatkan data variabel yang dikirimkan dari file php, seperti nama pemilik, tanggal lahir, telepon, email, dan foto pemilik. Lokasi penyimpanan foto pemilik yang tersimpan pada database digabungkan dengan ip address dan folder pada htdocs disimpan pada variabel URL\_FOTO PEMILIK. Selanjutnya, untuk menampilkan foto yang URL nya sudah ditampung pada variabel, digunakan Picasso dengan memanfaatkan fungsi get,

kemudian fungsi load dengan menyertakan URL yang ditampung pada URL\_FOTO\_PENCARI untuk memuat gambar, lalu fungsi into untuk memuat gambar yang sudah didapatkan ke dalam image view. Setelah didapatkan dengan membuat suatu JSONObject, data-data tersebut di set ke masing-masing text view yang sudah dideklarasikan. Apabila data gagal diambil, akan muncul pesan error dalam toast yang menyatakan bahwa Data Profil gagal diambil. onErrorResponse bermanfaat untuk memberikan respon tertentu seperti toast akan error yang didapatkan. Sedangkan di bawahnya, getParams() digunakan untuk mengirimkan variabel yang diperlukan dalam file php untuk melakukan query. Dalam file ini diperlukan IDPemilik agar query MySQL pada file php dapat berjalan dengan baik yaitu mendapatkan data AkunPemilikKost berdasarkan IDPemilik yang login. Bagian akhir, harus ditambahkan queue.getCache().clear(); dan queue.add(request); agar program dapat berjalan dan terupdate tampilannya.

## Coding Java Edit Profile Pemilik

```

package com.projectk1.kosthost;

import ...

public class EditProfilePemilik extends AppCompatActivity {
    private Button btnKembali, btnSimpan, btnDatePicker, btnChoose;
    int year, month, dayOfMonth;
    Calendar calendar;
    Bitmap bitmap;
    private ImageView imagePemilik;
    DatePickerDialog datePickerDialog;
    String fotoPemilik;
    String URL_FOTO_PEMILIK;
    final int CODE_GALLERY_REQUEST = 999;
    String URL_EDITPROFILE_PEMILIK = "http://192.168.1.7/kosthostmobile/editprofile-pemilik.php";
    private TextView txtNamaPemilik, txtTanggalLahir, txtTelepon, txtEmail, txtStatus;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_edit_profile_pemilik);
        btnKembali = findViewById(R.id.btnKembali);
        btnSimpan = findViewById(R.id.btnSimpan);
        txtNamaPemilik = findViewById(R.id.txtNamaPemilik);
        txtTanggalLahir = findViewById(R.id.txtTanggalLahir);
        btnDatePicker = findViewById(R.id.btnTimePicker);
        txtTelepon = findViewById(R.id.txtTelepon);
        txtEmail = findViewById(R.id.txtEmail);
        txtStatus = findViewById(R.id.txtStatus);
        btnChoose = findViewById(R.id.btnChoose);
        imagePemilik = findViewById(R.id.imagePemilik);

        Intent i = getIntent();
        String IDPemilik = i.getStringExtra("IDPemilik");
        String namaPemilik = i.getStringExtra("namaPemilik");
        String tanggalLahir = i.getStringExtra("tanggalLahir");
        String telepon = i.getStringExtra("telepon");
        String email = i.getStringExtra("email");
        String status = i.getStringExtra("status");
        String fotoPemilik = i.getStringExtra("fotoPemilik");

        URL_FOTO_PEMILIK = "http://192.168.1.7/kosthostmobile/" + fotoPemilik;
        Picasso.get().load(URL_FOTO_PEMILIK).into(imagePemilik);
        txtNamaPemilik.setText(namaPemilik);
        txtTanggalLahir.setText(tanggalLahir);
        txtTelepon.setText(telepon);
        txtEmail.setText(email);
        txtStatus.setText(status);

        btnChoose.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                ActivityCompat.requestPermissions(
                    EditProfilePemilik.this,
                    new String[]{Manifest.permission.READ_EXTERNAL_STORAGE},
                    CODE_GALLERY_REQUEST
                );
            }
        });
    }
}

```

```

btnKembali.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent i = new Intent(getApplicationContext(), ProfilePemilik.class);
        i.putExtra("IDPemilik", IDPemilik);
        startActivity(i);
        EditProfilePemilik.this.finish();
    }
});

btnDatePicker.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        calendar = Calendar.getInstance();
        year = calendar.get(Calendar.YEAR);
        month = calendar.get(Calendar.MONTH);
        dayOfMonth = calendar.get(Calendar.DAY_OF_MONTH);
        datePickerDialog = new DatePickerDialog(EditProfilePemilik.this,
                new DatePickerDialog.OnDateSetListener() {
                    @Override
                    public void onDateSet(DatePicker datePicker, int year, int month, int day) {
                        txtTanggalLahir.setText(year + "-" + (month + 1) + "-" + day);
                    }
                }, year, month, dayOfMonth);
        datePickerDialog.show();
    }
});

btnSimpan.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String namaPemilik = txtNamaPemilik.getText().toString();
        String tanggalLahir = txtTanggalLahir.getText().toString();
        String telepon = txtTelepon.getText().toString();
        editProfilePemilik(IDPemilik, namaPemilik, tanggalLahir, telepon);
    }
});
}

private void editProfilePemilik(String IDPemilik, String namaPemilik, String tanggalLahir, String telepon) {
    RequestQueue queue = Volley.newRequestQueue(EditProfilePemilik.this);
    StringRequest request = new StringRequest(Request.Method.POST, URL_EDITPROFILE PEMILIK, new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            try {
                JSONObject jsonObject = new JSONObject(response);
                int success = jsonObject.getInt("success");
                if (success == 1) {
                    Toast.makeText(EditProfilePemilik.this, "Profil anda berhasil diedit", Toast.LENGTH_SHORT).show();
                    Intent i = new Intent(getApplicationContext(), ProfilePemilik.class);
                    i.putExtra("IDPemilik", IDPemilik);
                    startActivity(i);
                    EditProfilePemilik.this.finish();
                } else {
                    Toast.makeText(EditProfilePemilik.this, "Profil anda gagal diedit", Toast.LENGTH_SHORT).show();
                }
            } catch (JSONException e) {
                Toast.makeText(EditProfilePemilik.this, e.getMessage(), Toast.LENGTH_SHORT).show();
            }
        }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            String message = null;
            if (error instanceof NetworkError) {
                message = "Cannot connect to Internet...Please check your connection!";
            } else if (error instanceof ServerError) {
                message = "The server could not be found. Please try again after some time!!";
            } else if (error instanceof AuthFailureError) {
                message = "Cannot connect to Internet...Please check your connection!";
            } else if (error instanceof ParseError) {
                message = "Parsing error! Please try again after some time!!";
            } else if (error instanceof NoConnectionError) {
                message = "Cannot connect to Internet...Please check your connection!";
            } else if (error instanceof TimeoutError) {
                message = "Connection Timeout! Please check your internet connection.";
            }
            Toast.makeText(EditProfilePemilik.this, message, Toast.LENGTH_SHORT).show();
        }
    });
    @Nullable
    @Override
    protected Map<String, String> getParams() throws AuthFailureError {
        Map<String, String> params = new HashMap<>();
        fotoPemilik = imageToString(bitmap);
        params.put("IDPemilik", IDPemilik);
        params.put("namaPemilik", namaPemilik);
        params.put("tanggalLahir", tanggalLahir);
        params.put("telepon", telepon);
        params.put("fotoPemilik", fotoPemilik);
        return params;
    }
};

queue.getCache().clear();
queue.add(request);
}
}

```

```

    @Override
    protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
        if (requestCode == CODE_GALLERY_REQUEST && resultCode == RESULT_OK && data != null) {
            Uri filePath = data.getData();

            try {
                InputStream inputStream = getContentResolver().openInputStream(filePath);
                bitmap = BitmapFactory.decodeStream(inputStream);
                imagePemilik.setImageBitmap(bitmap);

            } catch (FileNotFoundException e) {
                e.printStackTrace();
            }
        }
    }

    super.onActivityResult(requestCode, resultCode, data);
}

@Override
public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {
    if (requestCode == CODE_GALLERY_REQUEST) {
        if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
            Intent intent = new Intent(Intent.ACTION_PICK);
            intent.setType("image/*");
            startActivityForResult(intent.createChooser(intent, "Select Image"), CODE_GALLERY_REQUEST);
        } else {
            Toast.makeText(getApplicationContext(), "Tidak ada akses ke gallery", Toast.LENGTH_LONG).show();
        }
        return;
    }
    super.onRequestPermissionsResult(requestCode, permissions, grantResults);
}

private String imageToString(Bitmap bitmap) {
    ByteArrayOutputStream outputStream = new ByteArrayOutputStream();
    bitmap.compress(Bitmap.CompressFormat.JPEG, 100, outputStream);
    byte[] imageBytes = outputStream.toByteArray();

    String encodedImage = Base64.encodeToString(imageBytes, android.util.Base64.DEFAULT);
    return encodedImage;
}
}

```

**Code 74. Code Java Edit Profile Pemilik**

Code diatas merupakan code yang digunakan untuk pemilik mengedit profile mereka. Hal pertama yang dilakukan adalah mendeklarasikan Button btnKembali, btnSimpan, btnDatePicker, btnChoose dan ImageView imagePemilik. Berbagai TextView lainnya juga diinisialisasikan seperti txtNamaPemilik, txtTanggalLahir, txtTelepon, txtEmail, txtStatus. Selain itu juga dilakukan insialisasi terhadap berbagai Int yang digunakan seperti year, month, dayOfMonth. Deklarasikan variabel lain seperti Calendar dan DatePickerDialog. Lalu terdapat Bitmap dengan id bitmap yang berhubungan dengan imageKost. Kemudian final int Code\_Gallery\_Request yang digunakan untuk membaca data external. Terdapat 2 URL, pertama adalah URL\_FOTO\_PEMILIK sebagai lokasi file php yang akan digunakan untuk mendapatkan foto pemilik dari database. Selanjutnya adalah URL\_EDITPROFILE\_PEMILIK sebagai lokasi file php yang akan digunakan untuk melakukan EDIT foto pemilik dari android ke dalam database.

Setelahnya kita membuat fungsi @Override dengan method OnCreate yang berfungsi untuk melakukan inisialisasi pada saat kondisi awal Activity baru diciptakan, sehingga setContentView secara otomatis akan memperlihatkan halaman activity Edit Profile Pemilik. Get Intent digunakan untuk mengambil beberapa data dari variabel dari activity sebelumnya yang dikirim. Find view by id digunakan untuk melakukan inisialisasi variabel yang sudah dideklarasikan di awal agar dapat terkoneksi dengan komponen-komponen dalam activity ini.URL\_FOTO\_PEMILIK = "http://192.168.1.7/kosthostmobile/editprofile-pemilik.php" digunakan untuk membuat suatu URL yang tepat untuk mengambil foto profil pemilik dari database. Untuk

menetapkan foto tersebut dari URL ke dalam ImageView dapat menggunakan Picasso, yaitu dengan Picasso.get().load (URL\_FOTO\_PEMILIK).into(imagePemilik); Untuk menggunakan metode Picasso, harus menambahkan implementation 'com.squareup.picasso:picasso:2.71828' pada build.gradle. setOnClickListener pada button digunakan untuk mendeklarasikan aksi yang akan dijalankan setelah tombol ditekan, biasanya digunakan untuk menampilkan activity lain. RequestQueue dan StringRequest digunakan untuk menghubungkan coding yang kita buat dengan URL lokasi file php agar dapat terkoneksi dengan database MySQL.OnResponse digunakan sebagai penampung dari hasil proses di dalamnya. Proses di dalamnya meliputi try dan catch untuk menjalankan perintah sesuai dengan file php.

Setelah didapatkan response dengan membuat suatu JSONObject, nilai dari success = 1 menandakan bahwa proses berjalan dengan lancar sedangkan selain 1 berarti proses tidak berjalan dengan baik dan terdapat error. Jika success = 1, akan muncul toast bahwa kost berhasil di book dan melakukan intent ke activity. OnErrorResponse digunakan untuk memberikan respon tertentu seperti toast Error. Sedangkan di bawahnya, getParams() digunakan untuk mengirimkan variabel yang diperlukan dalam file php untuk melakukan query. Dalam file ini diperlukan fotoPemilik, IDPemilik, namaPemilik, tanggalLahir, dan telepon agar query MySQL pada file php dapat berjalan dengan baik yaitu mendapatkan data IDKost yang akan dibook berdasarkan IDPencari yang login. Bagian akhir, harus ditambahkan queue.getCache().clear(); dan queue.add(request); agar program dapat berjalan dan terupdate tampilannya. OnRequestPermissionsResult digunakan jika aplikasi perlu menggunakan sumber daya atau informasi di luar kotak pasirnya sendiri, kita dapat mendeklarasikan izin dan menyiapkan permintaan izin yang menyediakan akses ini. Ini adalah bagian dari alur kerja untuk menggunakan izin. OnActivityResult digunakan untuk memulai aktivitas lain, baik dalam aplikasi maupun dari aplikasi lain, tidak harus berupa operasi satu arah. kita juga dapat memulai aktivitas lain dan menerima hasilnya kembali. aplikasi dapat menerima foto yang diambil sebagai hasilnya. imageToString digunakan untuk mengubah file gambar menjadi String.

## Coding Java pada Aplikasi Bagian Pencari Kost

### Coding Java Dashboard Pencari Kost

```
package com.projectK1.kosthost;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageButton;

public class DashboardPencari extends AppCompatActivity {
    private Button btnSearch, btnBooked, btnFAQ, btnProfile;
    private ImageButton btnKH;
    String IDPencari;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_dashboard_pencari);
        btnSearch = findViewById(R.id.btnSearch);
        btnBooked = findViewById(R.id.btnBooked);
        btnFAQ = findViewById(R.id.btnFAQ);
        btnProfile = findViewById(R.id.btnProfile);
        Intent i = getIntent();
        IDPencari = i.getStringExtra("IDPencari");

        btnSearch.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new Intent(getApplicationContext(), SearchPencari.class);
                i.putExtra("IDPencari", IDPencari);
                startActivity(i);
            }
        });

        btnBooked.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new Intent(getApplicationContext(), BookedPencari.class);
                i.putExtra("IDPencari", IDPencari);
                startActivity(i);
            }
        });

        btnFAQ.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new Intent(getApplicationContext(), faqPencari.class);
                i.putExtra("IDPencari", IDPencari);
                startActivity(i);
            }
        });

        btnProfile.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new Intent(getApplicationContext(), ProfilePencari.class);
                i.putExtra("IDPencari", IDPencari);
                startActivity(i);
            }
        });

        btnKH = findViewById(R.id.btnKH);
        btnKH.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new Intent(getApplicationContext(), DashboardPencari.class);
                i.putExtra("IDPencari", IDPencari);
                startActivity(i);
            }
        });
    }
}
```

Code 75. Code Java Dashboard Pencari Kost

Pada coding Java diatas adalah coding yang digunakan untuk tampilan Dashboard Pencari. Hal pertama yang dilakukan adalah mendeklarasikan Button btnSearch, btnBooked, btnFAQ, btnProfile, dan ImageButton btnKH. Setelahnya kita membuat fungsi @Override dengan method OnCreate yang berfungsi untuk melakukan inisialisasi pada saat kondisi awal Activity baru diciptakan, sehingga setContentView secara otomatis akan memperlihatkan halaman activity DashboardPencari. Get Intent digunakan untuk mengambil beberapa data dari variabel dari activity sebelumnya yang dikirim. Variabel yang diterima di activity ini adalah IDPencari yang didapatkan dari LoginPencari. Find view by id digunakan untuk melakukan inisialisasi variabel yang sudah dideklarasikan di awal agar dapat terkoneksi dengan komponen-komponen dalam activity ini.

Setiap button yang sudah dideklarasikan dibuatkan setOnClickListener agar dapat menjalankan perintah tertentu. btnSearch jika ditekan akan menampilkan activity SearchPencari. btnBooked jika ditekan akan menampilkan activity BookedPencari. btnFAQ jika ditekan akan menampilkan activity faqPencari. btnProfile jika ditekan akan menampilkan activity ProfilePencari. btnKH jika ditekan akan menampilkan activity DashboardPencari kembali. Setiap menjalankan activity baru, intent IDPencari dikirimkan agar data di setiap activity sesuai dengan IDPencari yang telah log in.

## Coding Java Menu Search

```

package com.projectkl.kosthost;

import ...

public class SearchPencari extends AppCompatActivity {
    private ListView listHasilCarikost;
    private Button btnSearch;
    private ImageButton btnKH;
    private EditText edtSearch;
    ArrayList<HashMap<String, String>> listHasilCarikost;
    String URL_GET_KOST = "http://192.168.1.7/kosthostmobile/getkostbykota_pencari.php";
    String IDKost, namakost, alamatkost, kotaKost, jenisKost, jumlahKamar, hargakost, deskripsiFasilitas, fotoKost;
    String IDPencari;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_search_pencari);
        edtSearch = findViewById(R.id.edtSearch);
        listHasilCarikost = findViewById(R.id.listHasilCarikost);
        listHasilCarikost = new ArrayList<>();
        btnSearch = findViewById(R.id.btnSearch);
        Intent i = getIntent();
        IDPencari = i.getStringExtra("IDPencari");
        btnSearch.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String kataKunci = edtSearch.getText().toString();
                listHasilCarikost.setAdapter(null);
                RequestQueue queue = Volley.newRequestQueue(SearchPencari.this);
                StringRequest request = new StringRequest(Request.Method.POST, URL_GET_KOST,
                    new Response.Listener<String>() {
                        @Override
                        public void onResponse(String response) {
                            try {
                                JSONObject jsonObject = new JSONObject(response);
                                JSONArray jsonArray = jsonObject.getJSONArray("data");
                                for (int i = 0; i < jsonArray.length(); i++) {
                                    JSONObject kost = jsonArray.getJSONObject(i);
                                    IDKost = kost.getString("IDKost");
                                    namakost = kost.getString("namakost");
                                    alamatkost = kost.getString("alamatkost");
                                    kotaKost = kost.getString("kotaKost");
                                    jenisKost = kost.getString("jenisKost");
                                    jumlahKamar = kost.getString("jumlahKamar");
                                    hargakost = kost.getString("hargakost");
                                    fotoKost = kost.getString("fotoKost");
                                    double dhargakost = Double.parseDouble(hargakost);
                                    Locale locale = new Locale("in", "ID");
                                    NumberFormat formatKiprah = NumberFormat.getCurrencyInstance(locale);
                                    hargakost = formatKiprah.format(dhargakost);
                                    deskripsiFasilitas = kost.getString("deskripsiFasilitas");

                                    HashMap<String, String> map = new HashMap<>();
                                    map.put("IDKost", IDKost);
                                    map.put("namakost", namakost);
                                    map.put("alamatkost", alamatkost);
                                    map.put("kotaKost", kotaKost);
                                    map.put("jenisKost", jenisKost);
                                    map.put("jumlahKamar", jumlahKamar);
                                    map.put("hargakost", hargakost);
                                    map.put("deskripsiFasilitas", deskripsiFasilitas);
                                    map.put("fotoKost", fotoKost);
                                }
                            } catch (JSONException e) {
                                e.printStackTrace();
                            }
                        }
                    });
                request.setultipart(false);
                queue.add(request);
            }
        });
    }
}

```

## **Code 76. Code Java Menu Search**

Pada coding Java diatas adalah coding yang digunakan untuk tampilan Menu Search. Hal pertama yang dilakukan adalah mendeklarasikan listHasilCariKost, btnSearch, btnKH, edtSearch. edtSearch digunakan untuk memasukkan data berupa text pada aplikasi android, disini edtSearch meminta pengguna untuk memasukkan kata kunci pencarian berupa nama kota. btnSearch merupakan tombol yang dapat digunakan untuk memulai

aktivitas atau suatu proses. Pada tampilan ini, btnSearch digunakan untuk melakukan proses pencarian yang hasilnya akan ditampilkan dalam bentuk List pada listHasilCariKost.

## Coding Java Book Kost

```
package com.projectk1.kosthost;

import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import com.android.volley.AuthFailureError;
import com.android.volley.NetworkError;
import com.android.volley.NoConnectionError;
import com.android.volley.ParseError;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.RetryPolicy;
import com.android.volley.ServerError;
import com.android.volley.TimeoutError;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import com.squareup.picasso.Picasso;

import org.json.JSONException;
import org.json.JSONObject;

import java.util.HashMap;
import java.util.Map;

public class BookKost extends AppCompatActivity {
    private TextView txtNamakost, txtHargaKost, txtAlamatKost, txtKotaKost, txtJenisKost, txtJumlahKamar, txtFasilitas, txtIDKost;
    String IDPencari, namakost, alamatkost, kotakost, jeniskost, jumlahkamar, hargakost, deskripsifasilitas, IDKost, fotoKost;
    String URL_INSERT_PENYEWAAN = "http://192.168.1.7/kosthostmobile/insert-penyewaan.php";
    private Button btnBookKost;
    String URL_FOTO_KOST;
    private ImageView imageKost;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_book_kost);
        txtNamakost = findViewById(R.id.txtNamakost);
        txtHargaKost = findViewById(R.id.txtHargaKost);
        txtAlamatKost = findViewById(R.id.txtAlamatKost);
        txtKotaKost = findViewById(R.id.txtKotaKost);
```

```

txtJenisKost = findViewById(R.id.txtJenisKost);
txtJumlahKamar = findViewById(R.id.txtJumlahKamar);
txtFasilitas = findViewById(R.id.txtFasilitas);
txtIDKost = findViewById(R.id.txtIDKost);
btnBookKost = findViewById(R.id.btnBookKost);
imageKost = findViewById(R.id.imageKost);

Intent i = getIntent();
IDPencari = i.getStringExtra( name: "IDPencari");
namakost = i.getStringExtra( name: "namakost");
txtNamaKost.setText(namakost);
alamatKost = i.getStringExtra( name: "alamatKost");
txtAlamatKost.setText(alamatKost);
kotaKost = i.getStringExtra( name: "kotaKost");
txtKotaKost.setText(kotaKost);
jenisKost = i.getStringExtra( name: "jenisKost");
txtJenisKost.setText(jenisKost);
jumlahKamar = i.getStringExtra( name: "jumlahKamar");
txtJumlahKamar.setText(jumlahKamar);
hangakost = i.getStringExtra( name: "hangakost");
txtHangakost.setText(hangakost);
deskripsiFasilitas = i.getStringExtra( name: "deskripsiFasilitas");
txtFasilitas.setText(deskripsiFasilitas);
IDKost = i.getStringExtra( name: "IDKost");
txtIDKost.setText(IDKost);
fotoKost = i.getStringExtra( name: "fotoKost");
URL_FOTO_KOST = "http://192.168.1.7/kosthostmobile/" + fotoKost;
Picasso.get().load(URL_FOTO_KOST).into(imageKost);

btnBookKost.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        IDKost = txtIDKost.getText().toString();
        bookKost(IDKost, IDPencari);
    }
});

private void bookKost(String idkost, String idPencari) {
    RequestQueue queue = Volley.newRequestQueue( context: BookKost.this);
    StringRequest request = new StringRequest(Request.Method.POST, URL_INSERT_PENYEWAAN, 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: BookKost.this, text: "Kost berhasil dibook", Toast.LENGTH_SHORT).show();
                    Intent i = new Intent(getApplicationContext(), BookedPencari.class);
                    i.putExtra( name: "IDPencari", IDPencari);
                    startActivity(i);
                    BookKost.this.finish();
                } else {
                    Toast.makeText( context: BookKost.this, text: "Kost gagal dibook", Toast.LENGTH_SHORT).show();
                }
            } catch (JSONException e) {
                Toast.makeText( context: BookKost.this, e.getMessage(), Toast.LENGTH_SHORT).show();
            }
        }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            String message = null;
            if (error instanceof NetworkError) {
                message = "Cannot connect to Internet...Please check your connection!";
            } else if (error instanceof ServerError) {
                message = "The server could not be found. Please try again after some time!";
            } else if (error instanceof AuthFailureError) {
                message = "Cannot connect to Internet...Please check your connection!";
            } else if (error instanceof ParseError) {
                message = "Parsing error! Please try again after some time!";
            } else if (error instanceof NoConnectionError) {
                message = "Cannot connect to Internet...Please check your connection!";
            } else if (error instanceof TimeoutError) {
                message = "Connection TimeOut! Please check your internet connection.";
            }
            Toast.makeText( context: BookKost.this, message, Toast.LENGTH_SHORT).show();
        }
    });
    @Nullable
    @Override
    protected Map<String, String> getParams() throws AuthFailureError {
        Map<String, String> params = new HashMap<>();
        params.put("IDKost", IDKost);
        params.put("IDPencari", IDPencari);
        return params;
    }
};

queue.getCache().clear();
queue.add(request);
}
}

```

**Code 77. Code Java Book Kost**

Pada coding Java diatas adalah coding yang digunakan untuk tampilan Book Kost. Hal pertama yang dilakukan adalah mendeklarasikan Button btnBookKost dan ImageView imageKost. Terdapat URL\_INSERT\_PENYEWAAN sebagai lokasi file php yang akan digunakan untuk melakukan insert data penyewaan dari android ke dalam database dan URL\_FOTO\_KOST untuk mendapatkan foto kost dari database. Berbagai String lainnya juga diinisialisasikan seperti IDPencari, namaKost, alamatKost, kotaKost, jenisKost, jumlahKamar, hargaKost, deskripsiFasilitas, IDKost, dan fotoKost. Selain itu juga dilakukan inisialisasi terhadap berbagai TextView yang digunakan. Setelahnya kita membuat fungsi @Override dengan method OnCreate yang berfungsi untuk melakukan inisialisasi pada saat kondisi awal Activity baru diciptakan, sehingga setContentView secara otomatis akan memperlihatkan halaman activity BookKost. Get Intent digunakan untuk mengambil beberapa data dari variabel dari activity sebelumnya yang dikirim. Variabel yang diterima di activity ini adalah IDPencari, namaKost, alamatKost, kotaKost, jenisKost, jumlahKamar, hargaKost, deskripsiFasilitas, dan IDKost yang didapatkan dari Search Pencari. Find view by id digunakan untuk melakukan inisialisasi variabel yang sudah dideklarasikan di awal agar dapat terkoneksi dengan komponen-komponen dalam activity ini. Terdapat setText untuk menetapkan text untuk berbagai TextView berdasarkan data yang didapatkan dari intent getStringExtra.

URL\_FOTO\_KOST = "http://192.168.1.7/kosthostmobile/" + fotoKost; digunakan untuk membuat suatu URL yang tepat untuk mengambil foto kost dari database. Untuk menetapkan foto tersebut dari URL ke dalam ImageView dapat menggunakan Picasso, yaitu dengan Picasso.get().load(URL\_FOTO\_KOST).into(imageKost); Untuk menggunakan metode Picasso, harus menambahkan implementation 'com.squareup.picasso:picasso:2.71828' pada build.gradle. setOnClickListener pada button digunakan untuk mendeklarasikan aksi yang akan dijalankan setelah tombol ditekan, biasanya digunakan untuk menampilkan activity lain. RequestQueue dan StringRequest digunakan untuk menghubungkan coding yang kita buat dengan URL lokasi file php agar dapat terkoneksi dengan database MySQL.

OnResponse digunakan sebagai penampung dari hasil proses di dalamnya. Proses di dalamnya meliputi try dan catch untuk menjalankan perintah sesuai dengan file php. Setelah didapatkan response dengan membuat suatu JSONObject, nilai dari success = 1 menandakan bahwa proses berjalan dengan lancar sedangkan selain 1 berarti proses tidak berjalan dengan baik dan terdapat error. Jika success = 1, akan muncul toast bahwa kost berhasil dibook dan melakukan intent ke activity BookedPencari dengan mengirimkan IDPencari.

onErrorResponse bermanfaat untuk memberikan respon tertentu seperti toast akan error yang didapatkan. Sedangkan di bawahnya, getParams() digunakan untuk mengirimkan variabel yang diperlukan dalam file php untuk melakukan query. Dalam file ini diperlukan IDKost dan IDPencari agar query MySQL pada file php dapat berjalan dengan baik yaitu mendapatkan data IDKost yang akan dibook berdasarkan IDPencari

yang login. Bagian akhir, harus ditambahkan queue.getCache().clear(); dan queue.add(request); agar program dapat berjalan dan terupdate tampilannya.

## Coding Java Menu Booked

```
package com.projectk1.kosthost;

import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;

import android.app.AlertDialog;
import android.app.Dialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.ImageButton;
import android.widget.ListView;
import android.widget.SimpleAdapter;
import android.widget.TextView;
import android.widget.Toast;

import com.android.volley.AuthFailureError;
import com.android.volley.NetworkError;
import com.android.volley.NoConnectionError;
import com.android.volley.ParseError;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.ServerError;
import com.android.volley.TimeoutError;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;

public class BookedPencari extends AppCompatActivity {
    private ListView listBooked;
    private ImageButton btnH;
    ArrayList<HashMap<String, String>> listbooked;
    String URL_GET_BOOKEDKOST = "http://192.168.1.7/kosthostmobile/getbooked-pencari.php";
    String namakost, alamatkost;
    String IDPencari;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_booked_pencari);
        Intent i = getIntent();
        IDPencari = i.getStringExtra("IDPencari");
        listBooked = findViewById(R.id.listBooked);
        listbooked = new ArrayList<>();
        RequestQueue queue = Volley.newRequestQueue(this);
        StringRequest request = new StringRequest(Request.Method.POST, URL_GET_BOOKEDKOST,
            new Response.Listener<String>() {
                @Override
                public void onResponse(String response) {
                    try {
                        JSONObject jsonObject = new JSONObject(response);
                        JSONArray jsonArray = jsonObject.getJSONArray("data");

                        for (int i = 0; i < jsonArray.length(); i++) {
                            JSONObject kost = jsonArray.getJSONObject(i);
                            namakost = kost.getString("namakost");
                            alamatkost = kost.getString("alamatkost");

                            HashMap<String, String> map = new HashMap<>();
                            map.put("namakost", namakost);
                            map.put("alamatkost", alamatkost);

                            listbooked.add(map);
                        }

                        String[] from = {"namakost", "alamatkost"};
                        int[] to = {R.id.txtNamakost, R.id.txtAlamatkost};
                       ListAdapter adapter = new SimpleAdapter(this, listbooked, R.layout.list_bookedpencari, from, to);
                        listBooked.setAdapter(adapter);
                    }
                }
            } catch (JSONException e) {
                Toast.makeText(this, e.getMessage(), Toast.LENGTH_SHORT).show();
            }
        }, new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError error) {
                String message = null;
                if (error instanceof NetworkError) {
                    message = "Cannot connect to Internet...Please check your connection!";
                } else if (error instanceof ServerError) {
                    message = "The server could not be found. Please try again after some time!";
                } else if (error instanceof AuthFailureError) {
                    message = "Cannot connect to Internet...Please check your connection!";
                }
            }
        });
    }
}
```

```

        } else if (error instanceof AuthFailureError) {
            message = "Cannot connect to Internet...Please check your connection!";
        } else if (error instanceof ParseError) {
            message = "Parsing error! Please try again after some time!!";
        } else if (error instanceof NoConnectionError) {
            message = "Cannot connect to Internet..Please check your connection!";
        } else if (error instanceof TimeoutError) {
            message = "Connection TimeOut! Please check your internet connection.";
        }
        Toast.makeText( context, BookedPencari.this, message, Toast.LENGTH_SHORT).show();
    }

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

queue.getCache().clear();
queue.add(request);

btnKH = findViewById(R.id.btnKH);
btnKH.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent i = new Intent(getApplicationContext(), DashboardPencari.class);
        i.putExtra("name: " + IDPencari, IDPencari);
        startActivity(i);
        BookedPencari.this.finish();
    }
});
}
}

```

**Code 78. Code Java Booked Pencari**

Pada coding Java diatas adalah coding yang digunakan untuk tampilan Booked Pencari. Hal pertama yang dilakukan adalah mendeklarasikan ListView dengan variabel listBooked, ImageButton dengan variabel btnKH. Lalu inisialisasi ArrayList<HashMap<String, String>> dengan nama listBooked yang bertujuan untuk tempat penyimpanan berbagai data yang nantinya akan ditampilkan ke dalam listBooked. Terdapat URL\_GET\_BOOKEDKOST sebagai lokasi file php yang akan digunakan untuk melakukan CRUD dari android ke dalam database. Berbagai String lainnya juga diinisialisasikan seperti IDKost, IDSewa, namaKost, namaPencari, telepon, status, dan IDPencari. Setelahnya kita membuat fungsi @Override dengan method OnCreate yang berfungsi untuk melakukan inisialisasi pada saat kondisi awal Activity baru diciptakan, sehingga setContentView secara otomatis akan memperlihatkan halaman activity BookedPencari. Get Intent digunakan untuk mengambil beberapa data dari variabel dari activity sebelumnya yang dikirim. Variabel yang diterima di activity ini adalah IDPencari yang didapatkan dari Dashboard Pencari. Find view by id digunakan untuk melakukan inisialisasi variabel yang sudah dideklarasikan di awal agar dapat terkoneksi dengan komponen-komponen dalam activity ini. Setelah listBooked diinisialisasi dengan findViewById, array list listbooked juga diinisialisasikan. RequestQueue dan StringRequest digunakan untuk menghubungkan coding yang kita buat dengan URL lokasi file php agar dapat terkoneksi dengan database MySQL.

OnResponse digunakan sebagai penampung dari hasil proses di dalamnya. Proses di dalamnya meliputi try dan catch untuk mendapatkan data variabel yang dikirimkan dari file php, seperti namaKost dan alamatKost. Setelah didapatkan dengan membuat suatu JSONObject, berbagai data tersebut disimpan ke dalam suatu adapter untuk ditampilkan di listBooked. Langkahnya adalah dengan menggunakan HashMap lalu dilakukan put ke dalam variabel-variabel yang akan disimpan ke dalam array listbooked. String[] from dan int[] to harus saling berhubungan untuk menandakan dari mana data yang diambil dan ke mana data tersebut akan ditampilkan. Data akan ditampilkan ke dalam layout list\_bookedpencari. list\_bookedpencari berisi berbagai komponen yang akan ditampilkan ke dalam listBooked dengan adapter.

onErrorResponse bermanfaat untuk memberikan respon tertentu seperti toast akan error yang didapatkan. Sedangkan di bawahnya, getParams() digunakan untuk mengirimkan variabel yang diperlukan dalam file php untuk melakukan query. Dalam file ini diperlukan IDPencari agar query MySQL pada file php dapat berjalan dengan baik yaitu mendapatkan data booked berdasarkan IDPencari yang login. Bagian akhir, harus ditambahkan queue.getCache().clear(); dan queue.add(request); agar program dapat berjalan dan terupdate tampilannya.

Selanjutnya adalah menetapkan setOnClickListener untuk listBooked, jika listBooked ditekan maka akan melakukan intent ke activity DetailBookedPencari dengan mengirimkan berbagai variabel yang dikirimkan. Terdapat setOnClickListener pula untuk btnKH yaitu ketiga logo KH ditekan, pengguna akan diarahkan untuk kembali ke dashboard. Semua intent yang digunakan harus diakhiri dengan startActivity(i) agar intent dapat berjalan.

## Coding Java Menu FAQ

```

package com.projectK1.kosthost;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.ImageButton;

public class faqPencari extends AppCompatActivity {
    private ImageButton btnKH;
    String IDPencari;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_faq_pencari);
        Intent i = getIntent();
        IDPencari = i.getStringExtra("IDPencari");
        btnKH = findViewById(R.id.btnKH);
        btnKH.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new Intent(getApplicationContext(), DashboardPencari.class);
                i.putExtra("IDPencari", IDPencari);
                startActivity(i);
                faqPencari.this.finish();
            }
        });
    }
}

```

Code 79. Code Java FAQ Pencari

Pada coding Java diatas adalah coding yang digunakan untuk tampilan FAQ pencari Kost. Hal pertama yang dilakukan adalah mendeklarasikan image button dengan id btnKH, dan String IDPencari. Setelahnya kita membuat fungsi @Override dengan method OnCreate yang berfungsi untuk melakukan inisialisasi pada saat kondisi awal Activity baru diciptakan, sehingga setContentView secara otomatis akan memperlihatkan halaman activity FAQ pemilik. Kemudian kita menggunakan intent sebagai jembatan penghubung interaksi antar activity pada aplikasi android. Lalu, kita akan mendapatkan data tambahan berupa IDPencari yang direpresentasikan sebagai string sehingga digunakan fungsi i.getStringExtra. Kemudian memanggil button dari activity berdasarkan IDnya dengan method findViewById. Button KH ini akan kita set sehingga ketika di-klik akan merujuk ke aktivitas tertentu sehingga digunakan method setOnClickListener. Pada activity ini, button KH dibuat untuk mengarahkan user kembali ke dashboard pencari sehingga diperlukan penggunaan intent yang dipanggil dan disimpan dalam variabel i dan getApplicationContext yang merujuk ke activity class Dashboard Pencari. startActivity(i) berfungsi untuk memulai activity yang dituju. Pada akhir java ditulis Faqpencari.this.finish(); yang akan terpanggil ketika kita menekan tombol sehingga activity yang sekarang akan di destroy dan kembali ke activity sebelumnya.

## Coding Java Menu Profile

---

```
package com.projectK1.kosthost;

import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;
import com.android.volley.AuthFailureError;
import com.android.volley.NetworkError;
import com.android.volley.NoConnectionError;
import com.android.volley.ParseError;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.ServerError;
import com.android.volley.TimeoutError;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import com.squareup.picasso.Picasso;
import org.json.JSONException;
import org.json.JSONObject;
import java.util.HashMap;
import java.util.Map;

public class ProfilePencari extends AppCompatActivity {
    private Button btnKembali, btnKeluar, btnEditProfil;
    String URL_PROFILE_PENCARI = "http://192.168.1.7/kosthostmobile/profil-pencari.php";
    String namaPencari, tanggallahir, telepon, email, IDPencari, fotoPencari;
    private TextView txtNamaPencari, txtTanggallahir, txtTelepon, txtEmail, txtStatus;
    private ImageView imagePencari;
    String URL_FOTO_PENCARI;
```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_profile_pencari);
    btnKembali = findViewById(R.id.btnKembali);
    btnKeluar = findViewById(R.id.btnKeluar);
    btnEditProfil = findViewById(R.id.btnEditProfil);
    txtNamaPencari = findViewById(R.id.txtNamaPencari);
    txtTanggallahir = findViewById(R.id.txtTanggallahir);
    txtTelepon = findViewById(R.id.txtTelepon);
    txtEmail = findViewById(R.id.txtEmail);
    txtStatus = findViewById(R.id.txtStatus);
    imagePencari = findViewById(R.id.imagePencari);
    Intent i = getIntent();
    IDPencari = i.getStringExtra("IDPencari");
    profilePencari(IDPencari);

    btnKembali.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent i = new Intent(getApplicationContext(), DashboardPencari.class);
            i.putExtra("IDPencari", IDPencari);
            startActivity(i);
            ProfilePencari.this.finish();
        }
    });
    btnKeluar.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent i = new Intent(getApplicationContext(), MainActivity.class);
            startActivity(i);
        }
    });
    btnEditProfil.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent i = new Intent(getApplicationContext(), EditProfilePencari.class);
            i.putExtra("IDPencari", IDPencari);
            i.putExtra("namaPencari", namaPencari);
            i.putExtra("tanggallahir", tanggallahir);
            i.putExtra("telepon", telepon);
            i.putExtra("email", email);
            i.putExtra("status", txtStatus.getText().toString());
            i.putExtra("fotoPencari", fotoPencari);
            startActivity(i);
        }
    });
}

private void profilePencari(String IDPencari) {
    RequestQueue queue = Volley.newRequestQueue(ProfilePencari.this);
    StringRequest request = new StringRequest(Request.Method.POST, URL_PROFILE_PENCARI,
        new Response.Listener<String>() {
            @Override
            public void onResponse(String response) {
                try {
                    JSONObject jsonObject = new JSONObject(response);
                    int success = jsonObject.getInt("success");
                    if (success == 1) {
                        namaPencari = jsonObject.getString("namaPencari");
                        tanggallahir = jsonObject.getString("tanggallahir");
                        telepon = jsonObject.getString("telepon");
                        email = jsonObject.getString("email");
                        fotoPencari = jsonObject.getString("fotoPencari");
                        URL_FOTO_PENCARI = "http://192.168.1.7/kosthostmobile/" + fotoPencari;
                        Picasso.get().load(URL_FOTO_PENCARI).into(imagePencari);

                        txtNamaPencari.setText(namaPencari);
                        txtTanggallahir.setText(tanggallahir);
                        txtTelepon.setText(telepon);
                        txtEmail.setText(email);
                        txtStatus.setText("Pencari Kost");
                    } else {
                        Toast.makeText(ProfilePencari.this, "Data profil gagal diambil", Toast.LENGTH_SHORT).show();
                    }
                } catch (JSONException e) {
                    e.printStackTrace();
                }
            }
        }
    );
}

```

```

        }
    } catch (JSONException e) {
        Toast.makeText( context: ProfilePencari.this, e.getMessage(), Toast.LENGTH_SHORT).show();
    }

},
new Response.ErrorListener() {
@Override
public void onErrorResponse(VolleyError error) {
    String message = null;
    if (error instanceof NetworkError) {
        message = "Cannot connect to Internet...Please check your connection!";
    } else if (error instanceof ServerError) {
        message = "The server could not be found. Please try again after some time!!";
    } else if (error instanceof AuthFailureError) {
        message = "Cannot connect to Internet...Please check your connection!";
    } else if (error instanceof ParseError) {
        message = "Parsing error! Please try again after some time!!";
    } else if (error instanceof NoConnectionError) {
        message = "Cannot connect to Internet...Please check your connection!";
    } else if (error instanceof TimeoutError) {
        message = "Connection TimeOut! Please check your internet connection.";
    }
    Toast.makeText( context: ProfilePencari.this, message, Toast.LENGTH_SHORT).show();
}
});
}
@Nullable
@Override
protected Map<String, String> getParams() throws AuthFailureError {
    Map<String, String> params = new HashMap<>();
    params.put("IDPencari", IDPencari);
    return params;
}
};
queue.getCache().clear();
queue.add(request);
}
}

```

**Code 80. Code Java Profile Pencari**

Pada coding Java diatas adalah coding yang digunakan untuk tampilan Profile Pencari. Hal pertama yang dilakukan adalah mendeklarasikan Button yang digunakan pada activity yaitu btnKembali, btnKeluar, dan btnEditprofil, Text View yang digunakan pada activity yaitu txtNamaPencari, txtTanggalLahir, txtTelepon, txtEmail, dan txtStatus, serta Image View yang digunakan pada activity yaitu imagePencari. Terdapat URL\_PROFILE\_PENCARI sebagai lokasi file php yang akan digunakan untuk melakukan CRUD dari android ke dalam database dan String URL\_FOTO\_PENCARI yang akan digunakan untuk menampung url foto yang tersimpan pada database. Berbagai String lainnya juga dideklarasikan seperti namaPencari, tanggalLahir, telepon, email, IDPencari, dan fotoPencari. Setelahnya kita membuat fungsi @Override dengan method OnCreate yang berfungsi untuk melakukan inisialisasi pada saat kondisi awal Activity baru diciptakan, sehingga setContentView secara otomatis akan memperlihatkan halaman activity ProfilePencari. Kemudian dilakukan pemanggilan button, text view, dan image view yang digunakan pada activity dari tampilan berdasarkan ID menggunakan method findViewById. Get Intent digunakan untuk mengambil data dari activity sebelumnya yang dikirim. Variabel yang diterima di activity ini adalah IDPencari yang didapatkan dari Dashboard Pencari. Kemudian IDPencari dijadikan parameter pada pemanggilan method profilPencari yaitu method yang digunakan untuk menampilkan informasi akun pencari yang sedang log in dari database. Terdapat 3 button pada activity ini yang akan kita set sehingga ketika di-klik akan merujuk ke aktivitas tertentu menggunakan method setOnClickListener. Button pertama adalah btnKembali yang berfungsi mengarahkan user ke activity Dashboard Pencari sehingga diperlukan penggunaan intent yang dipanggil dan disimpan dalam variabel i dan getApplicationContext serta rujukan activity class selanjutnya yaitu DashboardPencari. Kemudian dilakukan parsing data IDPencari yang sebelumnya diterima oleh activity dan ditampung pada variabel String IDPencari dengan

menggunakan fungsi putExtra. startActivity(i) berfungsi untuk memulai activity yang dituju. Button berikutnya adalah btnKeluar yang digunakan untuk Log Out dari aplikasi dan kembali ke halaman utama aplikasi yaitu MainActivity sehingga diperlukan penggunaan intent yang dipanggil dan disimpan dalam variabel i dan getApplicationContext serta rujukan activity class selanjutnya yaitu MainActivity. startActivity(i) berfungsi untuk memulai activity yang dituju. Button terakhir adalah btnEditProfil yang berfungsi mengarahkan user ke activity Edit Profil Pencari sehingga diperlukan penggunaan intent yang dipanggil dan disimpan dalam variabel i dan getApplicationContext serta rujukan activity class selanjutnya yaitu EditProfilePencari. Kemudian dilakukan parsing data IDPencari, namaPencari, tanggalLahir, telepon, email, status, dan fotoPencari yang terdapat pada activity dengan fungsi putExtra. startActivity(i) berfungsi untuk memulai activity yang dituju.

Method profilePencari yang akan dipanggil merupakan method yang digunakan untuk menampilkan data akun pemilik dengan yang tersimpan pada database tabel AkunPencariKost. Pada method digunakan RequestQueue dan StringRequest yang digunakan untuk menghubungkan coding yang kita buat dengan URL lokasi file php agar dapat terkoneksi dengan database MySQL. OnResponse digunakan sebagai penampung dari hasil proses di dalamnya. Proses didalamnya meliputi try dan catch untuk mendapatkan data variabel yang dikirimkan dari file php, seperti nama pencari, tanggal lahir, telepon, email, dan foto pencari. Lokasi penyimpanan foto pencari yang tersimpan pada database digabungkan dengan ip address dan folder pada htdocs disimpan pada variabel URL\_FOTO\_PENCARI. Selanjutnya, untuk menampilkan foto yang URL nya sudah ditampung pada variabel, digunakan Picasso dengan memanfaatkan fungsi get, kemudian fungsi load dengan menyertakan URL yang ditampung pada URL\_FOTO\_PENCARI untuk memuat gambar, lalu fungsi into untuk memuat gambar yang sudah didapatkan ke dalam image view. Setelah didapatkan dengan membuat suatu JSONObject, data-data tersebut di set ke masing-masing text view yang sudah dideklarasikan. Apabila data gagal diambil, akan muncul pesan error dalam toast yang menyatakan bahwa Data Profil gagal diambil. onErrorResponse bermanfaat untuk memberikan respon tertentu seperti toast akan error yang didapatkan. Sedangkan di bawahnya, getParams() digunakan untuk mengirimkan variabel yang diperlukan dalam file php untuk melakukan query. Dalam file ini diperlukan IDPencari agar query MySQL pada file php dapat berjalan dengan baik yaitu mendapatkan data AkunPencariKost berdasarkan IDPencari yang login. Bagian akhir, harus ditambahkan queue.getCache().clear(); dan queue.add(request); agar program dapat berjalan dan terupdate tampilannya.

# Coding Java Edit Profil Pencari

```
package com.projectK1.kosthost;

import ...

public class EditProfilePencari extends AppCompatActivity {
    private Button btnKembali, btnSimpan, btnDatePicker, btnChoose;
    int year, month, dayOfMonth;
    Calendar calendar;
    Bitmap bitmap;
    private ImageView imagePencari;
    DatePickerDialog datePickerDialog;
    String fotoPencari;
    String URL_FOTO_PENCARI;
    final int CODE_GALLERY_REQUEST = 999;
    String URL_EDITPROFILE_PENCARI = "http://192.168.1.7/kosthostmobile/editprofile-pencari.php";
    private TextView txtNamaPencari, txtTanggallahir, txtTelepon, txtEmail, txtStatus;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_edit_profile_pencari);
        btnKembali = findViewById(R.id.btnKembali);
        btnSimpan = findViewById(R.id.btnSimpan);
        txtNamaPencari = findViewById(R.id.txtNamaPencari);
        txtTanggallahir = findViewById(R.id.txtTanggallahir);
        btnDatePicker = findViewById(R.id.btnTimepicker);
        txtTelepon = findViewById(R.id.txtTelepon);
        txtEmail = findViewById(R.id.txtEmail);
        txtStatus = findViewById(R.id.txtStatus);
        btnChoose = findViewById(R.id.btnChoose);
        imagePencari = findViewById(R.id.imagePencari);

        Intent i = getIntent();
        String IDPencari = i.getStringExtra( name: "IDPencari");
        String namaPencari = i.getStringExtra( name: "namaPencari");
        String tanggallahir = i.getStringExtra( name: "tanggallahir");
        String telepon = i.getStringExtra( name: "telepon");
        String email = i.getStringExtra( name: "email");
        String status = i.getStringExtra( name: "status");
        String fotoPencari = i.getStringExtra( name: "fotoPencari");

        URL_FOTO_PENCARI = "http://192.168.1.7/kosthostmobile/" + fotoPencari;
        Picasso.get().load(URL_FOTO_PENCARI).into(imagePencari);
        txtNamaPencari.setText(namaPencari);
        txtTanggallahir.setText(tanggallahir);
        txtTelepon.setText(telepon);
        txtEmail.setText(email);
        txtStatus.setText(status);

        btnChoose.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                ActivityCompat.requestPermissions(
                    activity: EditProfilePencari.this,
                    new String[]{Manifest.permission.READ_EXTERNAL_STORAGE},
                    CODE_GALLERY_REQUEST
                );
            }
        });

        btnKembali.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new Intent(getApplicationContext(), ProfilePencari.class);
                i.putExtra( name: "IDPencari", IDPencari);
                startActivity(i);
                EditProfilePencari.this.finish();
            }
        });

        btnDatePicker.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                calendar = Calendar.getInstance();
                year = calendar.get(Calendar.YEAR);
                month = calendar.get(Calendar.MONTH);
                dayOfMonth = calendar.get(Calendar.DAY_OF_MONTH);
                datePickerDialog = new DatePickerDialog(this, context: EditProfilePencari.this,
                    new DatePickerDialog.OnDateSetListener() {
                        @Override
                        public void onDateSet(DatePicker datePicker, int year, int month, int day) {
                            |   txtTanggallahir.setText(year + "-" + (month + 1) + "-" + day);
                            |
                        }
                    }, year, month, dayOfMonth);
                datePickerDialog.show();
            }
        });

        btnSimpan.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String namaPencari = txtNamaPencari.getText().toString();
                String tanggallahir = txtTanggallahir.getText().toString();
                String telepon = txtTelepon.getText().toString();
                editProfilePencari(IDPencari, namaPencari, tanggallahir, telepon);
            }
        });
    }
}
```

```

private void editProfilePencari(String IDPencari, String namaPencari, String tanggallahir, String telepon) {
    RequestQueue queue = Volley.newRequestQueue(context);
    StringRequest request = new StringRequest(Request.Method.POST, URL_EDITPROFILE_PENCARI, new Response.ListenerString() {
        @Override
        public void onResponse(String response) {
            try {
                JSONObject jsonObject = new JSONObject(response);
                int success = jsonObject.getInt("name: success");
                if (success == 1) {
                    Toast.makeText(EditProfilePencari.this, "Profil anda berhasil diedit", Toast.LENGTH_SHORT).show();
                    Intent i = new Intent(getApplicationContext(), ProfilPencari.class);
                    i.putExtra("name: IDPencari", IDPencari);
                    startActivity(i);
                    EditProfilePencari.this.finish();
                } else {
                    Toast.makeText(EditProfilePencari.this, "Profil anda gagal diedit", Toast.LENGTH_SHORT).show();
                }
            } catch (JSONException e) {
                Toast.makeText(EditProfilePencari.this, e.getMessage(), Toast.LENGTH_SHORT).show();
            }
        }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            String message = null;
            if (error instanceof NetworkError) {
                message = "Cannot connect to Internet...Please check your connection!";
            } else if (error instanceof ServerError) {
                message = "The server could not be found. Please try again after some time!!";
            } else if (error instanceof AuthFailureError) {
                message = "Cannot connect to Internet...Please check your connection!";
            } else if (error instanceof ParseError) {
                message = "Parsing error! Please try again after some time!!";
            } else if (error instanceof NoConnectionError) {
                message = "Cannot connect to Internet...Please check your connection!";
            } else if (error instanceof TimeoutError) {
                message = "Connection Timeout! Please check your internet connection.";
            }
            Toast.makeText(EditProfilePencari.this, message, Toast.LENGTH_SHORT).show();
        }
    });
    @Nullable
    @Override
    protected Map<String, String> getParams() throws AuthFailureError {
        Map<String, String> params = new HashMap<>();
        fotoPencari = imageToString(bitmap);
        params.put("IDPencari", IDPencari);
        params.put("namaPencari", namaPencari);
        params.put("tanggallahir", tanggallahir);
        params.put("telepon", telepon);
        params.put("fotoPencari", fotoPencari);
        return params;
    }
    queue.getCache().clear();
    queue.add(request);
}
@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
    if (requestCode == CODE_GALLERY_REQUEST && resultCode == RESULT_OK && data != null) {
        Uri filePath = data.getData();
        try {
            InputStream inputStream = getContentResolver().openInputStream(filePath);
            bitmap = BitmapFactory.decodeStream(inputStream);
            imagePencari.setImageBitmap(bitmap);
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        }
    }
}
super.onActivityResult(requestCode, resultCode, data);
}

@Override
public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {
    if (requestCode == CODE_GALLERY_REQUEST) {
        if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
            Intent intent = new Intent(Intent.ACTION_PICK);
            intent.setType("image/*");
            startActivityForResult(Intent.createChooser(intent, title: "Select Image"), CODE_GALLERY_REQUEST);
        } else {
            Toast.makeText(getApplicationContext(), text: "Tidak ada akses ke gallery", Toast.LENGTH_LONG).show();
        }
        return;
    }
}
super.onRequestPermissionsResult(requestCode, permissions, grantResults);
}

private String imageToString(Bitmap bitmap) {
    ByteArrayOutputStream outputStream = new ByteArrayOutputStream();
    bitmap.compress(Bitmap.CompressFormat.JPEG, quality: 100, outputStream);
    byte[] imageBytes = outputStream.toByteArray();

    String encodedImage = Base64.encodeToString(imageBytes, android.util.Base64.DEFAULT);
    return encodedImage;
}
}

```

**Code 81. Code Java Edit Profile Pencari**

Code diatas merupakan code yang digunakan untuk pencari mengedit profile mereka. Hal pertama yang dilakukan adalah mendeklarasikan Button btnKembali, btnSimpan, btnDatePicker, btnChoose dan ImageView imagePemilik. Berbagai TextView lainnya juga diinisialisasikan seperti txtNamaPencari, txtTanggalLahir, txtTelepon, txtEmail, txtStatus. Selain itu juga dilakukan insialisasi terhadap berbagai Int yang digunakan seperti year, month, dayOfMonth. Deklarasikan variabel lain seperti Calendar dan DatePickerDialog, Lalu terdapat Bitmap dengan id bitmap yang berhubungan dengan imageKost. Kemudian final int Code\_Gallery\_Request yang digunakan untuk membaca data external. Terdapat 2 URL, pertama adalah URL\_FOTO\_PENCARI sebagai lokasi file php yang akan digunakan untuk mendapatkan foto pencari dari database. Selanjutnya adalah URL\_EDITPROFILE\_PENCARI sebagai lokasi file php yang akan digunakan untuk melakukan EDIT foto pencari dari android ke dalam database.

Setelahnya kita membuat fungsi @Override dengan method OnCreate yang berfungsi untuk melakukan inisialisasi pada saat kondisi awal Activity baru diciptakan, sehingga setContentView secara otomatis akan memperlihatkan halaman activity Edit Profile Pemilik. Get Intent digunakan untuk mengambil beberapa data dari variabel dari activity sebelumnya yang dikirim. Find view by id digunakan untuk melakukan inisialisasi variabel yang sudah dideklarasikan di awal agar dapat terkoneksi dengan komponen-komponen dalam activity ini. URL\_FOTO\_PENCARI = "http://192.168.1.7/kosthostmobile/editprofile-pencari.php" digunakan untuk membuat suatu URL yang tepat untuk mengambil foto profil pencari dari database. Untuk menetapkan foto tersebut dari URL ke dalam ImageView dapat menggunakan Picasso, yaitu dengan Picasso.get().load (URL\_FOTO\_PENCARI).into(imagePencari); Untuk menggunakan metode Picasso, harus menambahkan implementation 'com.squareup.picasso:picasso:2.71828' pada build.gradle. setOnClickListener pada button digunakan untuk mendeklarasikan aksi yang akan dijalankan setelah tombol ditekan, biasanya digunakan untuk menampilkan activity lain.

RequestQueue dan StringRequest digunakan untuk menghubungkan coding yang kita buat dengan URL lokasi file php agar dapat terkoneksi dengan database MySQL. OnResponse digunakan sebagai penampung dari hasil proses di dalamnya. Proses di dalamnya meliputi try dan catch untuk menjalankan perintah sesuai dengan file php. Setelah didapatkan response dengan membuat suatu JSONObject, nilai dari success = 1 menandakan bahwa proses berjalan dengan lancar sedangkan selain 1 berarti proses tidak berjalan dengan baik dan terdapat error. Jika success = 1, akan muncul toast bahwa kost berhasil di book dan melakukan intent ke activity.

OnErrorResponse digunakan untuk memberikan respon tertentu seperti toast Error. Sedangkan di bawahnya, getParams() digunakan untuk mengirimkan variabel yang diperlukan dalam file php untuk melakukan query.

Dalam file ini diperlukan fotoPencari, IDPencari, namaPencari, tanggalLahir, dan telepon agar query MySQL Bagian akhir, harus ditambahkan queue.getCache().clear(); dan

queue.add(request); agar program dapat berjalan dan terupdate tampilannya. OnRequestPermissionsResult digunakan jika aplikasi perlu menggunakan sumber daya atau informasi di luar kotak pasirnya sendiri, kita dapat mendeklarasikan izin dan menyiapkan permintaan izin yang menyediakan akses ini. Ini adalah bagian dari alur kerja untuk menggunakan izin. OnActivityResult digunakan untuk memulai aktivitas lain, baik dalam aplikasi maupun dari aplikasi lain, tidak harus berupa operasi satu arah. kita juga dapat memulai aktivitas lain dan menerima hasilnya kembali. aplikasi dapat menerima foto yang diambil sebagai hasilnya. imageToString digunakan untuk mengubah file gambar menjadi String.

#### 4.4 Coding Android Manifest

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3      package="com.projectK1.kosthost">
4
5      <uses-permission android:name="android.permission.INTERNET" />
6      <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
7      <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
8
9      <application
10         android:allowBackup="true"
11         android:icon="@mipmap/ic_launcher"
12         android:label="Kost Host"
13         android:networkSecurityConfig="@xml/network_security_config"
14         android:roundIcon="@mipmap/ic_launcher_round"
15         android:supportsRtl="true"
16         android:theme="@style/Theme.KostHost"
17         android:usesCleartextTraffic="true">
```

Code 82. Code Android Manifest

Pada bagian Android Manifest ditambahkan permission yang digunakan pada aplikasi yaitu permission Interent, Access Network State, dan Read External Storage dengan cara yang terlihat pada baris 4 sampai 6. Kemudian pada bagian application ditambahkan usesCleartextTraffic dengan nilai true, networkSecurityConfig dengan nilai @xml/network\_security\_config.

#### 4.5 Coding Build Gradle (Module: Kost\_Host.app)

```
dependencies {  
    implementation 'androidx.appcompat:appcompat:1.2.0'  
    //noinspection GradleCompatible  
    implementation 'com.google.android.material:material:1.1.0'  
    implementation 'androidx.cardview:cardview:1.0.0'  
    implementation 'androidx.constraintlayout:constraintlayout:2.0.4'  
    testImplementation 'junit:junit:4.12'  
    androidTestImplementation 'androidx.test.ext:junit:1.1.2'  
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.3.0'  
    implementation 'com.squareup.picasso:picasso:2.71828'  
    implementation 'com.android.volley:volley:1.2.0'  
}
```

Code 83. Code Build Gradle (Module:Kost\_Host.app)

Pada bagian build gradle, terdapat beberapa tambahan implementation yang digunakan. Hal tersebut digunakan untuk mendukung berbagai metode yang digunakan dalam aplikasi Kost Host. Tambahan implementation tersebut meliputi:

- implementation 'androidx.cardview:cardview:1.0.0'
- Card view digunakan untuk tampilan dashboard, baik dashboard pemilik maupun dashboard pencari menggunakan card view agar dapat memberikan tampilan menu yang rapi dan memberikan *user experience* yang baik dalam penggunaannya yang mudah.
- implementation 'com.squareup.picasso:picasso:2.71828'

Picasso digunakan untuk mengambil gambar dari database dengan URL yang ditentukan dalam coding java ke dalam ImageView yang terdapat di beberapa activity, contohnya adalah dalam menampilkan gambar profil pada menu profil.

- implementation 'com.android.volley:volley:1.2.0'
- Volley digunakan untuk mengirim ataupun menerima data dari dan ke Android Studio/file PHP.

## **BAB V**

### **PERANAN**

Berikut peranan masing-masing anggota dalam project ini.

1. Untuk API database menggunakan file PHP yang digunakan dalam aplikasi, dikerjakan bersama-sama dengan membagi file PHP yang dibuat. Rincian pembagian tugas sebagai berikut:

Alexander Bryan Wiratman	- insert-pemilik.php - insert-pencari.php - deletebooked-pemilik.php - editprofile-pemilik.php - editprofile-pencari.php - minuskamar-pemilik.php - updatetestatus-penyewaan.php - tambahkamar-pemilik.php
Fiona Tanadi	- insert-kost.php - edit-kost.php - delete-kost.php
Juan Richard Harmel	- getbooked-pemilik.php - profil-pemilik.php - profil-pencari.php
Marco Viriyatama Lim	- login-pemilik.php - login-pencari.php - deletebooked-pemilik.php
Monica Cornelia	- getkostbykota-pencari.php - getmykostpemilik.php - insert-penyewaan.php

2. Untuk pembuatan layout atau design aplikasi dikerjakan bersama-sama dengan membagi layout yang dibuat. Rincian pembagian tugas sebagai berikut:

Alexander Bryan Wiratman	- activity_main.xml - activity_login.xml - activity_login_pemilik.xml - activity_login_pencari.xml
--------------------------	---

	<ul style="list-style-type: none"> <li>- activity_signup.xml</li> <li>- activity_signup_signup_pemilik.xml</li> <li>- activity_signup_pencari.xml</li> <li>- activity_detailbooked_pemilik.xml</li> </ul>
Fiona Tanadi	<ul style="list-style-type: none"> <li>- activity_faq_pencari.xml</li> <li>- activity_faq_pemilik.xml</li> <li>- activity_penjaga.xml</li> </ul>
Juan Richard Harmel	<ul style="list-style-type: none"> <li>- activity_profile_pencari.xml</li> <li>- activity_profile_pemilik.xml</li> <li>- activity_edit_kost.xml</li> </ul>
Marco Viriyatama Lim	<ul style="list-style-type: none"> <li>- activity_booked_pencari.xml</li> <li>- activity_booked_pemilik.xml</li> </ul>
Monica Cornelia	<ul style="list-style-type: none"> <li>- activity_search_pencari.xml</li> <li>- activity_dashboard_pemilik.xml</li> <li>- activity_kost_pemilik.xml</li> <li>- list_kost.xml</li> <li>- activity_penjaga.xml</li> <li>- activity_tambah_penjaga.xml</li> <li>- activity_edit_penjaga.xml</li> <li>- dialog_penjaga.xml</li> </ul>

3. Untuk pembuatan coding java dalam aplikasi dikerjakan bersama-sama dengan membagi layout yang dibuat. Rincian pembagian tugas sebagai berikut:

Alexander Bryan Wiratman	<ul style="list-style-type: none"> <li>- MainActivity</li> <li>- Login</li> <li>- LoginPemilik</li> <li>- LoginPencari</li> <li>- Signup</li> <li>- SignupPemilik</li> <li>- SignupPencari</li> <li>- DetailbookedPemilik</li> <li>- EditProfilePemilik</li> </ul>
Fiona Tanadi	<ul style="list-style-type: none"> <li>- EditKost</li> </ul>

	<ul style="list-style-type: none"> <li>- faqPencari</li> <li>- faqPemilik</li> <li>- <u>Penjaga</u></li> </ul>
Juan Richard Harmel	<ul style="list-style-type: none"> <li>- ProfilePencari</li> <li>- Profile Pemilik</li> <li>- SearchPencari</li> </ul>
Marco Viriyatama Lim	<ul style="list-style-type: none"> <li>- BookedPemilik</li> <li>- BookedPencari</li> </ul>
Monica Cornelia	<ul style="list-style-type: none"> <li>- DashboardPemilik</li> <li>- KostPemilik</li> <li>- PenjagaPenjagaKost</li> <li>- MyDBHandler</li> <li>- TambahPenjaga</li> <li>- EditPenjaga</li> <li>- TambahKostPemilik</li> </ul>

4. Untuk pembuatan laporan ini, semua anggota berpartisipasi dengan pembagian tugas yang jelas. Rincian pembagian dalam laporan sebagai berikut.
- Penjelasan Proses Bisnis dikerjakan oleh Monica Cornelia
  - Pembuatan gambar ERD dikerjakan oleh Juan Richard Harmel
  - Penjelasan Layout

Alexander Bryan Wiratman	<ul style="list-style-type: none"> <li>- Layout Pemilik - Booked – Available</li> <li>- Layout Pemilik - Booked – Booked</li> <li>- Layout Pemilik – Penjaga</li> <li>- Layout Pemilik - Penjaga – Tambah</li> <li>- Layout Pemilik - Penjaga - Dialog Edit/Delete</li> </ul>
Fiona Tanadi	<ul style="list-style-type: none"> <li>- Layout Edit Penjaga</li> <li>- Dashboard Pencari</li> <li>- Search Pencari</li> <li>- Detail Kost Pencari</li> <li>- Booked Pencari</li> </ul>
Juan Richard Harmel	<ul style="list-style-type: none"> <li>- FAQ Pemilik</li> <li>- Profil Pemilik</li> <li>- Edit Profil Pemilik</li> </ul>

	<ul style="list-style-type: none"> <li>- FAQ Pencari</li> <li>- Profil Pencari</li> <li>- Edit Profil Pencari</li> </ul>
Marco Viriyatama Lim	<ul style="list-style-type: none"> <li>- Login/SignUp</li> <li>- Login Pemilik</li> <li>- Login Pencari</li> <li>- Sign Up Pemilik</li> <li>- Sign Up Pemilik</li> </ul>
Monica Cornelia	<ul style="list-style-type: none"> <li>- Layout Dashboard Pemilik</li> <li>- Layout My Kost</li> <li>- Layout My Kost - Tambah Kost</li> <li>- Layout My Kost - Edit Kost</li> <li>- Layout Booked</li> </ul>

d. Penjelasan Coding

- Coding PHP

Alexander Bryan Wiratman	<ul style="list-style-type: none"> <li>- connect.php</li> <li>- getbooked-pemilik.php</li> <li>- getbooked-pencari.php</li> <li>- getkostbykota-pencari.php</li> </ul>
Fiona Tanadi	<ul style="list-style-type: none"> <li>- insert-penyewaan.php</li> <li>- login-pemilik.php</li> <li>- login-pencari.php</li> <li>- minuskamar-pemilik.php</li> </ul>
Juan Richard Harmel	<ul style="list-style-type: none"> <li>- profil-pemilik.php</li> <li>- profil-pencari.php</li> <li>- tambahkamar-pemilik.php</li> <li>- updatetestatus-penyewaan.php</li> </ul>
Marco Viriyatama Lim	<ul style="list-style-type: none"> <li>- login-pemilik.php</li> <li>- login-pencari.php</li> <li>- deletebooked-pemilik.php</li> </ul>
Monica Cornelia	<ul style="list-style-type: none"> <li>- getmykost-pemilik.php</li> <li>- insert-kost.php</li> </ul>

	<ul style="list-style-type: none"> <li>- insert-pemilik.php</li> <li>- insert-pencari.php</li> </ul>
--	--

- Coding XML

Alexander Bryan Wiratman	<ul style="list-style-type: none"> <li>- activity_book_kost.xml</li> <li>- activity_booked_pemilik.xml</li> <li>- activity_booked_pencari.xml</li> <li>- activity_dashboard_pemilik.xml</li> <li>- activity_dashboard_pencari.xml</li> <li>- activity_detailbooked_pemilik.xml</li> <li>- activity_edit_kost.xml</li> </ul>
Fiona Tanadi	<ul style="list-style-type: none"> <li>- activity_login.xml</li> <li>- activity_login_pemilik.xml</li> <li>- activity_login_pencari.xml</li> <li>- activity_main.xml</li> <li>- activity_penjaga.xml</li> <li>- activity_profile_pemilik.xml</li> <li>- activity_profile_pencari.xml</li> </ul>
Juan Richard Harmel	<ul style="list-style-type: none"> <li>- dialog_penjaga.xml</li> <li>- list_bookedpemilik.xml</li> <li>- list_bookedpencari.xml</li> <li>- list_kost.xml, toolbar.xml</li> <li>- toolbar_edit.xml</li> </ul>
Marco Viriyatama Lim	<ul style="list-style-type: none"> <li>- activity_booked_pencari.xml</li> <li>- activity_booked_pemilik.xml</li> </ul>
Monica Cornelia	<ul style="list-style-type: none"> <li>- activity_search_pencari.xml</li> <li>- activity_signup.xml</li> <li>- activity_signup_pemilik.xml</li> <li>- activity_signup_pencari.xml</li> <li>- activity_tambah_kost.xml</li> <li>- activity_tambah_penjaga.xml</li> </ul>

- Coding Java

Alexander Bryan Wiratman	<ul style="list-style-type: none"> <li>- BookedPemilik</li> <li>- BookedPencari</li> <li>- BookKost</li> <li>- DashboardPemilik</li> <li>- DashboardPencari</li> </ul>
Fiona Tanadi	<ul style="list-style-type: none"> <li>- faqPemilik</li> <li>- faqPencari</li> <li>- KostPemilik</li> <li>- Login</li> <li>- LoginPemilik</li> <li>- LoginPencari</li> </ul>
Juan Richard Harmel	<ul style="list-style-type: none"> <li>- DetailBookedPemilik</li> <li>- EditKost</li> <li>- EditPenjaga</li> <li>- EditProfilePemilik</li> <li>- EditProfilePencari</li> </ul>
Marco Viriyatama Lim	<ul style="list-style-type: none"> <li>- DetailBookedPemilik</li> <li>- EditKost</li> <li>- EditPenjaga</li> <li>- EditProfilePemilik</li> <li>- EditProfilePencari</li> </ul>
Monica Cornelia	<ul style="list-style-type: none"> <li>- MainActivity</li> <li>- MyDBHandler</li> <li>- Penjaga</li> <li>- PenjagaKost</li> <li>- ProfilePemilik</li> <li>- ProfilePencari</li> </ul>

- Coding Android Manifest dikerjakan oleh Monica Cornelia
- Coding Build Gradle (Module: Kost\_Host.app) dikerjakan oleh Alexander Bryan

5. Untuk pembuatan video demo aplikasi, semua anggota berpartisipasi dengan pembagian tugas yang jelas. Rincian pembagian dalam video sebagai berikut.

Alexander Bryan Wiratman	<ul style="list-style-type: none"> <li>- Dashboard Pemilik Kost</li> <li>- Menu My Kost Pemilik Kost</li> <li>- Menu Booked Pemilik Kost</li> </ul>
Fiona Tanadi	<ul style="list-style-type: none"> <li>- Dashboard Pencari Kost</li> <li>- Menu Search Pencari Kost</li> <li>- Menu Booked Pencari Kost</li> </ul>
Juan Richard Harmel	<ul style="list-style-type: none"> <li>- Menu FAQ Pencari Kost</li> <li>- Menu Profile Pencari Kost</li> </ul>
Marco Viriyatama Lim	<ul style="list-style-type: none"> <li>- Signup Pemilik Kost</li> <li>- Signup Pencari Kost</li> <li>- Login Pemilik Kost</li> <li>- Login Pencari Kost</li> </ul>
Monica Cornelia	<ul style="list-style-type: none"> <li>- Menu Penjaga Pemilik Kost</li> <li>- Menu FAQ Pemilik Kost</li> <li>- Menu Profile Pemilik Kost</li> </ul>