KEBIJAKAN KEAMANAN SERVER & JARINGAN

UNIVERSITAS ‘AISYIYAH YOGYAKARTA

Kampus Terpadu:
Jl. Siliwangi (Ring Road Barat) No. 63 Mlangi, Nogotirto, Gamping, Sleman, Yogyakarta. 55292 Telepon: (0274) 4469199 Fax.: (0274) 4469204
Email: info@unisayogya.ac.id

KEBIJAKAN KEAMANAN SERVER & JARINGAN UNIVERSITAS ‘AISYIYAH YOGYAKARTA 2021

disusun oleh:

Badan Pengembangan Teknologi dan Sistem Informasi

Kampus Terpadu:
Jl. Siliwangi (Ring Road Barat) No. 63 Mlangi, Nogotirto, Gamping, Sleman, Yogyakarta. 55292 Telepon: (0274) 4469199 Fax.: (0274) 4469204
Email: info@unisayogya.ac.id

Lembar Pengesahan

KEBIJAKAN KEAMANAN SERVER & JARINGAN
UNIVERSITAS 'AISYIYAH YOGYAKARTA
2021

Yogyakarta, 15 Februari 2021

Disahkan oleh : Rektor	Disiapkan oleh : Kepala BPTSI
Warsiti, S. Kp., M. Kep., Sp. Mat.	Basit Adhi Prabowo, S.T.

Daftar Isi

KEBIJAKAN KEAMANAN SERVER & JARINGAN

Bab I
Ketentuan Umum

Pasal 1
Terminologi

Dalam Kebijakan Keamanan Server & Jaringan ini, yang dimaksud dengan: --

1. UNISA Yogyakarta adalah Universitas 'Aisyiyah Yogyakarta --
2. BPTSI adalah Badan Pengembangan Teknologi Sistem Informasi UNISA Yogyakarta, atau dahulu bernama --
    a. Pusat Data dan Sistem Informasi (PDSI) --
    b. Bagian Pengembangan Teknologi Informasi (BPTI) --
    c. Electronic Data Processing (EDP) --
3. *Software* adalah perangkat lunak --
4. *Programmer* adalah tenaga kependidikan yang bertugas membuat *software* --
5. LAN adalah *Local Area Network*, koneksi jaringan secara lokal di dalam jaringan institusi --
6. WAN adalah *Wide Area Network*, koneksi jaringan dari/ke luar jaringan institusi --
7. *Remote* adalah kegiatan menjalankan sesuatu secara tidak langsung, baik dari LAN maupun WAN --
8. VPN adalah *Virtual Private Network*, komunikasi terenkripsi yang melalui jaringan lain menggunakan kunci yang hanya diketahui oleh pengirim dan penerima --
9. Basis data adalah media simpan data yang dapat diakses melalui jarigan --
10. *Cloud storage* adalah media simpan berkas yang dapat diakses melalui jaringan --
11. API adalah *Application Programming Interface* adalah seperangkat aturan dan spesifikasi tertentu yang dapat diikuti oleh program/perangkat lunak untuk berkomunikasi satu sama lain --
12. *Clean code* adalah kode *software* yang formatnya benar dan disusun dengan baik dan rapi agar kode programnya mudah dibaca, dimengerti, ditelusuri oleh

siapapun dan mudah untuk diubah oleh siapapun jika nantinya akan ada perubahan --

13. *Sanitizing/filtering* adalah proses untuk menyaring masukan agar sesuai format atau terbebas dari konten untuk menyerang/menembus keamanan --
14. *Salt* adalah data/kata acak yang digunakan sebagai tambahan masukan untuk membuat *hash* --
15. *Hash* adalah data/kata yang disandikan/diacak, semacam dengan enkripsi, tetapi tidak dapat dibalik --
16. *Firewall* adalah sistem keamanan pada jaringan komputer yang dipakai untuk melindungi jaringan/komputer dari serangan komputer luar --
17. *Tools* adalah *software* untuk membantu melakukan suatu aksi --
18. *Auto-update* adalah sistem pembaharuan yang terjadi secara otomatis --
19. *Port (logical)* adalah nomor jalur komunikasi pada jaringan yang dapat digunakan oleh perangkat lunak untuk menghubungkan layanannya dengan jaringan --
20. *Checklist* adalah daftar item untuk diperiksa dan ditandai apabila sudah memenuhi kualifikasi/lolos pemeriksaan --

Pasal 2
Ruang Lingkup

Kebijakan ini mencakup keamanan server dan jaringan, termasuk --

1. Pengelola --
2. Sistem operasi yang dipasang --
3. *Software* yang dipasang, baik dibuat sendiri maupun yang dibuat oleh pihak ketiga --
4. Data yang berada di *server* --
5. Perangkat keras *server* dan jaringan --
6. Ruangan *server* dan jaringan --

--

--

--

--

--

--

--

Bab II

Keamanan

Pasal 3

Pengelola

Setiap tenaga kependidikan BPTSI sebagai pengelola *server* dan jaringan: --

1. Wajib menandatangani Pakta Integritas dan mematuhi apa yang ada di dalamnya, yakni --
    a. menyimpan keamanan data Sistem Informasi, baik *software* maupun dokumen di unit-unit kerja UNISA Yogyakarta --
    b. tidak akan melakukan komunikasi yang mengarah terhadap pembocoran kode data *software* unit kerja di UNISA Yogyakarta kepada pihak manapun --
    c. menyimpan rahasia keamanan data Sistem Informasi UNISA Yogyakarta jika sudah tidak bekerja sebagai *programmer* di UNISA Yogyakarta --
2. Memenuhi kode etik pegawai UNISA Yogyakarta pada umumnya dan kode etik *programmer* pada khususnya --
3. Diperbolehkan membuat tutorial atau mempublikasikan material di web, tetapi wajib menyamarkan informasi sensitif (rahasia) seperti *username*, *password*, *path* dan informasi sensitif (rahasia) lainnya, kecuali yang sudah umum atau standar, misalnya *path* /var/www --
4. Menggunakan perangkat yang dipastikan bebas *malware* atau aplikasi yang rentan terhadap keamanan --

Pasal 4

Komunikasi *Server*

Ketentuan di bawah ini berlaku untuk komunikasi melalui LAN maupun WAN. Ketentuan tidak berlaku untuk komunikasi internal (*localhost*) --

1. Setiap komunikasi *server* harus menggunakan protokol keamanan, diantaranya adalah menggunakan --
    a. SSL, --
    b. TLS, --
    c. atau protokol lain yang setara --

2. Setiap komunikasi server harus menggunakan implementasi protokol keamanan, diantaranya adalah menggunakan --
    a. HTTPS, --
    b. SSH, --
    c. atau metode lain --
3. Setiap kegiatan *remote* semaksimal mungkin --
    a. melalui LAN, atau jika hanya dapat menggunakan WAN maka semaksimal mungkin menggunakan sambungan VPN --
    b. menggunakan perangkat milik sendiri --
4. Basis data dan *cloud storage* --
    a. tidak terekspos secara langsung melalui WAN --
    b. hanya bisa diakses melalui --
        i. LAN --
        ii. WAN menggunakan VPN --
        iii. *Server* --
        iv. API --

Pasal 5
Basis Data

1. Akses basis data dari pihak ketiga ke basis data UNISA Yogyakarta hanya boleh melalui API --
2. Akses basis data dari *software* yang dibuat sendiri tidak boleh menggunakan akun admin/root dari basis data --
3. Hak setiap akun basis data --
    a. Setiap skema/*database* dibuatkan *username* dan *password* sendiri dengan hak yang terbatas di bawah hak admin/root, antara lain hanya boleh[1] --
        i. *Delete*: menghapus baris pada tabel --
        ii. *Execute*: menjalankan *stored procedure* dan *function* --
        iii. *Insert*: menyisipkan baris ke tabel, kemampuan menjalankan *Analyze Table*, *Optimize Table* dan *Repair Table* --
        iv. *Lock Table*: mengunci tabel --
        v. *Select*: menampilkan data --
        vi. *Update*: memperbarui data pada tabel --

[1] Contoh untuk MySQL --

b. Apabila diperlukan dapat ditingkatkan haknya disertai dengan alasan, antara lain: --
   i. *Create*: membuat tabel --
   ii. *Drop*: menghapus tabel --
   iii. *Index*: membuat atau menghapus indeks pada tabel --
   iv. Hak lainnya --

Pasal 6

Software/Program Yang Dibuat Sendiri

1. Pesan *error* --
   a. Pada saat pengembangan boleh ditampilkan --
   b. Pada saat produksi/*live* dimatikan atau disimpan dalam *log* --
2. Mengimplementasikan *clean code* dan *sanitizing/filtering* untuk meminimalisir celah keamanan --
3. Mengimplementasikan *password* yang ter-*hash* dengan ketentuan --
   a. *Salt* dapat berupa *plain text*[2] atau *hashed text*[3]. *Salt* dapat berupa --
      i. *Salt* statis[4]: --
         sama setiap saat --
      ii. *Salt* dinamis[5]: --
         1. berubah-ubah setiap beberapa saat atau selalu berubah setiap proses --
         2. *salt* dikirim pada saat *request* --
   b. Fungsi *hash*[6] dapat berupa --
      i. Fungsi *hash* yang standar[7]: --
         wajib dengan *salt* --

--

--

---

[2] *salt* = '*abc*' --
[3] *salt* = *hash*('*abc*') --
[4] Umumnya ada di berkas konfigurasi, misalnya pada wp-config.php untuk *Wordpress* --
[5] Misal: --
salt = *uniqid*() --
pass = *hash*(*concat*(*password*, *salt*)) --
header(*concat*('salt: ', *salt*)) --
[6] Enkripsi dengan *salt*: *hash*(*concat*(*password*, *salt*)) --
[7] Hash standar *general purpose*: md5, sha256, dsb
Hash standar *password-hashing*: argon2, scrypt, bcrypt, dsb

ii. Fungsi *hash* yang dimodifikasi: --
sebaiknya menggunakan *salt* --
1. Menggunakan fungsi *hash* yang sama beberapa kali[8] --
2. Kombinasi dan/atau modifikasi beberapa fungsi *hash*[9] --

c. Metode pengiriman *password* atau kredensial lain tidak boleh melalui URL (metode GET, contoh: https://example.org?pass=123). Meskipun bagian GET di-enkripsi ketika dikirim, tetapi *password* bisa terekspos di riwayat klien. *Salt* boleh dikirim melalui URL (metode GET) --

4. Apabila ada aplikasi di sisi klien yang membutuhkan otentikasi, misalnya pada aplikasi *mobile*, maka: --
   a. menerapkan single sign-on, atau --
   b. boleh meninggalkan kode[10] di dalam berkas konfigurasi ditambah dengan *hash* menggunakan *salt+unique_identifier*[11]. *unique_identifier* didapatkan dari *device/apps* dan tidak boleh disimpan di dalam berkas konfigurasi agar berkas konfigurasi tidak bisa dipakai di *device* lain. Harapannya adalah otentikasi tidak perlu lagi dilakukan ke basis data, tetapi cukup mencocokkan kode dengan *hash+salt+unique_identifier* di API atau di aplikasi mobile itu sendiri[12] --

Pasal 7
Server

1. Menggunakan sistem operasi khusus *server*, kecuali untuk *software* untuk kepentingan eksternal yang tidak dapat dipasang di sistem operasi khusus *server* --
2. Dipasang anti-virus, *firewall* dan *tools* keamanan lainnya --
3. Dipasang mekanisme monitoring/*alert* otomatis --
4. Sistem Operasi dan *software* tidak dalam status *auto-update*, kecuali yang tidak berpotensi *crash*, seperti anti-virus. Proses update dilakukan setelah melakukan simulasi *update* atau membaca *release note* untuk melihat tidak ada *crash* setelah *update* --

[8] Misal: *hash1(hash1(password))* atau *hash1(hash1(concat(password, salt)))* --
[9] Misal: *left(hash1(hash2(password)), 12)* atau *left(hash1(hash2(concat(password, salt))), 12)* --
[10] Misal: nim, kodemahasiswa, dll
[11] Misal: 1810201001_ZkjMd, di mana ZkjMd merupakan *hash(concat('1810201001', salt, unique_identifier))* --
[12] Misal: 1810201001_ZkjMd, apakah *hash(concat('1810201001', salt, unique_identifier))* sama dengan ZkjMd

5. Tidak memasang *software* yang tidak diperlukan --
6. Menutup semua *port*, kecuali yang digunakan --
7. Tidak dapat di-*remote* dengan akun admin/root, setiap tendik BPTSI dibuatkan 1 akun untuk setiap *server* --
8. Menyimpan semua *password server* pada tempat yang aman --

Pasal 8

Ruang Server

1. Desain ruang *server* yang aman dan sesuai dengan standar --
2. Sumber daya aman (UPS, backup jaringan, dsb) --
3. Hak akses yang ketat ke ruang *server* secara fisik --

Pasal 9

Pencadangan (*Backup*)

Mekanisme pencadangan (*backup*)

1. Pencadangan basis data (*backup*) dilakukan secara reguler --
2. Menggunakan *cloud storage* untuk menyimpan data berupa media dan menerapkan pencadangan:
    a. Partisi 100% data, *backup* menggunakan *cloud storage* di tempat lain, atau
    b. Partisi 50% data, 50% *backup* --
3. Menerapkan *High Availability* (*cloud computing*), termasuk pencadangan *image server* --

Pasal 10

Uji Coba/Perawatan

1. Uji coba yang dapat dilakukan antara lain: --
    a. Uji ketahanan *server* terhadap beban (*load test*, *benchmark*), sekaligus menguji dan mengatur ulang pengaturan *server* --
    b. Uji perangkat keras dengan *tools* bawaan sistem operasi atau *tools* pihak ketiga berlisensi --
    c. Uji penetrasi (*pen test*) untuk memeriksa celah keamanan --
    d. Uji perangkat yang dibutuhkan *server* (jaringan, UPS, dll) --
    e. Uji *malware* dengan *server dummy* --

f. Dan uji lainnya yang dianggap perlu --

2. Uji coba dapat dilakukan sendiri atau kerjasama dengan pihak ketiga --
3. Uji coba merupakan bagian dari perawatan --

Bab III
Implementasi

Pasal 11
Bentuk dan Jadwal Implementasi

1. Kebijakan ini diwujudkan dalam bentuk *checklist* yang dapat digunakan satu kali atau rutin untuk setiap item. *Checklist* dibuat terpisah dari kebijakan ini, serta dapat ditinjau kembali dan diperbaharui --
2. Waktu dan pelaksanaan kebijakan ini disesuaikan dengan situasi dan kondisi --

Bab IV
Penutup

Pasal 12
Penutup

Kebijakan Keamanan Server & Jaringan ini agar dapat dijadikan panduan bagi tenaga kependidikan BPTSI dalam mengamankan *server* dan jaringan UNISA Yogyakarta. --
--
--
--
--
--
--
--
--
--