BAB III
PERANCANGAN PERANGKAT LUNAK

Segala rancangan yang dilalui akan dijelaskan secara mendalam pada bab ini. Penjelasan rancangan akan mengikuti siklus pembuatan *web service* yang mengacu pada Gambar 2.1. Rancangan memuat pemetaan kebutuhan mengenai segala kebutuhan untuk memenuhi kebutuhan dari sisi pengembang UTC berbasis *mobile*. Analisis dari apa yang telah dipetakan sebelumnya juga termasuk tahap dari rancangan. Berisi segala bentuk parameter yang ingin digunakan pada pengembangan *web service*. Selanjutnya rancangan berisi desain arsitektur yang tentu dibutuhkan untuk pemahaman lebih lanjut mengenai posisi arsitektur dari apa yang dibangun. Desain juga memuat purwarupa dari tampilan katalog yang akan dikembangkan.

3.1 Pemetaan Kebutuhan

Mengacu pada Gambar 2.2 *website* UTC memiliki fitur masuk atau daftar, beranda, jadwal pelatihan & sertifikasi, konfirmasi, program, dan berita terbaru. Penjelasan lebih lanjut mengenai fitur sebagai berikut :

a. Masuk atau daftar : fitur digunakan oleh calon peserta ketika akan melakukan pendaftaran pelatihan. Fitur masuk dilakukan dengan menggunakan akun Google.
b. Beranda : beranda menampilkan halaman utama *website* dengan fitur yang tersedia dan garis besar mengenai *website*, seperti pelatihan yang tersedia, berita terbaru hingga kontak yang dapat dihubungi.
c. Jadwal pelatihan & sertifikasi : memuat jadwal serta sertifikasi yang tersedia serta pendaftaran pelatihan.
d. Konfirmasi : fitur berkaitan dengan konfirmasi pembayaran terkait biaya pelatihan atau sertifikasi yang didaftar.
e. Berita terbaru : menampilkan berita seputar pelatihan dan sertifikasi terbaru.

Dengan sasaran UTC platform *mobile* adalah peserta dan sesuai latar belakang yang telah dijabarkan, pengembangan UTC *mobile* perlu dilakukan untuk memberi kemudahan serta layanan yang baik dalam melakukan administrasi pelatihan. Dalam hal ini pengembang *web service* mengajukan pertanyaan terbuka kepada pengembang platform *mobile* terkait pemetaan fitur apa saja yang dibutuhkan pada platform *mobile*. Didapatkan hasil bahwa pada

platform *mobile* membutuhkan fitur *login*, pendaftaran pelatihan, trainingku, dan akun. Lebih rinci terkait fitur sebagai berikut :

a. *Login :* pada aplikasi *mobile login* dilakukan dengan menggunakan Google.
b. Pendaftaran pelatihan : menampilkan detail pelatihan dan melakukan pendaftaran.
c. Trainingku : fitur memuat pelatihan yang telah berhasil didaftar oleh peserta.
d. Akun : fitur akun memuat informasi diri akun yang terdaftar.

Semua fitur yang telah dipetakan untuk dibangun pada platform *mobile* membutuhkan *web service* API untuk komunikasi data. Penjelasan kebutuhan *web service* API pada fitur UTC *mobile* sebagai berikut :

a. *Login* : dibutuhkan API untuk mengambil data *user* yang berhasil masuk ke aplikasi.
b. Pendaftaran pelatihan : untuk mendapat *id user*, *id* pelatihan, seluruh pelatihan yang tersedia.
c. Trainingku : menampilkan pelatihan yang telah berhasil didaftar oleh *user*.
d. Akun : untuk mendapat *username* dari basis data *user* yang telah berhasil *login* dan menyimpan data *user* ketika melakukan perubahan informasi pada fitur akun.

3.2 Analisis

Semua data yang dibutuhkan didapat dari pengembang *website* UTC. Dilakukan komunikasi dan pertanyaan terbuka mengenai *web* yang sedang dikembangkan serta basis data yang digunakan. Selanjutnya dilakukan analisis lebih dalam mengenai fitur *web* dan basis data untuk kebutuhan pengembangan API pada fitur *mobile*. Didapat delapan kebutuhan API yang harus dikembangkan untuk memenuhi kebutuhan platform *mobile*. Delapan kebutuhan tersebut dirincikan menjadi tiga fungsi API yaitu *trainings*, *user*, dan *participants*. Tiap API memiliki *method*, parameter masukan, dan keluaran yang beragam berdasarkan kebutuhan pengembang *mobile*. Oleh karena itu, dilakukan analisis *method* apa saja yang dibutuhkan, parameter masukan seperti apa yang cocok untuk tiap *method*, dan *response* seperti apa yang dibutuhkan untuk dikeluarkan. Tiga API tersebut adalah *trainings* yang berfungsi untuk menampilkan daftar pelatihan, *users* yang memiliki fungsi menyimpan informasi data diri, dan *participants* yang berfungsi untuk menyimpan data ketika pengguna berhasil melakukan pendaftaran pelatihan.

3.2.1 API *Trainings*

Web service trainings dibutuhkan pada *mobile* untuk pertukaran data dalam hal menampilkan data dari daftar pelatihan yang ada di *web*. Terdapat dua API secara fungsional

yaitu menampilkan daftar secara keseluruhan dan berdasar id. Dalam melakukan *request* dibutuhkan desain parameter masukan dan juga *response* untuk keluaran. Pada Tabel 3.1 memuat detail hasil analisis mengenai fungsi REST API *trainings* yang akan digunakan pada *web service*.

Tabel 3.1 Kebutuhan API *Trainings*

Kebutuhan	Method	Parameter Masukan URI	Keluaran
Menampilkan data pelatihan yang tersedia secara keseluruhan	GET	/trainings	- id - organization_id - name - description - location - price - fasilitas - materi - infoPembayaran - quota - start_time - finish_time - closed_at - status - created_at - updated_at - clicked
Menampilkan detail informasi pelatihan berdasar *id*	GET	/trainings/{id}	- id - organization_id - name - description - location - price - fasilitas - materi - infoPembayaran - quota - start_time - finish_time - closed_at - status - created_at - updated_at - clicked

3.2.2 API *Users*

Pada fitur *users* dibutuhkan *web service* untuk menampilkan data pengguna berdasar *e-mail*, berdasar *id*, menyimpan data *user* ketika berhasil masuk, dan menyimpan data ketika *user* melakukan perubahan informasi pada fitur akun. Ketika melakukan *request* diperlukan parameter masukan secara khusus sesuai *method* yang digunakan. Begitu pula dengan

response yang dihasilkan akan berbeda-beda. Tabel 3.2 sampai Tabel 3.4 menampilkan analisis mengenai kebutuhan, parameter masukan, hingga keluaran sesuai *method* yang dibutuhkan.

Tabel 3.2 Kebutuhan API *Users*

Kebutuhan	Method	Parameter Masukan URI	Keluaran
Menampilkan data pengguna berdasar *e-mail*	GET	/users/email/{email}	- id - email - google_id - name - avatar - phone - address - institusi - pekerjaan - created_at
Menampilkan data pengguna berdasar *id*	GET	/users/{id}	- id - email - google_id - name - avatar - phone - address - institusi - pekerjaan - created_at

Tabel 3.3 Kebutuhan API *Users*

Kebutuhan	Method	Parameter Masukan		Keluaran
		URI	Key	
Menyimpan data pengguna ketika berhasil masuk	POST	/users	email	- email - google_id - name - avatar - id
			google_id	
			name	
			avatar	

Tabel 3.4 Kebutuhan API *Users*

Kebutuhan	Method	Parameter Masukan		Keluaran
		URI	Key	
Menyimpan data pengguna ketika melakukan perubahan	PUT	/usersupdate/{id}	phone	- id - email - google_id - name
			address	
			institusi	

Kebutuhan	Method	Parameter Masukan		Keluaran
		URI	Key	
informasi pada fitur akun			pekerjaan	- avaztar - phone - address - institusi - pekerjaan - created_at

3.2.3 API *Participants*

API *participants* dibutuhkan pada *mobile* untuk menampilkan data peserta yang mendaftar pelatihan serta menyimpan data ketika peserta mendaftar pelatihan. Terdapat dua *method* yang dikembangkan pada API *participants*, maka ketika melakukan *request* diperlukan parameter masukan sesuai *method* yang digunakan. *Response* yang dihasilkan akan berbeda berdasarkan *request* yang diajukan. Detail kebutuhan, *method*, parameter masukan, dan keluaran terlihat pada Tabel 3.5 dan Tabel 3.6.

Tabel 3.5 Kebutuhan API *Participants*

Kebutuhan	Method	Parameter Masukan URI	Keluaran
Menampilkan data pengguna berdasar *id*	GET	/participants/{user_id}	- id - user_id - training_id - joined_at - is_paid - invoice_id - email - google_id - name - avatar - phone - address - institusi - pekerjaan - created_at - organization_id - description - location - price - fasilitas - materi - infoPembayaran - quota - start_time - finish_time - closed_at

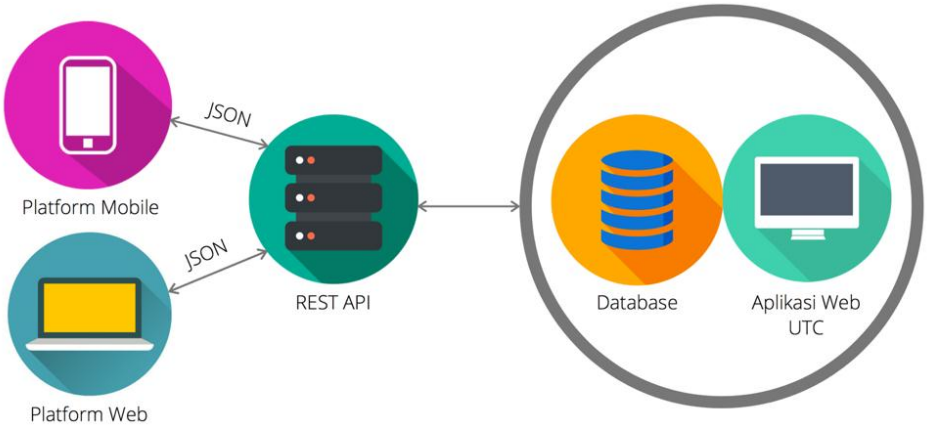
Kebutuhan	Method	Parameter Masukan URI	Keluaran
			- status - updates_at - clicked - filename

Tabel 3.6 Kebutuhan API *Participants*

Kebutuhan	Method	Parameter Masukan		Keluaran
		URI	Key	
Menyimpan data ketika pengguna melakukan pendaftaran pelatihan	POST	/participants	user_id training_id is_paid	- user_id - training_id - is_paid - invoice_id - id

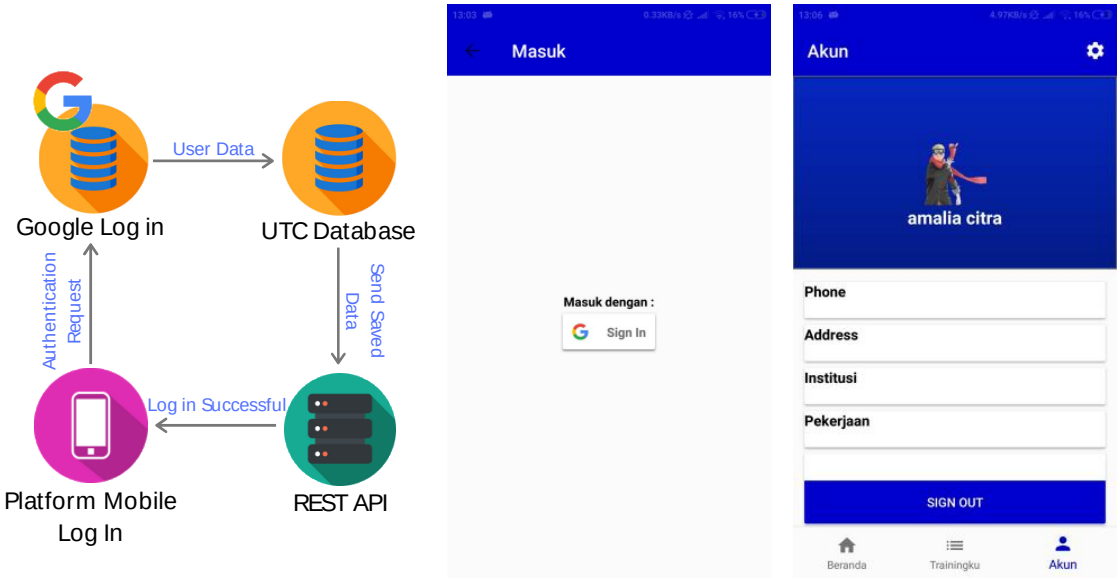
3.3 Desain

Setelah melakukan analisis kebutuhan maka dilakukan desain arsitektur dikembangkan untuk mengetahui lebih lanjut posisi REST API yang akan dikembangkan. Mengacu pada penelitian terdahulu (Pratama, 2017) tampilan dari desain arsitektur yang dikembangkan tampak seperti Gambar 3.1. Posisi *web service* API terletak di antara platform dan aplikasi *web* UTC. *Website* utama UTC berinteraksi dengan basis data lalu ketika terdapat platform lain yang ingin mengambil data dari basis data UTC maka komunikasi dilakukan dengan platform mengirim *request* ke REST API. Selanjutnya REST API akan berinteraksi dengan basis data dan memberikan *response* ke platform. Data yang dipertukarkan dalam komunikasi merupakan teks JSON.



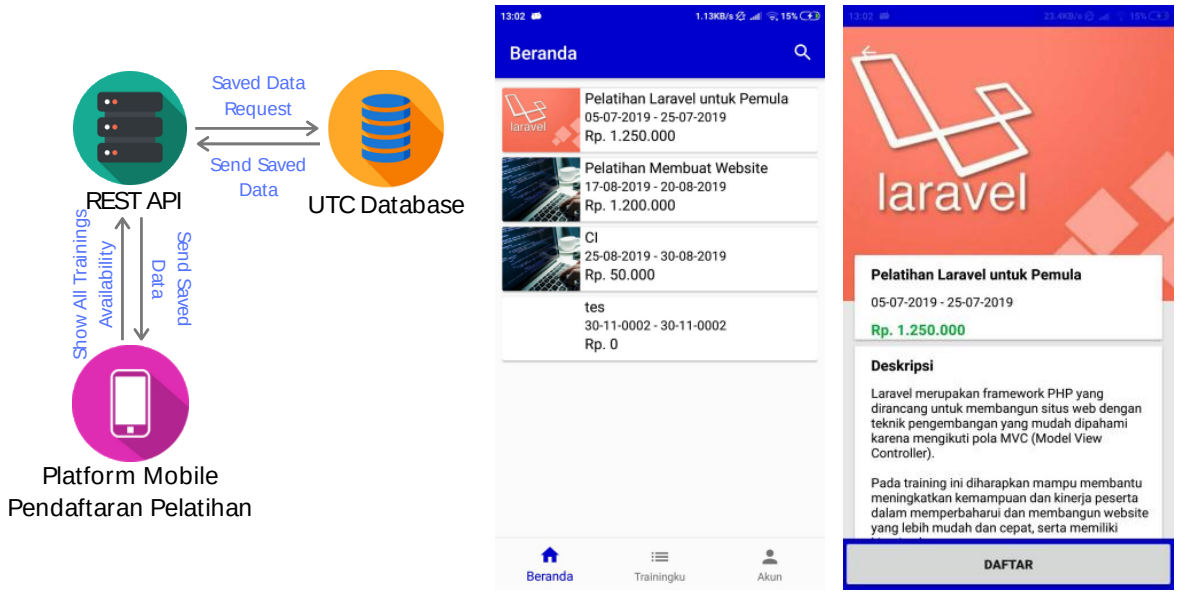
Gambar 3.1 Desain Arsitektur

Pemetaan kebutuhan menunjukkan beberapa kebutuhan *mobile* yang salah satunya adalah *login*. *Login* pada aplikasi digunakan menggunakan akun Google yang datanya akan berinteraksi dengan basis data UTC. Gambar 3.2 Desain Arsitektur dan Antarmuka *Mobile Log In* menunjukkan antarmuka *log in* dari aplikasi *mobile* dan desain arsitektur dari *login* yang terjadi pada aplikasi UTC *mobile*. Ketika UTC *mobile* melakukan *login* maka akan terjadi *request* ke Google sesuai *e-mail* yang terdaftar. Informasi yang diberikan oleh Google berupa *e-mail*, nama, google_id, dan *avatar* yang akan disimpan pada basis data UTC. Kemudian basis data berinteraksi dengan REST API UTC yang selanjutnya dapat digunakan oleh pengembang *mobile*.



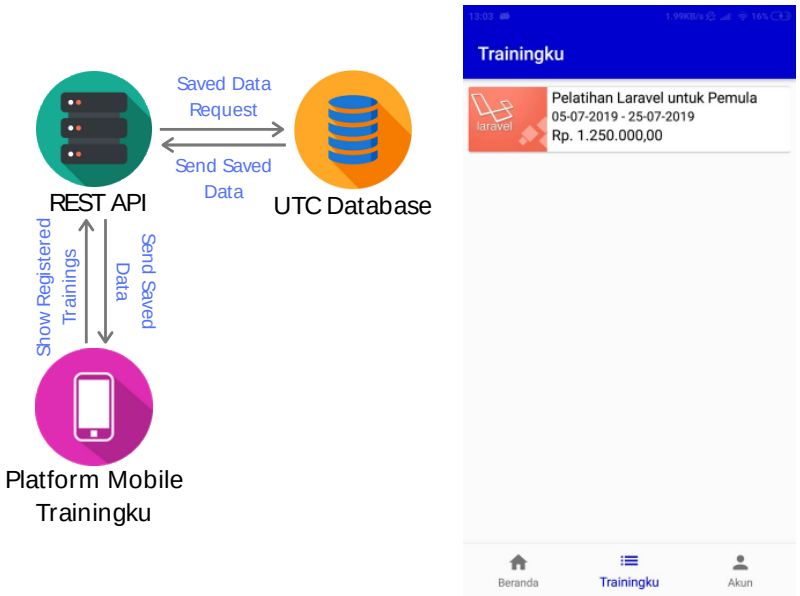
Gambar 3.2 Desain Arsitektur dan Antarmuka *Mobile Log In*

Pada pendaftaran pelatihan akan ditampilkan segala jenis pelatihan yang tersedia dan detail dari pelatihan tersebut. Desain arsitektur pada Gambar 3.3 menampilkan alur bagaimana aplikasi *mobile* memerlukan informasi data pada basis data UTC. Aplikasi platform lain tidak memiliki akses dan ijin untuk berinteraksi dengan basis data, maka dibutuhkan API yang dapat memberikan informasi yang serupa pada aplikasi UTC *web*. Sebuah aplikasi *mobile* menampilkan data dengan cara REST API melakukan *request* ke basis data yang kemudian basis data memberikan *response* berupa informasi terkait. Selain itu gambar juga menampilkan antarmuka pada *mobile*.



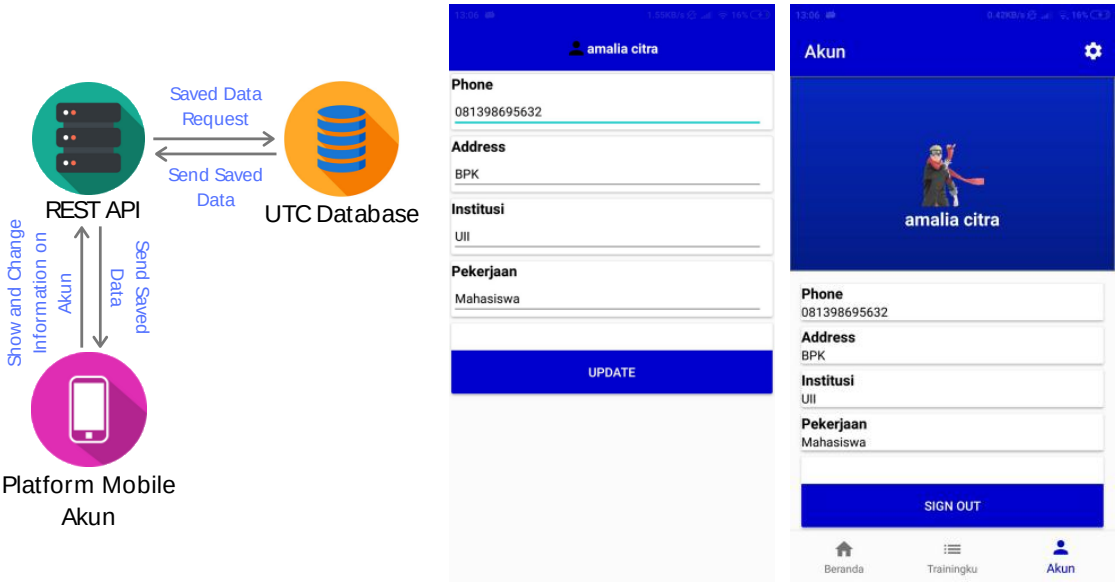
Gambar 3.3 Desain Arsitektur dan Antarmuka *Mobile* Pendaftaran Pelatihan

Fitur trainingku pada *mobile* membutuhkan REST API untuk menampilkan pelatihan yang telah berhasil didaftar oleh peserta seperti pada Gambar 3.4. Gambar juga menampilkan desain arsitektur kebutuhan trainingku. Data pelatihan yang telah berhasil didaftar terdapat pada basis data UTC yang dibutuhkan oleh *mobile*. Oleh karena itu, dibutuhkan REST API dalam proses pemanggilan data pelatihan yang telah berhasil didaftar pada basis data.



Gambar 3.4 Desain Arsitektur dan Antarmuka *Mobile* Trainingku

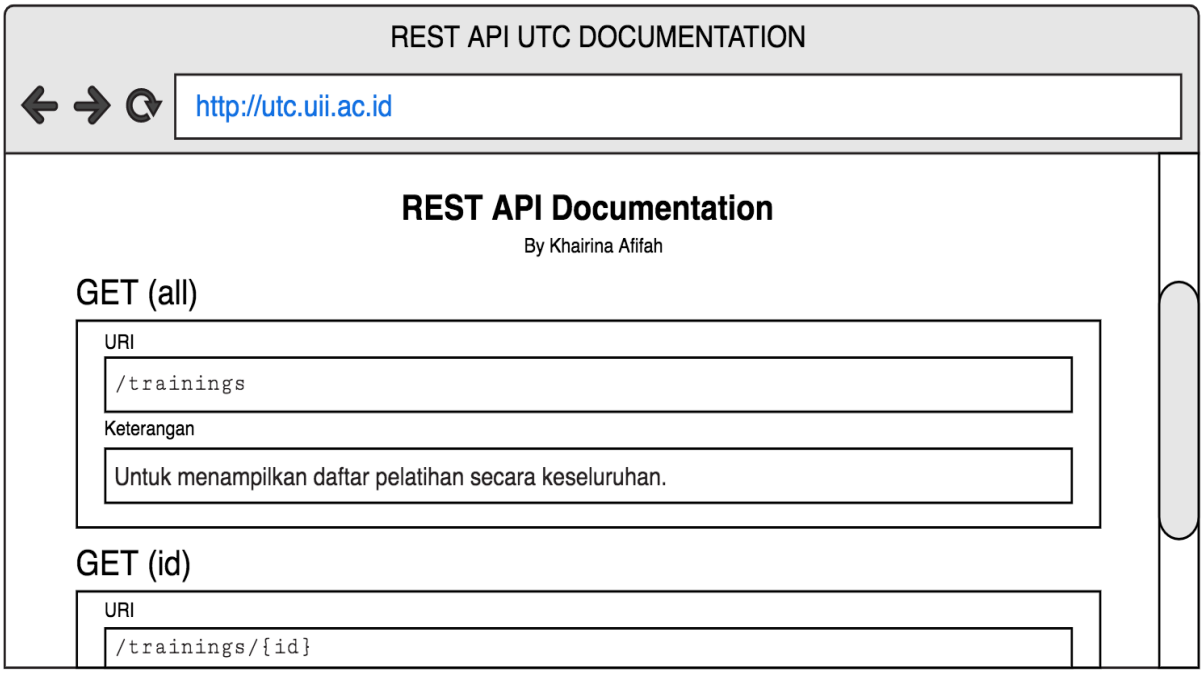
Peserta yang berhasil masuk ke aplikasi akan memiliki informasi data diri pada fitur akun seperti pada Gambar 4.x. Informasi yang terdapat pada antarmuka merupakan data yang sama pada saat melakukan *log in* namun, pengguna dapat menambahkan dan mengubah informasi berupa pekerjaan, institusi, nomor telepon, dan alamat. Gambar 3.5 merupakan antarmuka dari desain arsitektur ketika pengguna masuk ke fitur akun maka REST API dibutuhkan untuk mengirim data yang disediakan oleh Google yang telah tersimpan di basis data UTC pada saat *log in*. Begitu pula ketika pengguna melakukan perubahan pada informasi akun, seluruh perubahan akan disimpan dan dipanggil kembali ketika dibutuhkan di basis data melalui REST API sebagai jembatan komunikasi data.



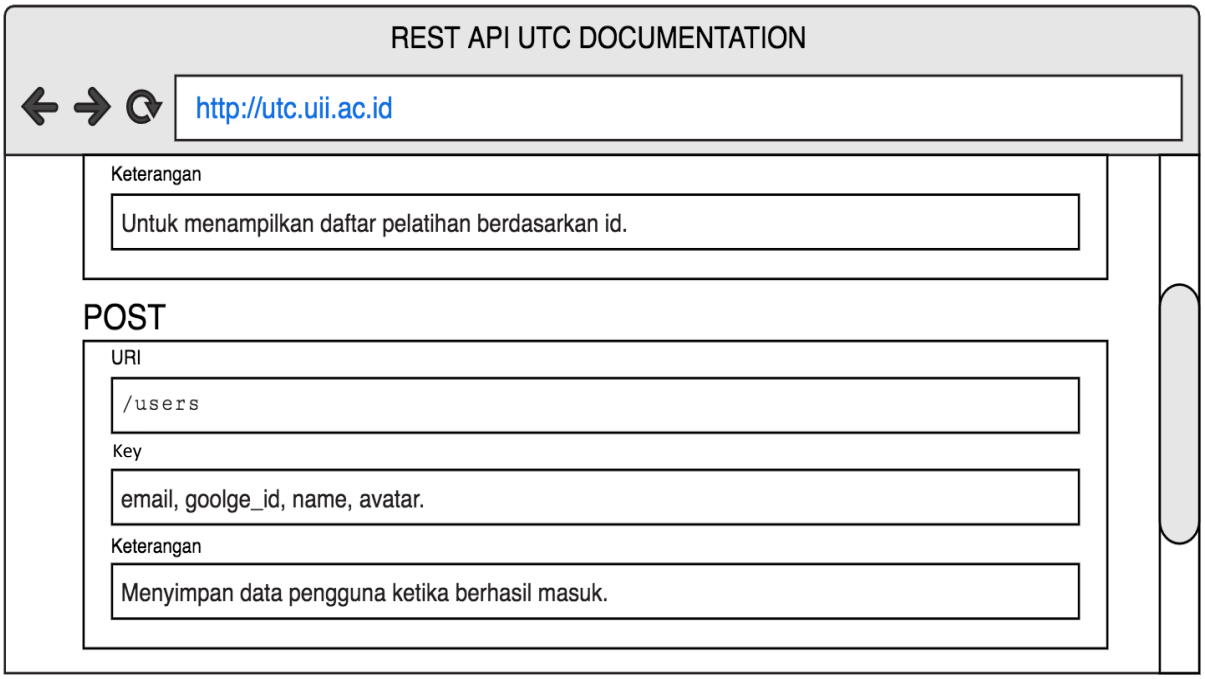
Gambar 3.5 Desain Arsitektur dan Antarmuka *Mobile* Akun

Setelah melakukan analisis kebutuhan, desain parameter masukan dan keluaran, hingga desain arsitektur, langkah selanjutnya dilakukan desain katalog yang akan dikembangkan. Pada tahap ini dilakukan pengembangan purwarupa dari katalog yang ingin dibangun. Tujuan dikembangkan purwarupa sebelum membangun sistem adalah untuk mendapatkan gambaran secara umum tampilan untuk selanjutnya dikembangkan sistem sungguhan yang lebih besar (Purnomo, 2017). Terdapat dua buah rancangan antarmuka yang dikembangkan dengan tujuan agar tidak terlalu banyak pekerjaan pada hal serupa. Berdasarkan rancangan pada tahap analisis kebutuhan sebelumnya, hasil akhir dari pengembangan fungsi akan memiliki lima GET, dua POST, dan satu PUT. Dikembangkan dua purwarupa dari *method* GET dan POST untuk memberi gambaran pada pengembangan sistem mengenai tampilan yang dibutuhkan.

Gambar 3.6 merupakan tampilan antarmuka awal dari dokumentasi REST API UTC. Pada tampilan dokumentasi ini memuat *method*, URI, dan penjelasan singkat mengenai GET *trainings* untuk memudahkan pengembang dalam mengakses data. Tampilan lanjutan purwarupa dapat dilihat pada Gambar 3.7 yang memuat *method* POST, *key*, dan keterangan secara singkat mengenai fungsi POST *users*.



Gambar 3.6 Purwarupa Katalog



Gambar 3.7 Purwarupa Katalog