

A LINK BIO (LINKSET)



**Oleh :
Kelompok 3**

12S20002_Yoga Sihombing
12S20005_Juan Evander Klose Sitompul
12S20007_Rufina Aprilina Sitorus
12S20010_Suprianto Dharma Sitompul
12S20014_Lidia Ginting
12S20021_Sintia Lolita Silaen
12S20030_Lamsihar Siahaan
12S20044_Hanna Dhea Christi Sihombing
12S20047_Ruth Christine Manurung
12S20052_Eka Rohani Gultom

**INSTITUT TEKNOLOGI DEL
FAKULTAS INFORMATIKA DAN TEKNIK
ELEKTRO**

		
INSTITUT TEKNOLOGI DEL FAKULTAS INFORMATIKA DAN TEKNIK ELEKTRO PROGRAM STUDI SARJANA SISTEM INFORMASI		
RENCANA TUGAS MAHASISWA		
MATA KULIAH	Nama	Pemrograman dan Pengujian Aplikasi Web
	Kode	12S3101
	Kredit	4 sks
	Semester	V
	Pengampu	Mario Simaremare (mario@del.ac.id)
TUGAS MAHASISWA	Minggu/sesi	12/2
	Bentuk	Tugas terstruktur (kelompok)
	Waktu	2 x 60 menit terstruktur dan 2 x 60 menit mandiri
	Judul	<i>Designing REST services: interaction</i>
SUBCAPAIAN PEMBELAJARAN MATA KULIAH (SubCPMK)		
SCPMK12	[C6] Mampu mengembangkan dan menggunakan web service berbasis REST.	
DESKRIPSI TUGAS		
Tugas ini adalah bagian dari pengembangan proyek kelompok anda. Pada tugas ini anda dan kelompok anda diminta untuk secara kolaboratif mendesain REST APIs yang akan disediakan pada aplikasi anda. REST APIs yang dirancang harus memperhatikan aturan (<i>rules</i>).		
METODE Pengerjaan Tugas		
Setiap kelompok menyelenggarakan pertemuan secara daring dan mendiskusikan <i>services</i> yang dapat disediakan pada aplikasi. Setiap group member akan bertanggung jawab pada sekumpulan <i>services</i> yang berasal dari suatu <i>collection</i> .		
BENTUK DAN FORMAT LUARAN		
1. Laporan yang berisi draft REST APIs yang disusun dengan memperhatikan aturan-aturan yang berlaku.		
INDIKATOR, KRITERIA DAN BOBOT PENILAIAN		
Untuk kedua jenis luaran akan dinilai menggunakan kriteria dan bobot sebagai berikut:		
1. Laporan ditulis dengan baik dan mampu menjabarkan pekerjaan yang dihasilkan.		
PENGUMPULAN		
Tanggal/ waktu	29 November 2020/ 22.00 WIB	
Tempat	https://ecourse.del.ac.id/mod/assign/view.php?id=18579	
LAIN-LAIN		
<i>Tidak ada</i>		
PUSTAKA		
Utama:		
[1] REST API Design Rulebook		
Pendukung:		

1. Deskripsi Proyek

Adanya kesulitan dalam menemukan berbagai tautan dalam suatu pencarian mengakibatkan turunnya *traffic* pada tautan yang telah dibuat. Untuk itu diperlukan sesuatu yang dapat mengkategorikan setiap link tersebut untuk dapat diakses dan dicari dengan mudah oleh pengguna. Setiap link yang telah terkategori harus dapat dibuat dan dibagikan dengan tahapan yang mudah tanpa merubah apa pun isi dari tautan.

LinkSet adalah sebuah website yang dapat membantu pengguna (user yang telah terdaftar pada database) dalam membuat satu tautan (link) yang mewakili kumpulan berbagai link yang diinginkan. User dapat mendaftarkan berbagai kumpulan link dengan memberi judul pada setiap link yang akan didaftarkan. Setiap kumpulan link tersebut dapat dibagikan melalui berbagai media sosial dengan melakukan copy pada link akun sebagai perwakilan dari link yang sudah didaftarkan sebelumnya.

Requirement

LinkSet dapat membuat daftar link ke dalam satu tautan. LinkSet dirancang agar dapat digunakan hanya oleh pengguna yang telah melakukan registrasi (pendaftaran). Setelah user berhasil melakukan registrasi dan terautentifikasi dengan login maka, user dapat menggunakan fitur yang ada di website LinkSet. Adapun fitur-fitur tersebut terkategori menjadi dua requirement, yaitu:

Functional

- Create new link. Fitur ini dapat mendaftarkan berbagai link dengan menginput "Link Name" dan "Link Url"
- Edit. Fitur ini digunakan untuk melakukan perubahan pada link yang sudah terdaftar sebelumnya. Adapun perubahan yang dapat dilakukan adalah deleted link untuk menghapus semua informasi yang telah terdaftar sebelumnya. Selanjutnya Update link dengan mengubah Link Url atau Link Name yang sudah terdaftar sebelumnya.
- Copy your links. Fitur ini bertujuan untuk dapat menyalin link akun yang dimiliki pengguna agar dapat dibagikan
- Register. Fitur ini memiliki fungsi sebagai autentifikasi sebuah akun. Sehingga website dibatasi hanya untuk pengguna yang telah melakukan pendaftaran sebelumnya.

Non-functional

- Memiliki User Interface yang mudah dipahami

2. Business Process Modeling Notation (BPMN) LinkSet

Pada bisnis proses ini, ketika user ingin menggunakan LinkSet, user harus mendaftarkan akun (sign up) terlebih dahulu. Untuk mendaftarkan akun (sign up), user akan mengisi atau melengkapi username, email, kata sandi (password), dan konfirmasi kata sandi (confirm password). Setelah user berhasil mendaftarkan akun LinkSet, user dapat melakukan login ke website LinkSet, dimana mereka akan mengisi atau memasukkan kembali alamat email dan kata sandi (password) yang telah didaftarkan ketika melakukan sign up sebelumnya. Akan tetapi, jika user telah memiliki akun LinkSet sebelumnya, user tidak perlu lagi untuk melakukan sign up, melainkan user dapat login langsung ke website Linkset untuk mengaplikasikan website tersebut.

Ketika user berhasil melakukan login dan masuk ke website LinkSet, user akan melihat tampilan awal (homepage) LinkSet yang berisi salah satu dari fiturnya yaitu "Create new link", dimana kegunaan dari fitur ini adalah untuk memasukkan atau pun menambahkan link (url) ke dalam suatu kumpulan link yang telah dibuat oleh user. Pada fitur ini, user akan mengisi atau memasukkan name dan url (link). Setelah user berhasil memasukkan kedua komponen tersebut, user juga dapat melihat total visits (jumlah orang yang mengunjungi link tersebut), last visit (waktu dan tanggal terakhir link diakses), dan action. Pada action, user dapat melakukan edit terhadap nama dan link yang telah

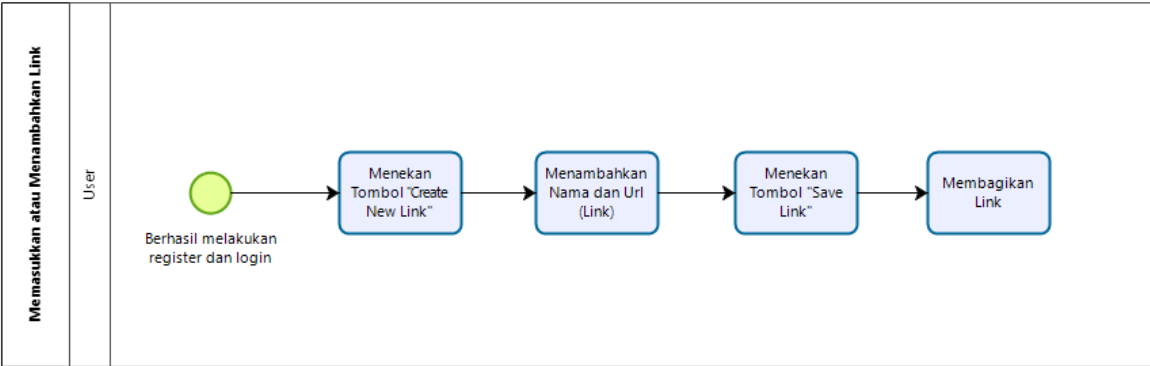
ditambahkan atau dimasukkan sebelumnya. Kemudian, setelah user berhasil menambahkan atau memasukkan nama dan link, user dapat menekan tombol "save link" dan akan terupload. Setelah itu, user dapat membagikan link tersebut. Selain menambahkan atau memasukkan link, user juga dapat menghapus link yang telah dibuat sebelumnya, yaitu dengan menekan tombol "edit", kemudian user menekan tombol "delete link".

Kemudian, user dapat masuk ke fitur selanjutnya, yaitu settings. Pada fitur ini, user dapat mengatur background color, text color, username, dan password. Pertama, user dapat mengatur background color dan text color untuk tampilan tempat link yang telah dibuat. User dapat memilih warna sesuai keinginan mereka. Setelah user selesai memilih warna tersebut, user dapat menekan tombol "save setting". Kedua, user dapat mengatur kembali username yang telah dibuat sebelumnya. Setelah user selesai mengatur username tersebut, user dapat menekan tombol "save profile". Ketiga, user dapat mengatur ulang kembali atau mengganti kata sandi mereka. Setelah selesai mengganti kata sandi tersebut, user juga harus mengisi kolom "confirm password" atau konfirmasi kata sandi. Setelah selesai mengisi kolom tersebut, user dapat menekan tombol "save password".

Selanjutnya, pada website ini ada terdapat fitur "search username" atau pencarian username. Kegunaan dari fitur ini adalah untuk memudahkan pengguna (user) dalam mencari link yang telah dibuat pada username yang telah didaftar sebelumnya. Misalnya, seorang pengguna memiliki tiga akun LinkSet dan user telah menambahkan atau memasukkan link di ketiga akun tersebut. Dengan adanya fitur ini, akan membantu user dengan mudah dalam mencari link yang telah dimasukkan tersebut dari akun-akun yang berbeda dengan memasukkan username pengguna (user) yang sudah didaftarkan ke dalam kolom pencarian. Setelah user memasukkan username tersebut, user dapat menekan tombol enter. Kemudian, user akan mendapatkan tampilan berupa tempat kumpulan link yang sudah ada pada username tersebut.

Apabila user telah selesai menggunakan website LinkSet, user dapat keluar dari akun Linkset nya dengan menekan tombol logout, maka user akan otomatis keluar dari akun LinkSet tersebut. Adapun BPMN dari LinkSet adalah sebagai berikut:

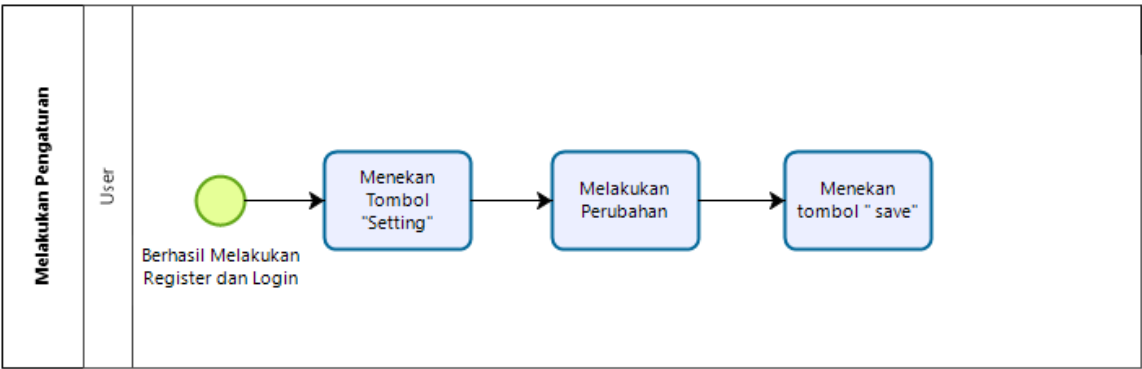
- BPMN Memasukan atau Menambahkan Link



Gambar 1 BPMN Memasukan atau Menambahkan Link

Pada BPMN ini setiap user yang telah berhasil melakukan register dan login dapat menekan tombol "Create New Link", kemudian user dapat menambahkan Nama dan URL (Link) yang sesuai, yang akan dimasukkan ke dalam daftar link. Selanjutnya, user dapat menekan button "Save Link" untuk menyimpan nama dan link yang akan didaftarkan. Terakhir. User dapat membagikan link akun sebagai perwakilan link yang telah didaftarkan sebelumnya.

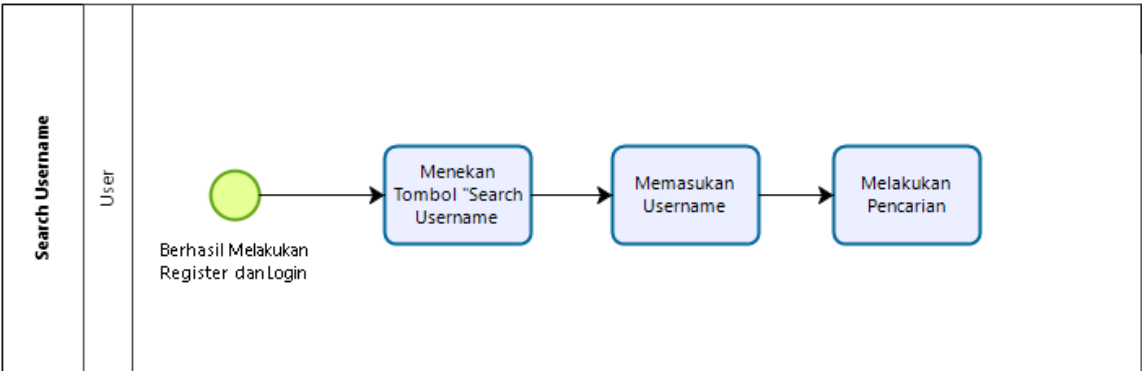
- BPMN Melakukan Pengaturan



Gambar 2 BPMN Melakukan Pengaturan

Pada BPMN ini setiap user yang telah berhasil melakukan register dan login dapat menekan tombol “setting” untuk dapat memulai pengaturan pada tampilan dan data yang tersimpan pada website. Setelah tombol setting ditekan maka user dapt melakukan perubahan dan dilanjut menekan button save untuk menyimpan perubahan.

- BPMN Search Username

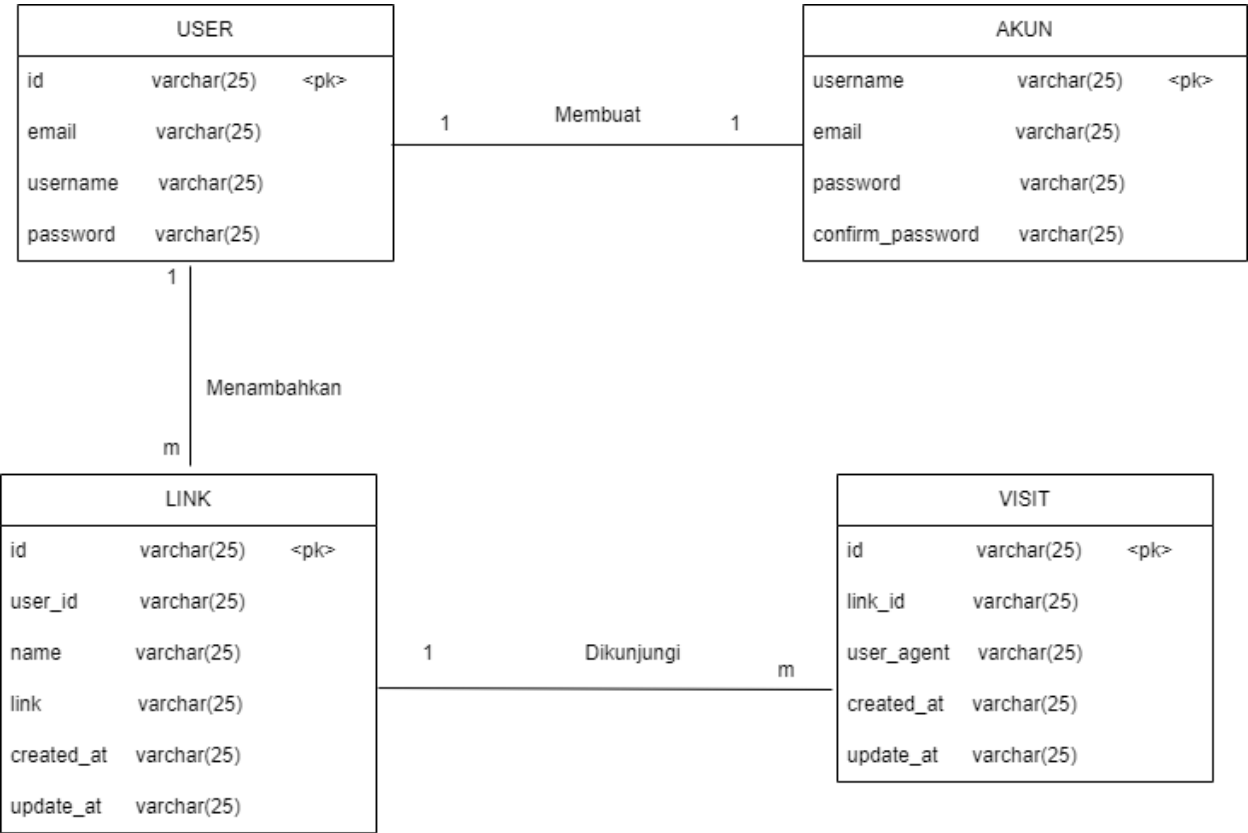


Gambar 3 BPMN Search Username

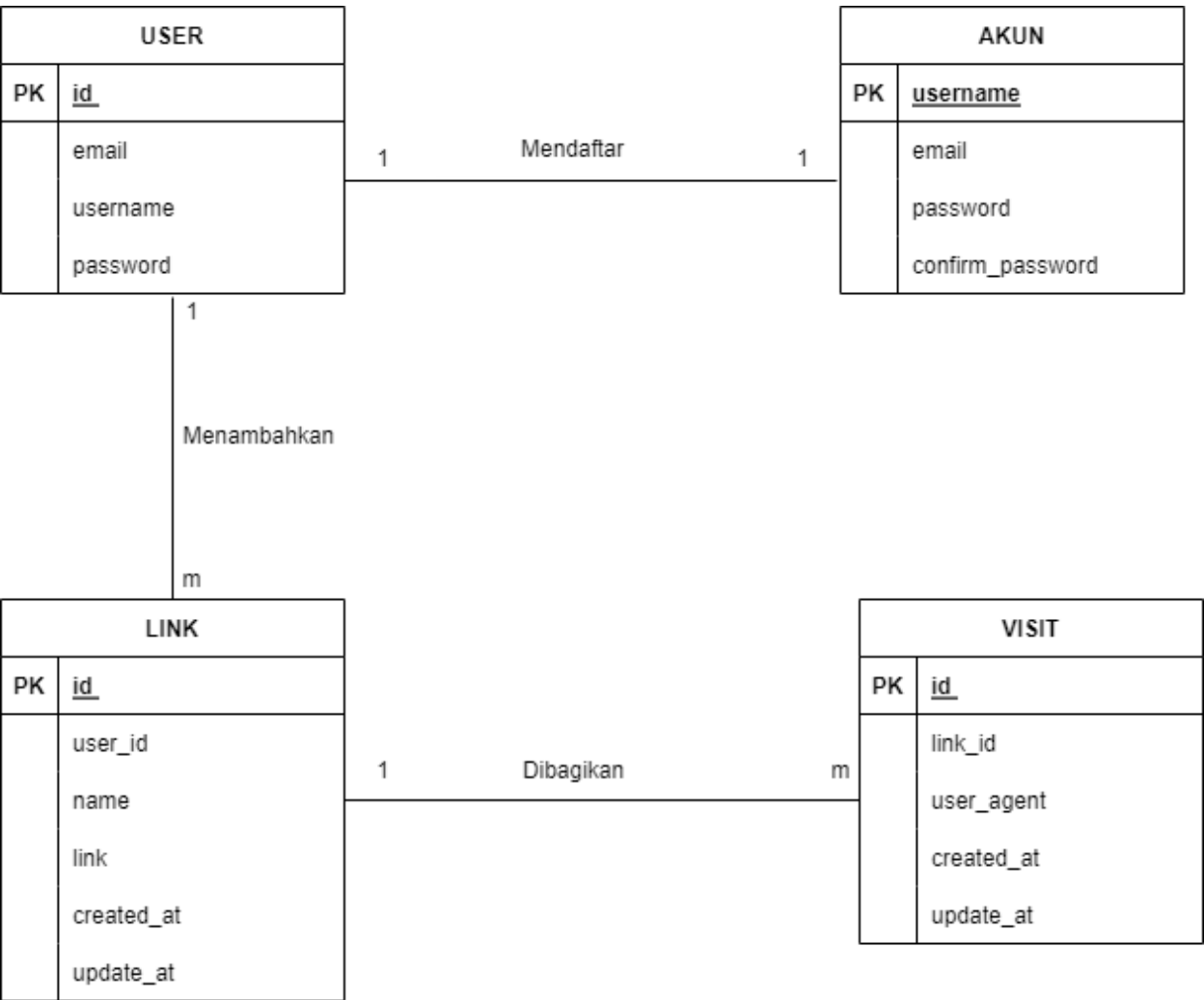
Pada BPMN ini setiap user yang telah berhasil melakukan register dan login dapat menekan tombol “Search Username” untuk mencari username yang telah terdaftar sebelumnya. Setelah menekan tombol “Search Username” selanjutnya user dapat memasukkan username yang akan dicari. Selanjutnya user dapat melihat hasil pencarian username

3. Conceptual Data Model (CDM) and Physical Data Model (PDM) LinkSet

CDM



PDM



4. API Design LinkSet

FR01 : Created

General Information	
Collection	: /users/{user-id}/links/{links-id}/created
Business process	: [FR01] Created
Tujuan	: Insert new link
Responsible Member	: Lidiaginting (@Lidiaginting)
Applied Rules	: #1: Forward slash “/” #2: No trailing forward slash”/” #5: Use lowercase
Request	
HTTP Verb	: POST
Authenticated?	: Yes, with token (<i>anonymous access is not allowed</i>)
User group	: Users
Has payload data	: Yes (JSON)
Data structure	: { “user_id” : “1” “name” : “ecourse” “link” : “https://ecourse.del.ac.id/my/” }
Applied Rules	: #19: Using POST to create a new links
Possible Response	
HTTP Status Code	: 201 (Created)
Has payload data	: Yes (JSON)
Data structure	: { “message”: { “text”: “Link created successfully” “type”: “CREATED” }, “data”: { “id” : “1” “user_id” : “1” “name” : “ecourse” “link” : “https://ecourse.del.ac.id/my/” “create_at”: “2022-11-04 11:03:32” “updated_at” : “2022-11-04 11:03:32” }, “links”: { “/users/{user-id}/links/created” } }
	: #26: Using 201 (create) for successfully insertion
Possible Response	
HTTP Status Code	: 400 (Bad Request)
Has payload data	: Yes (JSON)
Data structure	: { “message”: { “text”: “validation.min.string”, “validation.required” “type”: “Bad Request” }, “links”: { “/users/{user-id}” } }
Applied Rules	: #34: 400 (Bad Request) indicates a general failure that is caused by the

		user.
Possible Response	:	Response 3
HTTP Status Code	:	400 (Bad Request)
Has payload data	:	Yes (JSON)
Data structure	:	{ “message”: { “text”: “validation.url” “type”: “Bad Request” }, “links”: { “/users/{user-id}” } }
Applied Rules	:	#34: 400 (Bad Request) indicates a general failure that is caused by the user.
Possible Response	:	Response 4
HTTP Status Code	:	400 (Bad Request)
Has payload data	:	Yes (JSON)
Data structure	:	{ “message”: { “text”: “validation.required”, “validation.required” “type”: “Bad Request” }, “links”: { “/users/{user-id}” } }
Applied Rules	:	#34: 400 (Bad Request) indicates a general failure that is caused by the user.

FR02: Deleted

General Information		
Collection	:	/users/{user-id}/links/{links-id}/deleted
Business process	:	[FR02] Deleted
Tujuan	:	Delete link
Responsible Member	:	Rufinasitorus(@Rufinasitorus)
Applied Rules	:	#1: Forward slash “/” #2: No trailing forward slash”/” #5: Use lowercase
Request		
HTTP Verb	:	DELETE
Authenticated?	:	Yes, with token (<i>anonymous access is not allowed</i>)
User group	:	Users
Has payload data	:	Yes (JSON)
Data structure	:	{ “id” : “1” “name” : “ecourse” “link” : “https://ecourse.del.ac.id/my/” }
Applied Rules	:	#23: Using DELETE to delete link
Possible Response	:	Response 1
HTTP Status Code	:	200 (OK)
Has payload data	:	Yes (JSON)

Data structure	:	{ “message”: { “text”: “Deleted successfully with id 1” “type”: “OK” }, “links”: { “/users/{user-id}/links/{links-id}/deleted” } }
Applied Rules	:	#26: Using 200 (Ok) for successfully delete

FR03: Update Link

General Information		
Collection	:	/users/{user-id}/links/{links-id}/update-links
Business process	:	[FR03] Update links
Tujuan	:	Doing link updates and inserting new links
Responsible Member	:	LamsiharSiahaan(@LamsiharSiahaan)
Applied Rules	:	#1: Forward slash “/” #2: No trailing forward slash”/” #5: Use lowercase
Request		
HTTP Verb	:	PUT
Authenticated?	:	Yes, with token (<i>anonymous access is not allowed</i>)
User group	:	Users
Has payload data	:	Yes (JSON)
Data structure	:	{ “link_name” : “myecourse” “link_url” : “https://ecourse.del.ac.id/my/” }
Applied Rules	:	#19: Using PUT to update link
Possible Response	:	Response 1
HTTP Status Code	:	200 (OK)
Has payload data	:	Yes (JSON)
Data structure	:	{ “message”: { “type”: “OK” }, “data”: { “id” : “2” “name” : “myecourse” “link” : “https://ecourse.del.ac.id/my/” “create_at”: “2022-11-04 11:03:32” “updated_at” : “2022-11-04 11:03:32” }, “links”: { “/users/{user-id}/links/{links-id}/update-links” } }
Applied Rules	:	#26: Using 201 (create) for successfully updated
Possible Response	:	Response 2
HTTP Status Code	:	400 (Bad Request)
Has payload data	:	Yes (JSON)
Data structure	:	{ “message”: { “type”: “OK” }, “data”: { “id” : “2” “name” : “myecourse” “link” : “https://ecourse.del.ac.id/my/” “create_at”: “2022-11-04 11:03:32” “updated_at” : “2022-11-04 11:03:32” }, “links”: { “/users/{user-id}/links/{links-id}/update-links” } }

		<pre>“text”: “validation.min.string”, “validation.required” “type”: “Bad Request” }, “links”: { “/users/{user-id}” } }</pre>
Applied Rules	:	#34: 400 (Bad Request) indicates a general failure that is caused by the user.
Possible Response	:	Response 3
HTTP Status Code	:	400 (Bad Request)
Has payload data	:	Yes (JSON)
Data structure	:	<pre>{ “message”: { “text”: “validation.url” “type”: “Bad Request” }, “links”: { “/users/{user-id}” } }</pre>
Applied Rules	:	#34: 400 (Bad Request) indicates a general failure that is caused by the user.
Possible Response	:	Response 4
HTTP Status Code	:	400 (Bad Request)
Has payload data	:	Yes (JSON)
Data structure	:	<pre>{ “message”: { “text”: “validation.required”, “validation.required” “type”: “Bad Request” }, “links”: { “/users/{user-id}” } }</pre>
Applied Rules	:	#34: 400 (Bad Request) indicates a general failure that is caused by the user.

FR04: Read

Collection	:	/users/{user-id}/links/{links-id}/read
Business process	:	[FR04] Read
Tujuan	:	View the list of linksets that have been created
Responsible Member	:	juanevander (@juanevanderks)
Applied Rules	:	#1: Forward slash “/” #2: No trailing forward slash”/” #5: Use lowercase
Request		
HTTP Verb	:	GET

Authenticated?	:	No, without token
User group	:	Users
Has payload data	:	Yes (JSON)
Data structure	:	{ "name" : "myecourse" "link" : "https://ecourse.del.ac.id/my/" }
Applied Rules	:	#19: Using GET to see links that has been create before
Possible Response	:	Response 1
HTTP Status Code	:	200 (OK)
Has payload data	:	Yes (JSON)
Data structure	:	{ "message": { "type": "OK" }, "data": { "id" : "1" "user_id" : "2" "name" : "myecourse" "link" : "https://ecourse.del.ac.id/my/" "create_at": "2022-11-04 11:03:32" "updated_at" : "2022-11-04 11:03:32" } "links": { "users/{user-id}/links/{links-id}/read" } }
Applied Rules	:	#26: Using 200 (ok) for successfully read
Possible Response	:	Response 2
HTTP Status Code	:	404 (Not Found)
Has payload data	:	Yes (JSON)
Data structure	:	{ "message": { "type": "Not Found" } }
Applied Rules	:	#34: 404 (Not Found) the page are trying to open on the website cannot be found on the server.

FR05: Autentifikasi

General Information		
Collection	:	/ users/{user-id}
Business process	:	[FR05] Autentifikasi
Tujuan	:	Insert new account
Responsible Member	:	Lidiaginting(@Lidiaginting)
Applied Rules	:	#1: Forward slash “/” #2: No trailing forward slash”/” #5: Use lowercase
Request		
HTTP Verb	:	POST
Authenticated?	:	No, not with token
User group	:	Everyone
Has payload data	:	Yes (JSON)
Data structure	:	{ “username” : “lidia” “email” : “ lidia25@gmail.com ” “Password”: “lidiagin” “Confirm Password”: “lidiagin” }
Applied Rules	:	#19: Using POST to create a new account
Possible Response		
HTTP Status Code	:	201 (Created)
Has payload data	:	Yes (JSON)
Data structure	:	{ “message”: { “text”: “Link created successfully” “type”: “CREATED” }, “data”: { “id” : “1” “username” : “lidia” “email” : “ lidia25@gmail.com ” “create_at”: “2022-11-04 11:03:32” “updated_at” : “2022-11-04 11:03:32” }, “links”: { / users/{user-id} } }
	:	#26: Using 201 (create) for successfully insertion
Possible Response		
HTTP Status Code	:	401 (unauthorized)
Has payload data	:	Yes (JSON)
Data structure	:	{ “message”: { “text”: “validation.unique” “type”: “Bad Request” }, “links”: { “/users” } }
Applied Rules	:	#35 : Using 401 (Unauthorized) for not having valid authentication credentials for the requested resource

5. Test Plan LinkSet

NFR01 – Tes API Create link user

Test ID	NFR 01-01-API			
FR/NFR ID	NFR01 – Security: authentication			
Test Name	Create Link			
Objective	This test is intended to test whether this feature can function to create links.			
Description	This mechanism is for creating links from users.			
Precondition	The user already has a successfully logged in.			
Date	30 Desember 2022			
Tester	Sintia Silaen			
Testing Scenario				
1. Set the target URL (/api/CreatedLink).				
2. Set the request method to POST.				
3. Set a JSON-formatted data with the given configuration and send it.				
Evaluation Criteria				
1. The entered data is matched with the stored data.				
Test Cases				
ID	Input	Expected Behavior	Actual Behavior	Verdict
NFR 01-01-API-01	All data required when creating a new links is filled in. name : youtube link : https://www.youtube.com/	A 201-based response with message “Link Created Successfully”	A 201-based response with message “Link Created Successfully”	[x] passed [] failed
NFR 01-01-API-02	Not all data required when making new links is filled in. name : r link :	A 400- based response with message “validation.min.string”, and “validation.required”	A 400- based response with message “validation.min.string”, and “validation.required”	[x] passed [] failed
NFR 01-01-API-03	Not all data required when making new links is filled in. name : youtube link : youtube	A 400- based response with message “validation.url”	A 400- based response with message “validation.url”	[x] passed [] failed

NFR 01-01-API-04	Not all data required when making new links is filled in. name : link :	A 400- based response with message “validation.required”, and “validation.required”	A 400- based response with message “validation.required”, and “validation.required”	<input checked="" type="checkbox"/> passed <input type="checkbox"/> failed
Notes				
Fourth check it and the cases pass				

NFR02– Tes API Delete

Test ID	NFR 02-01-API			
FR/NFR ID	NFR02 – Security: authentication			
Test Name	Delete Link			
Objective	This test is intended to test whether the data link that has been created can be deleted			
Description	This mechanism is to remove link data from users after accessing the website.			
Precondition	The user already has an account and has successfully logged in.			
Date	30 December 2022			
Tester	Ruth Manurung			
Testing Scenario				
1. Set the target URL (/api/DeleteLink).				
2. Set the request method to GET.				
Evaluation Criteria				
1. The entered data are matched with the stored data.				
Test Cases				
ID	Input	Expected Behavior	Actual Behavior	Verdict
NFR 02-01-API-01	The URL to review is : name : “ecourse” link: : https://ecourse.del.ac.id/my/	A 200-based response display id links.	A 200-based response display id links.	[x] passed [] failed
Notes				
checked it and the cases pass.				

NFR03– Tes API Update Link

Test ID	NFR 03-01-API			
FR/NFR ID	NFR03 – Security: authentication			
Test Name	Update Link			
Objective	This test is intended to test whether data link updates can be performed on this feature.			
Description	This mechanism is for updating link data from users after link creation.			
Precondition	The user already has an account and has successfully logged in.			
Date	30 December 2022			
Tester	Hanna Dhea Sihombing			
Testing Scenario				
1. Set the target URL (/api//UpdateLink). 2. Set the request method to GET.				
Evaluation Criteria				
1. The entered data are matched with the stored data.				
Test Cases				
ID	Input	Expected Behavior	Actual Behavior	Verdict
NFR 03-01-API-01	All data required when update a new links is filled in. name : youtube link : https://www.youtube.com/	A 200-based response with message “Link Created Successfully”	A 200-based response with message “Link Created Successfully”	[x] passed [] failed
NFR 03-01-API-02	Not all data required when making update links is filled in. name : r link :	A 400- based response with message “validation.min.string”, and “validation.required”	A 400- based response with message “validation.min.string”, and “validation.required”	[x] passed [] failed
NFR 03-01-API-03	Not all data required when making update links is filled in. name : youtube link : youtube	A 400- based response with message “validation.url”	A 400- based response with message “validation.url”	[x] passed [] failed
NFR 03-01-API-04	Not all data required when making update links is filled in. name : link :	A 400- based response with message “validation.required”, and “validation.required”	A 400- based response with message “validation.required”, and “validation.required”	[x] passed [] failed

Notes
Fourth checked it and the cases pass.

NFR04– Tes API Read

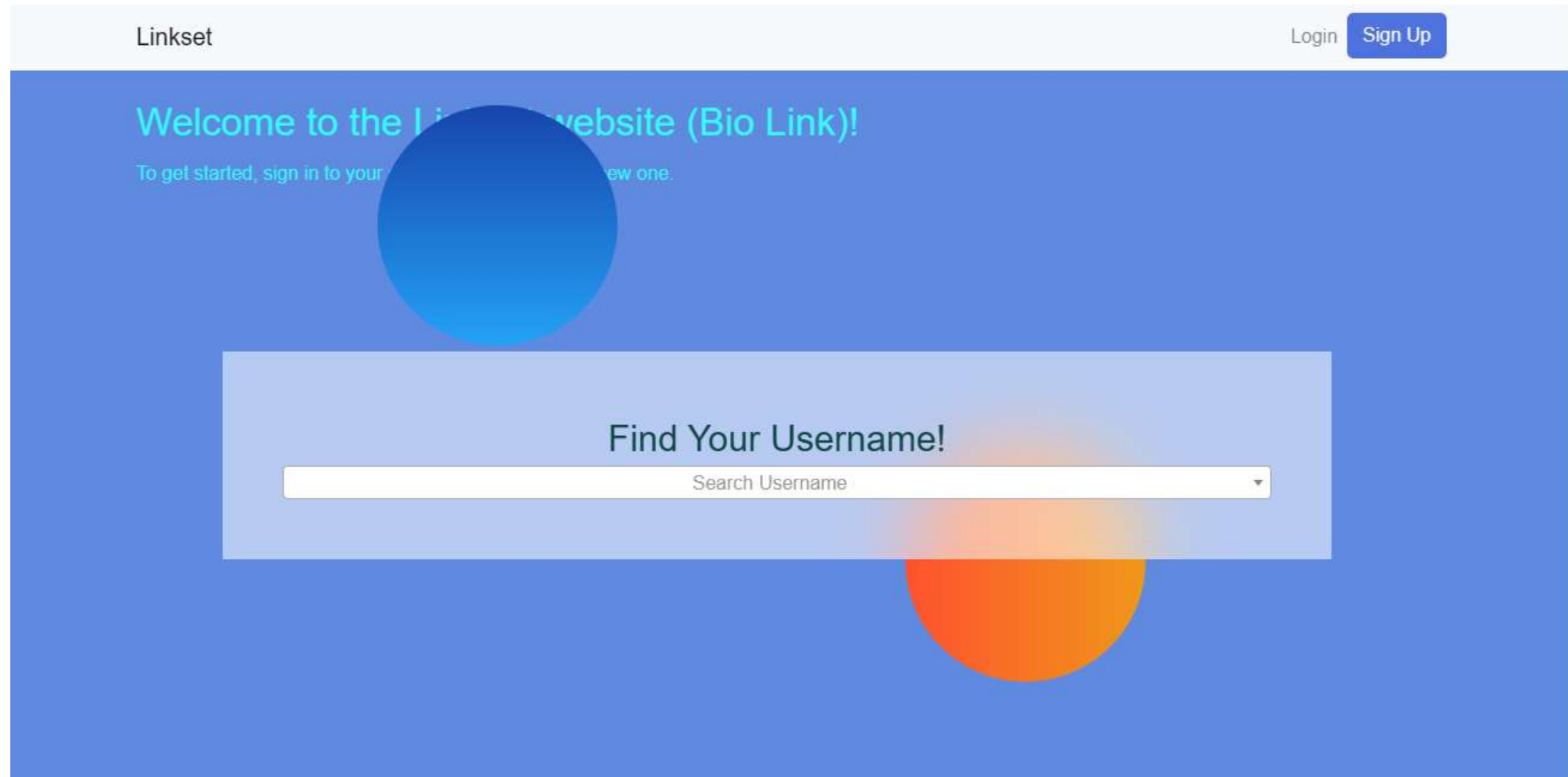
Test ID	NFR 04-01-API			
FR/NFR ID	NFR04 – Security: authentication			
Test Name	Read LINKS			
Objective	This test is intended to test whether the link can be read and viewed on this feature.			
Description	This mechanism is for reading data and viewing links that have been made after creating a link.			
Precondition	Visitors have links listed on the linkset			
Date	30 December 2022			
Tester	Eka Rohani Gultom			
Testing Scenario				
1. Set the target URL (/api/ReadLinks).				
2. Set the request method to Read.				
Evaluation Criteria				
1. The entered data are matched with the stored data.				
Test Cases				
ID	Input	Expected Behavior	Actual Behavior	Verdict
NFR 04-01-API-01	All data required when read a links is filled in. name : myecourse link :https://ecourse.del.ac.id/my/	200 based response by displaying data links	200 based response by displaying data links	[x] passed [] failed
Notes				
Checked it and the cases pass				

NFR05– Tes API Autentifikasi

Test ID	NFR 05-01-API			
FR/NFR ID	NFR05 – Security: authentication			
Test Name	Autentifikasi			
Objective	This test is intended to test whether the everyone can be created account			
Description	This mechanism is for creating accounts and getting tokens as authentication when accessing the website			
Precondition	Has successfully opened the LinkSet website page			
Date	30 December 2022			
Tester	Eka Rohani Gultom			
Testing Scenario				
1. Set the target URL (/api/Register). 2. Set the request method to Read.				
Evaluation Criteria				
1. The entered data are matched with the stored data.				
Test Cases				
ID	Input	Expected Behavior	Actual Behavior	Verdict
NFR 05-01-API-01	All data required when register a account is filled . Username : lidia Email : lidia25@gmail.com Password: lidiagin Confirm Password”: lidiagin	A 201-based response with message “Link Created Successfully”	A 201-based response with message “Link Created Successfully”	[x] passed [] failed
NFR 05-01-API-02	Not all data required when register a account is filled . Username : lidia Email : lidia25 Password: lidiagin Confirm Password”: lidiagin	A 401-based response with message “validation.unique”	A 401-based response with message “validation.unique”	[x] passed [] failed
Notes				
Double Checked it and the cases pass				

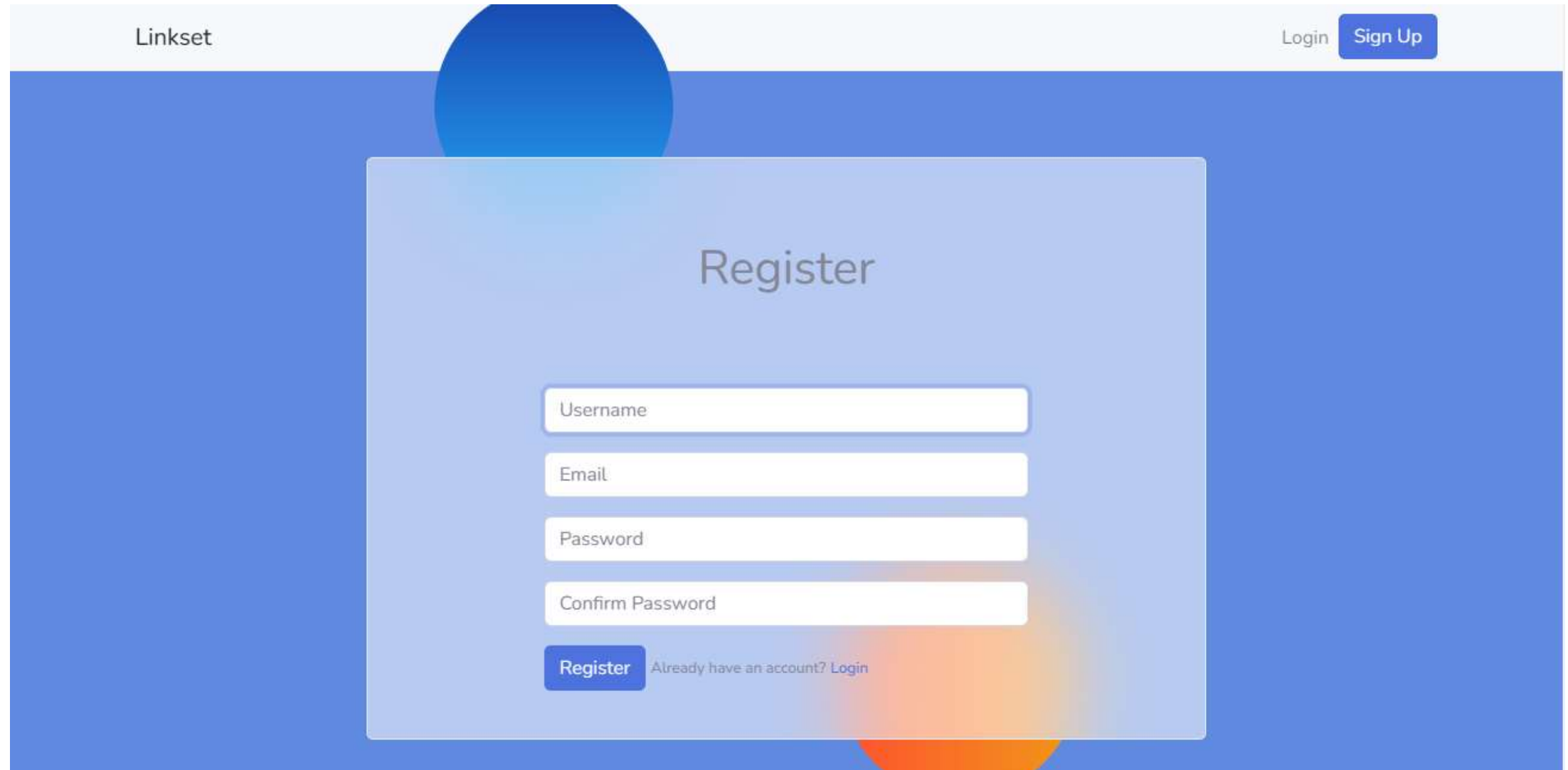
6. Tampilan Website LinkSet (<https://linkset.grup32022.my.id/>)

Tampilan awal LinkSet



Gambar 4 Tampilan awal LinkSet

Tampilan Register



The image shows a web registration form titled "Register". The form is centered on a blue background. At the top, there is a light blue header bar with the text "Linkset" on the left and "Login" and "Sign Up" buttons on the right. The registration form itself is a light blue rectangle with a white border. It contains four input fields: "Username", "Email", "Password", and "Confirm Password". Below these fields is a blue "Register" button and a link that says "Already have an account? Login".

Linkset

Login Sign Up

Register

Username

Email

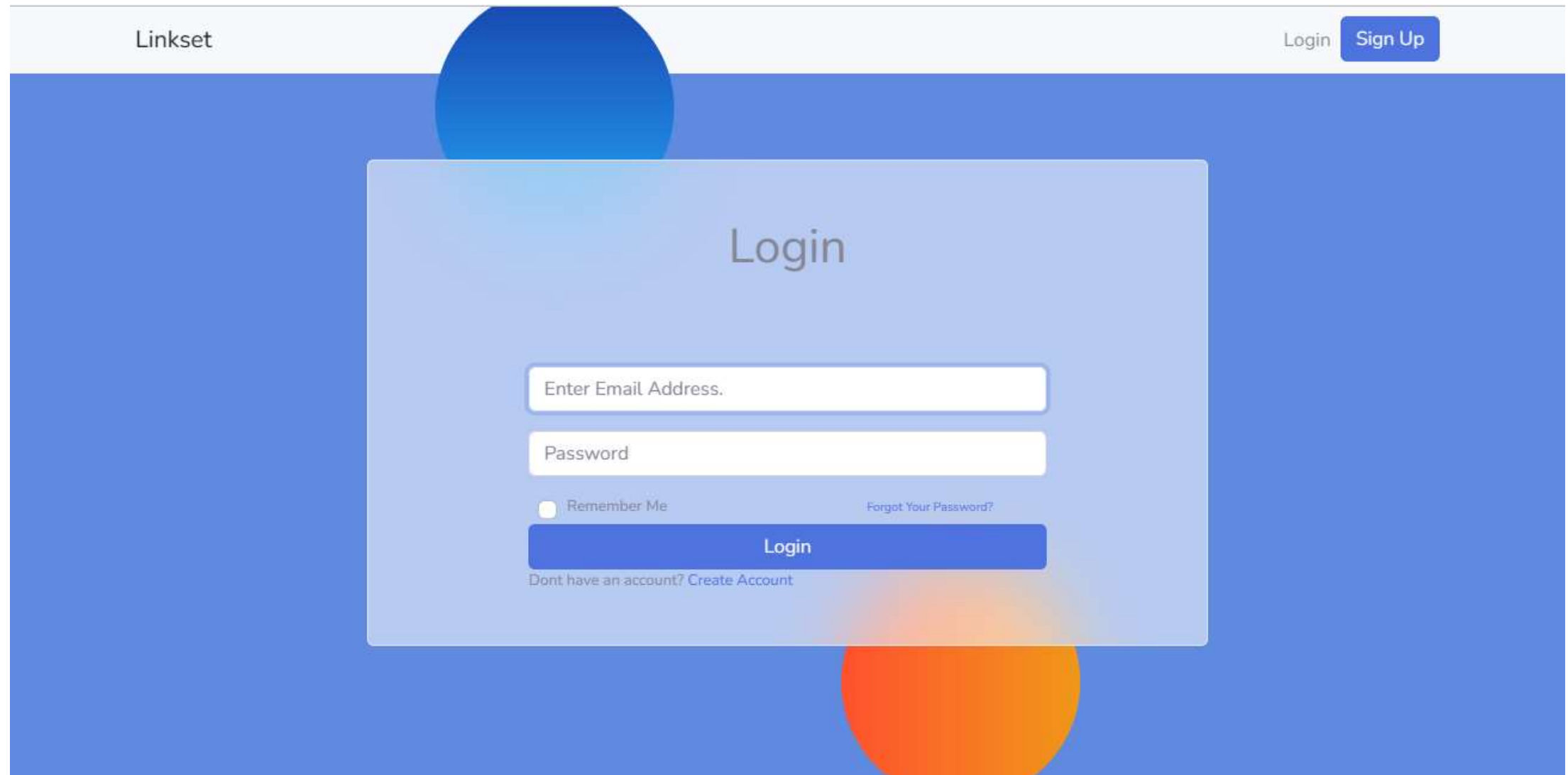
Password

Confirm Password

Register Already have an account? Login

Gambar 5 Tampilan Register

Tampilan Login



The image shows a web login interface. At the top, a light blue header bar contains the text "Linkset" on the left and "Login" and "Sign Up" on the right, with "Sign Up" enclosed in a blue button. The main background is a solid blue color. In the center, there is a light blue rectangular box containing the login form. The form has the title "Login" at the top. Below the title are two input fields: "Enter Email Address." and "Password". Under the "Password" field, there is a "Remember Me" checkbox and a "Forgot Your Password?" link. A blue "Login" button is positioned below these elements. At the bottom of the form box, there is a link that says "Dont have an account? Create Account".

Linkset

Login Sign Up

Login

Enter Email Address.

Password

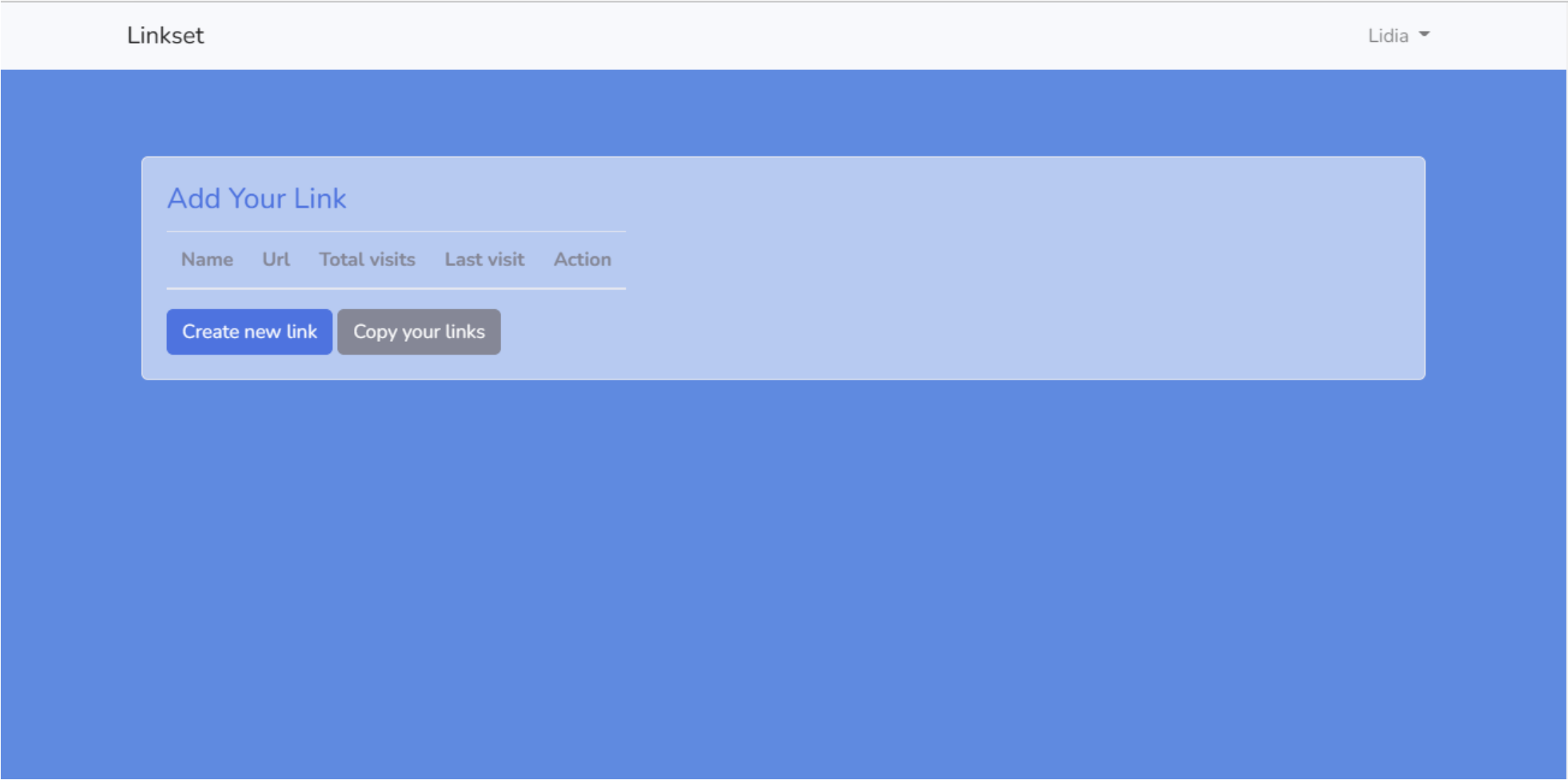
☐ Remember Me [Forgot Your Password?](#)

Login

[Dont have an account? Create Account](#)

Gambar 6 Tampilan Login

Tampilan Dashboard Linkset



Gambar 7 Tampilan Dashboard Linkset

Tampilan Created Link

Linkset

Lidia ▼

Create New Link

Link Name

Social Media Name

Link Url

https://your-links.com

Save Link

Gambar 8 Tampilan Created Link

Tampilan Edit pada LinkSet

Linkset

Lidia ▾

Edit Link

Link Name

Youtube

Link Url

https://www.youtube.com/channel/UCBR8-60-B28hp2BmDPdntcQ

Save Link

Delete Link

Gambar 9 Tampilan Edit pada LinkSet

Tampilan Setting

Linkset

Lidia ▾

Background Color

Text Color

Save Setting

Username

Lidia

Save Profile

New Password

Confirm Password

Save Password

Gambar 10 Tampilan Setting