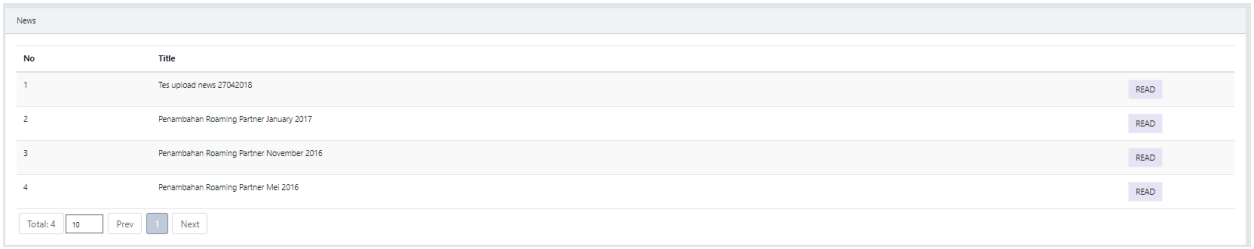
WEB SETIA REQUIREMENT

Fungsi-fungsi utama web SETIA :

1. CMS untuk news
2. Configurable menu dengan Page Iframe yang link ke halaman report/Dashboard InstaBI
3. CRUD untuk master data

1. CMS untuk News

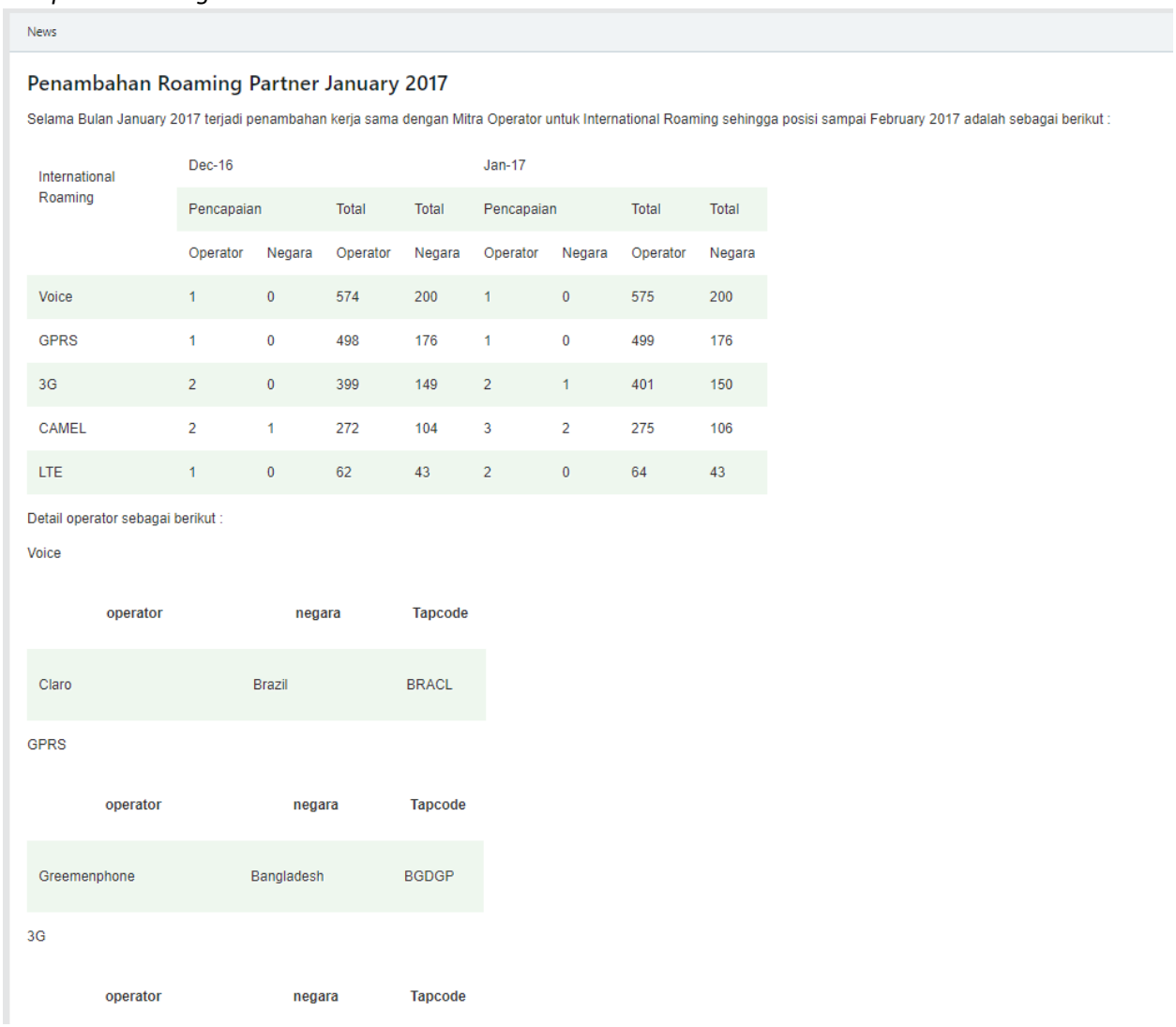
Tampilan Eksisting List News:



Perbaikan List News:

Di buat lebih menarik tampilannya dengan preview Judul serta snapshot Content News.

Tampilan Eksisting News:



News

Penambahan Roaming Partner January 2017

Selama Bulan January 2017 terjadi penambahan kerja sama dengan Mitra Operator untuk International Roaming sehingga posisi sampai February 2017 adalah sebagai berikut :

International Roaming	Dec-16				Jan-17			
	Pencapaian		Total	Total	Pencapaian		Total	Total
	Operator	Negara	Operator	Negara	Operator	Negara	Operator	Negara
Voice	1	0	574	200	1	0	575	200
GPRS	1	0	498	176	1	0	499	176
3G	2	0	399	149	2	1	401	150
CAMEL	2	1	272	104	3	2	275	106
LTE	1	0	62	43	2	0	64	43

Detail operator sebagai berikut :

Voice

operator	negara	Tapcode
Claro	Brazil	BRACL

GPRS

operator	negara	Tapcode
Greemenphone	Bangladesh	BGDGP

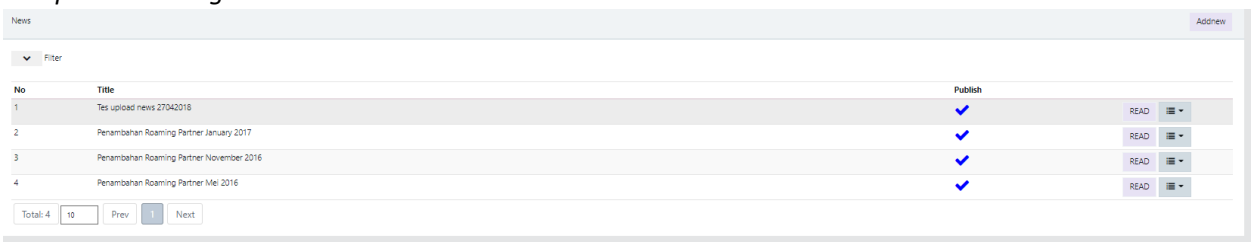
3G

operator	negara	Tapcode

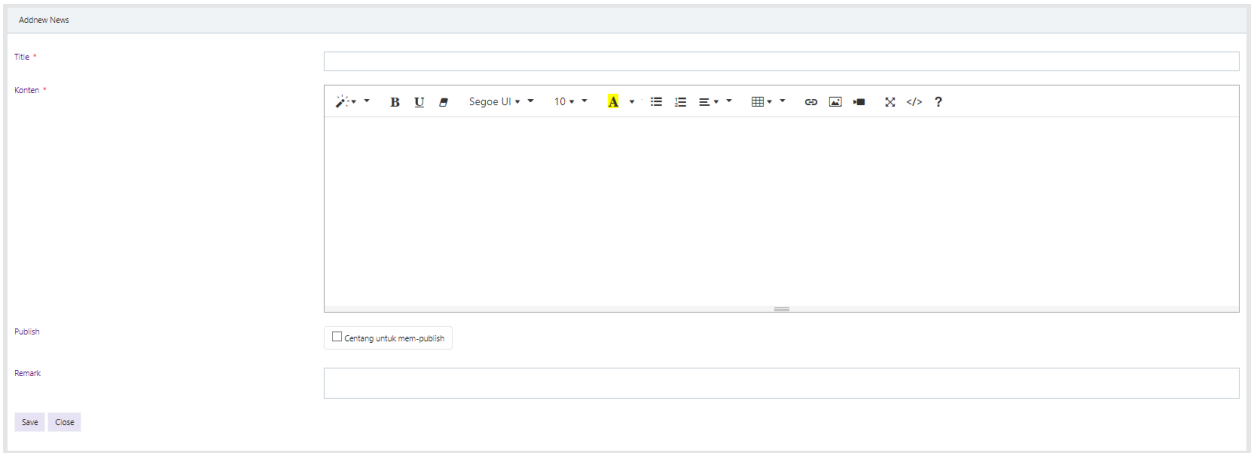
Perbaikan Eksisting News:

DIbuat lenih menarik dan interaktif

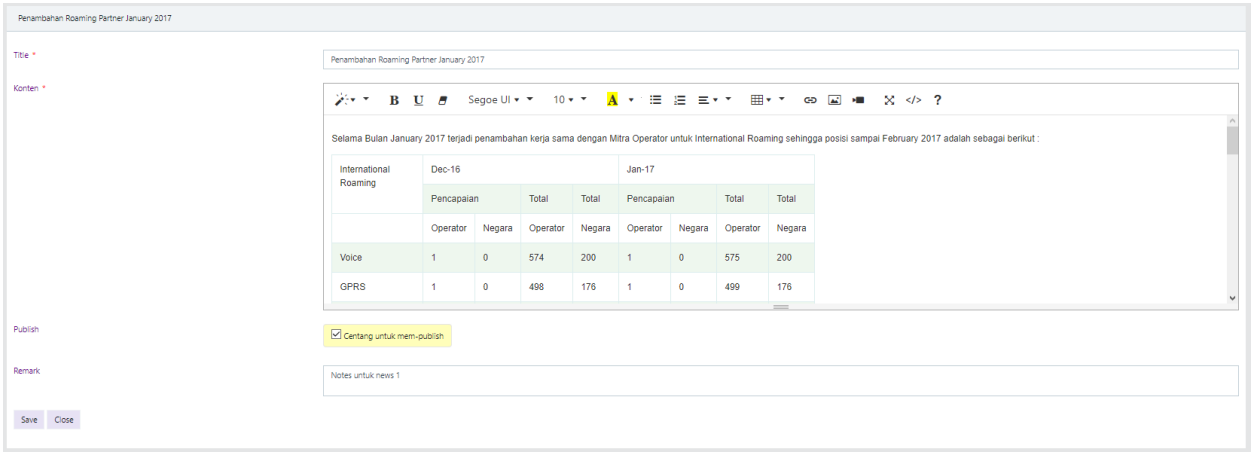
Tampilan Eksisting CRUD News:



Add News:

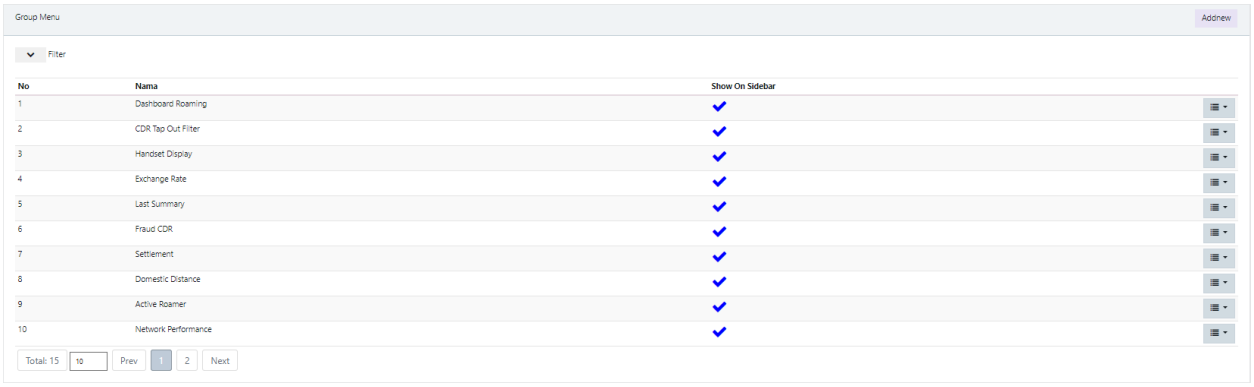


Edit News:

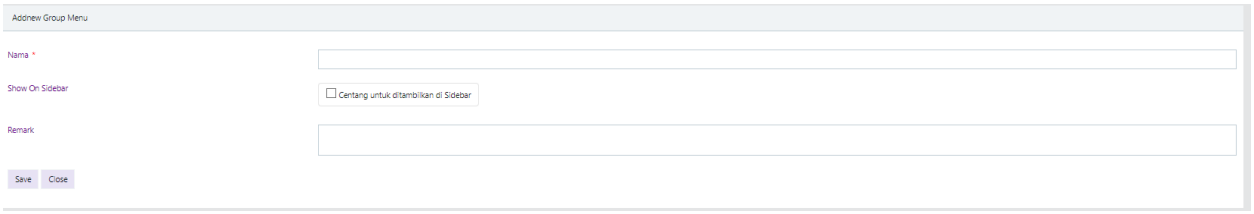


2. Configurable menu dengan Page Iframe yang link ke halaman report/Dashboard InstaBI

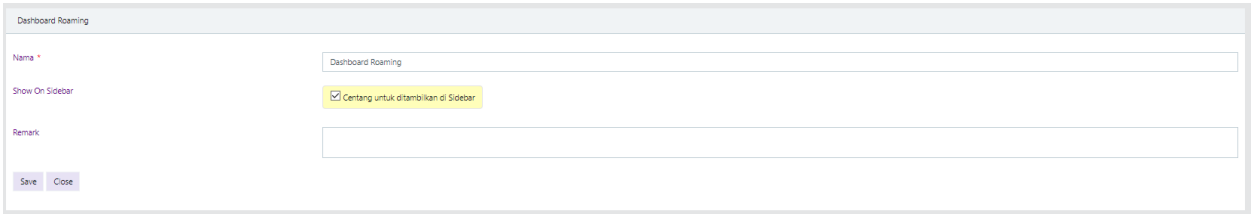
Tampilan Eksisting Konfigurasi Group Menu:



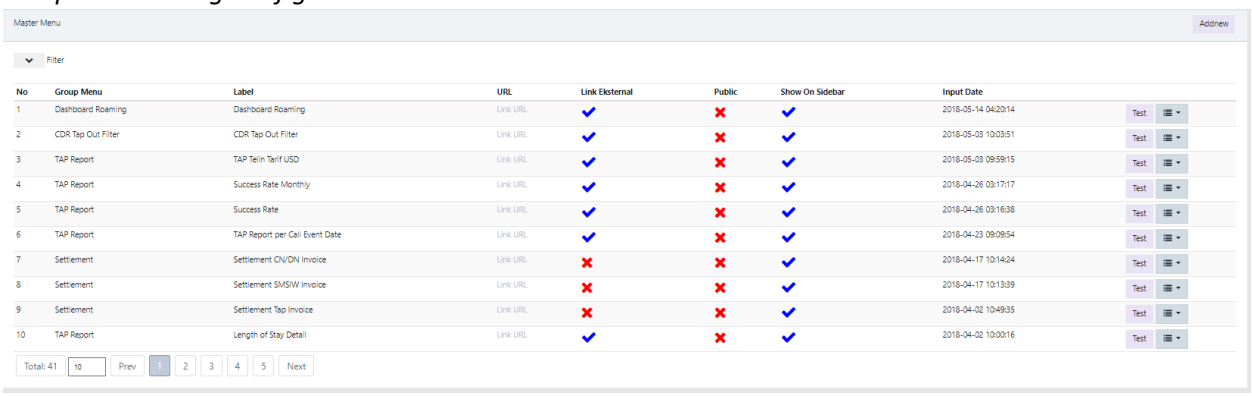
Add New Group Menu:



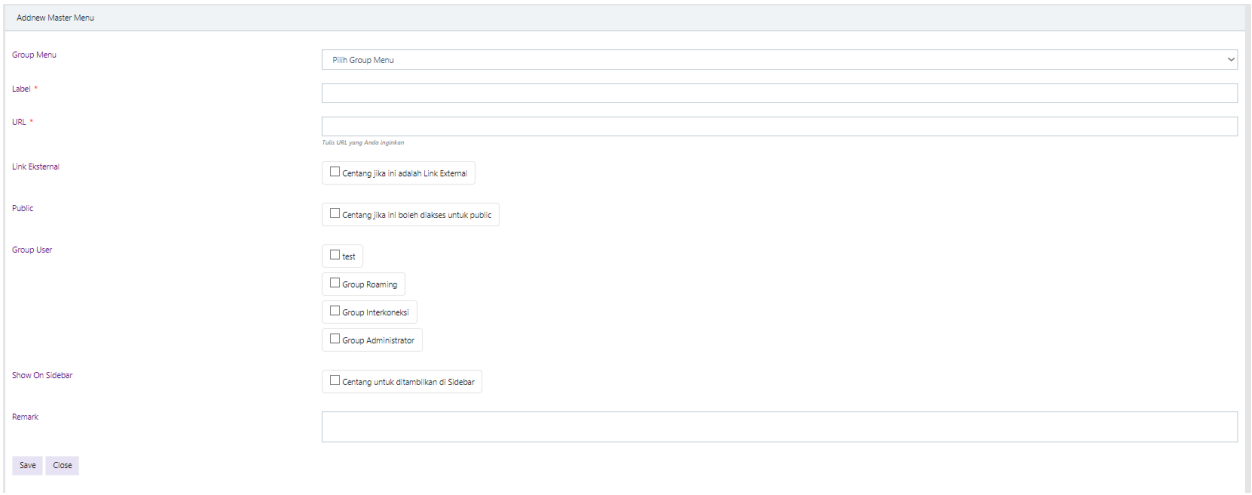
Edit Group Menu:



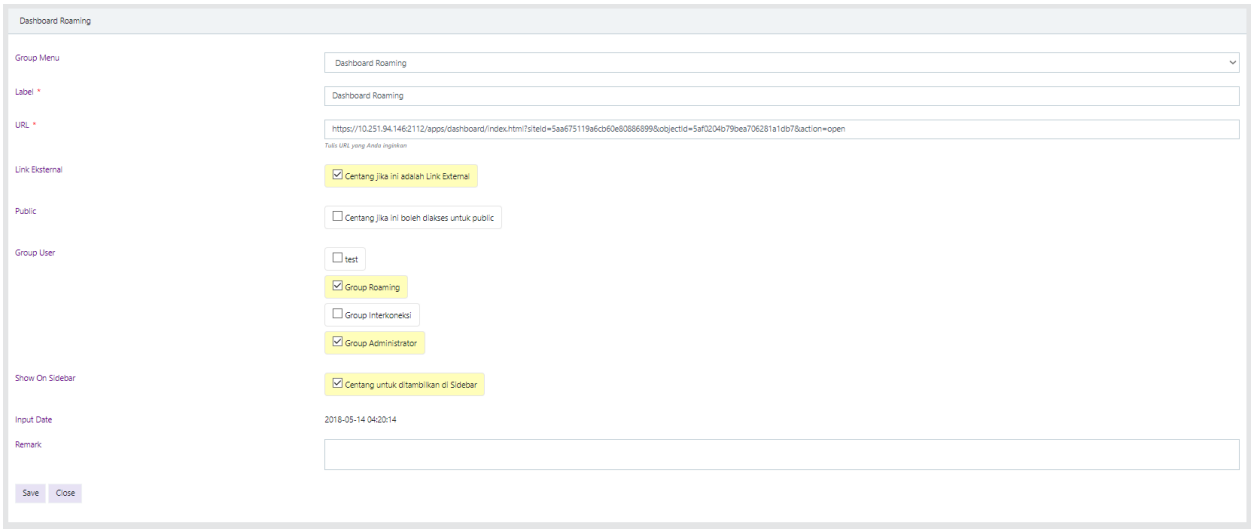
Tampilan Eksisting Konfigurasi Menu:



Add New Menu:



Edit Menu:



3. CRUD untuk master data

Untuk Membuat Page Create, Read, Update dan Delete dari master data:

a. Continent
b. Negara
c. Group Operator
d. Operator
e. Telin Tariff
f. Kota
g. MSC
h. Exchange Rate

4. Perbaikan-perbaikan lainnya

Comply with Web Application Security Standard

No	Standard	Notes
1	**Authentication**	
1.1	Authentication and authorization should be performed on server-side for any access to non-public resources.	Mandatory
1.2	Use multi-factor authentication. Typically this is based on something they have (e.g., smartcard, private key), something they know (e.g., a pin, passphrase), or something they are (e.g., data from a biometric reader).	Recommended
1.3	Users should be forced to change their initial password, which they get within an envelope or via email, by their first access to system.	Mandatory
1.4	A common message should be used for authentication failures to prevent user enumeration attacks. An example of such a message would be “Username and/or Password is wrong”.	Mandatory
1.5	Password complexity shall be enforced (include upper case, lower case letters, numbers or punctuation marks).	Mandatory
1.6	All successful and unsuccessful authentication attempts and access attempts to resources should be logged.	Mandatory
1.7	The current password should always be asked to users for password change functionalities.	Mandatory

1.8	Password retrieval functionalities should be supported with secret questions and similar arguments.	Mandatory
1.9	Password retrieval functionalities should not send user names and passwords back within emails. Instead, a link with certain lifetime should be sent that prompts a dialog for password change.	Mandatory
1.10	CAPTCHA or similar anti-automation security controls should be implemented within HTML forms to prevent DoS, brute-forcing and dictionary attacks.	Mandatory
1.11	The application must support disabling of accounts and terminating sessions when authorization ceases (e.g., Changes to role, employment status, business process, etc.)	Mandatory
1.12	Use only HTTP POST requests to transmit authentication credentials.	Mandatory
1.13	Default user accounts should be removed from applications, systems and databases.	Mandatory
2	**Session Management**	
2.1	HTTP/HTML attributes (e.g. autocomplete, cache-control, pragma) should be disable and configured accordingly to prevent storage of sensitive information like passwords within caches.	Mandatory
2.2	Unique values (e.g. session identifiers, token etc.) used for session management should be generated via secure random number generators.	Mandatory
2.3	Do not allow concurrent logins with the same user ID. Generate a new session identifier on any re-authentication.	Mandatory
2.4	Disallow persistent logins and enforce periodic session terminations, even when the session is active. Especially for critical applications. Note : Mass Destroy session every 24hours	Mandatory
2.5	Do not expose session identifiers in URLs. Session identifiers should only be located in the HTTP cookie header. For example do not pass session identifiers as GET parameters.	Mandatory
2.6	After each authentication and reauthentication, a new session id should be created and the old session id should be invalidated. After logging out, the session id should be invalidated as well.	Mandatory

2.7	The domain and path for cookies containing authenticated session identifiers should be set to an appropriately restricted value for the site.	Mandatory
2.8	*httponly* attribute should be set on cookies. In addition, *secure* attribute should be set on cookies for HTTPS communications.	Recommended
2.9	Logout links should be available within all pages of accessed applications.	Recommended
2.1	Sessions shall timeout after a specified period of inactivity. Session inactivity timeout is as short as possible, based on balancing risk and business functional requirements (five minutes is common).	Mandatory
3	**Input Validation**	
3.1	All user inputs should be validated on server-site. White-lists should be preferred for validation instead of black-lists. Each input should be encoded to a common character set before validation (canonicalization).	Mandatory
3.2	User inputs regarding arithmetic operations should be checked and validated for minimum and maximum values.	Mandatory
3.3	By the operation of a file upload, name, length, type and content of the file should be checked.	Mandatory
3.4	User inputs used for HTTP redirects should be validated by using whitelist method to prevent phishing attacks (open redirect problem).	Recommended
3.5	CR/LF characters sent within user inputs should not be directly appended within HTTP Responses on the application side (CRLF injection attack). User inputs should be properly sanitized.	Mandatory
3.6	A timeout for search functionalities should be enabled against SQL Wildcard attacks which force databases to perform CPU-intensive queries by using several search wildcards like “%” or “*”.	Mandatory
4	**Access Control**	

4.1	Sensitive links which should not be indexed by search engines should be listed within robots.txt files. On the other hand, if a critical webpage (e.g. administration panel) is not explicitly linked within the web application, it should not be included within robot.txt files as well.	Mandatory
4.2	Access to non-public resources (e.g. backup files, development test files) should be restricted for certain users and unnecessary applications (e.g. default web server sites, demo applications) should be removed.	Mandatory
4.3	Solutions like tokens, captchas should be integrated for critical operations in order prevent Cross-Site-Request-Forgery (CSRF) attacks.	Mandatory
4.4	Parameter manipulations within GET/POST requests should be taken into consideration and unauthorized access to legal user resources by attackers should be prevented.	Mandatory
4.5	Restricted URLs, functions, object references, services, application data, user information and security configuration files should be accessible for authorized users and roles (e.g. implementation of authentication and access control list based on roles).	Mandatory
4.6	Names of critical directories like administration panels should not be easily guessable (e.g. admin, administration).	Mandatory
4.7	It is recommended for application not to use port that become a default port for other services (e.g. 8080, 3128 – default port for web proxy)	Mandatory
4.8	Application must follow multi tiers architecture, (e.g. first tier is web server, the second tier is application server, and the third tier is database server). Additional security control must be addressed for each tiers (e.g. tier considered critical is placed behind firewall with proper rules)	Mandatory
5	**Error Handling & Logging**	
5.1	Error messages of web applications and application-server default error messages should not be displayed in details to clients.	Mandatory
5.2	Implement generic error messages and use custom error pages.	Recommended
5.3	Logging controls should support both success and failure of specified security events.	Mandatory

5.4	Ensure logs contain important log event data (time stamp from a trusted system component, Severity rating for each event, Tagging of security relevant events if they are mixed with other log entries, Identity of the account/user that caused the event, Source IP address associated with the request, Event outcome (success or failure), Description of the event).	Mandatory
5.5	Restrict access to logs to only authorized individuals.	Mandatory
5.6	Do not store sensitive information in logs, including unnecessary system details, session identifiers or passwords.	Mandatory
5.7	Log all input validation failures.	Recommended
5.8	Log all transactional events that take place made by the administrator.	Mandatory
5.9	Log all authentication attempts, especially failures.	Mandatory
5.10	Log all access control failures.	Recommended
5.11	Log all system exceptions.	Recommended
5.12	Log all administrative functions, including changes to the security configuration settings.	Mandatory
5.13	System log must support and integrated with Telkomsel centralized log management system.	Mandatory
6	**Data Protection**	
6.1	Sensitive data processed by application shall be identified. This data shall be encrypted (both at rest and in transit).	Mandatory
6.2	Implement least privilege; restrict users to only the functionality, data and system information that is required to perform their tasks.	Mandatory
6.3	Encrypt highly sensitive stored information, like authentication verification data, even on the server side.	Mandatory
6.4	Protect server-side source-code from being downloaded by a user	Mandatory
6.5	Do not store passwords, connection strings or other sensitive information in clear text or in any non- cryptographically secure manner on the client side.	Mandatory
6.6	Critical information about system components (e.g. server name, version, installed program versions, etc.) of web, application and database servers should be obscured and not revealed via HTTP responses or error messages.	Mandatory

6.7	When applications are transferred from a development/integration environment into a production environment, unnecessary resources (e.g. test codes, demo applications, backup files) should be excluded. Source files should be excluded as well if they are not required. Comments should be removed from source files.	Mandatory
6.8	Do not include sensitive information in HTTP GET request parameters.	Mandatory
6.9	Implement appropriate access controls for sensitive data stored on the server. This includes cached data, temporary files and data that should be accessible only by specific system users.	Mandatory
7	**Security Configuration**	
7.1	All security-relevant configuration information shall be stored in locations that are protected from unauthorized access.	Mandatory
7.2	All changes to the security configuration settings managed by the application shall be logged in the security event log.	Mandatory
7.3	The configuration store shall be able to be output in a human-readable format to facilitate audit.	Recommended
8	**Cryptography**	
8.1	Use proven cryptographic services, either provided by the platform or a trusted third party library.	Mandatory
8.2	Salt value should be used as well by the generation of password hashes.	Mandatory
8.3	Cryptographic keys should be managed securely. Establish and utilize a policy and process for how cryptographic keys will be managed.	Mandatory
8.4	Access to any master secret(s) shall be protected from unauthorized access (A master secret is an application credential stored as plaintext on disk that is used to protect access to security configuration information).	Mandatory
9	**Communication Security**	
9.1	Implement encryption for the transmission of all sensitive information (e.g. authentication credential). This should include TLS/SSL for protecting the connection and may be supplemented by discrete encryption for non-HTTP based connections.	Mandatory

9.2	TLS/SSL certificates should be valid and have the correct domain name, not be expired, and be installed with intermediate certificates.	Mandatory
9.3	Failed TLS/SSL connections should not fall back to an insecure connection.	Mandatory
9.4	Utilize TLS/SSL connections for all content requiring authenticated access and for all other sensitive information.	Mandatory
9.5	Utilize TLS/SSL for connections to external systems that involve sensitive information or functions.	Mandatory
10	**System Configuration**	
10.1	Ensure servers, frameworks and system components are running the latest approved version.	Mandatory
10.2	Ensure servers, frameworks and system components have all patches issued for the version in use.	Mandatory
10.3	Turn off directory listings.	Mandatory
10.4	Restrict the web server, process and service accounts to the least privileges possible.	Mandatory
10.5	Remove all unnecessary functionality and files.	Mandatory
10.6	Remove test code or any functionality not intended for production, prior to deployment.	Mandatory
10.7	Prevent disclosure of your directory structure in the robots.txt file by placing directories not intended for public indexing into an isolated parent directory. Then “Disallow” that entire parent directory in the robots.txt file rather than Disallowing each individual directory.	Mandatory
10.8	Disable unnecessary server side function (e.g. upload file or system execution) or HTTP methods expect GET and POST, such as WebDAV extensions. If an extended HTTP method that supports file handling is required, utilize a well-vetted authentication mechanism.	Mandatory
10.9	Remove unnecessary information from HTTP response headers related to the OS, web-server version and application frameworks.	Mandatory
10.10	Implement an asset management system and register system components and software in it.	Mandatory
10.12	Implement a software change control system to manage and record changes to the code both in development and production.	Recommended
11	**Database Security**	

11.1	Use strongly typed parameterized queries.	Mandatory
11.2	The application should use the lowest possible level of privilege when accessing the database.	Mandatory
11.3	Connection strings should not be hard coded within the application. Connection strings should be stored in a separate configuration file on a trusted system and they should be encrypted.	Mandatory
11.4	Remove or change all default database administrative passwords. Utilize strong passwords/phrases or implement multi-factor authentication.	Mandatory
11.5	Turn off all unnecessary database functionality (e.g., unnecessary stored procedures or services, utility packages, install only the minimum set of features and options required (surface area reduction)).	Recommended
11.6	Remove unnecessary default vendor content (e.g., sample schemas).	Mandatory
11.7	Disable any default accounts that are not required to support business requirements.	Mandatory
11.8	Database user should be able access to the database server only through the relevant application server IP address.	Recommended
12	**File Management**	
12.1	Do not pass user supplied data directly to any dynamic include function.	Mandatory
12.2	Require authentication before allowing a file to be uploaded.	Mandatory
12.3	Limit the type of files that can be uploaded to only those types that are needed for business purposes.	Mandatory
12.4	Validate uploaded files are the expected type by checking file headers. Checking for file type by extension alone is not sufficient.	Mandatory
12.5	Do not save files in the same web context as the application. Files should either go to the content server or in the database.	Mandatory
12.6	Prevent or restrict the uploading of any file that may be interpreted by the web server.	Mandatory
12.7	Turn off execution privileges on file upload directories.	Mandatory
12.8	Do not pass directory or file paths, use index values mapped to pre-defined list of paths.	Mandatory
12.9	Never send the absolute file path to the client.	Mandatory
12.10	Ensure application files and resources are read-only.	Mandatory

13	**General Coding Practice**	
13.1	Use tested and approved managed code rather than creating new unmanaged code for common tasks.	Recommended
13.2	Utilize task specific built-in APIs to conduct operating system tasks. Do not allow the application to issue commands directly to the Operating System, especially through the use of application initiated command shells.	Recommended
13.3	Explicitly initialize all your variables and other data stores, either during declaration or just before the first usage.	Recommended
13.4	Do not pass user supplied data to any dynamic execution function.	Mandatory
13.5	Restrict users from generating new code or altering existing code.	Mandatory
14	**Miscellaneous**	
14.1	Support High Availability (HA) architecture implementation.	Recommended
14.2	NIK (Nomor Induk Karyawan) or Employee Identity Number shall be used as a required User ID information.	Mandatory
14.3	Have an API for Telkomsel IDM (Identity Management) System Integration.	Mandatory
14.4	Follow deployment configuration standard for OS & Database.	Mandatory
14.5	For public or internet facing application, it is recommended to place in elkomsel.com domain	Recommended