

 **LAMPIRAN**

Lampiran 1. Surat Kesepakatan Bimbingan Pembimbing 1

SURAT KESEPAKATAN BIMBINGAN SKRIPSI

Kami yang bertanda tangan di bawah ini :

Pihak Pertama

Nama : Nur Khafidah
NIM : 19090075
Program Studi : DIV Teknik Informatika

Pihak Kedua

Nama : Dairoh, M.Sc.
Status : Dosen
NIDN : 0612108701
Jabatan Fungsional : Lektor
Pangkat/Golongan : III/D

Pada hari ini Selasa tanggal 07 Maret 2023 telah terjadi sebuah kesepakatan bahwa Pihak Kedua bersedia menjadi Pembimbing I Skripsi Pihak Pertama dengan syarat Pihak Pertama wajib melakukan bimbingan Skripsi minimal 8 kali kepada Pihak Kedua. Adapun waktu dan tempat pelaksanaan disepakati antar pihak.

Demikian kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Skripsi.

Tegal, 07 Maret 2023

Pihak Pertama



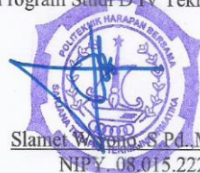
Nur Khafidah

Pihak Kedua



Dairoh, M.Sc.
NIPY. 04.014.178

Mengetahui
Ketua Program Studi D-IV Teknik Informatika



Slamet Widodo, S.Pd., M.Eng.
NIPY. 08.015.222

Lampiran 2. Surat Kesepakatan Bimbingan Pembimbing 2

SURAT KESEPAKATAN BIMBINGAN SKRIPSI

Kami yang bertanda tangan di bawah ini :

Pihak Pertama

Nama : Nur Khafidah
NIM : 19090075
Program Studi : DIV Teknik Informatika

Pihak Kedua

Nama : Ardi Susanto, S.Kom., M.Cs.
Status : Dosen
NIDN : 0629109301
Jabatan Fungsional : Asisten Ahli
Pangkat/Golongan : -

Pada hari ini Selasa tanggal 07 Maret 2023 telah terjadi sebuah kesepakatan bahwa Pihak Kedua bersedia menjadi Pembimbing II Skripsi Pihak Pertama dengan syarat Pihak Pertama wajib melakukan bimbingan Skripsi minimal 8 kali kepada Pihak Kedua. Adapun waktu dan tempat pelaksanaan disepakati antar pihak.

Demikian kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Skripsi.

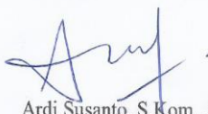
Tegal, 07 Maret 2023

Pihak Pertama



Nur Khafidah

Pihak Kedua



Ardi Susanto, S.Kom., M.Cs.
NIPY. 03.021.490

Mengetahui
Ketua Program Studi D-IV Teknik Informatika



Slamet Wijono, S.Pd., M.Eng.
NIPY. 08.015.222

Lampiran 3. Surat Pernyataan Pengajuan HKI

SURAT PERNYATAAN

Yang bertanda tangan di bawah ini, pemegang hak cipta :

1. Nama : Nur Khafidah
Kewarganegaraan : Indonesia
Alamat : Gg. Mbah Depok II, RT02/05, No.48, Desa Kepandean, Kec.Dukuhturi, Kab. Tegal
2. Nama : Dairoh, M.Sc.
Kewarganegaraan : Indonesia
Alamat : Jl. Garuda No 3 RT 03 RW 09 Randugunting Tegal Selatan Kota Tegal
3. Nama : Ardi Susanto, S.Kom., M.Cs.
Kewarganegaraan : Indonesia
Alamat : Jl. Banyuwangi RT 08 RW 01 Sumurpanggang, Margadana, Kota Tegal

Dengan ini menyatakan bahwa:

1. Karya Cipta yang saya mohonkan:
Berupa : Program Komputer
Berjudul : Aplikasi Undangan *Digital* Berbasis *Website* Menggunakan *Flask*
 - Tidak meniru dan tidak sama secara esensial dengan Karya Cipta milik pihak lain atau obyek kekayaan intelektual lainnya sebagaimana dimaksud dalam Pasal 68 ayat (2);
 - Bukan merupakan Ekspresi Budaya Tradisional sebagaimana dimaksud dalam Pasal 38;
 - Bukan merupakan Ciptaan yang tidak diketahui penciptanya sebagaimana dimaksud dalam Pasal 39;
 - Bukan merupakan hasil karya yang tidak dilindungi Hak Cipta sebagaimana dimaksud dalam Pasal 41 dan 42;
 - Bukan merupakan Ciptaan seni lukis yang berupa logo atau tanda pembeda yang digunakan sebagai merek dalam perdagangan barang/jasa atau digunakan sebagai lambang organisasi, badan usaha, atau badan hukum sebagaimana dimaksud dalam Pasal 65 dan;
 - Bukan merupakan Ciptaan yang melanggar norma agama, norma susila, ketertiban umum, pertahanan dan keamanan negara atau melanggar peraturan perundang-undangan sebagaimana dimaksud dalam Pasal 74 ayat (1) huruf d Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.

2. Sebagai pemohon mempunyai kewajiban untuk menyimpan asli contoh ciptaan yang dimohonkan dan harus memberikan apabila dibutuhkan untuk kepentingan penyelesaian sengketa perdata maupun pidana sesuai dengan ketentuan perundang-undangan.
3. Karya Cipta yang saya mohonkan pada Angka 1 tersebut di atas tidak pernah dan tidak sedang dalam sengketa pidana dan/atau perdata di Pengadilan.
4. Dalam hal ketentuan sebagaimana dimaksud dalam Angka 1 dan Angka 3 tersebut di atas saya / kami langgar, maka saya / kami bersedia secara sukarela bahwa:
 - a. permohonan karya cipta yang saya ajukan dianggap ditarik kembali; atau
 - b. Karya Cipta yang telah terdaftar dalam Daftar Umum Ciptaan Direktorat Hak Cipta, Direktorat Jenderal Hak Kekayaan Intelektual, Kementerian Hukum Dan Hak Asasi Manusia R.I dihapuskan sesuai dengan ketentuan perundang-undangan yang berlaku.
 - c. Dalam hal kepemilikan Hak Cipta yang dimohonkan secara elektronik sedang dalam perkara dan/atau sedang dalam gugatan di Pengadilan maka status kepemilikan surat pencatatan elektronik tersebut ditangguhkan menunggu putusan Pengadilan yang berkekuatan hukum tetap.

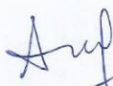
Demikian Surat pernyataan ini saya/kami buat dengan sebenarnya dan untuk dipergunakan sebagaimana mestinya.

Tegal, 8 Juli 2024



Nur Khafidah
Pemegang Hak Cipta*


Dairoh, M.Sc.
Pemegang Hak Cipta*


Ardi Susanto, S.Kom., M.Cs.
Pemegang Hak Cipta*

Pemegang Hak Cipta*

* Semua pemegang hak cipta agar menandatangani di atas materai.

Lampiran 4. Surat Pengalihan HKI

SURAT PENGALIHAN HAK CIPTA

Yang bertanda tangan dibawah ini:

1. Nama : Nur Khafidah
Kewarganegaraan : Indonesia
Alamat : Gg. Mbah Depok II, RT 02 RW 05, No.48, Desa Kepandean,
Kec.Dukuhturi, Kab. Tegal
2. Nama : Dairoh, M.Sc.
Kewarganegaraan : Indonesia
Alamat : Jl. Garuda No 3 RT 03 RW 09 Randugunting Tegal Selatan Kota
Tegal
3. Nama : Ardi Susanto, S.Kom., M.Cs.
Kewarganegaraan : Indonesia
Alamat : Jl. Banyuwangi RT 08 RW 01 Sumurpanggung, Margadana, Kota
Tegal

Adalah **Pihak I** selaku pencipta, dengan ini menyerahkan karya ciptaan saya kepada:

Nama : Pusat Penelitian dan Pengabdian Masyarakat (P3M) Politeknik
Harapan Bersama
Alamat : Jl. Mataram No. 9, Pesurungan Lor, Kota Tegal

Adalah **Pihak II** selaku pemegang Hak Cipta berupa Aplikasi Komputer dengan judul “**Aplikasi Undangan Digital Berbasis Website Menggunakan Flask**” untuk didaftarkan di Direktorat Hak Cipta dan Dosen Industri, Direktorat Jendral Kekayaan Intelektual, Kementerian Hukum dan Hak Asasi Manusia Republik Indonesia.

Demikianlah surat pengalihan hak ini kami buat, agar dapat dipergunakan sebagaimana mestinya.

Tegal, 8 Juli 2024

Pemegang Hak Cipta
Ketua P3M

(Dr. Aldi Badi Riyanta, S.Si., M.T.)



Pencipta

(Nur Khafidah)



(Dairoh, M.Sc.)

(Ardi Susanto, S.Kom., M.Cs.)

Lampiran 5. Sertifikat HKI

 REPUBLIK INDONESIA KEMENTERIAN HUKUM DAN HAK ASASI MANUSIA	
<h1>SURAT PENCATATAN CIPTAAN</h1>	
Dalam rangka perlindungan ciptaan di bidang ilmu pengetahuan, seni dan sastra berdasarkan Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta, dengan ini menerangkan:	
Nomor dan tanggal permohonan	: EC00202464786, 12 Juli 2024
Pencipta	
Nama	: Nur Khafidah, Dairoh, M.Sc. dkk
Alamat	: Gg. Mbah Depok II No 48 RT 02 RW 05, Desa Kepandean Kec.Dukuhturi Kab.Tegal Provinsi Jawa Tengah 52192, Dukuhturi, Tegal, Jawa Tengah, 52192
Kewarganegaraan	: Indonesia
Pemegang Hak Cipta	
Nama	: Pusat Penelitian dan Pengabdian Masyarakat (P3M) Politeknik Harapan Bersama
Alamat	: Jalan Mataram No. 9, Pesurungan Lor, Kecamatan Margadana 52142 , Margadana, Tegal, Jawa Tengah 52142
Kewarganegaraan	: Indonesia
Jenis Ciptaan	: Program Komputer
Judul Ciptaan	: Aplikasi Undangan Digital Berbasis Website Menggunakan Flask
Tanggal dan tempat diumumkan untuk pertama kali di wilayah Indonesia atau di luar wilayah Indonesia	: 12 Juli 2024, di Tegal
Jangka waktu perlindungan	: Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman.
Nomor pencatatan	: 000640141
adalah benar berdasarkan keterangan yang diberikan oleh Pemohon. Surat Pencatatan Hak Cipta atau produk Hak terkait ini sesuai dengan Pasal 72 Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.	
	a.n. MENTERI HUKUM DAN HAK ASASI MANUSIA DIREKTUR JENDERAL KEKAYAAN INTELEKTUAL u.b Direktur Hak Cipta dan Desain Industri
	 IGNATIUS M.T. SILALAH NIP. 196812301996031001
Disclaimer: Dalam hal pemohon memberikan keterangan tidak sesuai dengan surat pernyataan, Menteri berwenang untuk mencabut surat pencatatan permohonan.	

Lampiran 6. Pencipta HKI

LAMPIRAN PENCIPTA

No	Nama	Alamat
1	Nur Khafidah	Gg. Mbah Depok II No 48 RT 02 RW 05, Desa Kepandean Kec.Dukuhturi Kab.Tegal Provinsi Jawa Tengah 52192, Dukuhturi, Tegal
2	Dairoh, M.Sc.	Jl. Garuda No. 3 RT 02 RW 09, Randugunting Kec. Tegal Selatan Kota Tegal Provinsi Jawa Tengah 52131, Tegal Selatan, Tegal
3	Ardi Susanto, S.Kom., M.Cs.	Jl. Banyuwangi RT 08 RW 01, Sumurpanggang Kec.Margadana Kota Tegal Provinsi Jawa Tengah 52141, Margadana, Tegal



Lampiran 7. Lembar bimbingan dosen pembimbing I

IK	P2M	PHB	04.06.G.7.b.3
----	-----	-----	---------------



**SARJANA TERAPAN TEKNIK INFORMATIKA
POLITEKNIK HARAPAN BERSAMA**

LEMBAR BIMBINGAN SKRIPSI

Nama : Nur Khafidah
 NIM : 19090075
 No. Ponsel : 083861367245
 Judul TA : Aplikasi Undangan *Digital* Berbasis
Website Menggunakan Flask
 Dosen Pembimbing I : Dairoh, M.Sc

No	Tanggal	Pemeriksaan	Perbaikan Yang Perlu Dilakukan	Paraf Pembimbing
1	24/3/2023	Aplikasi	- User interface - Desain Interface - use case.	f
2	23/5/2023	- Usecas - Jurnal	Usec	f
3	14/6/2023	- Jurnal - Aplikasi	Metode 6573	f



SARJANA TERAPAN TEKNIK INFORMATIKA
POLITEKNIK HARAPAN BERSAMA

A.	23/06/2023	- Aplikasi - Jurnal	- Dmg jurnal - box :	9
5	6/9/2023	Aplikasi	- Aplikasi 2 back end & terapkan - front end - integrasi - selaraskan aplikasi - dan u/ mengajukan Hki	9
7	7/7/2023	Aplikasi	- template & unduh u/ - Setup jenis - standar - Tambahan menu u/ - dan tampilan - format u/ - format u/ - format u/	9
8	11/7/2023	Aplikasi	- template u/ - Setup u/ - unduh	9
9	12/7/2023	Aplikasi	- & tambahkan - step - proses penerapan	9
10	29/1/2024	Aplikasi	- desain fitur & terapkan algoritma	9

IK	P2M	PHB	04.06.G.7.b.3
----	-----	-----	---------------



**SARJANA TERAPAN TEKNIK INFORMATIKA
POLITEKNIK HARAPAN BERSAMA**

--	--	--	--	--

Tegal, 12 Juli 2023
Dosen Pembimbing I

Dairoh, M.Sc.
NIPY. 04.014.178

Lampiran 8. Lembar bimbingan dosen pembimbing II

IK	P2M	PHB	04.06.G.7.b.3
----	-----	-----	---------------



**SARJANA TERAPAN TEKNIK INFORMATIKA
POLITEKNIK HARAPAN BERSAMA**

LEMBAR BIMBINGAN SKRIPSI

Nama : Nur Khafidah
NIM : 19090075
No. Ponsel : 083861367245
Judul TA : Aplikasi Undangan *Digital* Berbasis
Website Menggunakan Flask
Dosen Pembimbing I : Ardi Susanto, S.Kom., M.Cs.

No	Tanggal	Pemeriksaan	Perbaikan Yang Perlu Dilakukan	Paraf Pembimbing
1)	03/4/2023	- konsep aplikasi - Use case diagram	- Use case diagram	
2)	13/6/2023	- Aplikasi to Use case diagram	- membuat template excel	
3.)	7/9/2023	- Aplikasi	- berjalankan aplikasi sampai selesai	



**SARJANA TERAPAN TEKNIK INFORMATIKA
POLITEKNIK HARAPAN BERSAMA**

4.	12/7.2023	- Aplikasi	- Membuat dashboard utk admin	
5.	29/01.2024	Aplikasi	- Backend digabung	
6.	26/06.2024	Aplikasi	- Sederhanakan konsep yang sekiranya sulit	
7.	02/7.2024	Aplikasi	- Perbaikan bug	
8.	5/7.2024	Aplikasi	- Acc	
9.	17/7.2024	- laporan simulasi	- Acc	

Tegal, 17 Juli 2024
Dosen Pembimbing II

Ardi Susanto, S.Kom., M.Cs.
NIPY. 04.014.178



CREAVITATION



MANUAL BOOK

APLIKASI UNDANGAN DIGITAL BERBASIS WEBSITE MENGGUNAKAN FLASK

Oleh:

- Nur Khafidah
- Dairoh, M.Sc.
- Ardi Susanto, S.Kom., M.Cs.



1. PENDAHULUAN

1.1. Tujuan Pembuatan Dokumen

Tujuan dibuatnya dokumen *user manual* aplikasi undangan *digital* berbasis *website* adalah sebagai berikut:

1. Mendefinisikan fitur aplikasi secara detail.
2. Sebagai panduan penggunaan aplikasi undangan *digital* (Creavitation).

1.2. Deskripsi Umum Sistem

Aplikasi undangan *digital* ini merupakan aplikasi berbasis *website* yang dikembangkan guna mempermudah masyarakat khususnya generasi muda dalam membuat dan mendistribusikan surat undangan untuk acara non-formal atau semi-formal. Dengan memanfaatkan aplikasi *web* ini, pengguna dapat menghemat biaya, waktu serta tenaga dalam proses pembuatan maupun distribusi surat undangan.

1.3. Deskripsi Dokumen

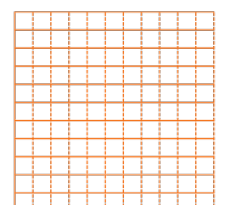
Manual book atau buku panduan merupakan dokumen yang dibuat dengan tujuan memudahkan pengguna dengan memberikan panduan aplikasi *web* undangan *digital*. Berikut merupakan daftar informasi yang tertera dibuku panduan ini:

1.3.1. BAB I

Bab I menjelaskan informasi umum dalam buku panduan yang merupakan bagian dari pendahuluan. Pada bagian ini, informasi yang tersedia meliputi tujuan pembuatan dokumen, penjelasan umum sistem secara keseluruhan, serta deskripsi dokumen.

1.3.2. BAB II

Bab II menjelaskan daftar perangkat yang dibutuhkan untuk menggunakan aplikasi. Pada bagian ini, informasi yang tersedia meliputi perangkat lunak dan perangkat keras yang diperlukan.





1.3.3. BAB III

Bab III menjelaskan panduan penggunaan aplikasi *web* undangan *digital* (Creavitation). Pada bagian ini, informasi yang disediakan berupa cara penggunaan dan penjelasan fitur-fitur pada aplikasi *web* Creavitation.

2. Perangkat yang Dibutuhkan

2.1. Perangkat Lunak

Perangkat lunak yang dibutuhkan pengguna untuk menjalankan aplikasi *web* Creavitation adalah *web browser*.

2.2. Perangkat Keras

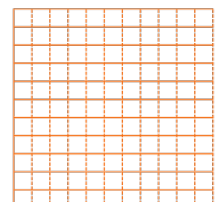
Pengguna cukup menyediakan perangkat keras berupa komputer, laptop ataupun *smartphone* yang telah terhubung ke jaringan internet yang stabil untuk menggunakan aplikasi *web* Creavitation.

3. Menu dan Cara Penggunaan

3.1. Struktur Menu

Struktur menu aplikasi *web* Creavitation terbagi menjadi 2 level, yaitu *level admin* dan *level user*. Berikut rincian menu yang terdapat didalam aplikasi

1. Menu Home
2. Sign In
3. Sign Up
4. Dashboard Admin:
 - a. Dashboard
 - b. Category
 - c. Template
 - d. User
5. Dashboard User:
 - a. Dashboard
 - b. Undanganku
 - c. Template



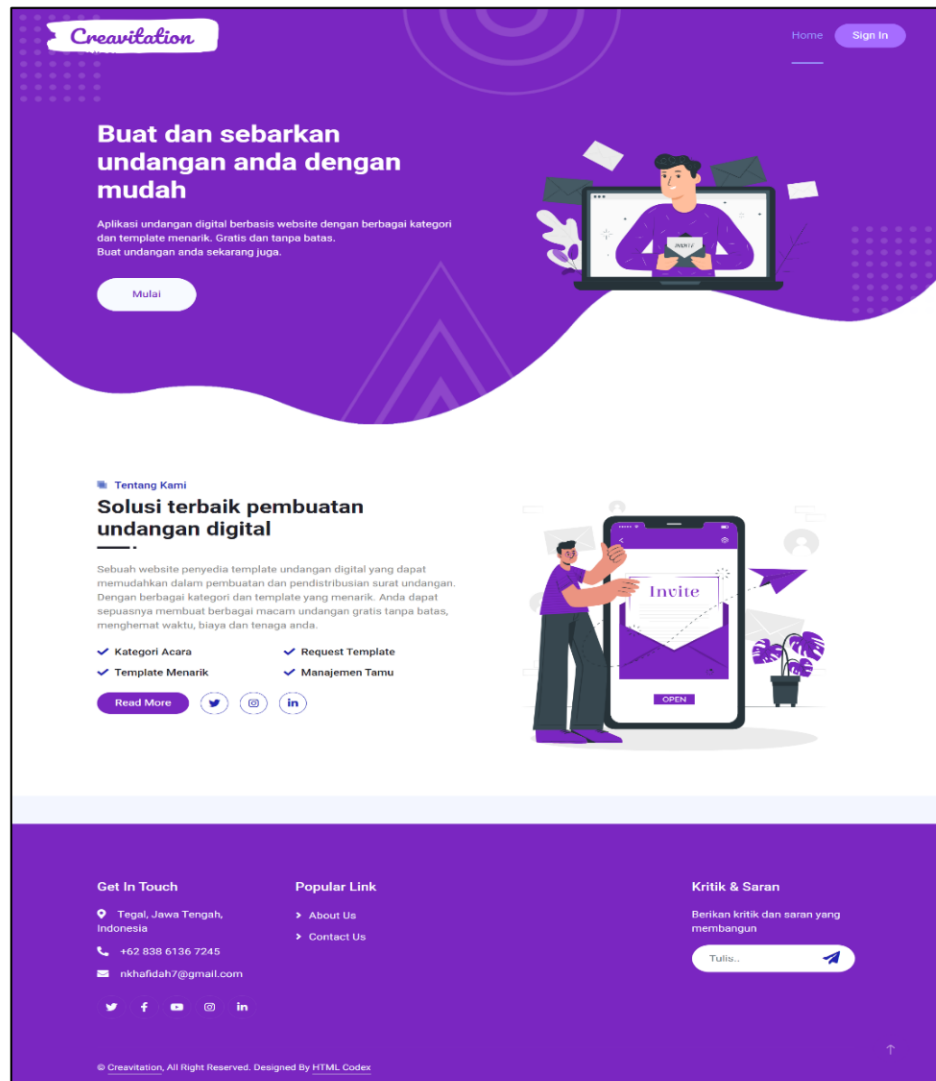


d. Tamu

3.2. Penggunaan

3.2.1. Cara Mengakses Aplikasi *Website* Creavitation

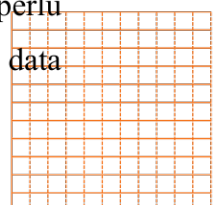
Sebelum mulai mengakses aplikasi *web* pastikan perangkat keras yang digunakan telah terhubung ke jaringan internet, usahakan untuk memilih jaringan internet yang stabil. Kemudian pengguna dapat membuka *link website* dengan menggunakan *web browser* yang tersedia diperangkat pengguna. Berikut adalah tampilan utama saat *link website*:



Gambar 3.1. Halaman *home* aplikasi *web* Creavitation

3.2.2. Cara Melakukan *Sign Up* dan *Sign In*

Untuk mulai membuat undangan *digital*, pengguna perlu melakukan registrasi dan *login*. Hal ini dilakukan guna memprivasi data

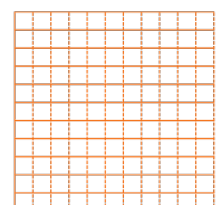


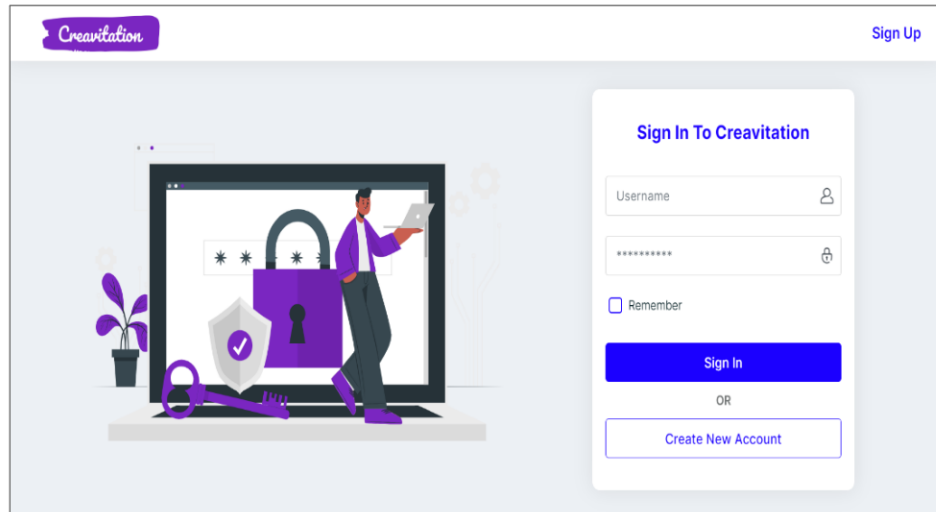


satu pengguna dengan pengguna lainnya. Berikut merupakan Langkah-langkah untuk melakukan *Sign Up* (registrasi akun) dan *Sign In* (login akun):

Gambar 3.2. Halaman *Sign Up*

1. Pada halaman utama aplikasi, pengguna akan diperlihatkan menu navigasi aplikasi antara lain: *Home* dan *Sign In*. Untuk melakukan *Sign In* atau *Sign Up* pengguna harus menekan tombol **Sign In**.
2. Pengguna akan diarahkan ke halaman *Sign In/Login* aplikasi. Bagi pengguna yang belum memiliki akun, diharuskan melakukan registrasi aplikasi. Untuk memulainya pengguna hanya perlu menekan tombol **Sign Up** yang terdapat dinavigasi menu login, atau tombol **Sign Up** yang berada dibawah tombol **Sign In**.
3. Pada Langkah berikutnya, pengguna hanya perlu memasukkan data diri untuk melakukan registrasi. Pastikan data yang diinputkan sudah benar, perhatikan kolom input yang memiliki **tanda bintang (*)** pada *form*, ini merupakan *mandatory* data atau data yang **wajib diisi**. Pengguna dapat menekan tombol **Sign Up** untuk lanjut kelangkah berikutnya.



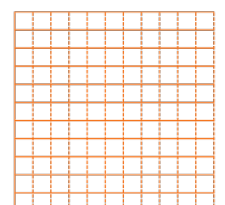


Gambar 3.3. Halaman *Sign In*

4. Setelah berhasil melakukan registrasi, pengguna akan kembali diarahkan ke menu *Sign In*. Masukkan *email* dan *password* yang telah didaftarkan. Kemudian tekan tombol **Sign In** untuk masuk kedalam *dashboard* aplikasi.
5. Saat pengguna berhasil *sign in* maka pengguna akan diarahkan ke halaman *dashboard* aplikasi yang berisikan beberapa informasi dan navigasi menu dibagian kiri layar. Perlu diingat bahwa menu **Sign Up** hanya diperuntukkan khusus pengguna *level User*, sedangkan pengguna *level Admin* hanya dapat melakukan **Sign In** dengan langkah-langkah yang sama.

3.2.3. Pengguna *Level Admin*

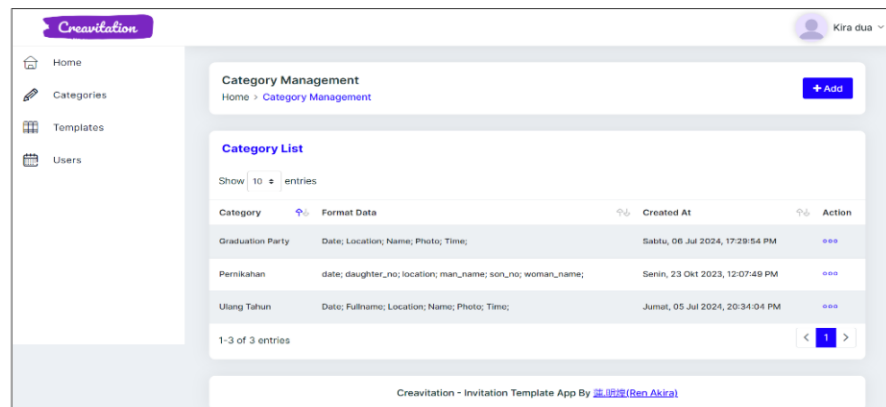
Kelebihan dari *level* akun Admin adalah dapat melakukan manajemen pada kategori acara dan *template* undangan yang tersedia didalam aplikasi, permintaan *template* pengguna yang dapat ditinjau melalui *dashboard*, serta akun pengguna yang terdaftar didalam aplikasi *web* Creavitation. Di halaman *dashboard*, admin juga dapat meninjau jumlah kategori, jumlah *template*, jumlah undangan, dan jumlah pengguna yang telah terdaftar aplikasi ini. Fitur ini memudahkan admin untuk mengetahui informasi secara cepat dan singkat.





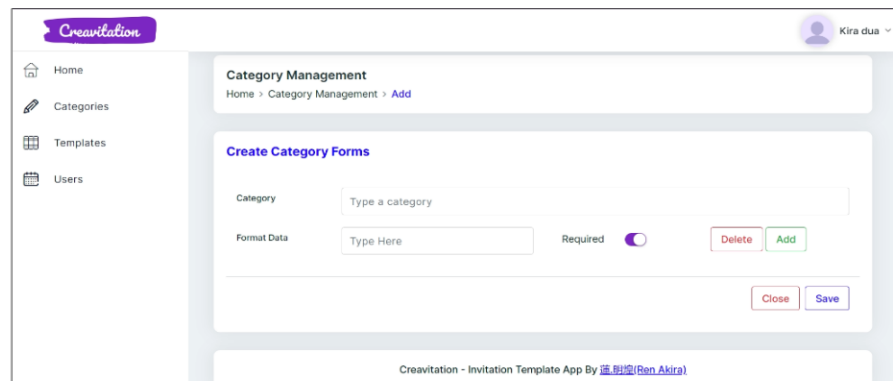
3.2.3.1. Manajemen Kategori Acara

Pada halaman kategori, admin dapat melakukan pengolahan data kategori acara mulai dari menambahkan data baru, mengubah data dan menghapus data yang ada. Admin juga disajikan data kategori acara dalam bentuk table untuk memudahkan pembacaan data pada halaman utama menu kategori.

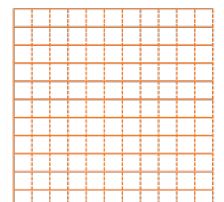


Gambar 3.4. Halaman kategori (khusus admin)

Untuk menambahkan kategori, admin hanya perlu menekan tombol **Add** yang berada di ujung kanan heading halaman. Selanjutnya admin akan diarahkan ke halaman baru yang berisi *form* pengisian data kategori yang baru. Isikan nama kategori baru dan format data yang dibutuhkan. Klik tombol **Save** untuk menyimpan data.



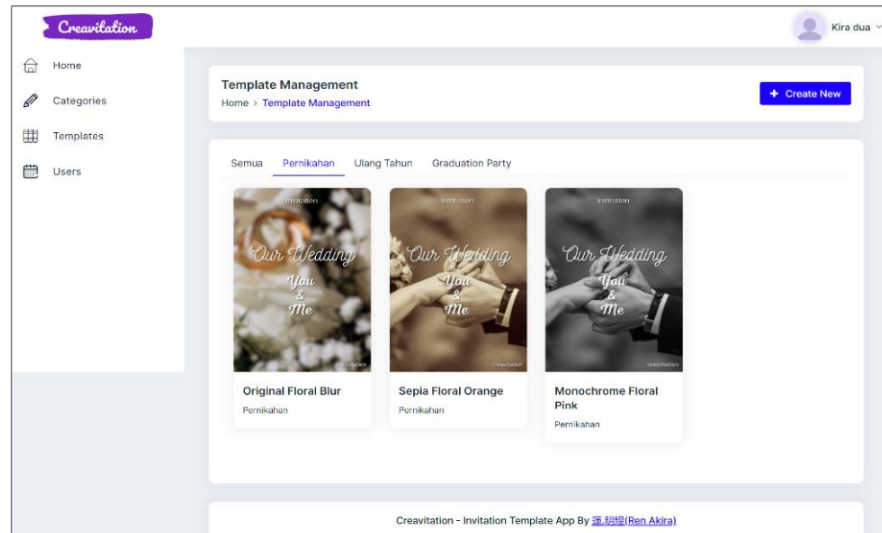
Gambar 3.5. *Form* tambah data kategori





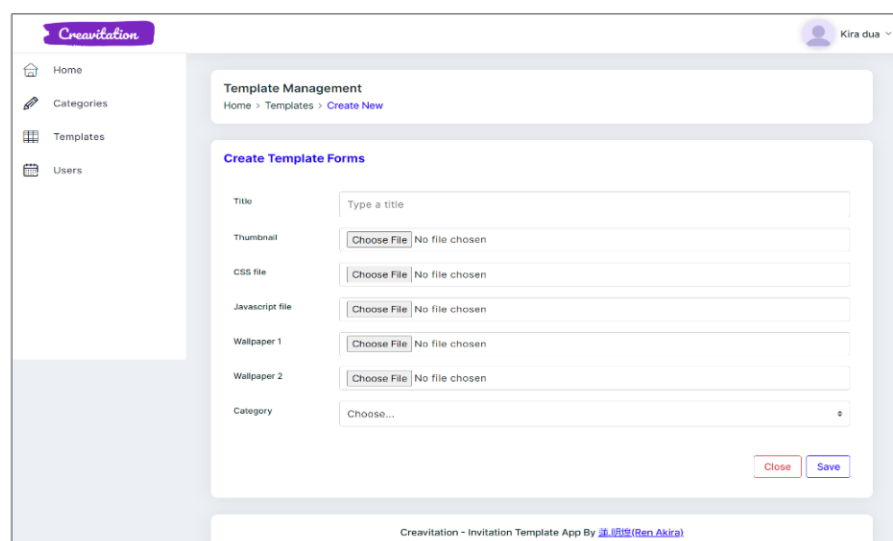
3.2.3.2. Manajemen *Template*

Didalam menu *template*, admin juga memiliki akses untuk menambahkan data baru, mengubah data, menghapus data, dan melihat data *template* undangan.



Gambar 3.6. Halaman daftar *template* (khusus admin)

Untuk menambahkan data *template* baru, klik tombol **Create New** yang terletak di ujung kanan *header* halaman. Kemudian admin akan diarahkan ke *form* tambah *template*. Pada halaman tersebut, admin hanya perlu mengisikan *form* input yang tersedia kemudia tekan tombol **Save** untuk menyimpan data.



Gambar 3.7. *Form* tambah data *template*

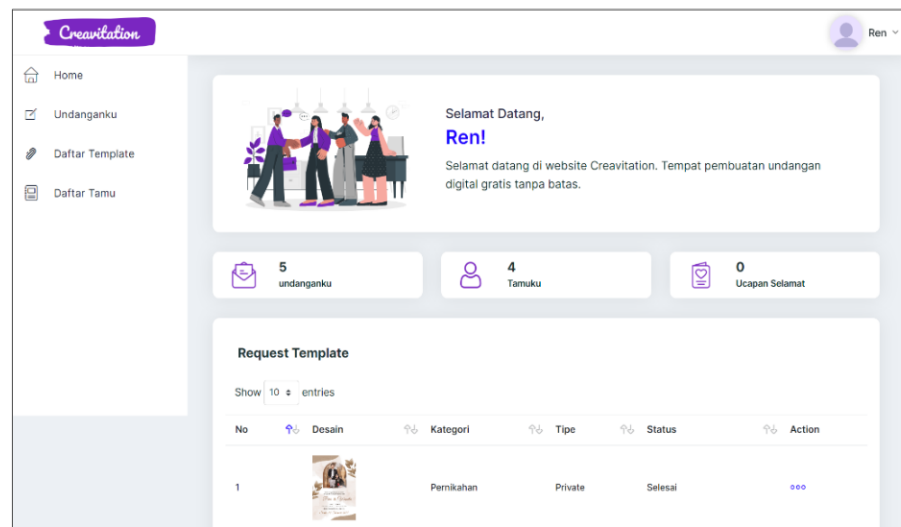




3.2.4. Pengguna *Level User*

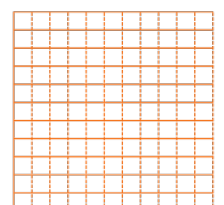
3.2.4.1. Cara Membuat Undangan

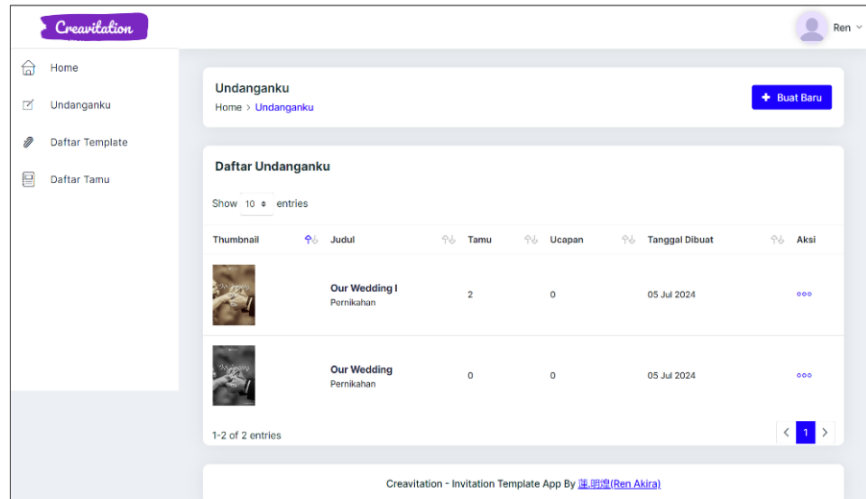
Pembuatan undangan *digital* menggunakan aplikasi ini cukup sederhana, pengguna hanya perlu memilih *template* dan kategori yang tersedia serta mengisi semua data yang diperlukan. Untuk langkah-langkah dalam membuat undangan adalah sebagai berikut ini:



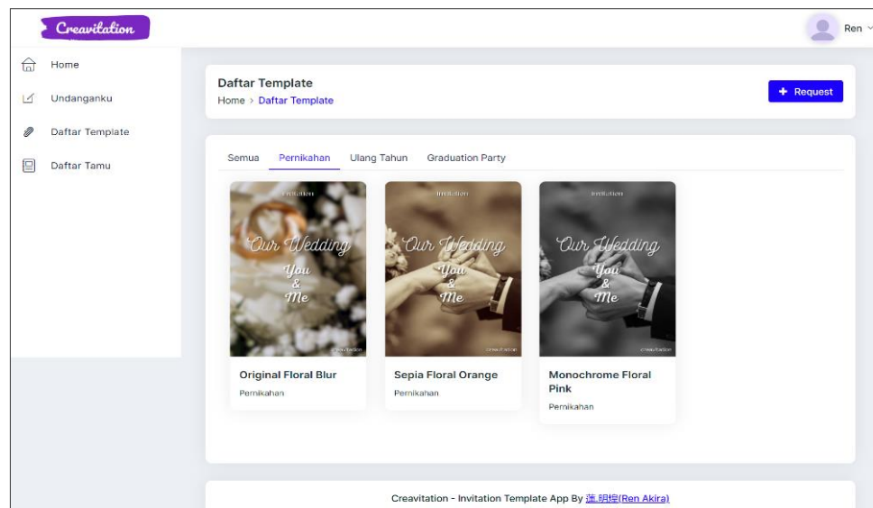
Gambar 3.8. Halaman *dashboard user*

1. Pada tampilan *dashboard* terdapat menu navigasi pada bagian kiri layar. Pengguna dapat masuk ke menu **Undanganku** atau menu **Daftar Template** untuk mulai membuat undangan.
2. Jika pengguna ingin mulai membuat undangan dari halaman **Undanganku**, pengguna hanya perlu menekan tombol **Buat Baru** yang terletak pada kanan atas halaman. Selanjutnya pengguna akan diarahkan ke menu halaman baru yang berisikan kategori dan *template* undangan yang tersedia di aplikasi. Pengguna hanya perlu memilih *template* yang ada menyesuaikan kategori acara yang dibutuhkan. Jika pengguna ingin memulainya dari halaman **Daftar Template**, pengguna bisa secara langsung memilih kategori dan *template* yang diinginkan.





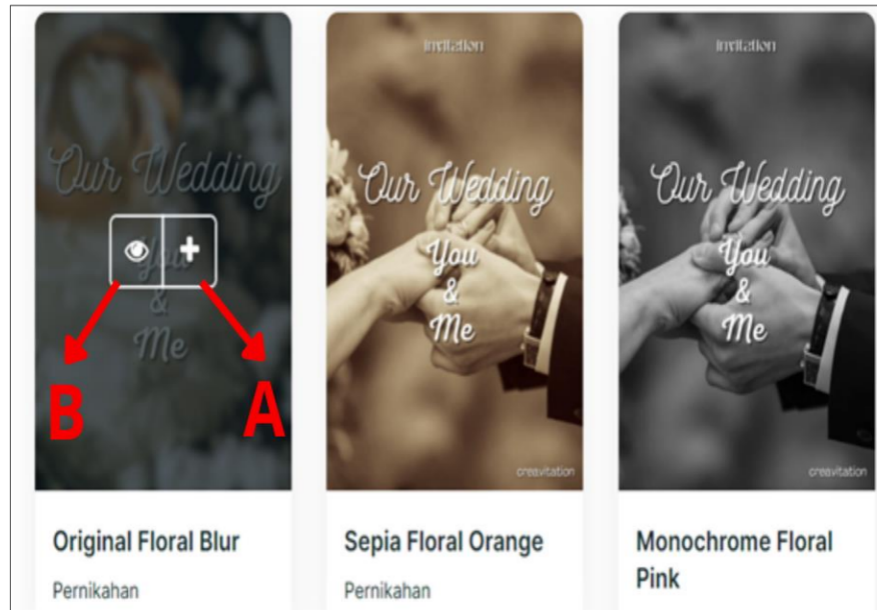
Gambar 3.9. Halaman undanganku



Gambar 3.10. Halaman daftar *template* (khusus user)

- Untuk mulai menggunakan *template*, klik *template* atau arahkan *pointer* ke *thumbnail template* kemudian tekan tombol dengan simbol **tambah** untuk lanjut ke tahap pembuatan undangan. Pengguna juga diberikan kesempatan untuk melihat tampilan dari *template* yang akan digunakan dengan menekan tombol bersimbol **mata** yang berada tepat disamping tombol **tambah**. Fitur ini berguna apabila pengguna ingin meninjau tampilan dan *style* dari *template* undangan yang tersedia.





Gambar 3.11. *Template* undangan. (A) Tombol untuk tambah undangan; (B) Tombol untuk menampilkan desain *template*.

- Setelah menekan tombol **tambah**, pengguna akan disajikan *form* untuk mengisi data diri dan detail informasi acara. Pastikan untuk mengisi kolom input yang memiliki tanda **bintang(*)**. Klik tombol **simpan** untuk *generate* undangan secara otomatis. Selanjutnya pengguna akan diarahkan ke halaman detail informasi undangan yang berhasil dibuat.

Gambar 3.12. *Form* pembuatan undangan *digital*

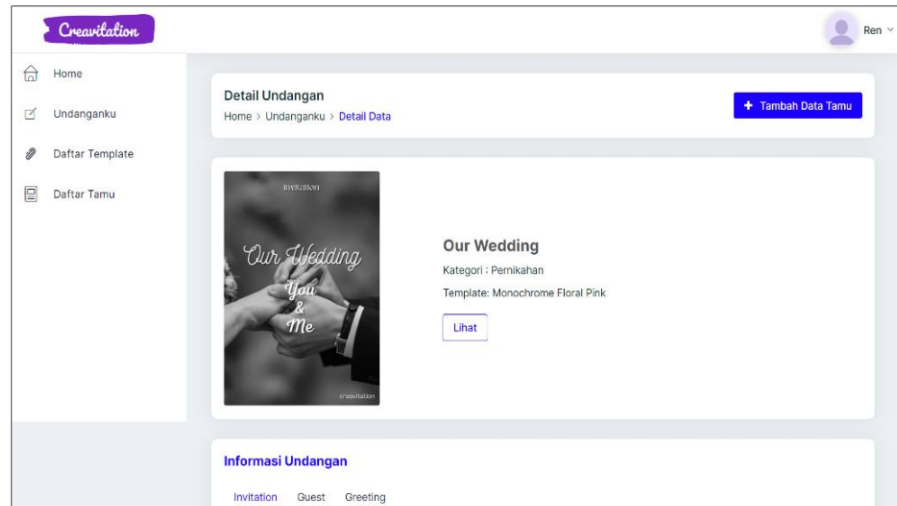
3.2.4.2. Menampilkan Undangan

Untuk menampilkan undangan yang telah dibuat, bisa





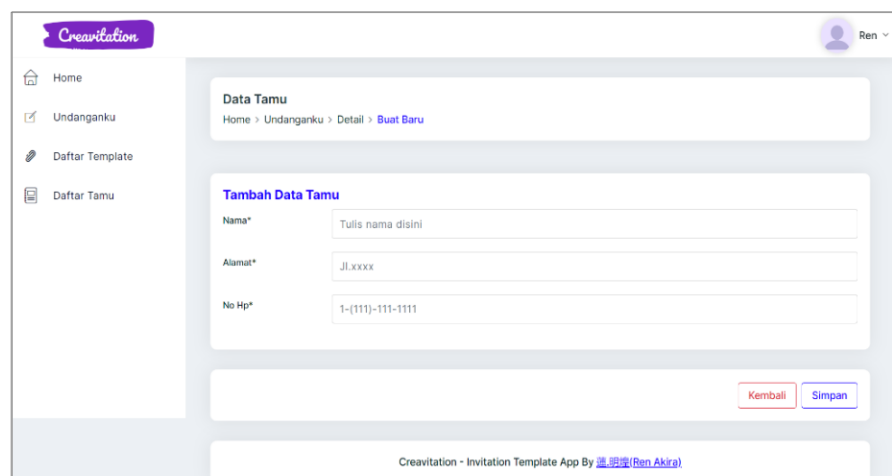
melalui halaman **Undanganku** dengan menekan tombol aksi lalu pilih **Lihat** atau pengguna juga dapat melakukannya melalui halaman **Detail Undangan** dengan menekan tombol **Lihat** yang terletak pada deskripsi singkat undangan.



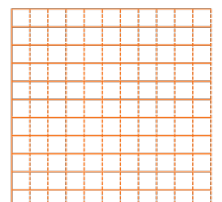
Gambar 3.13. Halaman detail undangan

3.2.4.3. Membagikan Undangan

Sebelum membagikan undangan perlu diperhatikan bahwa pengguna harus memasukkan data tamu terlebih dahulu. Pengguna hanya bisa menambahkan data tamu dari halaman **Detail Undangan** dengan menekan tombol **Tambah Data Tamu**.

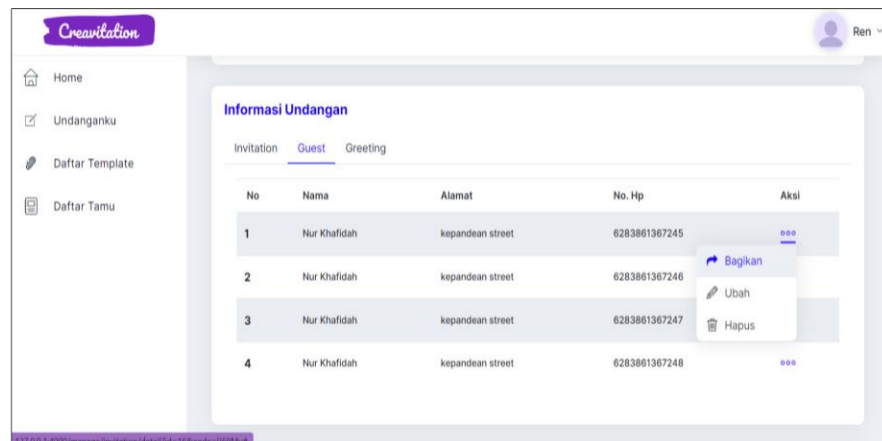


Gambar 3.14. Form tambah data tamu



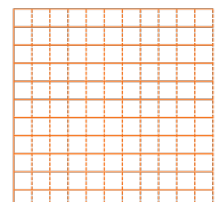


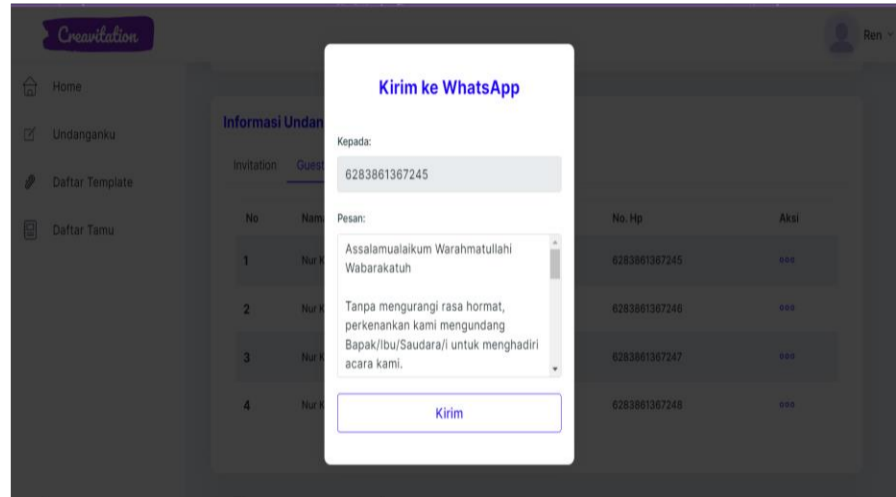
Dihalaman *form* pengisian data tamu, pengguna hanya perlu memasukkan nama tamu, alamat tamu, dan nomor telepon dari tamu yang akan diundang. Klik **simpan** untuk menambahkan data baru. Jika data berhasil tersimpan, maka data tamu akan terlihat pada tab **Daftar Tamu** di halaman detail undangan dalam bentuk data tabel. Pengguna bisa langsung membagikan undangan digital dengan menekan tombol pada kolom aksi kemudian klik **Bagikan**.



Gambar 3.15. Daftar data tamu

Selanjutnya pengguna akan disajikan *form* berisi nomor telepon dari tamu undangan beserta format pesan yang akan disampaikan kepada tamu undangan. Pengguna dapat merubah atau memodifikasi format pesan bawaan sesuai keinginannya. klik tombol **Kirim** kemudian pengguna akan diarahkan ke aplikasi *whatsapp*, selanjutnya pengguna hanya perlu melanjutkan pengiriman.

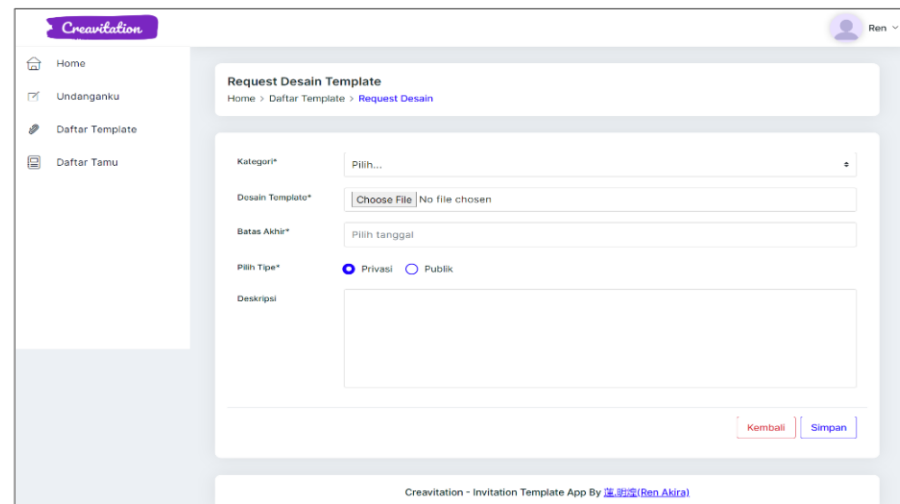




Gambar 3.16. Form pesan pembagian undangan digital

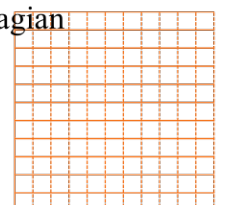
3.2.4.4. Request Desain Template

Selain memanfaatkan desain *template* undangan yang tersedia didalam aplikasi, pengguna juga diberikan kesempatan untuk mengirimkan permintaan desain *template* kepada admin. Untuk mengirimkan permintaan desain *template* undangan, pengguna perlu menyiapkan foto desain dari *template* yang diinginkan terlebih dahulu.



Gambar 3.17. Form permintaan desain template

Setelah menyiapkan foto desain, pengguna bisa mengirim permintaan melalui halaman **Daftar Template**. Pada bagian





ujung kanan header halaman, klik tombol **Request** untuk selanjutnya pengguna akan diarahkan ke *form* pengisian permintaan. Pastikan untuk mengisi input data yang bertanda **bintang (*)** termasuk deskripsi desain. Pengguna juga dapat menentukan tipe desain yang dikirimkan menjadi publik atau privasi. Klik **Simpan** untuk mengirimkan data.

Selanjutnya desain permintaan pengguna akan diterima oleh admin untuk dipilah dan proses. Perlu diingat bahwa pengguna hanya bisa mengirimkan permintaan desain 1 minggu sebelum tanggal yang ditentukan.





TECHNICAL DOCUMENT

APLIKASI UNDANGAN DIGITAL BERBASIS WEBSITE MENGGUNAKAN FLASK

Oleh:

- Nur Khafidah
- Dairoh, M.Sc.
- Ardi Susanto, S.Kom., M.Cs.



Profil

Aplikasi web *Creavitation* merupakan aplikasi pembuatan undangan digital berbasis *website* yang dibangun dengan tujuan untuk mempermudah Masyarakat khususnya generasi muda dalam membuat dan membagikan undangan untuk acara non-formal atau semi-formal. Website ini menyediakan beberapa template undangan untuk kategori acara yang berbeda, serta memiliki fitur request desain template dan manajemen daftar tamu undangan.

Latar Belakang

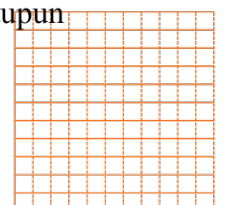
Surat undangan selalu menjadi bagian penting dalam perencanaan suatu acara baik acara formal, semi-formal, maupun acara non-formal. Selama ini informasi acara hanya disampaikan menggunakan undangan berbentuk kertas yang ditujukan kepada tamu yang diundang. Diera ini, cara tersebut dianggap memiliki banyak kekurangan seperti resiko kesalahan pencetakan informasi acara, undangan yang rentan rusak karena berbahan kertas, sulitnya menyebarkan undangan karena terkendala transportasi dan lokasi, serta masih banyak lagi.

Dengan pesatnya perkembangan teknologi saat ini, masyarakat mulai terbiasa menggunakan teknologi untuk membantunya dalam menyelesaikan tugas-tugas ataupun melakukan aktivitas sehari-hari. Maka dari itu dicetuskan sebuah solusi untuk mengatasi permasalahan yang ada dengan memanfaatkan teknologi yaitu dikembangkannya aplikasi undangan digital berbasis *website*. Selain dianggap bisa menyelesaikan permasalahan diatas, dengan adanya aplikasi undangan digital masyarakat dapat menghemat tenaga, waktu serta biaya dalam pembuatan dan pembagian surat undangan. Hal ini juga diharapkan memberikan manfaat lebih bagi masyarakat dan penulis.

Manfaat

Dengan dibangunnya aplikasi *web Creavitation* diharapkan memberikan manfaat bagi Masyarakat atau pengguna. Berikut adalah manfaat yang dapat diperoleh dengan adanya aplikasi *web Creavitation*, antara lain:

1. Pengguna dapat menghemat waktu, biaya, dan tenaga dalam pembuatan maupun





- pembagian undangan.
2. Meminimalisir kesalahan pencetakan informasi acara maupun informasi pemilik acara.
 3. Membantu pengguna untuk manajemen daftar tamu undangan untuk setiap acara.
 4. Menjadikan surat undangan lebih menarik dan animatif.

Spesifikasi

1. Perangkat Lunak
 - 1.1. Visual Studio Code
 - 1.2. XAMPP
 - 1.3. Postman
 - 1.4. Google Chrome
2. Framework : Flask
3. Database : MySQL
4. Source Code

4.1. Halaman Home

Berikut adalah baris kode program untuk menampilkan halaman home yang berisi informasi aplikasi dan informasi kontak developer.

1. Python Flask – Front End API

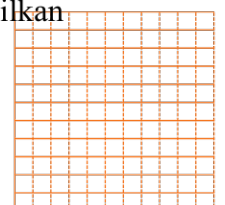
Kode dibawah ini merupakan endpoint Front End API dengan Bahasa pemrograman Python yang berfungsi untuk menjalankan halaman home agar dapat dioperasikan pada Flask Framework.

```
59  ## Route Home ----- Start
60  @app.route('/')
61  def home():
62  return render_template(
63      title=TITLE_HOME,
64      template_name_or_list='pages/index.html'
65  )
66  ## Route Home ----- Finish
67
```

Gambar 4.1. kode program python untuk endpoint home

2. HTML

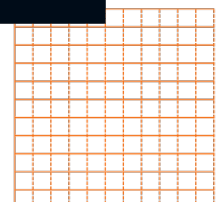
Kode dibawah merupakan baris kode HTML yang akan menghasilkan tampilan home pada aplikasi web Creavitation.





```
1 <!-- ===== -->
2 <!-- PAGE CONTENT - GUEST - USER -->
3 <!-- ===== -->
4 {% extends 'layouts/layout.html' %} {% block content %}
5 <div class="container-xxl bg-white p-0">
6
7 <!-- ===== -->
8 <!-- Spinner Start -->
9 {% include 'components/preloader.html' %}
10 <!-- Spinner End -->
11 <!-- ===== -->
12
13 <!-- ===== -->
14 <!-- Navbar & Hero Start -->
15 <div class="container-xxl position-relative p-0" id="top">
16 <!-- ===== -->
17 <!-- Navbar Start -->
18 {% include 'components/navbar.html' %}
19 <!-- Navbar End -->
20 <!-- ===== -->
21
22 <!-- ===== -->
23 <!-- Hero Start -->
24 <div class="container-xxl py-5 bg-primary hero-header mb-5">
25 <div class="container my-5 py-5 px-lg-5">
26 <div class="row g-5 py-5">
27 <div class="col-lg-6 text-center text-lg-start">
28 <h1 class="text-white mb-4 animated zoomIn">
29 Buat dan sebarlan undangan anda dengan mudah
30 </h1>
31 <p class="text-white pb-3 animated zoomIn">
32 Aplikasi undangan digital berbasis website dengan berbagai kategori
33 dan template menarik. Gratis dan tanpa batas. <br>Buat undangan anda sekarang juga.
34 </p>
35 <a href="{{ url_for('auth.sign_in') }}" class="btn btn-light py-sm-3 px-sm-5 rounded-pill a
36 </div>
37 <div class="col-lg-6 text-center text-lg-start">
38 
52 <div class="container px-lg-5">
53 <div class="row g-5">
54 <div class="col-lg-6 wow fadeInUp" data-wow-delay="0.1s">
55 <div class="section-title position-relative mb-4 pb-2">
56 <h2 class="position-relative text-primary ps-4">Tentang Kami</h2>
57 <h2 class="mt-2">Solusi terbaik pembuatan undangan digital</h2>
58 </div>
59 <p class="mb-4">
60 Sebuah website penyedia template undangan digital yang dapat memudahkan dalam pembuatan
61 dan pendistribusian surat undangan. Dengan berbagai kategori dan template yang menarik.
62 Anda dapat sepenuhnya membuat berbagai macam undangan
63 gratis tanpa batas, menghemat waktu, biaya dan tenaga anda.
64 </p>
65 <div class="row g-3">
66 <div class="col-sm-6">
67 <h6 class="mb-3"><i class="fa fa-check text-primary me-2"></i>Kategori Acana</h6>
68 <h6 class="mb-0"><i class="fa fa-check text-primary me-2"></i>Template Menarik</h6>
69 </div>
70 <div class="col-sm-6">
71 <h6 class="mb-3"><i class="fa fa-check text-primary me-2"></i>Request Template</h6>
72 <h6 class="mb-0"><i class="fa fa-check text-primary me-2"></i>Manajemen Tamu</h6>
73 </div>
74 </div>
75 <div class="d-flex align-items-center mt-4">
76 <a class="btn btn-primary rounded-pill px-4 me-3" href="">Read More</a>
77 <a class="btn btn-outline-primary btn-square me-3" href=""><i class="fab fa-twitter"></i></a>
78 <a class="btn btn-outline-primary btn-square me-3" href=""><i class="fab fa-instagram"></i></a>
79 <a class="btn btn-outline-primary btn-square" href=""><i class="fab fa-linkedin-in"></i></a>
80 </div>
81 </div>
82 <div class="col-lg-6">
83 <i class="bi bi-arrow-up"></i></a>
94 <!-- Back to Top -->
95 <!-- ===== -->
96 </div>
97 {% endblock %}
98 <!-- ===== -->
99 <!-- END PAGE CONTENT -->
100 <!-- ===== -->
```

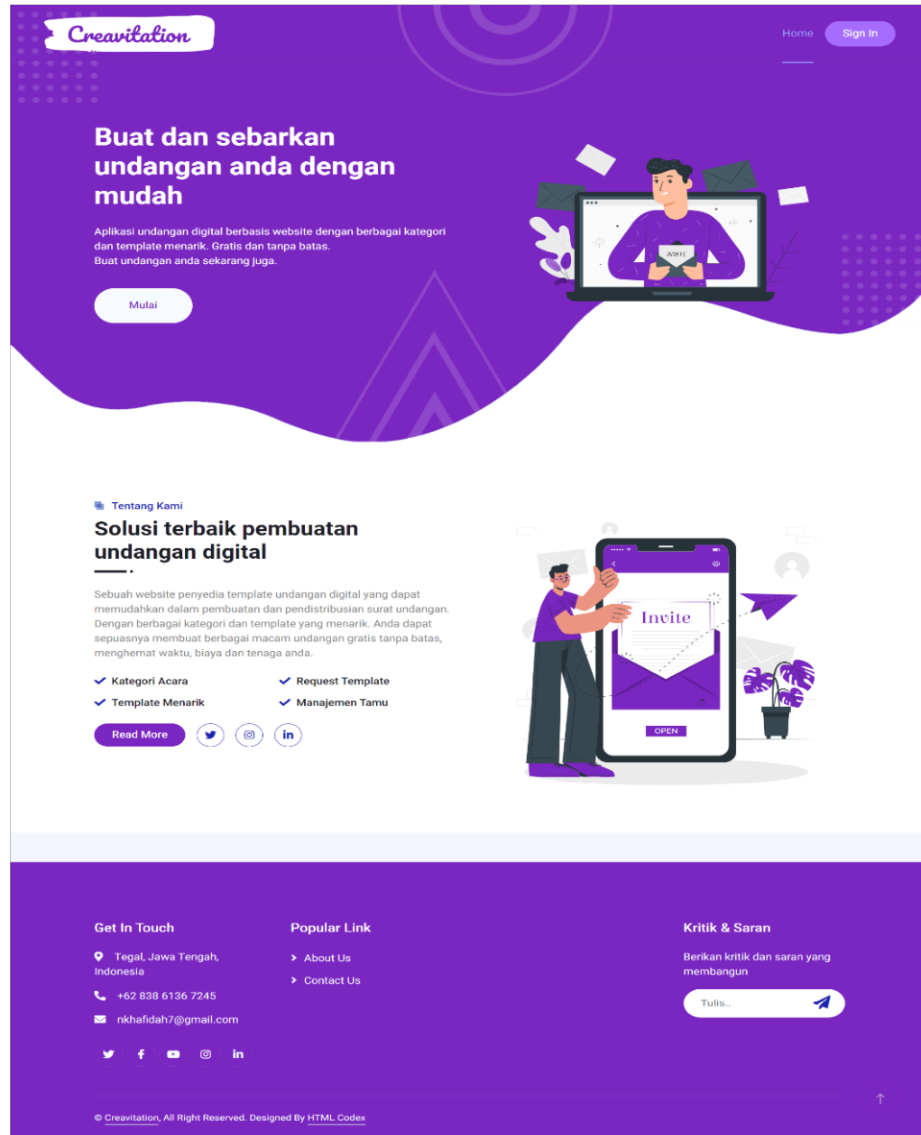
Gambar 4.2 Kode HTML untuk tampilan home





3. Hasil

Kode pada gambar 4.1 dan gambar 4.2 akan menghasilkan tampilan home seperti dibawah ini.



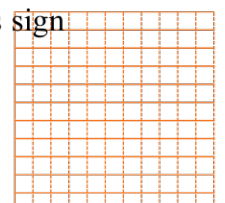
Gambar 4.3 Tampilan home yang dihasilkan

4.2. Halaman Sign In

Berikut adalah kode program untuk halaman Sign in yang berfungsi untuk menampilkan halaman Sign In dan melakukan login akun sebelum pengguna maupun admin menggunakan aplikasi web Creavitation.

1. Python Flask – Back end API

Kode program dari sign in admin dan user memiliki kesamaan dalam struktur dan penulisan. Dibawah ini merupakan contoh kode proses sign



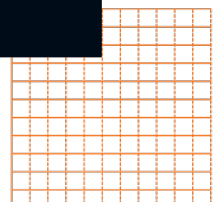


in dari pengguna level admin.

```
38 # SIGN IN ADMIN ===== Begin
39 # POST https://127.0.0.1:5000/admin/signin
40 @admin.post('/signin')
41 def signin():
42     try:
43         # Request Data =====
44         data = request.json
45
46         # Request Data =====
47         response = AdminModels.signin_admin(data)
48
49         # Request Data =====
50         return response
51
52     except Exception as e:
53         return bad_request(str(e))
54 # SIGN IN ADMIN ===== End
```

Gambar 4.4 Controller back end API

```
14 class AdminModels():
89 # SIGN IN ===== Begin
90 # Clear
91 def signin_admin(datas):
92     try:
93         # Checking Request Body ----- Start
94         if datas == None:
95             return invalid_params()
96
97         requiredData = ["email", "password"]
98         for req in requiredData:
99             if req not in datas:
100                 return parameter_error(f"Missing {req} in Request Body.")
101         # Checking Request Body ----- Finish
102
103         # Initialize Data Request ----- Start
104         email = datas["email"].strip().lower()
105         password = datas["password"].strip()
106         # Initialize Data Request ----- Finish
107
108         # Data Validation ----- Start
109         level = 1
110         checkResult, result, stts = vld_signin(email, password, level)
111         if len(checkResult) != 0:
112             return defined_error(checkResult, "Bad Request", statusCode=stts)
113         # Data Validation ----- Finish
114
115         # Update Data Last Active ----- Start
116         timestamp = int(round(time.time()*1000))
117         query = ADM_UPDATE_ACTIVE_QUERY
118         values = (timestamp, result[0]["id"])
119         DBHelper().save_data(query, values)
120         # Update Data Last Active ----- Finish
121
122         # Log Activity Record ----- Start
123         activity = f"Admin dengan id {result[0]['id']} telah berhasil log in."
124         query = LOG_ADD_QUERY
125         values = (result[0]["id"], level, activity, timestamp, )
126         DBHelper().save_data(query, values)
127         # Log Activity Record ----- Finish
128
129         # Generate File URL ----- Start
130         if len(result) >= 1:
131             detailRequestURL = str(request.url).find('?')
132             if detailRequestURL != -1:
133                 index = detailRequestURL
134                 request.url = request.url[:index]
135         for item in result:
136             item["photos"] = f"{request.url_root}admin/media/{item['photos']}"
137         # Generate File URL ----- Finish
```





```
138
139     # Data Payload ----- Start
140     jwt_payload = {
141         "id" : result[0]["id"],
142         "email" : email,
143         "name" : result[0]["username"],
144         "photos" : result[0]["photos"],
145         "role" : "ADMIN"
146     }
147     # Data Payload ----- Finish
148
149     # Access Token by Email =====
150     access_token = create_access_token(email, additional_claims=jwt_payload)
151
152     # Data Response ----- Start
153     response = {
154         "access_token" : access_token,
155         "role" : "ADMIN"
156     }
157     # Data Response ----- Finish
158
159     # Return Response =====
160     return success_data(response)
161
162     except Exception as e:
163         return bad_request(str(e))
164 # SIGN IN ===== End
```

Gambar 4.5 Kode model sign in admin

2. Python Flask – Front end API

Kode program sign in dari sisi front end terbagi menjadi 2, yaitu kode program untuk menampilkan form dan kode program untuk menjalankan proses sign in. Kode pada gambar 4.6 dibawah ini berfungsi untuk menampilkan form sign in, sedangkan gambar 4.7 menunjukkan bagaimana proses sign in bekerja.

```
50 @auth.get('/')
51 @auth.get('/signin')
52 def sign_in():
53     return render_template(
54         title=TITLE_SIGNIN,
55         template_name_or_list='sign_in.html'
56     )
```

Gambar 4.6. Source code yang berfungsi untuk render form Sign in



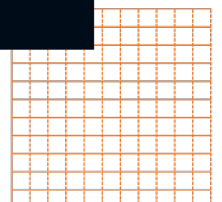


```
91 @auth.post('/signin')
92 def signin_proccess(is_admin=False): # Clear
93     # mengambil data masukkan
94     dataInput = request.form.to_dict()
95
96     # Membuat URL berdasarkan role user
97     url = app.config['BE_URL'] + ('/admin/signin' if is_admin else '/user/signin')
98
99     # request permintaan register akun baru ke api
100     payload = json.dumps(dataInput)
101     headers = {'Content-Type' : 'application/json'}
102     response = requests.request (
103         method='POST',
104         url=url,
105         headers=headers,
106         data=payload
107     )
108
109     # jika respon api tidak berhasil
110     if(response.status_code == 499):
111         flash (
112             message=response.json().get('message'),
113             category='danger'
114         )
115         return redirect(url_for('auth.sign_in'))
116
117     if(response.status_code != 200):
118         if(is_admin==False):
119             return signin_proccess(is_admin=True)
120         flash (
121             message='Username atau Password salah',
122             category='danger'
123         )
124         return redirect(url_for('auth.sign_in'))
125
126     # jika berhasil, arahkan ke halaman dashboard
127     else:
128         session['user'] = response.json().get('data')
129         return redirect(url_for('auth.sign_in'))
130
```

Gambar 4.7. Source code proses sign in

3. HTML

```
27 <form action="{{ url_for('auth.signin_proccess') }}" method="POST">
28     <!-- ===== -->
29     <!-- Alert -->
30     {% with messages = get_flashed_messages(with_categories=true) %}
31     {% include "components/alert.html" %}
32     {% endwith %}
33     <!-- End Alert -->
34     <!-- ===== -->
35
36     <div class="input-group custom">
37         <input type="text" id="email" name="email" class="form-control form-control-lg" placeholder="U
38         <div class="input-group-append custom">
39             <span class="input-group-text"><i class="icon-copy dw dw-user1"></i></span>
40         </div>
41     </div>
42     <div class="input-group custom">
43         <input type="password" id="password" name="password" class="form-control form-control-lg" plac
44         <div class="input-group-append custom">
45             <span class="input-group-text"><i class="dw dw-padlock1"></i></span>
46         </div>
47     </div>
```

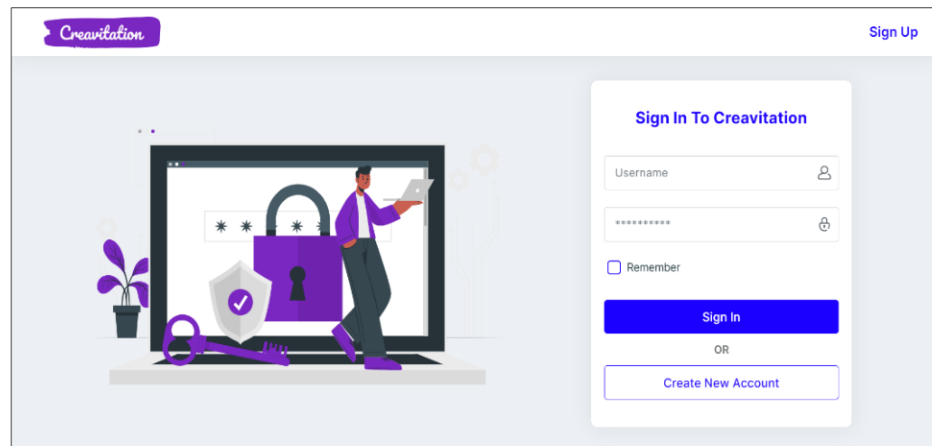




```
48 <div class="row">
49   <div class="col-sm-12">
50     <div class="input-group mb-0">
51       <button class="btn btn-primary btn-lg btn-block">Sign In</button>
52     </div>
53     <div class="font-16 weight-600 pt-10 pb-10 text-center" data-color="#707373">OR</div>
54     <div class="input-group mb-0">
55       <a class="btn btn-outline-primary btn-lg btn-block" href="{{ url_for('auth.sign_up') }}">
56     </div>
57   </div>
58 </div>
59 </form>
```

Gambar 4.8. Kode HTML form sign in

4. Hasil



Gambar 4.9. Halaman Sign In

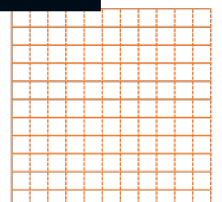
4.3. Halaman Sign Up

Dibawah ini merupakan baris kode untuk menampilkan halaman Sign Up aplikasi web Creavitation yang berfungsi untuk mendaftarkan akun pengguna baru.

1. Python Flask – Back end API

```
50 # Insert Data ----- Start
51 passEncrypt = hashPassword(password)
52 timestamp = int(round(time.time()*1000))
53 query = USR_ADD_QUERY
54 values = (firstName, email, passEncrypt, timestamp, timestamp, timestamp)
55 resReturn = DBHelper().save_return(query, values)
56 if resReturn == None:
57     return defined_error("Gagal menyimpan data.", "Bad Request", 400)
58 # Insert Data ----- Finish
59
60 # Insert Profile ----- Start
61 try:
62     data = {
63         "user_id": resReturn,
64         "level": 2, # 1 = Admin, 2 = User
65         "first_name": firstName,
66         "middle_name": middleName,
67         "last_name": lastName,
68         "phone": phone
69     }
70     profile = ProfileModels.create_profile(data)
71 except Exception as e:
72     return bad_request(str(e))
73 # Insert Profile ----- Finish
```

Gambar 4.10. Kode model sign up (resregistrasi)





2. Python Flask – Front end API

```
42 @auth.get('/signup')
43 def sign_up():
44     return render_template(
45         title=TITLE_SIGNUP,
46         template_name_or_list='sign_up.html'
47     )
```

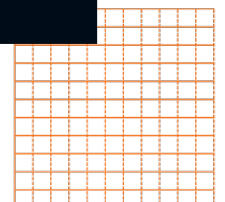
Gambar 4.11. Source code yang berfungsi untuk render form sign up

```
59 @auth.post('/signup')
60 def signup_proccess():
61     # mengambil data masukan
62     dataInput = request.form.to_dict()
63
64     # request permintaan register akun baru ke api
65     payload = json.dumps(dataInput)
66     headers = {'Content-Type' : 'application/json'}
67     response = requests.request (
68         method='POST',
69         url=app.config['BE_URL'] + '/user/',
70         headers=headers,
71         data=payload
72     )
```

Gambar 4.12. Source code proses sign up

3. HTML

```
54 <form action="{ url_for('auth.signup_proccess') }" method="post">
55     <!-- ===== -->
56     <!-- Alert -->
57     {% with messages = get_flashed_messages(with_categories=true) %}
58     {% include "components/alert.html" %}
59     {% endwith %}
60     <!-- End Alert -->
61     <!-- ===== -->
62
63     <div class="form-wrap max-width-600 mx-auto">
64         <div class="form-group row">
65             <label class="col-sm-4 col-form-label">First Name*</label>
66             <div class="col-sm-8">
67                 <input type="text" class="form-control">
68             </div>
69         </div>
70         <div class="form-group row">
71             <label class="col-sm-4 col-form-label">Middle Name</label>
72             <div class="col-sm-8">
73                 <input type="text" class="form-control">
74             </div>
75         </div>
76         <div class="form-group row">
77             <label class="col-sm-4 col-form-label">Last Name</label>
78             <div class="col-sm-8">
79                 <input type="text" class="form-control">
80             </div>
81     </div>
```

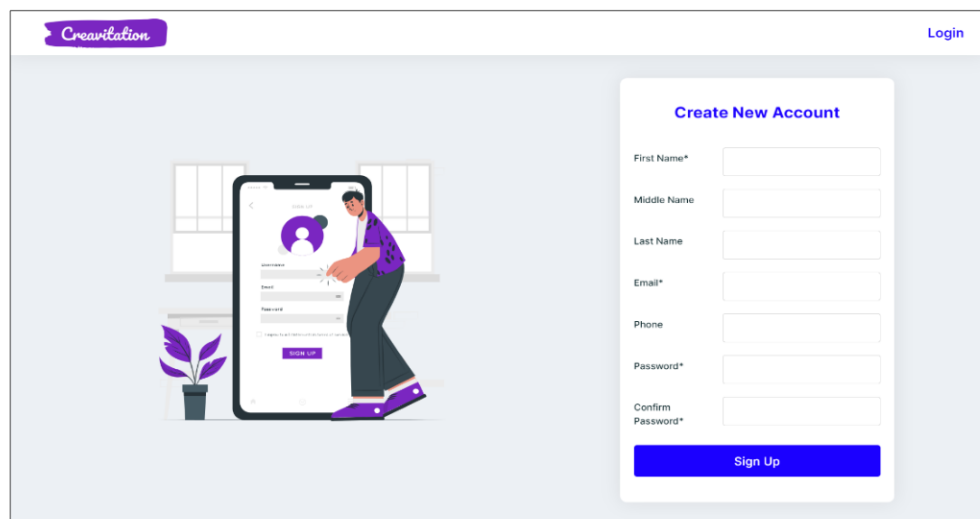




```
82 <div class="form-group row">
83   <label class="col-sm-4 col-form-label">Email*</label>
84   <div class="col-sm-8">
85     <input type="email" class="form-control">
86   </div>
87 </div>
88 <div class="form-group row">
89   <label class="col-sm-4 col-form-label">Phone</label>
90   <div class="col-sm-8">
91     <input type="text" class="form-control">
92   </div>
93 </div>
94 <div class="form-group row">
95   <label class="col-sm-4 col-form-label">Password*</label>
96   <div class="col-sm-8">
97     <input type="password" class="form-control">
98   </div>
99 </div>
100 <div class="form-group row">
101   <label class="col-sm-4 col-form-label">Confirm Password*</label>
102   <div class="col-sm-8">
103     <input type="password" class="form-control">
104   </div>
105 </div>
106 </div>
107 <div class="row">
108   <div class="col-sm-12">
109     <div class="input-group mb-0">
110       <button class="btn btn-primary btn-lg btn-block" type="submit">Sign Up</button>
111     </div>
112   </div>
113 </div>
114 </form>
```

Gambar 4.13. Kode HTML form sign up

4. Hasil



Gambar 4.14. Halaman Sign Up





4.4. Halaman Dashboard

Berikut ini adalah kode program yang berfungsi untuk menjalankan halaman dashboard admin dan dashboard pengguna.

4.4.1. Dashboard Admin

1. Python Flask – Front end API

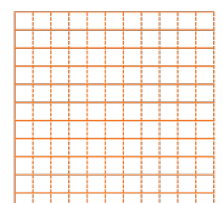
```
31 @dashboard.get('/')
32 def index():
33     data = dict()
34     data['count_invitation'] = get_count_invitation()
35     data['count_guest'] = get_count_guest()
36     data['count_greeting'] = get_count_greeting()
37     data['count_category'] = get_count_category()
38     data['count_template'] = get_count_template()
39     data['count_user'] = get_count_user()
40     data['template_request'] = get_request_template()
41
42     if session['user']['role'] == "ADMIN":
43         return render_template(
44             title=TITLE_DASHBD,
45             data=data,
46             template_name_or_list='dash_admin.html',
47             active='daashboard.adm_index'
48         )
49     elif session['user']['role'] == "USER":
50         return render_template(
51             title=TITLE_DASHBD,
52             data=data,
53             template_name_or_list='dash_user.html',
54             active='dashboard.index'
55         )
```

Gambar 4.15. Source code dashboard

2. HTML

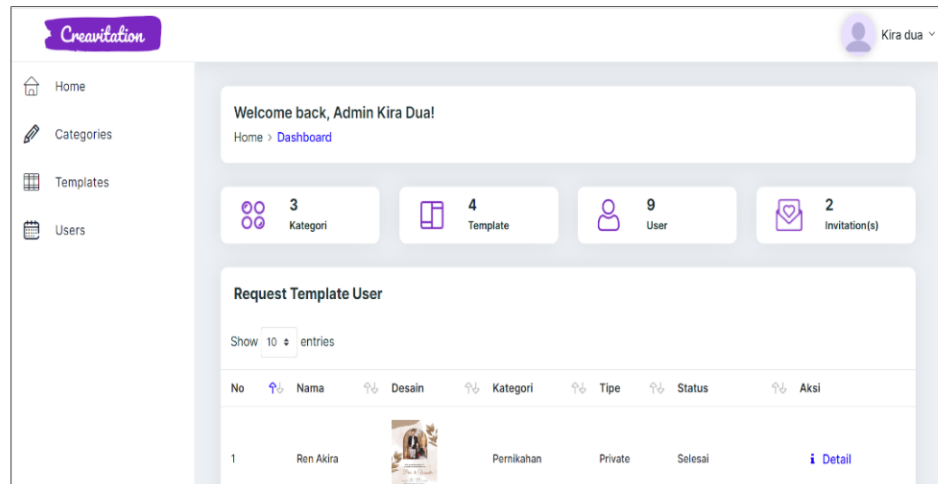
```
33 <!-- ===== -->
34 <!-- MAIN CONTENT - HOME - ADMIN -->
35 <!-- ===== -->
36 <div class="main-container">
37     <div class="xs-pd-20-10 pd-ltr-20">
38         <!-- ===== -->
39         <!-- Title Page & Breadcrumb Section -->
40         <div class="page-header">
41             <div class="row">
42                 <div class="col-md-6 col-sm-12">
43                     <div class="title">
44                         <h4>Welcome back, Admin {{ session['user']['name'].title() }}!</h4>
45                     </div>
46                     <nav aria-label="breadcrumb" role="navigation">
47                         <ol class="breadcrumb">
48                             <li class="breadcrumb-item"><a href="{{ url_for('dashboard.index') }}">Home
49                             <li class="breadcrumb-item active" aria-current="page">Dashboard</li>
50                         </ol>
51                     </nav>
52                 </div>
53             </div>
54         </div>
55         <!-- End Title Page & Breadcrumb -->
56     </div>
57 </div>
```

Gambar 4.16. Kode HTML halaman dashboard





3. Hasil



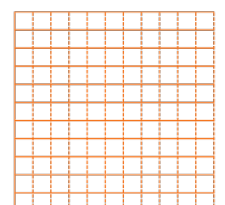
Gambar 4.17. Halaman dashboard admin

4.4.2. Dashboard User

1. HTML

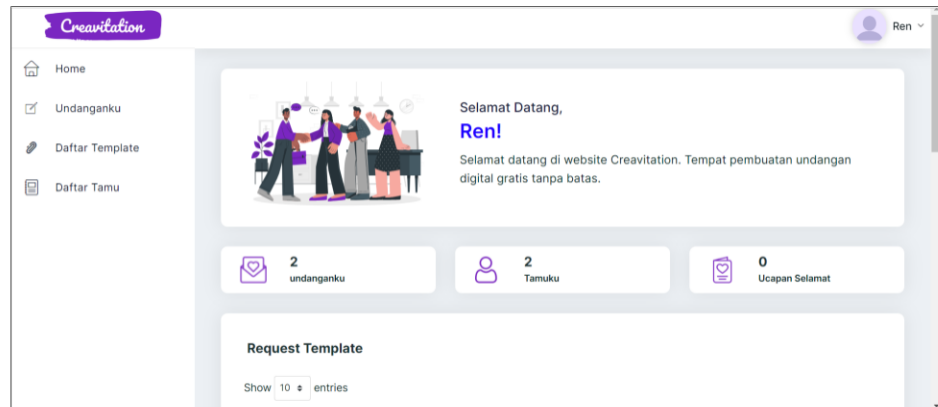
```
33 <!-- ===== -->
34 <!-- MAIN CONTENT - HOME - USER -->
35 <!-- ===== -->
36 <div class="main-container">
37   <div class="pd-ltr-20">
38     <!-- ===== -->
39     <!-- Banner Dashboard Section -->
40     <div class="card-box pd-20 height-100-p mb-30">
41       <div class="row align-items-center">
42         <!-- ===== -->
43         <!-- Image Banner Section -->
44         <div class="col-md-4">
45           
47         <!-- End Image Banner -->
48         <!-- ===== -->
49         <!-- Text Banner Section -->
50         <div class="col-md-8">
51           <h4 class="font-20 weight-500 mb-10 text-capitalize">
52             Selamat Datang,
53             <div class="weight-600 font-30 text-blue">
54               {{ session['user']['name'] }}!
55             </div>
56           </h4>
57           <p class="font-18 max-width-600">
58             Selamat datang di website Creavitation. Tempat pembuatan undangan
59             digital gratis tanpa batas.
60           </p>
61         </div>
62       </div>
63     </div>
64     <!-- End Text Banner -->
65     <!-- ===== -->
66   </div>
67 </div>
68 <!-- End Banner Dashboard -->
69 <!-- ===== -->
```

Gambar 4.18. Kode HTML halaman dashboard user





2. Hasil



Gambar 4.19. Halaman dashboard user

4.5. Halaman Kategori

Dibawah ini merupakan kode program untuk menu kategori acara. Menu ini hanya dapat diakses oleh pengguna level admin. Pada menu ini admin dapat mengelola kategori acara yang tersimpan.

4.5.1. Menambahkan Data Kategori

1. Python Flask – Back end API

```
32 # Data Validation ----- Start
33 category = datas["category"].strip()
34 formatData = datas["format_data"]
35 ctgrCheck = vld_category(category, formatData)
36 if len(ctgrCheck) != 0:
37     return defined_error(ctgrCheck, "Bad Request", 400)
38 # Data Validation ----- Finish
39
40 # Insert Data ----- Start
41 timestamp = int(round(time.time()*1000))
42 formatData = json.dumps(formatData)
43 query = CTGR_ADD_QUERY
44 values = (category, formatData, timestamp, user_id, timestamp, user_id)
45 DBHelper().save_data(query, values)
46 # Insert Data ----- Finish
```

Gambar 4.20. Kode model back end tambah kategori





2. Python Flask – Front end API

```
85     dataInput = {
86         "category" : dataInput["category"],
87         "format_data" : dumps4
88     }
89
90     url = app.config['BE_URL'] + '/category/'
91
92     payload = json.dumps(dataInput)
93     headers = {
94         'Authorization' : f'Bearer {session["user"]["access_token"]}',
95         'Content-Type' : 'application/json'
96     }
97     # return redirect(url_for('management.category.index'))
98     response = requests.request (
99         method='POST',
100         url=url,
101         headers=headers,
102         data=payload
103     )
```

Gambar 4.21. Source code proses tambah kategori

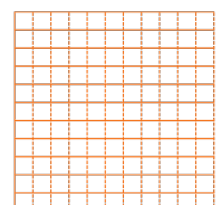
3. HTML

```
91 <form action="{{ url_for('management.category.add_process') }}" method="POST">
92 <div class="modal-body" id="dynamic-form">
93 <div class="form-group row">
94 <label class="col-sm-12 col-md-2 col-form-label">Category</label>
95 <div class="col-sm-12 col-md-10">
96 <input class="form-control" type="text" id="categoryNameInput" name="category" placeholder="Type Here" />
97 </div>
98 </div>
99 <div class="form-group row" id="form-group-1">
100 <label class="col-sm-12 col-md-2 col-form-label" for="input-fd-1">Format Data</label>
101 <div class="col-sm-12 col-md-4">
102 <input class="form-control" placeholder="Type Here" type="text" name="format-data-1" id="input-fd-1" />
103 </div>
104 <div class="col-sm-12 col-md-3">
105 <label class="col-sm-6 col-form-label" for="cb-fd-1" id="label-cb-fd">Required</label>
106 <input type="checkbox" checked="checked" class="form-control switch-btn" data-size="small" data-cb-fd="cb-fd-1" />
107 </div>
108 <div class="col-md-3 col-sm-12">
109 <button type="button" class="btn btn-outline-danger" onclick="removeFormField(1)">Delete</button>
110 <button type="button" class="btn btn-outline-success" id="rm-format-data" onclick="addFormatData()">Add</button>
111 </div>
112 </div>
113 </div>
114 <div class="modal-footer">
115 <a href="{{ url_for('management.category.index') }}" type="button" class="btn btn-outline-danger">Cancel</a>
116 <button type="submit" class="btn btn-outline-primary">Save</button>
117 </div>
118 </form>
```

Gambar 4.22. Kode HTML form tambah kategori

4. Hasil

Gambar 4.23. Tampilan form tambah data kategori





4.5.2. Menampilkan Data Kategori

1. Python Flask – Back end API

```
66 # Checking Data ----- Start
67 query = CTGR_GET_ALL_QUERY
68 result = DBHelper().execute(query)
69 if len(result) < 1 or result is None:
70     return not_found("Data kategori tidak dapat ditemukan.")
71 # Checking Data ----- Finish
72
73 # Response Data ----- Start
74 response = []
75 for rsl in result:
76     createdAt = split_date_time(datetime.fromtimestamp(rsl['created_at']/1000))
77     data = {
78         "category_id" : rsl["id"],
79         "category" : rsl["category"],
80         "format_data" : json.loads(rsl["format_data"]),
81         "created_at": createdAt
82     }
83     response.append(data)
84 # Response Data ----- Finish
85
86 # Return Response =====
87 return success_data(response)
```

Gambar 4.24. Source code back end untuk menampilkan kategori

2. Python Flask – Front end API

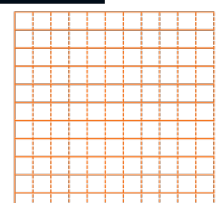
```
17 @mngcategory.get('/')
18 def index():
19     data = get_category()
20     return render_template(
21         title=TITLE_DASHBD,
22         template_name_or_list='category.html',
23         data=data
24     )
```

Gambar 4.25. Source code menu kategori

3. HTML

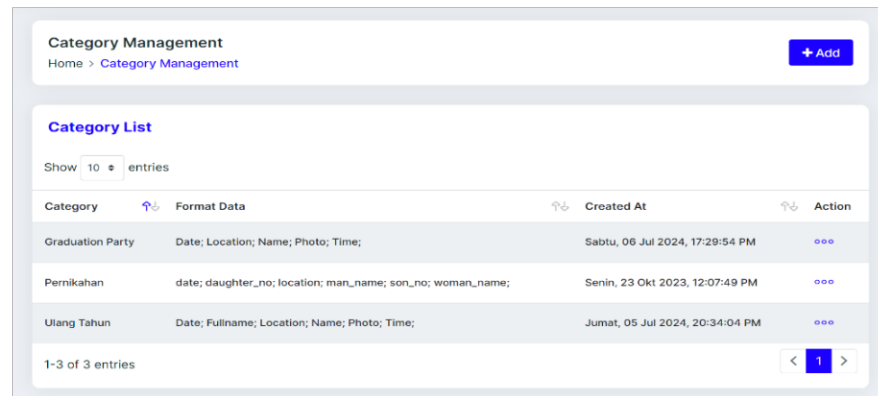
```
97 <div class="pd-20">
98     <h4 class="text-blue h4">Category List</h4>
99 </div>
100 <!-- End Table Title -->
101 <!-- ===== -->
102
103 <!-- ===== -->
104 <!-- Table Content -->
105 <div class="pb-20">
106     <table class="data-table table stripe hover nowrap">
107         <!-- ===== -->
108         <!-- Heading Table Column -->
109 > <thead>...
110 </thead>
111 <!-- End Heading Table Column -->
112 <!-- ===== -->
113
114 <!-- ===== -->
115 <!-- Body Table -->
116 <tbody>
117     {% for item in data %}
118     <!-- ===== -->
119     <!-- Template Popular - Row Box Data Section -->
120 > <tr>...
121 </tr>
122     <!-- End Template Popular - Row Box Data -->
123     <!-- ===== -->
124     {% endfor %}
125 </tbody>
126 <!-- End Body Table -->
127 <!-- ===== -->
128 </table>
129 </div>
```

Gambar 4.26. Kode HTML table daftar kategori





4. Hasil



Gambar 4.27. Halaman menu kategori

4.6. Halaman Daftar Template

Menu daftar template dapat diakses oleh pengguna level admin maupun level user. Akan tetapi pengguna level admin memiliki akses lebih seperti menambahkan, mengubah atau menghapus data template. Sedangkan level user hanya memiliki akses untuk melihat daftar template dan mengirimkan permintaan desain template yang diharapkan. Berikut adalah kode program untuk menambahkan data dan menampilkan data pada menu daftar template.

4.6.1. Menampilkan Data Template

1. Python Flask – Back end API

```
113 # Checking Data ----- Start
114 query = TPLT_GET_ALL_QUERY
115 result = DBHelper().execute(query)
116 if len(result) < 1:
117     return not_found("Data template tidak dapat ditemukan.")
118 # Checking Data ----- Finish
119
120 # Get Data Category ----- Start
121 query = CTGR_GET_ALL_QUERY
122 resultCtgr = DBHelper().execute(query)
123 if len(resultCtgr) < 1:
124     return not_found("Data kategori tidak dapat ditemukan.")
125 # Get Data Category ----- Finish
126
127 # Join Data ----- Start
128 for template in result:
129     for category in resultCtgr:
130         if template["category_id"] == category["id"]:
131             template["category"] = category["category"]
132 # Join Data ----- Finish
```

Gambar 4.28. Source code back end untuk menampilkan daftar template





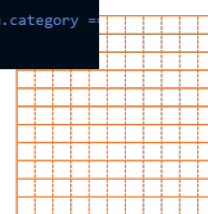
2. Python Flask – Front end API

```
17 @mngtemplate.get('/')
18 def index():
19     data = dict()
20     data['list_category'] = get_category()
21     data['list_template'] = get_template()
22     data['selected'] = 0
23
24
25     return render_template(
26         title=TITLE_DASHBD,
27         data=data,
28         template_name_or_list='template.html'
29     )
```

Gambar 4.29. Source code menu daftar template

3. HTML

```
144 <!-- All Data -->
145 <div class="tab-pane fade show active" id="semuaKategori" role="tabpanel">
146     <div class="pd-20">
147         {# If Template Null #} {% if data['list_template'] == None %}
148         <!-- ===== -->
149         <!-- Template Null Section -->
150 > <div class="tab-pane fade show active" id="invitationTabModal" role="tabpanel">...
151     </div>
152     <!-- End Template Null -->
153     <!-- ===== -->
154     {# If template already #} {% else %}
155     <!-- ===== -->
156     <!-- Template Ready Section -->
157     <div class="row clearfix">
158         {# Looping all data template #} {% for item in data['list_template'] %}
159         <div class="col-lg-3 col-md-6 col-sm-12 mb-30">
160             <div class="da-card">
161                 <div class="da-card-photo">
162                     
163                     <div class="da-overlay da-slide-left">
164                         <div class="da-social">
165                             <ul class="clearfix">
166                                 {# if session['user']['role'] == 'ADMIN' %}
167                                 <li><a href="{{ url_for('show.show_template', temId=item['templ
168                                 <li><a href="{{ url_for('management.template.edit') }}"><i clas
169                                 <li><a href="{{ url_for('management.template.delete', temId=item
170                                 {# else %}
171                                 <li><a href="{{ url_for('show.show_template', temId=item['templ
172                                 <li><a href="{{ url_for('management.invitation.add', temId=item
173                                 {# endif %}
174                             </ul>
175                         </div>
176                     </div>
177                 </div>
178                 <div class="da-card-content">
179                     <h5 class="h5 mb-10">{{ item['title'] }}</h5>
180                     <p class="mb-0">{{ item['category'] }}</p>
181                 </div>
182             </div>
183         </div>
184         {# endfor %}
185     </div>
186     <!-- End Template Ready -->
187     <!-- ===== -->
188     {# endif %}
189 </div>
190 <!-- All Data End -->
191
192 <!-- Specific Data -->
193 {# If Template Null #} {% if data['list_template'] != None %}
194 {# Looping data popular #} {% for items in data['list_category'] %}
195 <div class="tab-pane fade" id="{{ items['category'].replace(' ', '') }}" role="tabpanel">
196     <div class="pd-20">
197         <div class="row clearfix">
198             {# Looping all data template #} {% for item in data['list_template'] if item.category ==
199             <!-- ===== -->
200             <!-- Template ready Section -->
```

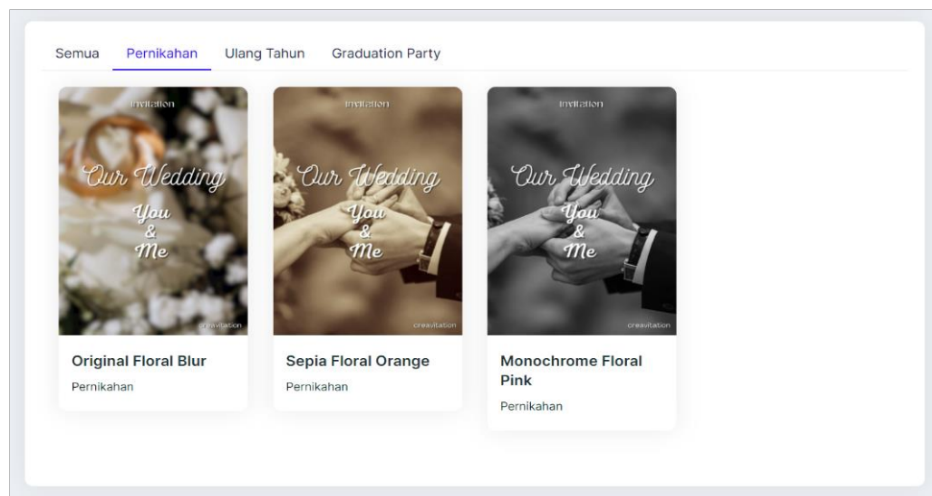




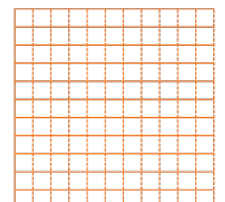
```
211 <div class="col-lg-3 col-md-6 col-sm-12 mb-30">
212 <div class="da-card">
213 <div class="da-card-photo">
214 
215 <div class="da-overlay da-slide-left">
216 <div class="da-social">
217 <ul class="clearfix">
218 <li>{% if session['user']['role'] == 'ADMIN' %}</li>
219 <li><a href="{{ url_for('show.show_template', temId=item['templid']) }}"></li>
220 <li><a href="{{ url_for('management.template.edit', temId=item['templid']) }}"></li>
221 <li><a href="{{ url_for('management.template.delete', temId=item['templid']) }}"></li>
222 <li>{% else %}</li>
223 <li><a href="{{ url_for('show.show_template', temId=item['templid']) }}"></li>
224 <li><a href="{{ url_for('management.invitation.add', id=item['templid']) }}"></li>
225 </ul>
226 </div>
227 </div>
228 </div>
229 </div>
230 <div class="da-card-content">
231 <h5 class="h5 mb-10">{{ item['title'] }}</h5>
232 <p class="mb-0">{{ item['category'] }}</p>
233 </div>
234 </div>
235 </div>
236 <!-- End Template ready -->
237 <!-- =====>
238
239 {% else %} (# if data template on this category is null #)
240 <!-- =====>
241 <!-- Template Null Section -->
242 <div class="col pd-20">...
243 </div>
244 <!-- End Template Null -->
245 <!-- =====>
246 {% endfor %}
247 </div>
248 </div>
249 </div>
250 </div>
251 </div>
252 </div>
253 </div>
254 </div>
255 </div>
256 </div>
257 </div>
258 </div>
```

Gambar.4.30. Kode HTML untuk daftar template

4. Hasil



Gambar 4.31. Halaman daftar template





4.6.2. Menambahkan Data Template

1. Python Flask – Back end API

```
52 # Saving File ----- Start
53 # Thumbnail
54 thumbFileName = secure_filename(time.strftime("%Y-%m-%d %H:%M:%S")+ "_" +randomNumber+"_thumbnai
55 thumbPath = os.path.join(app.config['TEMPLATE_THUMBNAIL_PHOTOS'], thumbFileName)
56 saving_image(thumbnail, thumbPath)
57 # CSS
58 cssFileName = secure_filename(time.strftime("%Y-%m-%d %H:%M:%S")+ "_" +randomNumber+"_style.css"
59 cssPath = os.path.join(app.config['TEMPLATE_CSS_FILE'], cssFileName)
60 saving_file(css, cssPath)
61 # JS
62 jsFileName = ""
63 if js != "":
64     jsFileName = secure_filename(time.strftime("%Y-%m-%d %H:%M:%S")+ "_" +randomNumber+"_script.
65     jsPath = os.path.join(app.config['TEMPLATE_JS_FILE'], jsFileName)
66     saving_file(js, jsPath)
67 # Wallpaper 1
68 wallpaperFileName1 = secure_filename(time.strftime("%Y-%m-%d %H:%M:%S")+ "_" +randomNumber+"_wallpap
69 wallpaperPath = os.path.join(app.config['TEMPLATE_WALLPAPER_PHOTOS'], wallpaperFileName1)
70 saving_image(wallpaper1, wallpaperPath)
71 # wallpaper 2
72 wallpaperFileName2 = ""
73 if wallpaper2 != "":
74     wallpaperFileName2 = secure_filename(time.strftime("%Y-%m-%d %H:%M:%S")+ "_" +randomNumber+"_wal
75     wallpaperPath = os.path.join(app.config['TEMPLATE_WALLPAPER_PHOTOS'], wallpaperFileName2)
76     saving_image(wallpaper2, wallpaperPath)
77 # Saving File ----- Finish
78
79 # Insert Data ----- Start
80 timestamp = int(round(time.time()*1000))
81 query = TMPLT_ADD_QUERY
82 values = (title, thumbFileName, cssFileName, jsFileName, wallpaperFileName1, wallpaperFileName2, ctgr_
83 DBHelper().save_data(query, values)
84 # Insert Data ----- Finish
```

Gambar 4.32. Source code back end untuk menyimpan data template

2. Python Flask – Front end API

```
91 dataInput = {
92     "category_id" : dataInput["category"],
93     "thumbnail" : thum_base64,
94     "css_file" : css_base64,
95     "js_file" : js_base64,
96     "wallpaper_1" : wall1_base64,
97     "wallpaper_2" : wall2_base64,
98     "title" : dataInput["title"]
99 }
100
101
102 url = app.config['BE_URL'] + '/template/'
103
104 payload = json.dumps(dataInput)
105 headers = {
106     'Authorization' : f'Bearer {session["user"]["access_token"]}',
107     'Content-Type' : 'application/json'
108 }
109
110 response = requests.request (
111     method='POST',
112     url=url,
113     headers=headers,
114     data=payload
115 )
```

Gambar 4.33. Source code proses tambah data template





3. HTML

```
93 <form action="{{ url_for('management.template.add_process') }}" method="post" enctype="multipart/form-
94 <div class="modal-body">
95 <!-- title -->
96 <div class="form-group row">
97 <label class="col-sm-12 col-md-2 col-form-label">Title</label>
98 <div class="col-sm-12 col-md-10">
99 <input class="form-control" type="text" placeholder="Type a title" name="title" id="tit
100 </div>
101 </div>
102 <!-- thumbnail -->
103 <div class="form-group row">
104 <label class="col-sm-12 col-md-2 col-form-label">Thumbnail</label>
105 <div class="col-sm-12 col-md-10">
106 <input type="file" class="form-control-file form-control height-auto" accept=".jpg,.jpe
107 </div>
108 </div>
109 <!-- css -->
110 <div class="form-group row">
111 <label class="col-sm-12 col-md-2 col-form-label">CSS file</label>
112 <div class="col-sm-12 col-md-10">
113 <input type="file" class="form-control-file form-control height-auto" accept=".css" na
114 </div>
115 </div>
116 <!-- js -->
117 <div class="form-group row">
118 <label class="col-sm-12 col-md-2 col-form-label">Javascript file</label>
119 <div class="col-sm-12 col-md-10">
120 <input type="file" class="form-control-file form-control height-auto" accept=".js" nam
121 </div>
122 </div>
123 <!-- wallpaper -->
124 <div class="form-group row">
125 <label class="col-sm-12 col-md-2 col-form-label">Wallpaper 1</label>
126 <div class="col-sm-12 col-md-10">
127 <input type="file" class="form-control-file form-control height-auto" accept=".jpg,.jpe
128 </div>
129 </div>
130 <!-- wallpaper -->
131 <div class="form-group row">
132 <label class="col-sm-12 col-md-2 col-form-label">Wallpaper 2</label>
133 <div class="col-sm-12 col-md-10">
134 <input type="file" class="form-control-file form-control height-auto" accept=".jpg,.jpe
135 </div>
136 </div>
137 <!-- category -->
138 <div class="form-group row">
139 <label class="col-sm-12 col-md-2 col-form-label">Category</label>
140 <div class="col-sm-12 col-md-10">
141 <select class="custom-select col-12" name="category" id="category">
142 <option selected="None">Choose...</option>
143 {% for item in data %}
144 <option value="{{ item['category_id'] }}">{{ item['category'] }}</option>
145 {% endfor %}
146 </select>
147 </div>
148 </div>
149 </div>
150 <div class="modal-footer">
151 <a href="{{ url_for('management.template.index') }}" type="button" class="btn btn-outline-dange
152 <button type="submit" id="categoryAddBtn" class="btn btn-outline-primary">Save</button>
153 </div>
154 </form>
```

Gambar 4.34. Kode HTML untuk form tambah data template





4. Hasil

Gambar 4.35. Form tambah data template

4.6.3. Mengirimkan Permintaan Desain Template

1. Python Flask – Back end API

```
51 # Saving File ----- Start
52 randomNumber = str(random_number(5))
53 # Design Photos
54 designFileName = secure_filename(time.strftime("%Y-%m-%d %H:%M:%S")+"_"+randomNumber+"_desi
55 designPath = os.path.join(app.config['TEMPLATE_REQUEST_DESIGN'], designFileName)
56 saving_image(tempDesign, designPath)
57 # Saving File ----- Finish
58
59 # Insert Data ----- Start
60 status = 0 # 0 = waiting | 1 = checked | 2 = acc | decline
61 timestamp = int(round(time.time()*1000))
62 query = REQ_ADD_QUERY
63 values = (user_id, acclevel, catId, designFileName, description, deadline, types, status, tin
64 DBHelper().save_data(query, values)
65 # Insert Data ----- Finish
```

Gambar 4.36. Source code back end untuk menyimpan request desain template

2. Python Flask – Front end API

```
308 dataInput = {
309     "category_id" : dataInput["category"],
310     "template_design" : base64_string,
311     "description" : dataInput["description"],
312     "deadline" : dataInput["deadline"],
313     "type" : dataInput["selTypeTemp"]
314 }
315
316
317 url = app.config['BE_URL'] + '/template/request'
318
319 payload = json.dumps(dataInput)
320 headers = {
321     'Authorization' : f'Bearer {session["user"]["access_token"]}',
322     'Content-Type' : 'application/json'
323 }
324
325 response = requests.request (
326     method='POST',
327     url=url,
328     headers=headers,
329     data=payload
330 )
```

Gambar 4.37. Source code proses request template





3. HTML

```
90 <form action="{{ url_for('management.template.request_create_proccess') }}" method="POST" enctype="mult
91 <div class="modal-body">
92 <!-- Category Input -->
93 <div class="form-group row">
94 <label class="col-sm-12 col-md-2 col-form-label">Kategori*</label>
95 <div class="col-sm-12 col-md-10">
96 <select class="custom-select col-12" name="category">
97 <option selected="" value="0">Pilih...</option>
98 {% for item in data['list_category'] %}
99 <option value="{{ item['category_id'] }}">{{ item['category'] }}</option>
100 {% endfor %}
101 </select>
102 </div>
103 </div>
104
105 <!-- Photo Design -->
106 <div class="form-group row">
107 <label class="col-sm-12 col-md-2 col-form-label">Desain Template*</label>
108 <div class="col-sm-12 col-md-10">
109 <input type="file" class="form-control-file form-control height-auto" accept=".jpg,.jpe
110 </div>
111 </div>
112
113 <!-- Deadline -->
114 <div class="form-group row">
115 <label class="col-sm-12 col-md-2 col-form-label">Batas Akhir*</label>
116 <div class="col-sm-12 col-md-10">
117 <input class="form-control date-picker" placeholder="Pilih tanggal" type="text" name="d
118 </div>
119 </div>
120
121 <!-- Design Type -->
122 <div class="form-group row">
123 <label class="col-sm-12 col-md-2 col-form-label">Pilih Tipe*</label>
124 <div class="col-sm-12 col-md-10">
125 <div class="row pt-2">
126 <div class="custom-control custom-radio mx-2">
127 <input type="radio" checked id="privateRadio" name="selTypeTemp" class="custom-
128 <label class="custom-control-label" for="privateRadio">Privasi</label>
129 </div>
130 <div class="custom-control custom-radio mx-2">
131 <input type="radio" id="publicRadio" name="selTypeTemp" class="custom-control-i
132 <label class="custom-control-label" for="publicRadio">Publik</label>
133 </div>
134 </div>
135 </div>
136 </div>
137
138 <!-- Description -->
139 <div class="form-group row">
140 <label class="col-sm-12 col-md-2 col-form-label">Deskripsi</label>
141 <div class="col-sm-12 col-md-10">
142 <textarea class="form-control" name="description"></textarea>
143 </div>
144 </div>
145 </div>
146
147 <!-- Button -->
148 <div class="modal-footer">
149 <a href="{{ url_for('management.template.index') }}" type="button" class="btn btn-outline-dange
150 <button type="submit" class="btn btn-outline-primary">Simpan</button>
151 </div>
152 </form>
```

Gambar 4.38. Kode HTML untuk form request template





4. Hasil

Gambar 4.39. Form request desain template

4.7. Halaman Undanganku

Halaman undanganku merupakan halaman web yang hanya bisa diakses melalui pengguna level user. Pengguna dapat menambahkan undangan digital melalui halaman ini. Berikut ini merupakan kode program untuk halaman ini.

4.7.1. Menambahkan Data Undangan

1. Python Flask – Back end API

```
52 # Saving File ----- Start
53 # wallpaper
54 wpFileName = ""
55 # if len(personalData) > 0
56 if personalData['womans_photo'] != "":
57     wpFileName = secure_filename(time.strftime("%Y-%m-%d %H:%M:%S")+"_"+invCode+"_woman_p
58     wpPath = os.path.join(app.config['USER_INVITATION_FILE'], wpFileName)
59     saving_image(personalData['womans_photo'], wpPath)
60     personalData['womans_photo'] = wpFileName
61 if personalData['mans_photo'] != "":
62     mpFileName = secure_filename(time.strftime("%Y-%m-%d %H:%M:%S")+"_"+invCode+"_man_phot
63     mpPath = os.path.join(app.config['USER_INVITATION_FILE'], mpFileName)
64     saving_image(personalData['mans_photo'], mpPath)
65     personalData['mans_photo'] = mpFileName
66 # Saving File ----- Finish
67
68 # Insert Data ----- Start
69 timestamp = int(round(time.time()*1000))
70 titlelink = title.replace(' ', '-')
71 invlink = titlelink
72 personalData = json.dumps(personalData)
73 invSett = json.dumps(invSett)
74 query = INV_ADD_QUERY
75 values = (accLevel, user_id, categoryId, templateId, title, personalData, invSett, invCod
76 resReturn = DBHelper().save_return(query, values)
77 # Insert Data ----- Finish
```

Gambar 4.40. Source code back end untuk membuat undangan





2. Python Flask – Front end API

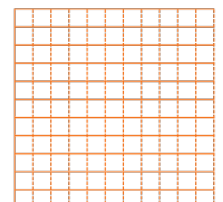
```
115 # Payload
116 payload = json.dumps(dataInput)
117
118 # Set URL
119 url = app.config['BE_URL'] + '/invitation/'
120
121 # Set Header
122 headers = {
123     'Authorization' : f'Bearer {session["user"]["access_token"]}',
124     'Content-Type' : 'application/json'
125 }
126 response = requests.request (
127     method="POST",
128     url=url,
129     headers=headers,
130     data=payload
131 )
132 data = response.json().get('data')
```

Gambar 4.41. Source code proses pembuatan undangan

3. HTML

```
76 <form action="{ url_for('management.invitation.add_process') }" method="post" enctype="multipart/form-
77 <div class="pd-20 card-box mb-30">
78 <div class="clearfix">
79 <div class="pull-left">
80 <h4 class="text-blue h4">Data Undangan</h4>
81 </div>
82 </div>
83 <div class="form-group row">
84 <label class="col-sm-12 col-md-2 col-form-label">Kategori</label>
85 <div class="col-sm-12 col-md-10">...
86 </div>
87 </div>
88 <div class="form-group row">
89 <label class="col-sm-12 col-md-2 col-form-label">Template</label>
90 <div class="col-sm-12 col-md-10">...
91 </div>
92 </div>
93 <div class="form-group row">
94 <label class="col-sm-12 col-md-2 col-form-label">Judul* </label>
95 <div class="col-sm-12 col-md-10">...
96 </div>
97 </div>
98 </div>
99 <div class="form-group row">
100 <div class="col-sm-12 col-md-10">...
101 </div>
102 </div>
103 </div>
104 {% if data['template']['category'] == "Pernikahan" %}
105 <!-- Data diri mempelai wanita -->
106 <div class="pd-20 card-box mb-30">
107 <div class="clearfix">...
108 </div>
109 <div class="form-group row">...
110 </div>
111 </div>
112 </div>
113 <div class="form-group row">...
114 </div>
115 </div>
116 </div>
117 </div>
118 </div>
119 </div>
120 </div>
121 </div>
122 </div>
123 </div>
124 </div>
125 </div>
126 </div>
127 </div>
128 </div>
129 </div>
130 </div>
131 </div>
132 </div>
133 </div>
134 </div>
135 </div>
136 </div>
137 </div>
138 </div>
139 </div>
140 </div>
141 </div>
142 </div>
143 </div>
144 </div>
145 </div>
146 </div>
147 </div>
```

Gambar 4.42. Kode HTML untuk form data undangan





4. Hasil

Data Undangan

Kategori: Pernikahan

Template: Original Floral Blur

Judul*: Tulis disini

Data Pribadi Mempelai Wanita

Nama Lengkap Wanita*: Tulis disini

Nama Panggilan Wanita*: Tulis disini

Foto: Choose File No file chosen

Gambar 4.43. Form pembuatan undangan (contoh: undangan pernikahan)

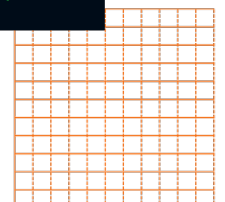
4.8. Halaman Detail Undangan

Halaman detail undangan juga merupakan halaman yang hanya dapat diakses oleh pengguna level user. Hal ini dilakukan untuk memberikan privasi pada pengguna. Pada halaman ini, pengguna disajikan informasi detail undangan, menampilkan undangan digital, menambahkan data tamu baru, dan membagikan undangan digital kepada tamu undangan. Berikut adalah kode program untuk menampilkan dan menjalankan menu detail undangan.

4.8.1. Menampilkan Undangan Digital

1. Python Flask – Back end API

```
361 # Checking Data Invitation ----- Start
362 query = INV_GET_BY_ID_QUERY
363 values = (invId,)
364 result = DBHelper().get_data(query, values)
365 if len(result) < 1 :
366     return not_found(f"Data undangan dengan Id {invId} tidak dapat ditemukan.")
367 # Checking Data Invitation ----- Finish
368
369 # Get Data Category ----- Start
370 invitation = result[0]
371 query = CTGR_GET_BY_ID_QUERY
372 values = (invitation['category_id'], )
373 category = DBHelper().get_data(query, values)
374 if len(category) < 1 :
375     return not_found(f"Data kategori dengan Id {invitation['category_id']} tidak dapat ditemuka
376 # Get Data Category ----- Finish
377
378 # Get Data Template ----- Start
379 query = TMPLT_GET_BY_ID_QUERY
380 values = (invitation['template_id'], )
381 templates = DBHelper().get_data(query, values)
382 if len(templates) < 1 :
383     return not_found(f"Data template dengan Id {invitation['template_id']} tidak dapat ditemuka
384 # Get Data Template ----- Finish
```





Gambar 4.44. Source code untuk menampilkan detail data undangan

2. Python Flask – Front end API

```
30 @show.get('<code>/')
31 def show_invitation(code):
32     data = dict()
33     data['invitation'] = get_detail_invitation()
34     wall1_url = data['invitation']['template_wall']
35     wall2_url = data['invitation']['template_wall2']
36
37     return render_template(
38         title="Creavitation",
39         data = data,
40         background1_url= wall1_url,
41         background2_url= wall2_url,
42         template_name_or_list='base_wedding.html'
43     )
```

Gambar 4.45. Source code untuk render undangan digital

3. HTML

```
22 <section id="hero" class="hero w-100 h-100 p-3 mx-auto text-center d-flex justify-content-center align-3
23     <main>
24         <h4>Kepada Bapak/Ibu/Saudara/i,</h4>
25         <h1>{{ data['invitation']['personal_data']['man_name'].title() }} & {{ data['invitation']['perso
26         <p>Akan melaksanakan resepsi pernikahan dalam:</p>
27         <div class="simply-countdown" data-years="{{ data['invitation']['personal_data']['reception']['y
28         <a href="#home" class="btn btn-lg mt-4 show-invitation" data-codes="{{ data['invitation']['invi
29     </main>
30 </section>
31 <!-- End Hero -->
32
33 <!-- Navbar -->
34 <nav class="navbar navbar-expand-md bg-transparent sticky-top mynav"> ...
35 </nav>
36 <!-- End Navbar -->
37
38 <!-- Home -->
39 <section id="home" class="home" style="background-image: url('{{background2_url}}');">
40     <div class="container">
41         <div class="row justify-content-center">
42             <div class="col-md-8 text-center">
43                 <h2>Acara Pernikahan</h2>
44                 <h3>Diselenggarakan pada {{ data['invitation']['personal_data']['reception']['date'] }}.
45                 <p>Oleh karena itu, dengan segala hormat, kami mengundang Bapak/Ibu/Saudara/i, untuk ha
46             </div>
47         </div>
48     </div>
49 </section>
```

Gambar 4.46. Kode HTML untuk undangan digital (contoh: undangan pernikahan)





4. Hasil



Gambar 4.47. Tampilan undangan digital

4.8.2. Menambahkan Data Tamu

1. Python Flask – Back end API

```
39 # Data Validation ----- Start
40 checkResult, phone = vld_guest(name, address, phone, invCode)
41 if len(checkResult) != 0:
42     return defined_error(checkResult, "Bad Request", 400)
43 # Data Validation ----- Finish
44
45 # Get User ID ----- Start
46 query = INV_CHK_CODE_QUERY
47 values = (invCode,)
48 result = DBHelper().get_data(query, values)
49 > if len(result) < 1: ...
51 # Get User ID ----- End
52
53 # Insert Data ----- Start
54 inv = result[0]
55 timestamp = int(round(time.time()*1000))
56 query = GUEST_ADD_QUERY
57 values = (user_id, accLevel, inv['category_id'], invCode, name, address, phone)
58 DBHelper().save_data(query, values)
59 # Insert Data ----- Finish
```

Gambar 4.48. Source code back end untuk menambahkan data tamu baru





2. Python Flask – Front end API

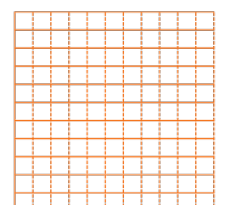
```
75     dataInput = {
76         "invitation_code" : dataInput['inv_code'],
77         "name" : dataInput['name'],
78         "address" : dataInput['address'],
79         "phone" : dataInput['phone']
80     }
81
82     # Set URL
83     url = app.config['BE_URL'] + '/guest/'
84
85     # Set Header
86     headers = {
87         'Authorization' : f'Bearer {session["user"]["access_token"]}',
88         'Content-Type' : 'application/json'
89     }
90
91     # Payload
92     payload = json.dumps(dataInput)
93
94     response = requests.request (
95         method='POST',
96         url=url,
97         headers=headers,
98         data=payload
99     )
```

Gambar 4.49. Source code proses tambah tamu baru

3. HTML

```
77 <form action="{{ url_for('guest.add_process') }}" method="post">
78 <div class="form-group row">
79 > <div class="col-sm-12 col-md-10">...
84 </div>
85 </div>
86
87 <!-- Data tamu -->
88 <div class="pd-20 card-box mb-30">
89 <div class="clearfix">
90 <div class="pull-left">
91 <h4 class="text-blue h4">Tambah Data Tamu</h4>
92 </div>
93 </div>
94 <div class="form-group row">
95 <label class="col-sm-12 col-md-2 col-form-label">Nama* </label>
96 <div class="col-sm-12 col-md-10">
97 <input class="form-control" type="text" placeholder="Tulis nama disini" name="name" id="
98 </div>
99 </div>
100 <div class="form-group row">
101 <label class="col-sm-12 col-md-2 col-form-label">Alamat* </label>
102 <div class="col-sm-12 col-md-10">
103 <input class="form-control" type="text" placeholder="Jl.xxxx" name="address" id="addres
104 </div>
105 </div>
106 <div class="form-group row">
107 <label class="col-sm-12 col-md-2 col-form-label">No Hp* </label>
108 <div class="col-sm-12 col-md-10">
109 <input class="form-control" placeholder="1-(111)-111-1111" type="number" name="phone"
110 </div>
111 </div>
112 </div>
113
114 <div class="pd-20 card-box mb-30 text-right">
115 <a href="{{ url_for('management.invitation.detail', id=data['invitation']['invitation_id'], co
116 <button type="submit" id="invitationAddBtn" class="btn btn-outline-primary">Simpan </button>
117 </div>
118 </form>
```

Gambar 4.50. Kode HTML form tambah tamu baru





4. Hasil

Tambah Data Tamu

Nama*

Alamat*

No Hp*

Gambar 4.51. Form tambah data tamu baru

4.8.3. Membagikan Undangan

1. Python Flask – Front end API

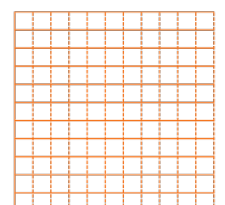
```
280 data['message'] = set_message("")
281 if data['invitation'] != None:
282     data['message'] = set_message(data['invitation']['invitation_link'])
```

Gambar 4.52. Source code untuk setting pesan pembagian undangan

2. Javascript

```
1 $('message-modal').on('show.bs.modal', function (event) {
2     var button = $(event.relatedTarget); // Button that triggered the modal
3     var phone = button.data('phone'); // Extract info from data-* attributes
4     var message = button.data('message'); // Extract info from data-* attributes
5     var modal = $(this);
6     modal.find("#to").val(phone); // Update the modal's input with the phone number
7     modal.find("message").val(message); // Update the modal's input with the phone number
8 });
9
10 function sendToWhatsApp() {
11     // Ambil nomor telepon tujuan
12     const phoneNumber = document.getElementById("to").value; // Ganti dengan nomor tujuan
13     // Ambil pesan dari input form
14     const message = document.getElementById("message").value;
15     // const message = "cek send this message";
16     // Encode pesan untuk URL
17     const encodedMessage = encodeURIComponent(message);
18     // Buat URL whatsapp
19     const whatsappURL = `https://wa.me/${phoneNumber}?text=${encodedMessage}`;
20     // Buka URL di tab/jendela baru
21     window.open(whatsappURL, '_blank');
22 }
```

Gambar 4.53. Kode javascript untuk membagikan undangan via whatsapp





3. HTML

```
366 <div class="login-title">
367   <h2 class="text-center text-primary">Kirim ke WhatsApp</h2>
368 </div>
369 <form>
370   <div class="form-group">
371     <label>Kepada: </label>
372     <input type="text" class="form-control" value="" disabled="" name="to" id="to">
373   </div>
374   <div class="form-group">
375     <label>Pesan: </label>
376     <textarea class="form-control" name="message" id="message"></textarea>
377   </div>
378   <div class="row">
379     <div class="col-sm-12">
380       <div class="input-group mb-0">
381         <a href="#" class="btn btn-outline-primary btn-lg btn-block" onclick="sendToWhatsApp()">
382           </div>
383       </div>
384     </div>
385   </form>
```

Gambar 4.54. Kode HTML form pembagian undangan

4. Hasil

Gambar 4.55. Form pembagian undangan digital via whastapp

