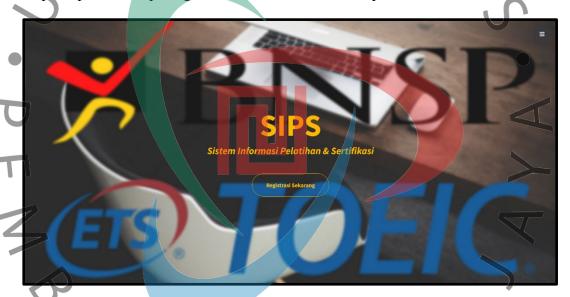
# BAB V HASIL DAN PEMBAHASAN

Pada kegiatan-kegiatan penelitian tentang algoritma kriptografi yang telah dikerjakan oleh penulis, maka perlu dilakukan pembahasan mengenai hasil dari uraian perancangan dan pembahasan tentang uraian dari uji coba.

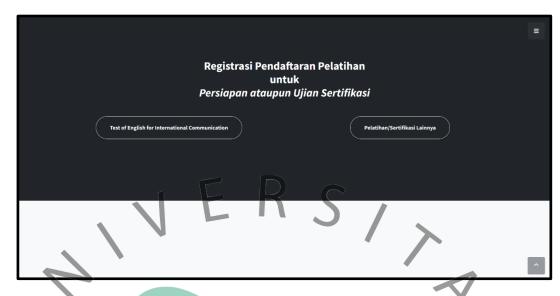
### 5.1. Hasil

Pada bagian ini penulis membahas hasil dari perancangan, di mana pada halaman utama, yaitu halaman *guest* terdapat *menu* untuk registrasi bagi pendaftar yang memiliki kepentingan untuk mendaftar sebagai peserta pelatihan atau sertifikasi. Dapat diperhatikan pada gambar 5.1. halaman utama aplikasi.



Gambar 5.1. Halaman Utama Aplikasi

Pada menu registrasi terdapat dua opsi pendaftaran, di mana salah satu dikhususkan untuk *Test of English for International Communication* (TOEIC), dan opsi lain ditujukan untuk pendaftaran terhadap jenis sertifikat atau pelatihan dari bidang profesi yang berbasis pada program studi, seperti dari Badan Nasional Sertifikasi Profesi (BNSP). Dapat diperhatikan pada gambar 5.2. opsi registrasi pelatihan dan sertifikasi.



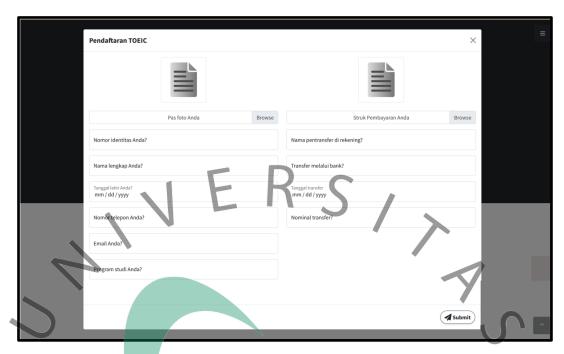
Gambar 5.2. Opsi Registrasi Pelatihan dan Sertifikasi

Untuk opsi *TOEIC* terdapat 12 *field* yang harus diisi oleh pengguna, sedangkan untuk opsi terhadap bidang profesi berbasis pada program studi terdapat 13 *field* yang harus diisi oleh pengguna. Dapat diperhatikan pada tabel 5.1. keterangan *field* pada *form* registrasi.

Tabel 5.1. Keterangan Field pada Form Registrasi

	Tabel 5.1. Keterangan Fi	eld pad	a Form Registrasi
No.	TOEIC	No.	Bidang Profesi
1	Pas foto	1	Nomor identitas
2	Nomor identitas	2	Nama lengkap
3	Nama lengkap	3	Nama instansi
4	Tanggal lahir	4	Pekerjaan
5	Nomor telepon	5	Program studi
6	Email	6	Nomor telepon
7	Program studi	7	Email
8	Bukti pembayaran	8	Bukti pembayaran
9	Nama pentransfer pada bukti	9	Nama pentransfer pada bukti
	pembayaran		pembayaran
10	Nama bank pentransfer	10	Nama bank pentransfer
11	Tanggal transfer	11	Tanggal transfer
12	Nominal transfer	12	Nominal transfer
	GII	13	Nama pelatihan atau sertifikasi yang
			akan diikuti

Pada setiap *field* maka informasi yang diisi merupakan data yang bersifat pribadi dan sensitif, apabila data yang *disubmit* terdapat kesalahan karena *human error* maka hanya *administrator* yang dapat memperbaiki data tersebut. Oleh karena itu, *administrator* akan memeriksa data yang telah *disubmit* oleh peserta apakah *valid* atau *invalid*. Dapat diperhatikan pada gambar 5.3. *TOEIC registration form*.



Gambar 5.3. TOEIC Registrastion Form

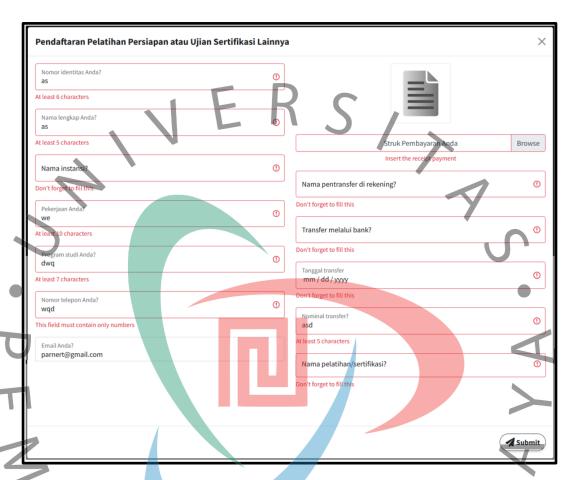
Pada *form* registrasi *online* untuk bidang profesi berbasis pada program studi, dapat diperhatikan pada gambar 5.4. *profession based registration form*.



Gambar 5.4. Profession Based Registration Form

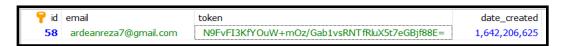
Terdapat sistem validasi yang mana berguna untuk memvalidasi setiap data yang disubmit oleh pengguna dari setiap field yang tertera pada online form tersebut, di mana sistem validasi ini berguna untuk mencegah SQL injection attack, dan untuk

memberikan *rules* tentang kondisi yang diperbolehkan dalam pengisian data pada setiap *field* yang tertera. Hal ini untuk mencegah kesalahan dalam pengisian data. Dapat diperhatikan ilustrasinya pada gambar 5.5. sistem validasi data.



Gambar 5.5. Sistem Validasi Data

Adapun hasil dari data yang berhasil disubmit pada sistem, maka sistem akan langsung memproses pembuatan akun peserta (user account), dan mengenerate token dengan random byte 32 characters menggunakan algoritma base64 encoding, di mana hal ini ditujukan untuk web security dan network security. Setelah akun dan token telah digenerate maka sistem akan mengrimkan pesan autentikasi untuk aktivasi akun melalui email yang ditujukan kepada alamat email peserta yang tertera pada kolom email. Dapat diperhatikan untuk ilustrasi token pada gambar 5.6. token base64 encoding.



Gambar 5.6. Token Base64 Encoding

Di mana token tersebut disematkan dalam link activate (hyperlink) dengan batas waktu expire 24 jam, apabila peserta telat dalam mengaktifkan akun user tersebut maka token yang tersematkan pada hyperlink sudah tidak dapat digunakan, sehingga hanya administrator yang dapat mengaktifkan akus user tersebut, dan token yang disematkan hanya dapat dipakai sekali saja oleh peserta ketika mengaktifkan akun user tersebut, sehingga hal ini dapat mencegah serangan Distributed Denial of Service (DDoS) pada server aplikasi. Dapat diperhatikan pada gambar 5.7. autentikasi akun peserta.

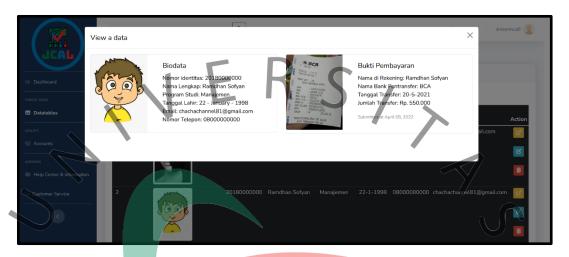


Gambar 5.7. Autentikasi Akun Peserta

Data peserta yang telah *disubmit* pada sistem dapat diakses oleh pengurus pada halaman *administrator*, di mana pada halaman ini data terbagai menjadi 2 *menu*, yaitu data peserta TOEIC dan data peserta berbasis bidang profesi dari program studi. Dapat diperhatikan pada gambar 5.8. halaman *administrator*.

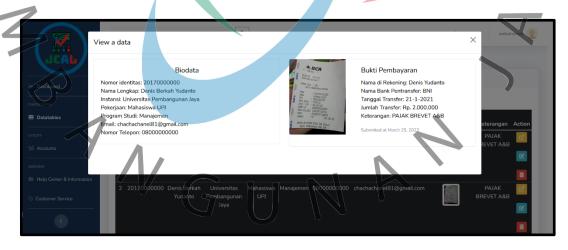


Adapun hasil dari data peserta *TOEIC* yang telah selesai diproses dapat diakses pada halaman *administrator*, data yang ditampilkan merupakan data yang telah *didecipher* sehingga informasi yang ditampilkan berupa *plaintext*. Untuk ilustrasi dapat diperhatikan pada gambar 5.9. data peserta *TOEIC*.



Gambar 5.9. Data Peserta TOEIC

Adapun hasil dari data peserta berbasis bidang profesi yang telah diproses dapat diakses pada halaman *administrator*, data yang ditampilkan merupakan data yang telah *didecipher* sehingga informasi yang ditampilkan berupa *plaintext*. Untuk ilustrasi dapat diperhatikan pada gambar 5.10. data peserta berbasis bidang profesi.



Gambar 5.10. Data Peserta Berbasis Bidang Profesi

Untuk data peserta berupa *plaintext* akan dienkripsi oleh sistem menggunakan algoritma *AES* bertujuan untuk mengubah data tersebut menjadi *ciphertext*, selanjutnya sistem *mengencoding ciphertext* hasil enkripsi menggunakan algoritma *base64* sehingga data yang disimpan dalam database merupakan data dalam bentuk *index base64*. Dalam hal ini untuk *mendechiper* informasi tersebut kembali menjadi *plaintext* maka terdapat dua fase yang harus dilalui, yaitu *decoding* dan *decrypting*. Untuk ilustrasi dapat diperhatikan pada gambar 5.11. *encipher*.

nama	telepon	pekerjaan	transaction
Z1UEn9dQCOoh2TGxehU	GC2zh99RTg8f7HkFfs+3	ms5botc8NqEPOzVXwC8	2fhF4JztwBRfilw3n5NOpi
WL+jVz1lUEEUq/F2iYpY8	BTT33n61PndSlV87pX7r	b37hSPitObJhYhe/yTK+	+rMoQc+XIsBFXHV/sUK
CcJW+X3g8o3onvYdSSc	tSk85Mx31/c3xpgANBF5	mfNzhVdciqz8VL0tFpzhs	plbtVLxB6+XBYkrKfgGE2
pxfuohMuoVbWc00cf9ri5	iqEv8uHwrETRh4d8z976	JC4jR75kyHolQUTFquJA5	Hkg5llcoLOGy4O0OyjAHR
oUu/rEYSbUvVo4A7u1Jm	bZ5oq9UarzIwVkBePqRN	4E/cV/BKrwli6MmATET5	R+HheCnPqN2P4/Cf8HD
rnaQthDUU8BMgE5y60tf	wvjS/TMsg43tB/1kOnr3q	d9NB24QVUXwox8Rj221	YjQx7+W1KTGiU/83D4H
tHY7jX3iHMd0FUc05Z0Kf	kcRFmUusrIsY5yKZRl2dv	2CHeEDAYUxUx6zRy0TQ	b9q8OfN2MltRLHdb32qd
0G0PH+ODaIDjw8qS6LM	bGw6aEYh280YwW4fBG	rh63mW23i+/70ykzvF2R	X3jmoMEvoVSFk5YCSMQ
7scN3u/mVtsQYvq3qZt/Y	XCdHAocHW/gTFUifd5p4	D3wQ0ThKJGI6RpVTxak	K9ioGEau6Et3VsF1GgT29
NdhtcNy9eBRMvdIK7I0R	t7wrXVsNmNx5jzf++0oO	MPR8NRUGJ9v0RxfvyDN	qVhG+o2Qn3x5YfbKjBpx

Gambar 5.11. Encipher

## 5.1.1. Penerapan Algoritma AES dan Base 64

Hasil dari enkripsi (*encipher*) menggunakan algoritma *AES* 256 *bit* memiliki tingkat kerumitan dan sensitivitas yang tinggi, di mana hasil dari enkripsi yang terjadi adalah berupa *ciphertext*, berikut merupakan *ciphertext* "�q��k�\*�|�?���\$82�6f��j" yang mana dalam hal ini proses enkripsi dan dekripsi membutuhkan *cipher key* yang sama (*match*) apabila *cipher key* yang dipakai pada proses enkripsi berbeda (*mismatch*) dengan yang dipakai ketika dilakukan proses dekripsi maka sistem tidak akan dapat mendekripsi pesan tersebut.

Pada proses *encoding* pesan menggunakan algoritma *base64* maka dalam kasus ini penulis menerapkannya untuk *mengencode ciphertext* ke dalam karakter *index base64*, berikut merupakan karakter dari hasil *encode* ke dalam bentuk *index base64* "QXJkZWFuIFJlemEgQW5na2FzYQ=", dengan demikian maka data mudah untuk diolah dan disimpan dalam database. Dari hasil yang terjadi penulis mendapatkan bahwa hasil enkripsi memiliki tingkat sesitivitas yang tinggi terhadap perubahan kalimat yang terjadi, apabila suatu kalimat dari hasil enkripsi ada yang

dihapus maupun diubah, maka proses dekripsi pesan tidak dapat dilakukan dengan kata lain gagal dengan keterangan "authentication failed". Untuk ilustrasi dapat diperhatikan pada tabel 5.2. hasil enkripsi.

Tabel 5.2. Hasil Enkripsi

Nama	Telepon	Nomor	Tanggal	Prodi	Nama	Nama di	Tanggal	Nominal
		ID	Lahir		Bank	Rekening	Transfer	
kmZW2	cHuuca	WBq54	JrbNQ	gMWFa	ooVFq	FxueUy	d1hLVl	hyd0iP
PX2RzlL	T7ocHS	Cm5X2	kOB2y	7F1GFv	LvW3	Mt7mlV	YwR/2	TKW8c
Ks3KUD	L37Zm	rOcGR	qEYa9	9HVAo	39frj3t	bjjfOFp	owxdyf	6hgEN8
KVwa+	3t+Wa	yfl4XJ	<b>bcCmw</b>	tQCg+g	/zPvO	hw/GVb	5KAsX	amoQv
QkjmFA	OjHXe	5c6u+S	MC4jN	hocXtV	gOey/	ooUCx5	1a3HqJ	0AZJK
0PYf/T	p5Ysjx	ev9BPh	A+P9I	n3ZDbf	OSKM	NwnTyi	NBQE	XZcNi
WW3Ltv	n9ciMk	c6+gf0	HLVh	eBg1Ve	61ON5	ZHg5vn	VKqS2	Kf+obk
Mz6+By	QjUUn	da5Piu	B35413	FWOlg	u/pEH	gCJHgS	utL9Vv	3dvOc
5NTiE0	pwg1Cl	MrVsu	fB4Z9	oy2J0p	7C3Ia	R6+AA	reR6q4	OayN6
GKWqjv	jkhXVb	0enK4b	TWsS	hS45/y	FT0yo	v3IJ9sT	HEZPh	BgHjtg
25QN5v	gJIQ8I	Wuqa+	V2jaC/	TjJOKE	RfW/6	9NJ8dvg	yODsc <sub>■</sub>	emg+1h
ElxWLc	WyLK7	jhGAZ	jNYxr	JCE7xe	AGFR	LD80eV	uaecO	8ywxsg
4NzcXT	hse3nY	womR	TC663f	R6QArl	yKNiv	vfNcy9	mvO8I	Ps8cs0
3iY9iG7	q1WNp	miNuS	5veYzv	EO02x2	nID47	UtIMSU	n1szL3	DWiF
yWxBm	9RfbXk	4ZlxPX	Qlp+D	RB86y	FyCjv	nhrergrk	8ggsL/	Wi0uX
+/tqYYe	e2MTz	uJUNr	sBp8C	EjHbno	OUQ/y	fiyU8U	RgbLC	YXxvM
K42imG	5xHnX	mf5t/s	myuks	7qgC/G	gwCSi	QdDSZ3	b3wxh	TTDjE/
QFjb9R7	BM5F0	BgVq7	GRnE6	bScwn	D8GU	hC2gHb	1eW8g	wH7+o
ZfwFTf9	4CJ++y	ShEUX	z56T/V	XFWB1	GgnMf	54ms5V	hDyenI	ql3Wp
$_{ m BhJFRZ}$	yoUZ8g	6zJTRS	G3nq2	P+rU/T	NeQ1S	keH4/81	1KAaX	YpYsup
yUx/rRj	tYqUx2	PlrjeN	pv+P9	ux+Hlc	62Jyw	KWsocp	rpBQlS	Z4WuN
UKPzf3S	UzUNQ	RirclFl	Rusvzj	8fabwK	g87snv	AsbOQJ	kyJghe	E91cws
Qg==	==	w==	C2u+v	c=	Fk=	A==	P	byw==
			V					

Berikut hasil *decipher* dapat diperhatikan pada tabel 5.3. hasil dekripsi, di mana pada hasil dekripsi ini walaupun kalimat teks (*plaintext*) yang dipakai sama namun apabila dienkripsi kembali maka akan memberikan hasil enkripsi yang berbeda dari sebelumnya, sehingga hasil enkripsi dari algoritma *AES* ini tidak *passive*, berbeda dengan algoritma *base64* di mana apabila kalimat teks yang sama akan dilakukan *encoding* maka akan menghasilkan output yang sama persis alias *passive*. Oleh karena itu, dengan penggabungan algoritma *AES* 256 *bit* dan algoritma *base64 encoding* maka hasil enkripsi akan memberikan output yang berbeda-beda walaupun kalimat yang akan dienkripsi sama dengan sebelumnya.

Tabel 5.3. Hasil Dekripsi

Nama	Telepon	Nomor ID	Tanggal Lahir	Prodi	Nama Bank	Nama di Rekening	Tanggal Transfer	Nominal
Fayrell Faza Putra Pratama	080000	2017000 0000	28 - March - 1999	Arsite ktur	BCA	ANDRIA TI ENDANG PRATIWI	18-5- 2021	Rp 550.000



#### 5.2. Pembahasan

Hal yang dijadikan bahan oleh penulis sebagai pembahasan didapatkan dari tahapan pengujian yang dilakukan, pengujian sendiri dilakukan pada sisi *client* dan *server*, tahapan pengujian yang dilakukan berupa *white box testing*, dan *black box testing*. Lebih rinci tentang pembahasan terhadap tahapan pengujian adalah sebagai berikut ini.

# 5.2.1. White Box Testing

Pada *white box testing*, maka penulis melakukan pengujian terhadap kode program *PHP* v8 yang dirancang pada *framework codeigniter* v4 dalam penerapan algoritma *AES* dan output dijadikan sebagai hasil riset pada pengujian.

Tabel 5.4. White Box Testing Algoritma AES (1) Hasil No. Kode Program Yang Indikator Diuji Keberhasilan Pengujian public function save2(){ Data Dapat Berhasil \$model2 = new Datareceipt();pribadi dienkripsi \$encrypter = | Config | Services::encrypter() peserta (encipher) \$data = array(disubmit dan disimpan 'nama'=> \$encrypter->encrypt(\$thisdalam pada database >request->getPost('nama2')), bentuk 'id number'=> \$encrypter->encrypt(\$thisplaintext >request->getPost('id number2')), 'nominal'=> \$encrypter->encrypt(\$this->request->getPost('nominal2')), 'telepon'=> \$encrypter->encrypt(\$this->request->getPost('telepon')), 'account'=> \$encrypter->encrypt(\$this->request->getPost('account2')), prodi'=> \$encrypter->encrypt(\$this->request->getPost('prodi')), 'bank name'=> \$encrypter->encrypt(\$this->request->getPost('bank name2')), 'transaction'=> \$encrypter->encrypt(\$this->request->getPost('transaction2')), >request->getPost('instansi')), 'pekerjaan'=> \$encrypter->encrypt(\$this->request->getPost('pekerjaan')) \$model2->saveProduct(\$data);

public function saveProduct(\$data){
\$query = \$this->db->table('pendaftaran')-

>insert(\$data); return \$query;

Tabel 5.5.	White B	ox Testing	Algoritma	AES(2)
I door J.J.	TI TILLE D	ON I COUNTY	1 II CITUIL	1100 (2)

No.	Kode Program	Yang Diuji	Indikator	Hasil
2	php foreach(\$pendaftaran as</td <td>Data hasil</td> <td>Keberhasilan Dapat</td> <td>Pengujiai Berhasil</td>	Data hasil	Keberhasilan Dapat	Pengujiai Berhasil
2	\$row):\$row['id'];?>	enkripsi	didekripsi	Demasii
	<h5>Biodata</h5>	•	(decipher)	
		(encipher) diolah oleh	sehingga data	
	Nomor identitas: = \$encrypter-</td <td>sistem pada</td> <td></td> <td></td>	sistem pada		
	>decrypt(\$row['id number']); ?>	halaman	yang ditampilkan	
	Nama Lengkap: = \$encrypter-</td <td>administrator</td> <td>pada halaman</td> <td></td>	administrator	pada halaman	
	>decrypt(\$row['nama']); ?	aaministrator	administrator	
	Instansi: = \$encrypter-</td <td></td> <td>berupa</td> <td></td>		berupa	
	>decrypt(\$row['instansi']); ?>		plaintext	
	Pekerjaan: = \$encrypter-</td <td></td> <td>piainiexi</td> <td></td>		piainiexi	
	>decrypt(\$row['pekerjaan']); ?>			
	Program Studi: = \$encrypter-</td <td></td> <td></td> <td></td>			
4	>decrypt(\$row['prodi']); ?>			
	Nomor Telepon: = \$encrypter-</td <td></td> <td></td> <td>7</td>			7
	>decrypt(\$row['telepon']); ?>		Y	
	<h5>Bukti Pembayaran</h5>			
	Nama di Rekening: = \$encrypter-</p			U·
	>decrypt(\$row['account']); ?>			
	Nama Bank Pentransfer: = \$encrypter-</td <td></td> <td></td> <td></td>			
	>decrypt(\$row['bank name']); ?>			
	Tanggal Transfer: = (date('d-n-Y',</td <td>_</td> <td></td> <td></td>	_		
	strtotime (\$encrypter-			
	>decrypt(\$row['transaction'])))); ?>			
	Jumlah Transfer: = \$encrypter-</td <td></td> <td></td> <td></td>			
	>decrypt(\$row['nominal']); ?>			
l	php endforeach;?			
			,	
	4 / G 11			
	/ / .			
	/1/		•	

Tabel 5.6. White Box Testing Algoritma AES (3)

No.	Kode Program	Yang Diuji	Indikator	Hasil
3	public function update(){	Data	Keberhasilan	Pengujian Berhasil
3	\$model = new Datareceipt();	diperbarui	Dapat dienkripsi	Delliasii
	<b>2</b> 0	•		
	Sencrypter =	( <i>edit</i> ) dalam bentuk	( <i>encipher</i> ) kemudian	
	Config   Services::encrypter();			
	\$id = \$this->request->getPost('id');	plaintext	disimpan	
	\$data = array(		pada database	
	'instansi'=> \$encrypter->encrypt(\$this-			
	>request->getPost('instansi')),			
	'pekerjaan' => \$encrypter->encrypt(\$this-			
	>request->getPost('pekerjaan')),			
	'prodi' => \$encrypter->encrypt(\$this-			
	>request->getPost('prodi')),	~		
	'nominal' => \$encrypter->encrypt(\$this-			
	<pre>&gt;request-&gt;getPost('nominal')), 'bank name'=&gt; \$encrypter-</pre>			7
	>encrypt(\$this->request-		Y	
	>getPost('bank_name')), 'transaction'=> \$encrypter-			
	>encrypt(\$this->request-			(),
	>getPost('transaction')),			
	'account'=> Sencrypter->encrypt(Sthis-			
	>request->getPost('account')),			
	'nama'=> \$encrypter->encrypt(\$this-			
	>request->getPost('nama2')),			
	'id number'=> \$encrypter-			
,	>encrypt(\$this->request-			
	>getPost('id_number')),			
١	'telepon'=> \$encrypter->encrypt(\$this-			
	>request->getPost('telepon'))			
	);			
	), \$model->updateProduct(\$data, \$id);			
	smoder-rupadierroduci(sadia, sta),			
				V
	<pre>public function updateProduct(\$data, \$id){</pre>	7		
	\$query = \$this->db-			
	table('pendaftaran')->update(\$data,			
	array('id' => \$id));			
	return Squery;		<b>A</b>	
	return oquery,			
		•	1	
	′ / / ,			
	/1/	_	,	

Tabel 5.7. White Box Testing Algoritma AES (4)

No.	Kode Program	Yang Diuji	Indikator	Hasil
	S	8	Keberhasilan	Pengujian
4	class Encryption extends BaseConfig	Melakukan	Tidak dapat	Berhasil
	{	dekripsi	melakukan	
	<pre>public \$key = 'empty';</pre>	(decipher)	dekripsi	
		data pribadi	(decipher)	
	<pre>public \$driver= 'OpenSSL';</pre>	peserta tanpa	data, sehingga	
		memiliki	data aman dari	
	<pre>public \$blockSize = 16;</pre>	cipher key	data	
		yang sesuai	breaching dan	
	$\mathcal{I}$	(match) pada	data leakage	
		proses		
	\ \ \	enkripsi		
~	and the second s	(encipher)		D 1 11
5	\$key=	Kode	Sistem dapat	Berhasil
	CodeIgniter\Encryption\Encryption::cre	program	mengenerate	7
	ateKey(); echo \$key;	PHP pada codeigniter	random ciphertext	7
	echo skey;	v4 untuk	characters	
		mengenerate	yang mana	
		random	untuk	0.
		ciphertext	dijadikan	
		characters	sebagai <i>cipher</i>	
		citar actors	key pada	
			proses	
			enkripsi	
			(encipher) dan	
			dekripsi	
			(decipher)	
				•
			<u> </u>	
	**			
	′ // /			
	/// ~	- 1	<b>\</b>	
	V / _			
	A V G U	11 ,		

Berikut merupakan *white box testing* terhadap kode program *PHP* v8 yang dirancang pada *framework codeigniter* v4 untuk penerapan algoritma *base64* dan output dijadikan sebagai hasil riset pada pengujian.

Tabel 5.8. White Box Testing Algoritma Base64 (1)

No.	Kode Program	Yang Diuji	Indikator	Hasil
			Keberhasilan	Pengujian
1	public function save2(){	Data hasil	Dapat	Berhasil
	\$model2 = new Datareceipt();	enkripsi	diencode	
	\$encrypter =	(encipher)	(encoding)	
	Config Services::encrypter();	diencode	dalam bentuk	
	\$data = array(	(encoding)	index base64	
	'nama'=> base64_encode(\$encrypter-	menggunakan	kemudian	
	>encrypt(\$this->request-	base64	hasil	7
	>getPost('nama2'))),	encoding	disimpan	
	'id number'=>		pada databas <del>e</del>	
	base64_encode(\$encrypter-			
	>encrypt(\$this->request-			O,
	>getPost('id_number2'))),			
	'nominal'=> base64 encode(\$encrypter-			
	>encrypt(\$this->request-			
	>getPost('nominal2'))),			
	'telepon'=> base64_encode(\$encrypter-			
	>encrypt(\$this->request-			
	>getPost('telepon'))),			
	'account'=> base64 encode(\$encrypter-			
	>encrypt(\$this->request-			
11	>getPost('account2'))),			
	'prodi' => base64_encode(\$encrypter-			
	>encrypt(\$this->request-			
	>getPost('prodi'))),			
4	'bank name'=>			
	base64_encode(\$encrypter-			
	>encrypt(\$this->request-			
	getPost('bank_name2'))),			
O	transaction'=>			•
	base64_encode(\$encrypter-			
	>encrypt(\$this->request-			
	getPost('transaction2'))),			
	'instansi'=> base64 encode(\$encrypter-			
	>encrypt(\$this->request-			
	>getPost(*instansi'))),	NI		
	'pekerjaan'=>	1/1 '		
	base64_encode(\$encrypter-			
	>encrypt(\$this->request-			
	>getPost('pekerjaan')))			
	Our della same Dura de est (O deste ).			
	\$model2->saveProduct(\$data);			
	7			
	public function saveProduct(\$data){			
	\$query = \$this->db-			
	>table('pendaftaran')->insert(\$data);			
	return \$query;			
	}			
			•	

0.	Kode Program	Yang Diuji	Indikator Keberhasilan	Hasil Pengujia
,	php foreach(\$pendaftaran as</td <td>Data hasil</td> <td>Dapat</td> <td>Berhasil</td>	Data hasil	Dapat	Berhasil
	\$row):\$row['id'];?>	encoding	didecode	
	< <i>h5&gt;Biodata</i> <i h5>	base64	(decoding)	
	< <i>p&gt;</i>	didecode	sehingga data	
	Nomor identitas: =</td <td>(decoding)</td> <td>yang</td> <td></td>	(decoding)	yang	
	<pre>base64 decode(\$row['id number']); ?&gt;</pre>		ditampilkan	
	Nama Lengkap: =</td <td></td> <td>berupa</td> <td></td>		berupa	
	base64_decode(\$row['nama']); ?>		plaintext	
	Instansi: =</td <td></td> <td></td> <td></td>			
	base64_decode(\$row['instansi']); ?>			
	Pekerjaan: =</td <td></td> <td></td> <td></td>			
	base64_decode(\$row['pekerjaan']); ?>			
_	Program Studi: =</td <td></td> <td></td> <td></td>			
	base64_decode(\$row['prodi']); ?>			7
	Nomor Telepon: =</td <td></td> <td></td> <td></td>			
	base64_decode(\$row['telepon']); ?>			
J	<h5>Bukti Pembayaran</h5>			U,
	Nama di Rekening: =<br base64 decode(\$row['account']); ?>			
	Nama Bank Pentransfer: =</td <td></td> <td></td> <td></td>			
	base64 decode(\$row['bank name']); ?>			
	Tanggal Transfer: = (date('d-n-Y',</td <td></td> <td></td> <td></td>			
	strtotime			
	(base64 decode(\$row['transaction']))));			
	?>			
	Jumlah Transfer: =</td <td></td> <td></td> <td></td>			
	base64 decode(\$row['nominal']); ?>			
	php endforeach;?			
-				•
			_	
	* Y A			
	VGU	1		
	///	n 1		
	v ( \		,	
	<b>9</b> ()	11		
		-		

Tabel 5.10. White Box Testing Algoritma Base64 (3)

37	Tabel 3.10. White Box Testir			
No.	Kode Program	Yang Diuji	Indikator	Hasil
			Keberhasilan	Pengujian
3	public function update(){	Data	Dapat	Berhasil
	<pre>\$model = new Datareceipt();</pre>	diperbarui	diencode	
	Sencrypter =	(edit) dalam	(encoding)	
	\Config\Services::encrypter();	bentuk	kemudian	
	\$id = \$this->request->getPost('id');	plaintext	disimpan	
	\$data = array(	piumiexi	pada database	
	'instansi'=> base64_encode(\$encrypter-		pada dalabase	
	>encrypt(\$this->request-			
	>getPost('instansi'))),			
	'pekerjaan'=>			
	base64 encode(Sencrypter-			
	>encrypt(\$this->request-			
	>getPost('pekerjaan'))),			
1	prodi => base64 encode(\$encrypter-			7
	>encrypt(\$this->request-			
	>getPost('prodi'))),			
	'nominal'=> base64 encode(\$encrypter-			
	>encrypt(\$this->request-			( ) '
	>getPost('nominal'))),			
	'inform'=> \$this->request-			
	>getPost('inform'),			
	'bank name'=>			
	base64_encode(Sencrypter-			
	>encrypt(\$this->request-	1		
	>getPost('bank name'))),			
	'transaction' =>			
	base64 encode(\$encrypter-			
	>encrypt(\$this->request-			
	>getPost('transaction'))),			
	'account' => base64 encode(\$encrypter-			
	>encrypt(\$this->request-			
	>getPost('account'))),			
	'nama'=> base64 encode(\$encrypter-			
	>encrypt(\$this->request-			
	getPost('nama2'))),			
	"id number'=>			
	base64_encode(\$encrypter-			
	>encrypt(\$this->request-			
	>geiPost('id_number'))),			
	'telepon'=> base64_encode(\$encrypter-			
	>encrypt(\$this->request-			
	>getPost('telepon')))	n 1		
	);		_	
	\$model->updateProduct(\$data, \$id);	1 4		
	}	•		
	,			
	public function updateProduct(\$data, \$id){			
	\$query = \$this->db-			
	>table('pendaftaran')->update(\$data,			
	array('id' => \$id));			
	return \$query;			
	j			

Tabel 5.11. White Box Testing Algoritma Base64 (4)

No.	Tabel 5.11. White Box Testing  Kode Program	Yang	Indikator	Hasil	
110.	Roue Hogram	Diuji	Keberhasilan	Pengujian	
4	php foreach(\$pendaftaran as</td <td>Hasil</td> <td>Dapat</td> <td>Berhasil</td>	Hasil	Dapat	Berhasil	
	\$row):\$row['id'];?>	enkripsi	didecode		
	<h5>Biodata</h5>	(encipher)	(decoding)		
	< <i>p&gt;</i>	data	dan didekripsi		
	Nomor identitas: = \$encrypter-</td <td>pribadi</td> <td>(decipher)</td> <td></td>	pribadi	(decipher)		
	>decrypt(base64 decode(\$row['id numbe	peserta	oleh sistem		
	r'])); ?>	diencode	sehingga data		
	Nama Lengkap: = \$encrypter-</td <td>(encoding)</td> <td>pribadi peserta</td> <td></td>	(encoding)	pribadi peserta		
	>decrypt(base64 decode(\$row['nama']));	dalam	yang		
	?>	bentuk	ditampilkan		
	Instansi: = \$encrypter-</td <td>index</td> <td>berupa</td> <td></td>	index	berupa		
	>decrypt(base64_decode(\$row['instansi']	base64	plaintext pada		
	)), ?>		halaman		
	Pekerjaan: = \$encrypter-</td <td></td> <td>administrator</td> <td>7</td>		administrator	7	
	<pre>&gt;decrypt(base64_decode(\$row['pekerjaan')]). 2</pre>				
	'])); ?> Program Studi: = \$encrypter-</td <td></td> <td><b>/</b></td> <td></td>		<b>/</b>		
	>decrypt(base64 decode(\$row['prodi']));				
	?>			O,	
	Nomor Telepon: = \$encrypter-</td <td></td> <td></td> <td>_</td>			_	
	>decrypt(base64 decode(\$row['telepon'])				
	); ?>				
	<h5>Bukti Pembayaran</h5>				
	Nama di Rekening: = \$encrypter-</td <td></td> <td></td> <td></td>				
	>decrypt(base64_decode(\$r <mark>ow['account']</mark>				
	)); ?>				
7	Nama Bank Pentransfer: = \$encrypter-</td <td></td> <td></td> <td></td>				
	>decrypt(base64_decode(\$row['bank_na				
	me'])); ?>				
	Tanggal Transfer: $ (date('d-n-Y',$				
	strtotime (Sencrypter-				
	<pre>&gt;decrypt(base64_decode(\$row['transactio n'])))); ?&gt;</pre>				
	Jumlah Transfer: = \$encrypter-</td <td></td> <td></td> <td></td>				
	>decrypt(base64 decode(\$row['nominal']				
U	)); ?>			•	
			•		
	php endforeach;?				
5	\$token=	Kode	Base64	Berhasil	
	base64 encode(random bytes(32));	program	encoding		
		PHP pada	dapat		
	v ( -	codeigniter	mengenerate		
	9 ()	v4 untuk	random bytes		
		mengenera	32 characters		
		te random	untuk		
		bytes 32	membuat		
		characters	token pada		
		dalam	proses		
		membuat	autentikasi		
		token	akun <i>user</i> dan		
		sebagai	administrator		
		proses			
		autentikasi			

Berikut merupakan *white box testing* terhadap kode program *PHP* v8 yang dirancang pada *framework codeigniter* v4 untuk percobaan *validation testing*, di mana output dijadikan sebagai hasil riset pada pengujian.

Tabel 5.12. White Box Testing pada Validation Testing (1)

No.	Kode Program	Yang	Indikator	Hasil
	_	Diuji	Keberhasilan	Pengujian
1	\$rules = [	Membuat	Sistem dapat	Berhasil
	'nama2'=>[	kode	melakukan	
	'rules'=>'alpha numeric punct',	program	validasi dan	
	'errors'≠>[	untuk SQL	menolak data	
	'alpha numeric punct'=>'This field may	injection	yang diisi	
	contain only alphanumeric characters,	attack pada	pada <i>¶ield</i>	
	spaces, and $\sim ! # $ \% & * - + =   : .$	field yang	(data invalid)	7
	characters']],	ada dan		
	'id number2'=> [	submit		
	'rules'=>'alpha numeric punct',	data		
	'errors'=>[			O,
	'alpha numeric punct'=>'This field may			
	contain only alphanumeric characters,			
	spaces, and $\sim ! # $ % & * - + =   : .$			
	characters']],			
	'instansi'=> [			
	'rules'=>'alpha numeric p <mark>unct',</mark> 'errors'=>[			
	'alpha numeric punct'=>'This field may			
	contain only alphanumeric characters,			
	spaces, and ~! #\$% & *+= :.			
1 1	characters']],			
	'pekerjaan'=> [			
	'rules'=>'alpha numeric punct',			
	'errors'=>[			
	'alpha numeric punct'=>'This field may			
	contain only alphanumeric characters,			
	spaces, and ~! #\$ % & *- +=  :.			
	characters']],			,
	'account2'=> [			
	'rules'=>'alpha_numeric_punct',			
	'errors'=>[			
	'alpha numeric punct'=>'This field may			
	contain only alphanumeric characters,			
	spaces, and $\sim 1 \# \% & \Longrightarrow + =  :.$	$\sim$		
	characters']],			
	'inform2'=> [	1 4		
	'rules'=>'alpha numeric punct',			
	'errors'=>[			
	'alpha numeric punct'=>'This field may			
	contain only alphanumeric characters,			
	spaces, and $\sim ! # $ % & *- +=   : .$			
	characters']]			
	_];			

Tabel 5.13. White Box Testing pada Validation Testing (2)

No.	Kode Program	Yang	Indikator	Hasil
		Diuji	Keberhasilan	Pengujian
2	<i>\$rules</i> = [	Mengkoso	Sistem dapat	Berhasil
	'nama2'=> [	ngkan	melakukan	
	'rules'=>'required',	setiap data	validasi dan	
	'errors'=>[	pada <i>field</i>	memberikan	
	'required'=>"Don't forget to fill this"]],	yang	feedback data	
	'id number2'=> [	tertera	in these fields	
	'rules'=>'required',	pada	are required	
	'errors'=>[	online	-	
	'required'=>"Don't forget to fill this"]],	form dan		
	'prodi'=> [	submit		
	'rules'=>'required',	data		
	'errors'=>f	4		
	'required'=>"Don't forget to fill this"]],			
	'instansi'=> [		•	
	'rules'=>'required',			
	'errors'=>[		<b>&gt;</b>	7
	'required'=>"Don't forget to fill this"]],		•	
	'pekerjaan'=> [			
	'rules'=>'required',			V
	'errors'=>f			
	'required'=>"Don't forget to fill this"]],			
	'account2'=> [			
	'rules'=>'required',			
-	'errors'=>[			
)	'required'=>"Don't forget to fill this"]],			
	'bank name2'=> [			
	'rules'=>'required',			
1	'errors'=>[			
1	'required'=>"Don't forget to fill this"]],			
	'transaction2'=> [			
	'rules'=>'required',			
	'errors'=>[			
	'required'=>"Don't forget to fill this"]],			
	'email4'=> [			
	'rules'=>'required',			
	'errors'=>[			
U	'required'=>"Don't forget to fill this"]],			•
	'telepon'=> [		•	
	'rules'=>'required',			
	'errors'=>[			
	'required' => "Don't forget to fill this"]],	1		
	'nominal2'=>/	- 1		
	'rules'=>'required',			
	'errors'=>[	IJI	•	
	'required'=>"Don't forget to fill this"]],	1 -		
	'inform2'=> [			
	'rules'=>'required',			
	<u> </u>			
	'errors'=>[ 'required'=>"Don't forget to fill this"]]			

Tabel 5.14. White Box Testing pada Validation Testing (3)

No.	Kode Program	Yang Diuji	Indikator Keberhasilan	Hasil Pengujian
3	\$rules = [	Mengisi	Sistem dapat	Berhasil
_	'id_number2'=> [	alphabet	melakukan	
	'rules'=>'numeric',	pada <i>field</i>	validasi dan	
	'errors'=>[	nomor	memberikan	
	'numeric'=>"This field must contain only	telepon,	feedback data	
	numbers"]],	nomor	in these fields	
	'telepon'=> [	identitas,	must numeric	
	'rules'=>'numeric'.	nominal		
	'errors'=>[	transfer		
	'numeric'=>"This field must contain only	dan submit		
	numbers"]],	data		
	'nominal2'=> [	4		
	rules'=>'mimeric',			
4	'errors'=>[		•	
`	numeric'=>'Numbers character only']]			
	1;			
4	\$rules = [	Mengupload	Sistem dapat	Berhasil
	'pict2'=> [	file excel	melakukan	
	'rules'=>	pada <i>field</i>	validasi dan	V
	'uploaded[pict2] mime_in[pict2,image/jpg,	browse a	memberikan	
	image/jpeg,image/png]  max_size[pict2,20	file yang	feedback data	
	24] is_image[pict2]',	tertera	in these fields	
	'errors'=>[	pada	must jpg, jpeg,	
ı	'uploaded'=> 'Insert the receipt payment',	online	or png	
	'max_size'=> 'Size is too big, max size 2	form dan		
	MB',	submit		
	'is_image'=> 'The file you were uploaded	data		
	is not an image',			
l	'mime_in'=> 'Must jpg/jpeg/png']]			
	];			
5	public function save2(){	Submit	Sistem dapat	Berhasil
	\$model2 = new Datareceipt();	data	merename file	
	file2 = file	berupa <i>file</i>	tersebut	
	\$filename2=\$file2->getRandomName();	gambar	dengan	
	\$file2-	dengan	random name	
	>move(ROOTPATH.'/public/receipt',\$filen	format jpg,	untuk	,
V	ame2);	<i>jpeg</i> , atau	keamanan data	
	\$data = array(	png	kemudian	
	'struk' => \$filename2,		menyimpannya	
	);		pada database	
	\$model2->saveProduct(\$data);	1	dan pada	
	, , , , , , , , , , , , , , , , , , ,	R 1	folder	
			directory yang	
	public function saveProduct(\$data){	11	telah	
	\$query = \$this->db->table('pendaftaran')-	_	ditentukan	
	>insert(\$data);			
	return \$query;			
	. }			

## 5.2.2. Black Box Testing

Pada *black box testing*, maka penulis melakukan pengujian terhadap aplikasi *web* yang dirancang pada *framework codeigniter* v4 dengan penerapan algoritma *AES* dan output dijadikan sebagai hasil riset pada pengujian.

Tabel 5.15. Black Box Testing Algoritma AES

_	No. Pengujian Yang Diuji Indikator Hasil yang Kesimpul					Kesimpulan
_	NO.	Halaman	Yang Diuji	Keberhasilan	Diharapkan	
	1	Menu registrasi	Online form	Data pribadi peserta	Berhasil	Hasil
			V	disubmit dalam		pengujian
				bentuk <i>plaintext</i>		valid
				dienkripsi (encipher)		
				kemudian disimpan pada database		7
	2	Halaman	Datatable	Data hasil enkripsi	Berhasil	Hasil
4		administrator –	Dalalable	(encipher) diolah oleh	Demasii	pengujian
		menu read data		sistem, selanjuntya		valid
		peserta		didekripsi (decipher)		
		F		sehingga data yang		
				ditampilkan pada		
				halaman		
				administrator berupa		
				plaintext		
	3	Halaman	Server	Data pribadi peserta	Berhasil	Hasil
		administrator -	database	diperbarui (edit)		pengujian
		menu update	aplikasi	dalam bentuk		valid
		data peserta		plaintext, selanjutnya		
-				dienkripsi (encipher)		
				kemudian disimpan pada database		
	1	Menu	Data hasil	Melakukan dekripsi	Berhasil	Hasil
	Ż	cryptography	registrasi	(decipher) data	Demasii	pengujian
		tool	peserta yang	pribadi peserta tanpa		valid
		^	telah	memiliki cipher key		
			terenkripsi	yang sesuai (match)		,
				pada proses enkripsi		
		1		(encipher) sehingga		
				tidak dapat		
		Ι Λ.		melakukan dekripsi		
	_	/ / /		(decipher) data		"
	5	Halaman	Sistem pada	Sistem dapat	Berhasil	Hasil
		administrator -	aplikasi <i>web</i>	mengenerate random		pengujian
		menu view data	dalam au <i>l</i> ainai	ciphertext characters		valid
		peserta	enkripsi (encipher)	yang mana untuk dijadikan sebagai		
			dan <i>dekripsi</i>	cipher key pada		
			(decipher)	proses enkripsi		
			data	(encipher) dan		
				dekripsi (decipher)		
_						

Berikut merupakan *black box testing* terhadap aplikasi *web* yang dirancang pada *framework codeigniter* v4 untuk penerapan algoritma *base64* dan output dijadikan sebagai hasil riset pada pengujian.

Tabel 5.16. Black Box Testing Algoritma Base64

_	No.	Pengujian	Yang Diuji	Indikator	Hasil yang	Kesimpulan
_		Halaman		Keberhasilan	Diharapkan	
	1	Menu registrasi	Online form	Data hasil enkripsi	Berhasil	Hasil
				(encipher) diencode	<i>b</i>	pengujian
				(encoding) dalam		valid
			<b>J</b>	bentuk index base64,		
				selanjutnya hasil		
				disimpan pada		
				database		1
	2	Halaman	Datatable	Data hasil encoding	Berhasil	Hasil
4		administrator –		base64 didecode	•	pengujian
		menu read data		(decoding) sehingga		valid
		peserta		data yang ditampilkan		U,
			_	berupa <i>plaintext</i>		
	3	Halaman	Server	Data diperbarui (edit)	Berhasil	Hasil
		administrator -	database	dalam bentuk		pengujian
		menu update	aplikasi	plaintext, selanjuntya		valid
		data peserta		diencode (encoding)		
				untuk disimpan pada		
				database		
	4	Halaman	Data hasil	Hasil enkripsi	Berhasil	Hasil
		administrator –	registrasi	(encipher) data		pengujian
1 1		menu view data	peserta yang	pribadi peserta		valid
		peserta	telah	diencode (encoding)		
			terenkripsi	dalam bentuk index		
				base64, selanjutnya didecode (decoding)		
				dan didekripsi		
				(decipher) oleh		
				sistem sehingga data		
				pribadi peserta yang		)
`	U			ditampilkan berupa		
				plaintext pada		
				halaman		
		7 1		administrator	13	
	5	Server database	Token	Sistem dapat	Berhasil	Hasil
		aplikasi	base64	menerapkan <i>base64</i>	- Constant	pengujian
		up III usi	encoding	encoding dalam		valid
			yang	mengenerate random		
			dikirimkan	bytes 32 characters		
			pada proses	untuk membuat token		
			autentikasi	sebagai proses		
			akun melalui	autentikasi akun <i>user</i>		
			pesan email	dan administrator		
_		•	_	•		

Berikut merupakan *black box testing* terhadap aplikasi *web* yang dirancang pada *framework codeigniter* v4 untuk percobaan *validation testing*, di mana output dijadikan sebagai hasil riset pada pengujian.

Tabel 5.17. Black Box Testing pada Validation Testing (1)

-				: Testing pada Validati		
	No.	Pengujian	Yang Diuji	Indikator	Hasil yang	Kesimpulan
-		Halaman	0.1:	Keberhasilan	Diharapkan	TT 1
	1	Menu registrasi	Online form	Membuat kode	Berhasil	Hasil
			1 1	program untuk SQL		pengujian
				injection attack pada		valid
			V	field yang ada dan		
				submit data,		
				selanjutnya sistem		
				memvalidasi dan		
				menolak data yang		Y
4				diisi pada field (data	•	
	2	Manual and interest	Ouling Com	invalid)	Darkarit	TTavil
	2	Menu registrasi	Online form	Mengkosongkan	Berhasil	Hasil
				setiap data pada field		pengujian
				yang tertera pada		valid
				online form dan		
				submit data, selanjutnya sistem		
				memvalidasi dengan		
				memberikan feedback		
				data in these fields		
				are required		
	3	Menu registrasi	Online form	Mengisi alphabet	Berhasil	Hasil
וון	1	mena registrasi	Omine John	pada <i>field</i> nomor	Demasii	pengujian
		4		telepon, nomor		valid
				identitas, nominal		
				transfer dan submit		
				data, selanjutnya		
				sistem memvalidasi		
		^		dengan memberikan		
				feedback data in		)
	V			these fields must		
				numeric		
	4	Menu registrasi	Online form	Mengupload file excel	Berhasil	Hasil
		, y ,	,	pada field browse a	13	pengujian
		' // /	•	file yang tertera pada		valid
		/ //		online form dan		
		V		submit data,	· ·	
				selanjutnya		
				sistem memvalidasi		
				dengan memberikan		
				feedback data in		
				these fields must jpg,		
				jpeg, or png		

			Testing pada Validati		
No.	Pengujian	Yang Diuji	Indikator	Hasil yang	Kesimpulan
	Halaman  Menu registrasi,	Online form	Keberhasilan Submit data berupa	<b>Diharapkan</b> Berhasil	Hasil
5	server database	Online Joint	file gambar dengan	Demasii	pengujian
	aplikasi, dan		format jpg, jpeg, atau		valid
	halaman		png, selanjutnya		
	administrator-		sistem merename file		
	menu view data		tersebut dengan		
			random name untuk		
			keamanan data		
		1 -	kemudian		
			menyimpannya pada database dan pada		
		<b>V</b>	folder directory yang		
			telah ditentukan		
				₩	
					(),
,					
-					
J					
-	`				
٦ .					
_					
	^				
					,
	1				
	7		II N	17	
	' /	•	1		
	/1/		1		
	V			<b>,</b>	
				-	