

**LAPORAN KERJA PRAKTEK**

**PEMBANGUNAN PUSTAKA PERANGKAT LUNAK**

**DOMAIN CHECKER UNTUK PT.TARGET SUKSES SINERGI**

Laporan ini disusun sebagai salah satu syarat pada mata Kuliah Kerja Praktek

pada Program Studi Teknik Informatika – S1 Fakultas Ilmu Komputer

Universitas Dian Nuswantoro

**Oleh :**

**NAMA : MUHAMMAD HAFIDZ**

**NIM : A11.2014.08602**

**Program Studi : Teknik Informatika-S1**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS DIAN NUSWANTORO**

**SEMARANG**

**2018**

# PERSETUJUAN PEMBIMBING KERJA PRAKTEK

PEMBANGUNAN PUSTAKA PERANGKAT LUNAK DOMAIN CHECKER

UNTUK PT. TARGET SUKSES SINERGI

Oleh :

**Nama : Muhammad Hafidz**

**NIM : A11.2015.09000**

**Jurusan : Teknik Informatika-S1**

Telah diperiksa dan disetujui oleh pembimbing kerja praktek untuk memenuhi salah satu syarat mata kuliah Kerja Praktek pada program studi Teknik Informatika-S1 Fakultas Ilmu Komputer Universitas Dian Nuswantoro Semarang.

|  |  |
| --- | --- |
| **Semarang, 29 Mei 2018** | |
| **Menyetujui** | |
| **Penyelia**  **VINCENSIUS WICAKSONO A.N.**  **LEAD BACKEND DEVELOPER** | **Pembimbing KP**  **FAHRI FIRDAUSILLAH S.Kom M.CS**  **NPP: 0686.11.2009.354** |

# ABSTRAK

Email merupakan media komunikasi bagi pengguna dan penyedia jasa Internet yang efektif. Perusahaan-perusahaan besar mayoritas menggunakan email sebagai media komunikasi dengan dengan para pelangganya. Namun tidak semua email yang dikirim dapat sampai ke kotak masuk email para pelanggan. Banyak faktor yang mempengaruhi hal tersebut, diantaranya karena konten yang tidak sesuai kaidan penulisan yang baik, alamat email yang tidak valid, domain pengguna yang terdaftar dalam Blacklist dan sebagainya. Berdasarkan hal tersebut, diperlukan adanya perangkat lunak yang digunakan untuk pengecekan data email yang akan dikirm untuk meningkatkan kemungkinan email sampai ke pelanggan. Penelitian ini menghasilkan sebuah pustaka perangkat lunak validator yang dikembangkan menggunakan bahasa pemrograman Kotlin. Melalui pustaka ini, PT. Target Sukses Sinergi dapat dimudahkan dalam proses pengiriman email marketing.

Kata kunci: pustaka perangkat lunak, filter spam, email.

# KATA PENGANTAR

Puji syukur kepada Tuhan yang Maha Esa atas segala berkat yang telah diberikan-Nya, sehingga Laporan Kerja Praktek ini dapat diselesaikan.Laporan Kerja Praktek dengan judul “Pembangunan Pustaka Perangkat Lunak Untuk PT. Target Sukses Sinergi” ini ditujukan untuk memenuhi sebagian persyaratan akademik guna menyelesaikan studi di Program Studi Teknik Informatika Strata Satu Universitas Dian Nuswantoro Semarang.

Penulis menyadari bahwa tanpa bimbingan, bantuan, dan doa dari berbagai pihak, Laporan Kerja Praktek ini tidak akan dapat diselesaikan tepat waktunya. Oleh karena itu, penulis mengucapkan terima kasih yang sebesar-besarnya kepada semua pihak yang telah membantu dalam proses penulisan Laporan Kerja Praktek ini, yaitu kepada:

1. Bapak Prof., Dr., Ir Edi Noersasongko M.Kom., selaku Rektor Universitas Dian Nuswantoro.
2. Bapak Dr Drs Abdul Syukur MM., selaku Dekan Fakultas Ilmu Komputer.
3. Bapak Heru Agus Santoso Ph.D., selaku Ketua Program Studi Teknik Informatika-S1.
4. Fahri Firdausillah S.Kom, M.CS selaku Dosen Pembimbing Kerja Praktek yang telah memberikan masukan kepada penulis.
5. Masas Dani, selaku CTO PT. Target Sukses Sinergi yang berkenan memberi izin penulis untuk melaksanakan kerja praktik di tempat yang Bapak pimpin.
6. Vincensius Wicaksono Adi Nugroho, selaku pembimbing kerja praktik di PT. Target Sukses Sinergi yang sudah sabar memberi panduan serta masukan selama kerja praktik berlangsung.
7. Ibu, Bapak, kakak serta adik yang telah memberikan dukungan moril, doa, dan kasih sayang.
8. Semua teman-teman yang namanya tidak dapat disebutkan satu per satu.

Akhir kata, penulis menyadari bahwa mungkin masih terdapat kekurangan dalam Laporan Kerja Praktek ini. Oleh karena itu, kritik dan saran dari pembaca sangat bermanfaat bagi penulis. Semoga Laporan Kerja Praktek ini dapat bermanfaat bagi semua pihak yang membacanya.

Semarang, 29 Mei 2018

            Muhammad Hafidz

# DAFTAR ISI

[PERSETUJUAN PEMBIMBING KERJA PRAKTEK ii](#_Toc521532111)

[ABSTRAK iii](#_Toc521532112)

[KATA PENGANTAR iv](#_Toc521532113)

[DAFTAR ISI vi](#_Toc521532114)

[DAFTAR GAMBAR viii](#_Toc521532115)

[DAFTAR TABEL ix](#_Toc521532116)

[DAFTAR LAMPIRAN x](#_Toc521532117)

[BAB I PENDAHULUAN 1](#_Toc521532118)

[1.1 Latar Belakang 1](#_Toc521532119)

[1.2 Rumusan Masalah 1](#_Toc521532120)

[1.3 Batasan Masalah 2](#_Toc521532121)

[1.4 Tujuan Kerja Praktek 2](#_Toc521532122)

[1.5 Manfaat Kerja Praktek 2](#_Toc521532123)

[BAB II LANDASAN TEORI 3](#_Toc521532124)

[2.1 Pustaka Perangkat Lunak 3](#_Toc521532125)

[2.2 Kotlin 3](#_Toc521532126)

[2.3 DNS dan Tipe DNS 4](#_Toc521532127)

[2.4 Jenis-jenis catatan DNS 5](#_Toc521532128)

[2.5 DNSBL 6](#_Toc521532139)

[BAB III TEMPAT KERJA PRAKTEK 7](#_Toc521532140)

[3.1 Profil Perusahaan 7](#_Toc521532141)

[3.2 Visi dan Misi 7](#_Toc521532142)

[3.2.1 Visi Perusahaan 7](#_Toc521532143)

[3.2.2 Misi Perusahaan 8](#_Toc521532144)

[3.3 Struktur Organisasi 8](#_Toc521532145)

[3.4 Layanan Perusahaan 8](#_Toc521532146)

[BAB IV HASIL DAN PEMBAHASAN 11](#_Toc521532147)

[4.1 Gambaran Umum Sistem 11](#_Toc521532148)

[4.2 Software & Layanan yang Digunakan 11](#_Toc521532149)

[4.3 Implementasi 13](#_Toc521532150)

[4.3.1 Ruang Lingkup Perangkat Keras 13](#_Toc521532151)

[4.3.2 Ruang Lingkup Perangkat Lunak 13](#_Toc521532151)

[BAB V PENUTUP 14](#_Toc521532152)

[5.1 Kesimpulan 14](#_Toc521532153)

[5.2 Saran 14](#_Toc521532154)

[DAFTAR PUSTAKA 15](#_Toc521532155)

[Lampiran 1. Foto Kegiatan 16](#_Toc521532156)

# DAFTAR GAMBAR

Use Case Diagram 15

Class Diagram 16

*Sequence Diagram : Add DNSBL* 16

*Sequence Diagram : Delete DNSBL* 17

*Sequence Diagram : List DNSBL* 17

*Sequence Diagram : Init DNSBL* 18

*Sequence Diagram : Check DNSBL* 18

*Halaman Domain Checker* 20

*Halaman DNSBL* 20

Pengujian Domain Checker pada Web 22

*Kode Program Domain Checker* 22

*Implementasi Kode Program Domain Checker pada Unit Testing* 23

*Log Hasil Pengecekan Domain 1* 23

*Log Hasil Pengecekan Domain 2* 24

# DAFTAR LAMPIRAN

Lampiran 1. Foto Kegiatan 33

# BAB I PENDAHULUAN

## 1.1 Latar Belakang

Email merupakan media komunikasi bagi pengguna dan penyedia jasa Internet yang efektif. Perusahaan-perusahaan besar mayoritas menggunakan email sebagai media komunikasi dengan dengan para pelangganya. Namun tidak semua email yang dikirim dapat sampai ke kotak masuk email para pelanggan. Banyak faktor yang mempengaruhi hal tersebut, salah satunya adalah domain pengguna yang terdaftar dalam Domain Blacklist.

Domain Blacklist merupakan adalah daftar alamat IP yang dicurigai mengirim spam dan digunakan untuk mencegah pesan email yang tidak diinginkan mencapai penerima yang tidak curiga. Satu hal yang penting untuk disebutkan adalah blacklist itu sebenarnya tidak memblokir pesan Anda, tetapi justru penyedia kotak surat yang sebenarnya. Penyedia ini menggunakan informasi ini dari berbagai layanan daftar hitam bersama dengan metrik internal untuk membuat keputusan tentang apakah atau tidak untuk memblokir pesan.

Saat ini, produk MailTarget dibawah pengembangan PT. Target Sukses Sinergi, belum memiliki pustaka perangkat lunak untuk pengecekan data domain berbasis pengecekan Domain Blacklist. Sedangkan perangkat lunak yang serupa hanya tersedia selain pada bahasa pemrograman Kotlin.

Berdasarkan hal tersebut, diperlukan adanya pustaka perangkat lunak berbahasa pemrograman kotlin yang digunakan untuk pengecekan data domain berbasis pengecekan Domain Blacklist yang akan dikirm untuk meningkatkan kemungkinan email sampai ke pelanggan. Penelitian ini menghasilkan sebuah pustaka perangkat lunak validator yang dikembangkan menggunakan bahasa pemrograman Kotlin. Melalui pustaka ini, PT. Target Sukses Sinergi dapat dimudahkan dalam proses pengiriman email marketing.

## 1.2 Rumusan Masalah

Permasalahan yang akan diselesaikan adalah bagaimana membangun modul pengecekan domain sehingga dapat meningkatkan untuk meningkatkan kemungkinan email sampai ke pelanggan pengguna layanan PT Target Sukses Sinergi.

## 1.3 Batasan Masalah

Adapun batasan masalah yang akan dibahas adalah :

1. Modul yang dibangun berbasis pustaka perangkat lunak dengan menggunakan bahasa pemrograman *Kotlin.*
2. Modul ini akan digunakan pada PT. Target Sukses Sinergi.

## 1.4 Tujuan Kerja Praktek

Adapun tujuan yang ingin dicapai dari penelitian ini adalah membangun modul pengecekan domain menggunakan bahasa pemrograman Kotlin sehingga dapat meningkatkan kemungkinan email sampai ke pelanggan pengguna layanan PT Target Sukses Sinergi dari awal proses bisnis hingga selesai.

## 1.5 Manfaat Kerja Praktek

Setelah melaksanakan kerja praktek, ada beberapa manfaat yang dirasakan antara lain :

1. Bagi Objek Kerja Praktek

* Pustaka perangkat lunak pengecekan domain sehingga dapat meningkatkan validitas data pengguna layanan PT Target Sukses Sinergi.

# BAB II LANDASAN TEORI

## 2.1 Surat Elektronik

Surat elektronik (akronim: ratel, ratron, surel, atau surat-e) atau pos elektronik (akronim: pos-el.) atau imel (bahasa Inggris: *email*) adalah sarana kirim mengirim surat melalui jalur jaringan komputer (misalnya Internet).

Struktur alamat surel, sebagai contoh:

|  |
| --- |
| surelsaya@surabaya.vibriel.net.id |

Keterangan:

* surelsaya: nama kotak surat (*mailbox*) atau nama pengguna (*username*) yang ingin dituju dalam *mailserver*
* surabaya.vibriel.net.id: nama *mailserver* tempat pengguna yang dituju, rinciannya:
  + surabaya: *subdomain* (milik pemegang nama *domain*), biasanya merujuk ke suatu komputer dalam lingkungan pemilik *domain*
  + vibriel: nama *domain*, biasanya menunjukkan nama perusahaan/organisasi/perorangan (Vibriel)
  + net: *second level domain*, menunjukkan bahwa *domain* ini termasuk kategori *networking* (net)
  + id: *top level domain*, menunjukkan bahwa *domain* ini terdaftar di otoritas *domain* Indonesia (id)

Untuk mengirim surat elektronik kita memerlukan suatu program *mail-client*. Surat elektronik yang kita kirim akan melalui beberapa poin sebelum sampai di tujuan. Mulai dari surat elektronik dikirim → Internet → POP3 server penyedia surel penerima → *mail client* (di komputer si penerima) → surat elektronik dibaca si penerima

Terlihat surat elektronik yang terkirim hanya melalui 5 poin (selain komputer pengirim dan penerima). Sebenarnya lebih dari itu sebab setelah surat elektronik meninggalkan POP3 Server maka itu akan melalui banyak server-server lainnya. Tidak tertutup kemungkinan surat elektronik yang kita kirim disadap orang lain. Maka dari itu bila surat elektronik yang kita kirim mengandung isi yang sensitif sebaiknya kita melakukan tindakan pencegahan, dengan mengacak (enkripsi) data dalam surat elektronik tersebut (contohnya menggunakan PGP, sertifikat digital, dan lain-lain).[1]

Post Office Protocol 2 (POP3) adalah sebuah mail access protocol yang digunakan oleh aplikasi klien untuk membaca pesan dari mail server. Pesan yang diterima biasanya dihapus dari server. POP mendukung fitur unduh dan hapus untuk mengakses kotak masuk.[2]

## 

## 2.2 Email Marketing

Email Marketing adalah email yang biasanya dikirim ke sekelompok orang dengan tujuan komersial. Email marketing sendiri biasanya memiliki konten berupa iklan, pengajuan kegiatan bisnis, permintaan penjualan atau donasi, dan dimaksudkan untuk membangun kesetiaan, kepercayaan, atau brand awareness. Email marketing dapat dikirim ke daftar prospek yang dibeli atau database pelanggan saat ini. Email marketing biasanya mengacu pada pengiriman pesan email dengan tujuan meningkatkan hubungan pedagang dengan pelanggan saat ini atau sebelumnya, mendorong loyalitas pelanggan dan mengulang bisnis, memperoleh pelanggan baru atau meyakinkan pelanggan saat ini untuk membeli sesuatu dengan segera, dan berbagi iklan pihak ketiga.

Email marketing populer di kalangan perusahaan karena lebih murah dan lebih cepat daripada surat tradisional. Bisnis dan organisasi yang mengirim email bervolume tinggi dapat menggunakan ESP (Email Service Provider) untuk mengumpulkan informasi tentang perilaku penerima. Hasil yang diberikan oleh tanggapan konsumen terhadap email marketing dapat membantu bisnis dan organisasi memahami dan memanfaatkan perilaku konsumen. Hampir separuh pengguna internet di Amerika memeriksa atau mengirim email pada hari-hari biasa,[4] dengan email yang dikirim antara pukul 1 pagi hingga 5 pagi waktu setempat mengalahkan yang dikirim pada waktu lain.[5][6]

## 2.3 Bounce Message

Bounce Message adalah email otomatis yang memberi tahu pengirim pesan sebelumnya bahwa pesan itu belum terkirim (atau masalah pengiriman lainnya terjadi). Pengirim terkadang menerima bounce message dari server emailnya sendiri, melaporkan bahwa ia tidak dapat mengirim pesan, atau meskipun telah menerima pesan tersebut, sekarang pesan itu tidak dapat dikirim, juga menerima tanggung jawab untuk mengirimkan DSN jika pengiriman gagal. Karena berbagai alasan, terutama spam dan email virus, pengguna mungkin menerima pesan pentalan yang salah dikirim sebagai tanggapan terhadap pesan yang sebenarnya tidak pernah mereka kirim.[4]

## 2.4 Pustaka Perangkat Lunak

**Pustaka** (atau ***Library*** dalam *Bahasa Inggris*), dalam ilmu komputer adalah koleksi dari rutin-rutin program yang digunakan untuk membangun dan mengembangkan perangkat lunak. Pustaka, umumnya mengandung kode program dan data pembantu (banyak *programmer* menyebutnya sebagai *helper*), yang menyediakan layanan-layanan kepada program-program independen. Hal ini mengizinkan kode dan data dapat digunakan bersama-sama dan dapat diubah dengan menggunakan paradigma pemrograman moduler. Beberapa berkas yang dapat dieksekusi, umumnya berjenis program yang berdiri sendiri atau juga pustaka, tetapi kebanyakan pustaka tidak dapat dieksekusi. Program komputer dapat merujuk kepada pustaka dengan menggunakan sebuah proses yang disebut *linking*, yang mampu menghubungkan satu dengan lainnya, yang dapat dilakukan dengan *linker*.

Sebagian besar sistem operasi modern menyediakan banyak pustaka untuk mengimplementasikan banyak layanan-layanan sistem operasi. Windows API adalah contohnya. Pustaka jenis ini akan menyediakan kebutuhan yang dibutuhkan oleh aplikasi dari sistem operasi dimana aplikasi tersebut berjalan. Karenanya, sebagian besar kode digunakan oleh aplikasi modern disediakan dalam pustaka sistem operasi.[4]

## 

## 2.5 Kotlin

Kotlin adalah sebuah bahasa pemrograman dengan pengetikan statis yang berjalan pada Mesin Virtual Java ataupun menggunakan kompiler LLVM yang dapat pula dikompilasikan kedalam bentuk kode sumber JavaScript. Pengembang utamanya berasal dari tim programer dari JetBrains yang bermarkas di Rusia. [10] Meskipun sintaksisnya tidak kompatibel dengan bahasa Java, Kotlin didesain untuk dapat bekerja sama dengan kode bahasa Java dan bergantung kepada kode bahasa Java dari Kelas Pustaka Java yang ada, seperti berbagai framework Java yang ada. Tim Pengembang memutuskan menamakannya Kotlin dengan mengambil nama dari sebuah pulau di Rusia, sebagaimana Java yang mengambil nama dari pulau Jawa di Indonesia.[11] Setelah Google mengumumkan bahwa Kotlin menjadi bahasa kelas satu bagi Android, maka bersama Java dan C++, Kotlin menjadi bahasa resmi untuk pengembangan aplikasi-aplikasi Android. [12]

## 

## 2.6 DNS dan Tipe DNS

Sistem Penamaan Domain (bahasa Inggris: (*Domain Name System*; *DNS*) adalah sebuah sistem yang menyimpan informasi tentang nama host ataupun nama domain dalam bentuk basis data tersebar (*distributed database*) di dalam jaringan komputer, misalkan: Internet. DNS menyediakan alamat IP untuk setiap nama host dan mendata setiap server transmisi surat (*mail exchange server*) yang menerima surel (*email*) untuk setiap domain. Menurut browser Google Chrome, DNS adalah layanan jaringan yang menerjemahkan nama situs web menjadi alamat internet.

DNS menyediakan pelayanan yang cukup penting untuk Internet, ketika perangkat keras komputer dan jaringan bekerja dengan alamat IP untuk mengerjakan tugas seperti pengalamatan dan penjaluran (routing), manusia pada umumnya lebih memilih untuk menggunakan nama host dan nama domain, contohnya adalah penunjukan sumber universal (URL) dan alamat surel. Analogi yang umum digunakan untuk menjelaskan fungsinya adalah DNS bisa dianggap seperti buku telepon internet di mana saat pengguna mengetikkan www.indosat.net.id di peramban web maka pengguna akan diarahkan ke alamat IP 124.81.92.144 (IPv4) dan 2001:e00:d:10:3:140::83 (IPv6).

## 

## 2.7 Jenis-jenis DNS Record

## Beberapa kelompok penting dari data yang disimpan di dalam DNS adalah sebagai berikut:

## A record atau catatan alamat memetakan sebuah nama host ke alamat IP 32-bit (untuk IPv4).

## AAAA record atau catatan alamat IPv6 memetakan sebuah nama host ke alamat IP 128-bit (untuk IPv6).

## CNAME record atau catatan nama kanonik membuat alias untuk nama domain. Domain yang di-alias-kan memiliki seluruh subdomain dan rekod DNS seperti aslinya.

## MX record' *atau* catatan pertukaran surat memetakan sebuah nama domain ke dalam daftar *mail exchange server* untuk domain tersebut.

## PTR record atau catatan penunjuk memetakan sebuah nama host ke nama kanonik untuk host tersebut. Pembuatan rekod PTR untuk sebuah nama host di dalam domain in-addr.arpa yang mewakili sebuah alamat IP menerapkan pencarian balik DNS (*reverse DNS lookup*) untuk alamat tersebut. Contohnya (saat penulisan / penerjemahan artikel ini), www.icann.net memiliki alamat IP 192.0.34.164, tetapi sebuah rekod PTR memetakan 164.34.0.192.in-addr.arpa ke nama kanoniknya: referrals.icann.org.

## NS record atau catatan server nama memetakan sebuah nama domain ke dalam satu daftar dari server DNS untuk domain tersebut. Pewakilan bergantung kepada rekod NS.

## SOA record atau catatan otoritas awal (*Start of Authority*) mengacu server DNS yang menyediakan otorisasi informasi tentang sebuah domain Internet.

## SRV record adalah catatan lokasi secara umum.

## Catatan TXT mengizinkan administrator untuk memasukan data acak ke dalam catatan DNS; catatan ini juga digunakan di spesifikasi *Sender Policy Framework*.

Jenis catatan lainnya semata-mata untuk penyediaan informasi (contohnya, catatan LOC memberikan letak *lokasi* fisik dari sebuah host, atau data ujicoba (misalkan, catatan WKS memberikan sebuah daftar dari server yang memberikan servis yang dikenal (*well-known service*) seperti HTTP atau POP3 untuk sebuah domain.[13]

## 

## 2.8 DNSBL

Domain Name System-based Blackhole List (DNSBL) atau Real-time Blackhole List (RBL) merupakan daftar hitam dari domain yang terdeteksi mengirim email spam. Sebagian besar perangkat lunak mail server dapat dikonfigurasi untuk menolak atau menandai pesan yang berasal dari domain yang sudah tercatat dalam daftar blokir. Istilah “Blackhole List” sendiri berasal dari istilah “blacklist” dan “blocklist”.

Terdapat lusinan DNSBL yang ada, dimana menggunakan array besar yang berisi kriteria dari alamat yang terdaftar dan tidak. DNSBL dapat berisi alamat komputer zombie atau mesin  yang dikhususkan untuk mengirim spam, Penyedia Layanan Internet/Internet Service Provider (ISP) yang bersedia menjadi host spammer atau yang mengirim spam ke sistem honeypot.[14]

## 2.9 Jenis jenis DNSBL

## DNSBL sendiri dapat dikelompokkan berdasarkan fokus blaclist dan cara pemeliharaan domain yang terblaclist sebagai berikut :

|  |  |  |
| --- | --- | --- |
| Fokus Blacklist | Cara Pemeliharaan | Blacklist |
| Spammer | Konservatif | sbl.spamhaus.org |
| Proxy terbuka | Konservatif | opm.blitzed.org |
| Relay terbuka | Konservatif | rbl.maps.vix.com, list.dsbl.org, multihop.dsbl.org, relays.mail-abuse.org, relays.osirusoft.com,  relays.visi.com, relays.orbs.org, relays.ordb.org |
| Relay terbuka | Agresif | unconfirmed.dsbl.org, dnsbl.sorbs.net |
| Serangan Virus/Exploitasi | Agresif | xbl.spamhaus.org, cbl.abuseat.org |
| Netblock ISP/Negara | Agresif | {argentina,att,...}.blackholes.us, dul.maps.vix.com, dul.dnsbl.sorbs.net, dynablock.easynet.nl, blackholes.easynet.nl, dialups.mail-abuse.org |
| RFC  Violators | Mix | {dsn,ipwhois,whois,abuse,postmaster,bogusmx.rfc-ignorant.org |
| Mix | Mix | sbl-xbl.spamhaus.org, bl.spamcop.net, dnsbl.njabl.org |
| Commercial | Commercial | hil.habeas.com, sa-hil.habeas.com, query.bondedsender.org, sa-other.bondedsender.org, sa-trusted.bondedsender.org |
| Unknown | Unknown | rbl.dorkslayers.com, rbl.debian.net |

Table 1: Klasifikasi DNSBL yang digunakan CSAIL (Computer Science and Artificial Intelligence Laboratory)

## 2.10 Cara Kerja Pengecekan Domain berbasis DNSBL

Berikut merupakan alur proses pengecekan domain berbasis DNSBL :

1. DNS lookup tipe A pada domain yang akan di cek untuk mendapatkan ip. Sebagai contoh DNS lookup pada google.com akan mendapatkan ip *172.217.194.113*.
2. Ubah susunan ip dari a.b.c.d menjadi d.c.b.a. Sehingga susunanya menjadi *113.194.217.172*.
3. Konkat dengan DNSBL yang ingin di test. Sebagai contoh DNSBL yang akan dipakai adalah sbl.spamhaus.org. Sehingga susunanya menjadi *113.194.217.172.sbl.spamhaus.org.*
4. DNS lookup tipe TXT pada ip yang sudah di konkat. Apabila hasilnya sukses maka domain tersebut telah terblokir pada DNSBL yang terkait dan sebaliknya. [15]

## 2.11 Laman Web

Laman web adalah sumber dokumen atau informasi yang sesuai untuk [Waring Wera Wanua](https://id.wikipedia.org/w/index.php?title=Waring_Wera_Wanua&action=edit&redlink=1) dan dapat diakses melalui peramban web dan ditampilkan pada monitor atau perangkat seluler. Informasi ini biasanya dalam format [HTML](https://id.wikipedia.org/wiki/HTML) atau [XHTML](https://id.wikipedia.org/wiki/XHTML), dan bisa menyediakan navigasi ke halaman web lainnya melalui tautan [hiperteks](https://id.wikipedia.org/wiki/Hiperteks). Laman web yang sering menggolongkan sumber daya lain seperti *style sheet*, [skrip](https://id.wikipedia.org/w/index.php?title=Skrip&action=edit&redlink=1) dan citra ke dalam presentasi akhir mereka.

Laman web dapat diambil dari komputer lokal atau dari remot server web. Server web dapat membatasi mengakses hanya ke jaringan pribadi misalnya intranet perusahaan, atau mungkin mempublikasikan halaman di Waring Wera Wanua. Laman web yang meminta dan dilayani dari server web menggunakan *Hypertext Transfer Protocol* (HTTP). Laman web dapat terdiri dari berkas teks statis dan konten lainnya yang disimpan di dalam sistem berkas [server web](https://id.wikipedia.org/wiki/Server_web) (halaman web statis), atau mungkin dibangun oleh server-[perangkat lunak](https://id.wikipedia.org/wiki/Perangkat_lunak) sisi ketika mereka diminta (Laman web dinamis). Skrip-sisi klien dapat membuat halaman web lebih responsif terhadap masukan dari pengguna sekali di [peramban](https://id.wikipedia.org/wiki/Peramban) klien.[16]

## 2.12 Web Scraping

Web scraping, web harvesting atau web data extraction adalah data scraping untuk mengekstrak data dari website.[17] Perangkat lunak web scrapper akan mengakses World Wide Web melalui protokol HTTP atau melalui web browser. Web scrapping dapat dilakukan menggunakan bot atau web crawler. [18]

# BAB III TEMPAT KERJA PRAKTEK

## 3.1 Profil Perusahaan

MailTarget.co adalah perusahaan startup IT dengan model bisnis SaaS (Software as a Service) dari Jakarta. Kami membangun email marketing automation software berbasis cloud, untuk membantu usaha kecil, menengah hingga perusahaan besar dalam mengembangkan bisnis mereka.

MailTarget dengan tujuan perancangan agar mudah digunakan siapa saja, cocok untuk menjadi starting point bagi perusahaan konvensional ketika ingin mulai mendigitalkan bisnisnya, ataupun perusahaan yang sudah establish untuk lebih berkembang dengan merangkul pelanggan yang lebih luas.

Dalam satu tempat, MailTarget dapat memenuhi semua kebutuhan digital marketing untuk berbagai latar belakang bisnis dan tujuan. MailTarget dengan fitur-fitur unggulannya mendukung para pelaku usaha mulai dari membuat landing page untuk profil bisnis atau event, membuat katalog produk, membuat formulir untuk mengumpulkan data subscribers, RSVP event & melakukan survey, mengelola kontak pelanggan & segmentasi, mengirim email otomatis untuk membuat email berseri & email ucapan ulang tahun, menyebar email newsletter / promosi ke masif audiens secara reguler atau melalui A/B Testing dengan analisa pengiriman yang dapat di track & dikelola serta dukungan integrasi antar API aplikasi web untuk implementasi email transaksi dengan teknologi D.I.A - MailTarget Artificial Intelligence.

## 

## 3.2 Visi dan Misi

### 3.2.1 Visi Perusahaan

Kami memiliki visi untuk mendigitalkan Indonesia melalui marketing automation tools & website builder MailTarget.

### 3.2.2 Misi Perusahaan

1. Menciptakan produk yang dapat membantu pelaku bisnis konvensional di Indonesia agar mudah mendigitalkan bisnisnya.
2. Menciptakan produk yang dapat meningkatkan lead market bisnis secara eksponensial dan terukur.
3. Menciptakan produk yang dapat memenuhi semua kebutuhan digital marketing, email marketing dan automation dalam satu tempat, dengan fitur yang paling lengkap dan harga yang ekonomis.

## 

## 3.3 Struktur Organisasi

Adapun peranan atau pembagian tugas dalam struktur tersebut adalah sebagai berikut :

1. Chief
2. Lead Technology
3. Software Engineer
4. Designer
5. Dev Ops
6. Content Writer
7. Digital Marketing
8. Customer Success
9. Personal Assistant

## 3.4 Layanan Perusahaan

Berikut layanan yang diberikan oleh PT. Widya Solusi Utama :

1. Membuat Landing Page & Website Dengan Mudah

MailTarget dapat digunakan untuk membuat halaman landing page & website dengan fitur MailTarget Page builder. Dengan fungsi komponen yang lengkap, membuat halaman web untuk profil bisnis, event ataupun katalog produk dapat dilakukan dengan sangat mudah.

1. Mengumpulkan Data Pelanggan Dengan Form API

Formulir merupakan ‘pintu’ untuk melakukan pertukaran data audiens. Di MailTarget, fitur Form API dapat digunakan untuk merancang formulir yang pada prosesnya data yang dimasukkan akan tersimpan ke Contact Book MailTarget sesuai label spesifik yang sudah ditentukan.

1. Digital Marketing

Adalah layanan untuk pemasaran produk secara digital yang komprehensif untuk menjaga tujuan klien. Dengan cara mengemas produk secara menarik yang memanfaatkan Social Media, SEO, SEM, dan Virtual Marketing.

1. Mengirim Penawaran Langsung Ke Inbox Pelanggan Dengan Email Marketing

Jika tren sebelum digital bisnis dan marketing menyebar brosur di pusat keramaian untuk melakukan pemasaran, untuk saat ini dan masa yang akan datang email marketing akan lebih efektif dalam melakukan hal tersebut. Dengan MailTarget membuat ‘digital brochure’ jauh lebih mudah, hemat dan efisien karena dapat dilakukan dengan mandiri, dikirim ke audiens tertentu dan hasil akhir yang dapat dianalisa serta disegmentasikan.

1. Otomatis Mengirim Email Campaign Dengan Email Automation

Salah satu fitur unggulan MailTarget adalah Email Automation. Email Automation MailTarget dapat digunakan untuk mengirim email otomatis dengan konten email yang sudah disiapkan di awal untuk dikirim berdasarkan tujuan campaign. Dengan Email Automation MailTarget banyak hal bisa dilakukan, diantaranya mengirim rentetan email berseri, mengirim email berskenario yang dikirim berdasarkan respon audiens terhadap email yang dikirim sebelumnya, mengirim email respon otomatis dari form yang dibuat di Form API MailTarget hingga email otomatis untuk memberi ucapan selamat ulang tahun ke pelanggan.

1. Integrasi Sistem Email Dengan Transactional Email MailTarget

Salah satu fitur terbaru MailTarget adalah Transactional Email yang berfungsi untuk mengirim email yang dikirim berdasar trigger interaksi pengguna dengan sistem aplikasi web, seperti email permintaan untuk reset password, rangkuman struk transaksi dan lain sebagainya. Transactional Email MailTarget dihadirkan untuk mempermudah pengembangan sistem email internal yang lebih responsif dan efisien dengan penggunaan environment email marketing MailTarget mulai dari compose email, template hingga tracking laporan pengiriman.

# BAB IV HASIL DAN PEMBAHASAN

## 4.1 Gambaran Umum Sistem

Pustaka yang akan dibuat adalah pustaka untuk pengecekan alamat domain web. Pustaka ini memiliki tujuan utama yaitu untuk meningkatkan validitas data pengguna layanan PT Target Sukses Sinergi.

## 

## 4.2 Software & Layanan yang Digunakan

Penulis menggunakan software untuk membuat sistem manajemen sales, adapun beberapa software yang digunakan yaitu :

1. Linux Ubuntu 16.04 LTS

Penulis menggunakan sistem operasi Linux Ubuntu dikarenakan sistem operasi ini sangat fleksibel dan tidak perlu membayar lisensi dalam pemakaianya.

1. Intellij IDEA Ultimate Edition

Untuk IDEpenulis menggunakan Intellij IDEA Ultimate Edition. IDEini sangat *powerfull* untuk pengembangan aplikasi berbasis JVM. Intellij IDEA Ultimate Edition juga mendukung berbagai *autocomplete* untuk bahasa pemrograman maupun framework terkenal seperti Spring, html, css, maupun javascript. Selain itu, Inte;lij IDEA Ultilmate Edition juga memiliki banyak *plugin* dan tema yang bisa disesuaikan dengan keperluan programmer. Penulis memanfaatkan student pack dari email mahasiswa udinus untuk lisensi IDE Intellij IDEA Ultimate Edition.

1. Git

Untuk memudahkan penulis dalam hal *version control* dari web yang penulis bangun, penulis menggunakan Git version Control versi 2.17.0 yang sangat kompatibel dengan Linux Ubuntu 18.04 LTS.

1. Gitlab

Untuk layanan hosting repositori git dari sistem management sales, penulis menggunakan layanan premium dari Gitlab. Layanan tersebut difasilitasi dari pihak tempat penulis melakukan kerja praktek.

## 4.3 Model – model Perancangan Sistem

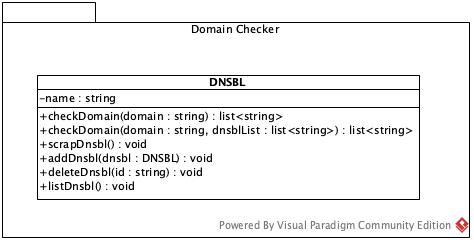
## 4.3.1. Use Case Diagram

## ../VPProjects/use_case.jpg

*Gambar 1 Use Case Diagram*

Pada gambar diatas *Use Case Domain Checker* menunjukan bahwa terdapat 1 aktor yang terlibat, yaitu User. User mampu mengecek domain, menginisialisasi data dnsbl dengan scrapping, melihat data dnsbl, menambah dnsbl, dan menghapus dnsbl.

**4.3.2. Class Diagram**

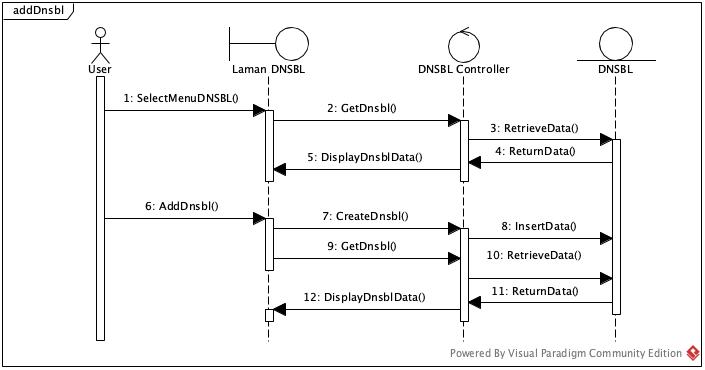


*Gambar 2 Class Diagram*

Pada gambar diatas Class Diagram menunjukkan bahwa terdapat 1 variabel “name” bertipe data string yang mewakili alamat tiap dnsbl. Selain itu terdapat beberapa fungsi untuk mengecek domain apakah tidak tercantum pada DNSBL dengan memasukkan nama domain dengan atau tidak mencantumkan list dnsbl, fungsi untuk menginisialisasi data DNSBL dengan web scraping, dan fungsi untuk menulis, membaca serta menghapus data DNSBL ke file.

**4.3.3. Sequence Diagram**

**4.3.3.1. Add DNSBL**



*Gambar 3 Sequence Diagram : Add DNSBL*

Pada gambar diatas Sequence Diagram menunjukkan proses menambahkan data DNSBL dari web hingga data ditulis ke file. Pertama setelah user mengunjungi halaman DNSBL maka web akan mengambil data DNSBL dari database melalui controller dan menampilkanya. Kemudian user menambahkan DNSBL melalui menu dari web, dan diteruskan controller untuk ditulis ke file. Kemudian web akan memuat ulang tampilan dengan data yang terbaru.

## 4.3.3.2. Delete DNSBL

## ../VPProjects/deleteDnsbl.jpg

## *Gambar 4 Sequence Diagram : Delete DNSBL*

Pada gambar diatas Sequence Diagram menunjukkan proses penghapusan data DNSBL dari web hingga dihapus dari file. Pertama setelah user mengunjungi halaman DNSBL maka web akan mengambil data DNSBL dari database melalui controller dan menampilkanya. Kemudian user menghapus DNSBL melalui menu dari web, dan diteruskan controller untuk dihapus dari file. Kemudian web akan memuat ulang tampilan dengan data yang terbaru.

## 4.3.3.3. List DNSBL

## ../VPProjects/listDnsbl.jpg

## *Gambar 5 Sequence Diagram : List DNSBL*

## Pada gambar diatas Sequence Diagram menunjukkan proses pengambilan data DNSBL dari file hingga ditampilkan ke web. Setelah

## user mengunjungi halaman DNSBL maka web akan mengambil data DNSBL dari database melalui controller dan menampilkanya.

## 4.3.3.4. Init DNSBL

## 

## *Gambar 6 Sequence Diagram : Init DNSBL*

## Pada gambar diatas Sequence Diagram menunjukkan proses inisialisasi data data DNSBL dari file hingga ditampilkan ke web. Setelah user mengunjungi halaman DNSBL maka web akan mengambil data DNSBL dari database melalui controller dan menampilkanya. Kemudian user dapat menginisialisasi data DNSBL melalui menu dari web, yang kemudian perintahnya akan diteruskan ke controller. Controller melakukan http request ke *dnsbl.info* untuk mendapatkan laman. Kemudian laman tersebut di lakukan proses scrapping untuk mendapatkan data DNSBL. Data DNSBL itu kemudian di simpan kedalam file. Kemudian web akan memuat ulang tampilan dengan data yang terbaru.

## 4.3.3.5. Check Domain

## ../VPProjects/checkDomain.jpg

## *Gambar 7 Sequence Diagram : Check DNSBL*

## Pada gambar diatas Sequence Diagram menunjukkan proses pengecekan domain berbasis DNSBL melalui web. Ketika user mengunjungi halaman utama maka akan ditampilkan laman Domain Checker. Kemudian user dapat memasukkan alamat domain ke dalam menu, yang kemudian perintahnya akan diteruskan ke controller. Controller mengambil data DNSBL dari file diamana di tiap domain tersebut dilakukan pengecekan domain berbasis DNSBL sesuai domain yang di inputkan. Untuk tiap DNSBL yang memblokir domain tersebut maka akan ditampung untuk ditampilkan ke web apabila proses pengecekan untuk semua DNSBL selesai.

## 4.4 Implementasi

Bab ini akan membahas tentang pengujian dan analisa hasil program yang telah dibuat. Tujuan dari pengujian ini adalah untuk mengetahui apakah aplikasi yang telah dibuat sesuai dengan perancangannya. Selain itu juga untuk mengetahui detail jalannya aplikasi serta kesalahan yang ada untuk pengembangan dan perbaikan lebih lanjut. Pada proses pembangunan Pustaka Perangkat Lunak Pengecekan Alamat Domain membutuhkan beberapa peralatan, baik berupa perangkat keras maupun perangkat lunak.

**4.4.1 Ruang Lingkup Perangkat Keras**

Dalam pembangunan Sistem Manajemen Sales penulis menggunakan perangkat keras Asus dengan spesifikasi sebagai berikut:

* Prosesor Intel Core i3
* RAM 4GB
* HDD 500GB

### 4.4.2 Ruang Lingkup Perangkat Lunak

Perangkat lunak yang digunakan untuk membangun Sistem Managemen Sales sebagai berikut:

* Sistem Operasi TeaLinuxOS 9 Chamomile 16.04 LTS 64 bit (Linux)
* Java
* Gradle
* Intellij IDEA

### 4.4.3 Implementasi Kode Program

#### 4.4.3.1 Fungsi checkDomain

**fun** checkDomain(domain: String, dnsbl: String): Observable<Boolean> {  
 **return** Observable.from(Lookup(domain, Type.*A*).run())

.subscribeOn(Schedulers.newThread())

.flatMap **{**

Observable.just(**it**)

.subscribeOn(Schedulers.newThread())

.map **{**

**var** result = **false  
 it as** ARecord  
 **val** ip = reverseIp(**it**.*address*.*hostAddress*)  
 **val** lookup2 = Lookup(**"$**ip**.$**dnsbl**"**, Type.*TXT*)  
 **val** resolver = SimpleResolver()  
 lookup2.setResolver(resolver)  
 lookup2.setCache(**null**)  
 lookup2.run()  
  
 *///blocked* **if** (lookup2.*result* == Lookup.*SUCCESSFUL*) {  
 result = **true** }  
 *///not blocked* **else if** (lookup2.*result* == Lookup.*HOST\_NOT\_FOUND*) {  
 result = **false** }  
**return**@map result  
 **}  
 }**.toList().map **{  
 return**@map !**it**.*joinToString*().*contains*(**"false"**)  
 **}**}

*Gambar 8. Kode Program Check Domain*

Gambar diatas merupakan potongan kode program *Check Domain*. Fungsi tersebut memiliki 2 parameter yaitu domain dan dnsbl. Untuk setiap domain yang di proses akan dilakukan pengecekan apakah terdaftar dalam dnsbl. Proses pengecekan dilakukan secara asyncronus untuk menghemat waktu. Domain yang terindikasi bermasalah maka program akan mengembalikan nilai true dan sebaliknya.

#### 4.4.3.2 Fungsi scrapDnsbl

**fun** scrapDnsbl(fileName : String) : String {  
 *//init dnsbl list* **var** dnsblList : List<Dnsbl> = *readFromCsv*(fileName)  
  
 *//scrap* **val** doc = Jsoup.connect(**"https://www.dnsbl.info/dnsbl-list.php"**).get()  
 **val** dnsbls : MutableList<Dnsbl> = *mutableListOf*()  
 doc.select(**"td[width='33%']"**).*forEach* **{** dnsbls.add(Dnsbl(  
 name = **it**.select(**"a"**).text()  
 ))  
 **}** *//read from csv and update* dnsblList += dnsbls  
 dnsblList = dnsblList.*distinctBy* **{ it**.**name }** *//write to csv* **return** *writeToCsv*(fileName,dnsblList)  
}

*Gambar 9. Kode Program Scrap DNSBL*

Gambar diatas merupakan potongan kode program *Scrap DNSBL*. Fungsi tersebut memiliki 1 parameter yaitu fileName yaitu nama file untuk menyimpan data hasil scrapping. Proses scrapping dilakukan pada alamat web *https://www.dnsbl.info/dnsbl-list.php*. Program akan mecari tag html *td* dengan *width* bernilai 33 %, kemudian akan mengambil nilai untuk setiap *a* dari *td* tersebut. Hasil dari scrapping menghasilkan list dnsbl yang kemudian ditulis kedalam file.

#### 4.4.3.3 Fungsi getDnsbl

**fun** getDnsbl(fileName : String) : MutableList<Dnsbl>{  
 **return** *readFromCsv*(fileName)  
}

*Gambar 10. Kode Program Get DNSBL*

Gambar diatas merupakan potongan kode program *Get DNSBL*. Fungsi tersebut akan mengambil data dari file csv dan mengolahnya menjadi list DNSBL.

#### 4.4.3.4 Fungsi addDnsbl

**fun** addDnsbl(fileName : String, dnsbl : Dnsbl) : List<Dnsbl> {  
 *//read from csv* **var** dnsblList = *readFromCsv*(fileName).*toList*()  
 dnsblList += dnsbl  
 dnsblList = dnsblList.*distinctBy* **{ it**.**name }** *//write to csv  
 writeToCsv*(fileName, dnsblList)  
  
 **return** dnsblList  
}

*Gambar 11. Kode Program Add DNSBL*

Gambar diatas merupakan potongan kode program *Add DNSBL*. Fungsi tersebut akan menambah data DNSBL dan menulisnya kedalam file csv.

#### 4.4.3.5 Fungsi deleteDnsbl

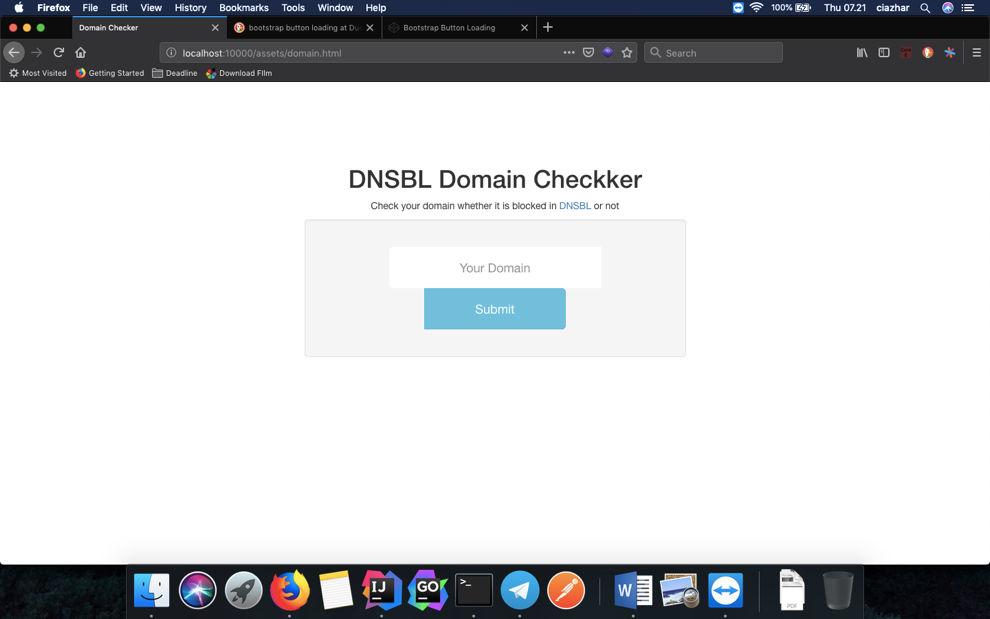
**fun** deletednsbl(id : String, fileName: String) {  
 **var** idInt = 0  
 **try** {  
 idInt = id.*toInt*()  
 } **catch** (nfe: NumberFormatException) {  
 *// not a valid int, handle this as you wish* }  
  
 *//delete line  
 removeLines*(fileName, idInt + 1, 1)  
}

## *Gambar 12. Potongan kode Delete DNSBL*

Gambar diatas merupakan potongan kode program *Delete DNSBL*. Fungsi tersebut akan menghapus data DNSBL dari file csv.

### 4.4.4 Implementasi Desain Interface

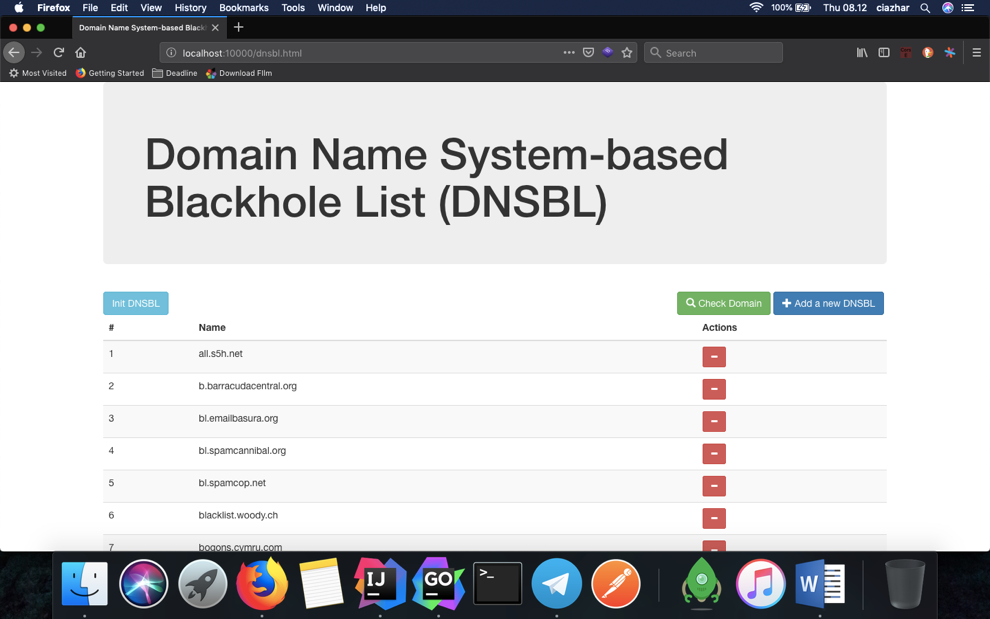
* Halaman Domain Checker



## *Gambar 13 Halaman Domain Checker*

## Gambar diatas merupakan Halaman Domain Chcker yang berisi menu untuk pengecekan domain berbasis DNSBL melalui web. Dengan memasukkan nama domain pada kolom *Your Domain* dan menekan tombol submit, program akan memproses dan menampilkan hasilnya di bagian bawah menu. Hasil pengecekan domain dapat berupa daftar dnsbl yang memblokir domain tersebut untuk domain bermasalah atau pesan bahwa domain tesebut aman.

* Halaman DNSBL



*Gambar 14 Halaman DNSBL*

Gambar diatas merupakan Halaman DNSBL yang berisi daftar DNSBL yang akan digunakan untuk mengecek apakah domain yang akan di test. DNSBL dapat diperoleh dari proses scrapping dengan menekan tombol *Init DNSBL* atau dengan menginputkan manual nama DNSBL. Selain itu dimungkinkan juga menghapus DNSBL yang sudah ada.

### 4.4.5 Pengujian

Pengujian dilakukan untuk mengetahui apakah aplikasi yang telah dibangun dapat berjalan dengan baik dan memenuhi spesifikasi yang telah ditentukan.

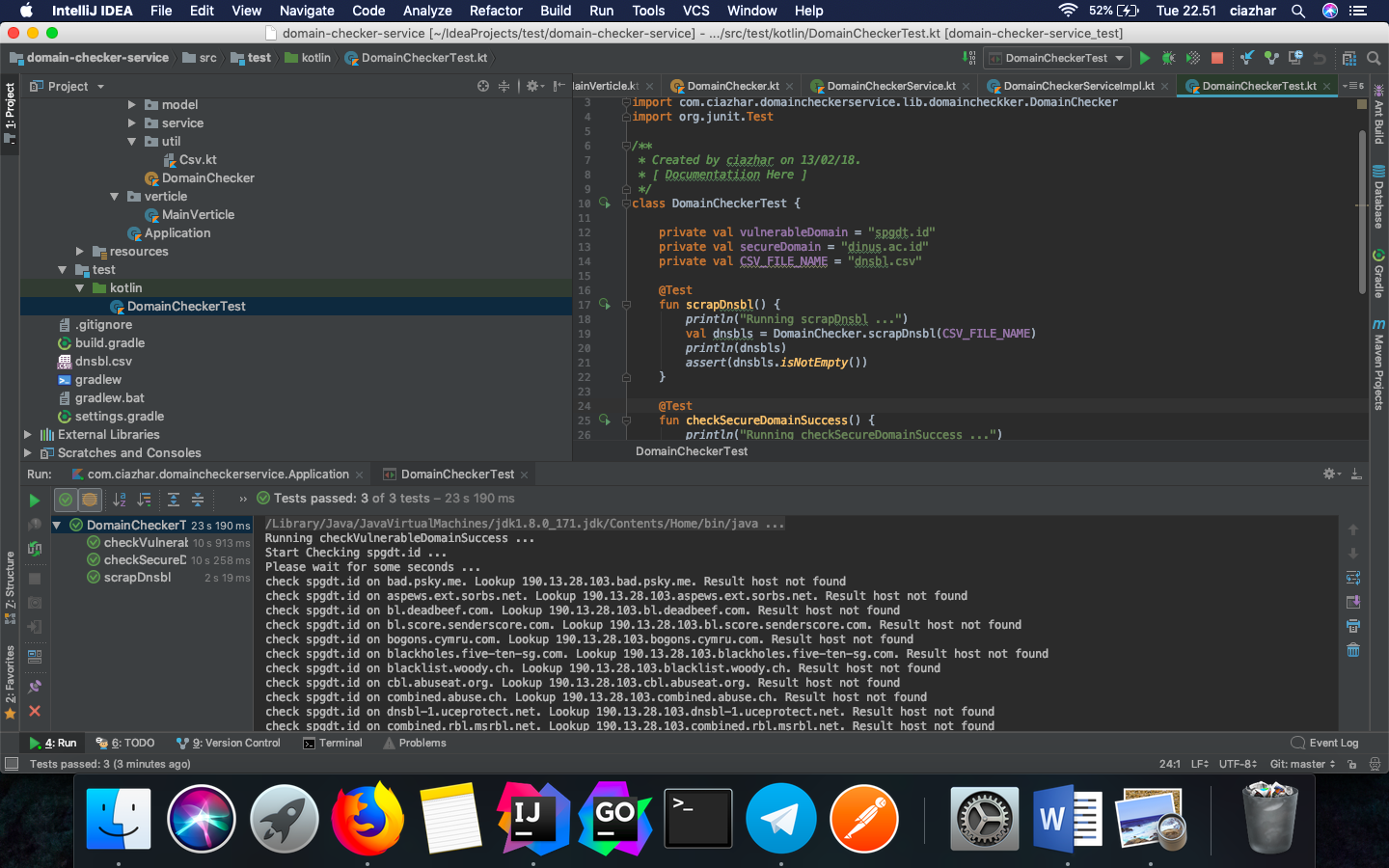
**4.4.5.1 White Box Testing**

Untuk melakukan pengujian *white-box* penulis menulis kode unit testing. Pengujian ini dilakukan pada bagian fungsi yang paling penting, yaitu fungsi checkDomain dan scrapDnsbl.

**class** DomainCheckerTest {  
  
 **private val vulnerableDomain** = **"spgdt.id"  
 private val secureDomain** = **"dinus.ac.id"  
 private val CSV\_FILE\_NAME** = **"dnsbl.csv"** @Test  
 **fun** scrapDnsbl() {  
 *println*(**"Running scrapDnsbl ..."**)  
 **val** dnsbls = DomainChecker.scrapDnsbl(**CSV\_FILE\_NAME**)  
 *println*(dnsbls)  
 *assert*(dnsbls.*isNotEmpty*())  
 }  
  
 @Test  
 **fun** checkSecureDomainSuccess() {  
 *println*(**"Running checkSecureDomainSuccess ..."**)  
 **val** dnsbls = DomainChecker.getDnsbl(**CSV\_FILE\_NAME**).*map* **{ it**.**name }**.*toMutableList*()  
 **val** listResult = DomainChecker.checkDomain(**secureDomain**,dnsbls)  
 *println*(listResult)  
 *assert*(listResult.isEmpty())  
 }  
  
 @Test  
 **fun** checkVulnerableDomainSuccess() {  
 *println*(**"Running checkVulnerableDomainSuccess ..."**)  
 **val** dnsbls = DomainChecker.getDnsbl(**CSV\_FILE\_NAME**).*map* **{ it**.**name }**.*toMutableList*()  
 **val** listResult = DomainChecker.checkDomain(**vulnerableDomain**,dnsbls)  
 *println*(listResult)  
 *assert*(listResult.*isNotEmpty*())  
 }  
}

## *Gambar 15 Potongan kode Unit Testing*

Gambar diatas merupakan potongan kode Unit Testing. Dalam kode tersebut terdapat 3 pengujian yaitu pengujian untuk domain aman menggunakan domain dinus.ac.id, pengujian untuk domain yang bermasalah menggunakan domain spgdt.id dan pengujian scrapping dnsbl. Dari unit testing dapat dihasilkan hasil testing sebagai berikut.



## *Gambar 16 Hasil Unit Testing Melaui IntelliJ Test Runner*

Dari gambar diatas dapat dilihat seluruh modul uji lulus uji. Sehingga dapat dihasilkan table hasil testing sebagai berikut.

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Modul | Hasil yang diharapkan | Hasil Pengujian |
| 1. | ScrapDnsbl | Mendapatkan daftar DNSBL dari alamat https://www.dnsbl.info/ dnsbl-list.php  dan menyimpanya pada file dnsbl.csv | Sesuai |
| 2. | CheckSecureDomainSuccess | Membuktikan bahwa domain dinus.ac.id tidak bermasalah. | Sesuai |
| 3. | CheckVulnerableDomainSuccess | Membuktikan bahwa domain spgdt.id bermasalah | Sesuai |

**4.4.5.1 Black Box Testing**

Black-Box Testing Domain Checker dilakuan melalui Web. Di bawah ini merupakan tabel hasil pengujian aplikasi menggunakan *Black-Box Testing*.

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Modul | Hasil yang diharapkan | Hasil Pengujian |
| 1. | Laman Check Domain | Mengecek apakah domain bermasalah atau tidak    dan menampilkan hasilnya di bawah menu. | Sesuai |
| 2. | Laman DNSBL | Menampilkan halaman DNSBL yang berisi daftar DNSBL. | Sesuai |
| 3. | Init DNSBL | Pada laman DNSBL terdapat tombol Init DNSBL, ketika ditekan akan menambahkan daftar DNSBL dari alamat https://www.dnsbl.info/dnsbl-list.php dan memasukkanya ke daftar DNSBL | Sesuai |
| 4. | Tambah DNSBL | Pada laman DNSBL terdapat tombol “Add a new DNSBL”, ketika ditekan akan muncul pop up sebagai berikut    Data yang di inputkan akan masuk kedalam daftar DNSBL | Sesuai |
| 5. | Delete DNSBL | Pada laman DNSBL, di setiap baris DNSBL terdapat tombol merah, apabila ditekan maka data akan dihapus dari daftar DNSBL. | Sesuai |

# BAB V PENUTUP

## 5.1 Kesimpulan

Dari permasalahan yang dihadapi oleh pihak PT. Target Sukses Sinergi maka dapat diambil kesimpulan bahwa dengan adanya pustaka perangkat lunak Domain Checker dapat mempermudah pihak PT dalam melakukan pengecekan data milik klien.

## 5.2 Saran

Penulis sadari bahwa apa yang penulis buat saat ini belumlah sempurna, oleh karena itu penulis memberikan saran kepada pengembang pustaka perangkat lunak Domain Checker tahap selanjutnya, untuk menambah fitur dan melakukan optimasi sehingga pustaka tersebut menjadi lebih sempurna.

# DAFTAR PUSTAKA

1. https://id.wikipedia.org/wiki/Surat\_elektronik
2. Allen, David (2004). [*Windows to Linux*](https://books.google.com/books?id=UD0h_GqgbHgC&printsec=frontcover&dq=network%2B+guide+to+networks&hl=en&src=bmrr&ei=hMnATfmmA8j00gGMsOC2Cg&sa=X&oi=book_result&ct=result&resnum=1&ved=0CE8Q6AEwAA#v=onepage&q&f=false). Prentice Hall. p. 192. [*Archived*](https://web.archive.org/web/20161226051835/https://books.google.com/books?id=UD0h_GqgbHgC&printsec=frontcover&dq=network%2B+guide+to+networks&hl=en&src=bmrr&ei=hMnATfmmA8j00gGMsOC2Cg&sa=X&oi=book_result&ct=result&resnum=1&ved=0CE8Q6AEwAA#v=onepage&q&f=false) from the original on 2016-12-26.
3. <https://en.wikipedia.org/wiki/Email_marketing>
4. Pew Internet & American Life Project, ["Tracking surveys"](http://www.pewinternet.org/trends.asp), March 2000 – March 2009
5. [How Scheduling Affects Rates](http://www.mailermailer.com/resources/metrics/2012/how-scheduling-affects-rates.rwp). Mailermailer.com (July 2012). Retrieved on July 28, 2013.
6. *BtoB Magazine*, ["Early Email Blasts Results in Higher Click & Open Rates"](http://www.btobonline.com/article/20110901/EMAIL13/309019997/early-morning-email-blasts-pay-off-with-strong-opens-clicks) [Archived](https://web.archive.org/web/20111122001746/http://www.btobonline.com/article/20110901/EMAIL13/309019997/early-morning-email-blasts-pay-off-with-strong-opens-clicks) 2011-11-22 at the [Wayback Machine](https://en.wikipedia.org/wiki/Wayback_Machine)., September 2011
7. https://en.wikipedia.org/wiki/Bounce\_message
8. https://id.wikipedia.org/wiki/Pustaka\_perangkat\_lunak
9. <https://id.wikipedia.org/wiki/Kotlin_(bahasa_pemrograman)>
10. Heiss, Janice (April 2013). [*"The Advent of Kotlin: A Conversation with JetBrains' Andrey Breslav"*](http://www.oracle.com/technetwork/articles/java/breslav-1932170.html). oracle.com. Oracle Technology Network*. Diakses tanggal February 2, 2014*.
11. Mobius (2015-01-08), *Андрей Бреслав — Kotlin для Android: коротко и ясно, diakses tanggal 2017-05-28*
12. https://developer.android.com/kotlin/index.html
13. https://id.wikipedia.org/wiki/Sistem\_Penamaan\_Domain
14. https://en.wikipedia.org/wiki/DNSBL
15. Jaeyeon Jung. 2004. An Empirical Study of Spam Traffic and the Use of DNS Black Lists
16. https://id.wikipedia.org/wiki/Laman\_web
17. Boeing, G.; Waddell, P. (2016). "New Insights into Rental Housing Markets across the United States: Web Scraping and Analyzing Craigslist Rental Listings". Journal of Planning Education and Research (0739456X16664789).
18. https://en.wikipedia.org/wiki/Web\_scraping

Lampiran 1. Foto Kegiatan