

BAB II

TINJAUAN PUSTAKA

2.1. Tinjauan Teori Umum

2.1.1 Pengembangan Sistem Berorientasi Objek

Menurut Sutarman (2022) “Sistem adalah suatu kesatuan yang terdiri atas komponen atau elemen yang terorganisasi, saling berinteraksi, saling tergantung satu sama lain dan terpadu untuk mencapai suatu tujuan.”. Menurut Sutabri (2022) “Informasi merupakan suatu data yang telah diolah, diklasifikasikan dan diinterpretasikan serta digunakan untuk proses pengambilan keputusan.”. Menurut Kenneth C. Laudon dan Jane P. Laudon (2022) "Sistem informasi adalah suatu rangkaian komponen yang saling berhubungan yang mengumpulkan, memproses, menyimpan, dan menyebarkan informasi untuk mendukung pengambilan keputusan dan koordinasi dalam suatu organisasi." Menurut Turban, McLean, dan Wetherbe (2019), sistem informasi adalah suatu kumpulan komponen yang saling berhubungan yang mengumpulkan, memproses, menyimpan, dan menyebarkan informasi untuk mendukung pengambilan keputusan dan koordinasi dalam suatu organisasi. Adapun pengembangan Sistem Berbasis Objek (PBO) adalah metodologi pengembangan perangkat lunak yang berfokus pada objek, yang merupakan entitas yang menggabungkan data (atribut) dan perilaku (metode). Pendekatan ini berbeda dengan metodologi tradisional yang berfokus pada fungsi dan prosedur (Soni, 2018). Dalam pemrograman berorientasi objek (OOP), terdapat beberapa konsep penting yang harus diketahui, yaitu:

1. Objek (*Object*) adalah entitas yang memiliki keadaan (*attribute*) dan

perilaku (*method*). Objek dianggap sebagai instansi dari suatu kelas dan dapat berinteraksi satu sama lain.

2. Kelas (*Class*) adalah blueprint atau template yang mendefinisikan atribut dan perilaku yang dimiliki oleh objek. Objek-objek yang memiliki sifat dan perilaku yang sama dikelompokkan dalam satu kelas.
3. Metode (*Method*) adalah tindakan atau perilaku yang dapat dilakukan oleh objek. Method mendefinisikan tugas atau operasi yang dapat dilakukan oleh objek dari suatu kelas.
4. Atribut (*Attribute*) adalah data atau keadaan yang dimiliki oleh objek. Atribut mewakili karakteristik objek yang dapat berupa variabel atau konstanta.
5. Abstraksi (*Abstraction*) adalah konsep untuk menyederhanakan objek dengan menyoroti fitur pentingnya dan menyembunyikan detail implementasinya. Abstraksi memungkinkan penggunaan objek tanpa perlu mengetahui implementasi internalnya.
6. Enkapsulasi (*Encapsulation*) adalah konsep yang menggabungkan atribut dan method yang berkaitan dalam suatu objek, dan menyembunyikan akses langsung ke data dari luar objek. Enkapsulasi melindungi data dan menyediakan metode publik untuk berinteraksi dengan objek.
7. Pewarisan (*Inheritance*) adalah konsep di mana kelas baru (kelas turunan) dapat mewarisi atribut dan method dari kelas yang sudah ada (kelas induk atau superclass). Pewarisan memungkinkan penggunaan kembali kode yang ada dan memfasilitasi hierarki kelas.

8. Polimorfisme (*Polymorphism*) adalah konsep di mana objek dapat memiliki banyak bentuk atau perilaku yang berbeda tergantung pada konteks pemanggilan. Polimorfisme memungkinkan penggunaan metode yang sama dengan nama yang sama pada kelas yang berbeda dalam hierarki pewarisan.
9. Antarmuka (*Interface*) adalah kontrak yang menggambarkan sekumpulan method yang harus diimplementasikan oleh kelas tertentu. Antarmuka memungkinkan adanya polimorfisme dan digunakan untuk menerapkan abstraksi lebih lanjut.
10. Paket (*Package*) adalah mekanisme yang digunakan untuk mengorganisir dan mengelompokkan kelas dan objek menjadi unit yang lebih besar. Paket membantu dalam mengatur kode secara terstruktur dan menghindari konflik nama.

Menurut (Boucher, G., 2022) , Konsep pemrograman berorientasi objek menawarkan beberapa keuntungan dalam pengembangan perangkat lunak, antara lain:

1. Modularitas dan pemeliharaan yang lebih baik. Pemrograman berorientasi objek memungkinkan pemisahan sistem menjadi objek-objek yang mandiri. Ini memfasilitasi modularitas, di mana setiap objek dapat dikembangkan dan diuji secara terpisah. Perubahan pada satu objek tidak akan mempengaruhi secara langsung objek lainnya, sehingga pemeliharaan dan perbaikan sistem menjadi lebih mudah.
2. Dapat digunakannya secara berulang. Objek dan kelas dapat digunakan kembali dalam proyek-proyek berbeda. Hal ini menghemat waktu dan

upaya dalam pengembangan perangkat lunak baru serta mengurangi kesalahan yang mungkin terjadi karena kode yang sudah teruji dan terbukti.

3. Peningkatan Produktivitas dan Efisiensi, Pemrograman berorientasi objek memungkinkan tim pengembang untuk bekerja secara terpisah dalam pengembangan yang paralel. Konsep Pemrograman berorientasi objek juga mengurangi kompleksitas kode dengan menyediakan struktur yang terorganisir, yang membuat kode lebih mudah dibaca, dimengerti, dan dikelola. Hal ini dapat meningkatkan produktivitas pengembang dan efisiensi dalam pengembangan perangkat lunak.
4. Pewarisan dan Polimorfisme, Pewarisan memungkinkan penggunaan kembali kode yang ada dan memperluas fungsionalitas dengan membuat kelas turunan dari kelas yang sudah ada. Polimorfisme memungkinkan objek dengan tipe yang berbeda untuk merespons metode dengan cara yang berbeda, tergantung pada konteks pemanggilan. Kedua konsep ini memberikan fleksibilitas dan kemampuan untuk memodelkan hierarki kelas yang kompleks.
5. Abstraksi dan enkapsulasi. Abstraksi membantu dalam menyederhanakan kompleksitas dengan berfokus pada aspek penting dari objek dan menyembunyikan detail implementasinya. Enkapsulasi melindungi data dan metode dari akses langsung oleh pihak luar. Hal ini menjaga keamanan dan integritas data serta memungkinkan perubahan dalam implementasi objek tanpa mempengaruhi cara penggunaan objek tersebut.

“*Unified Modelling Language (UML)* merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem”

(Hendini, 2016).

Maka dari itu dapat ditarik kesimpulan bahwa *Unified Modelling Language (UML)* merupakan suatu bahasa standar yang digunakan untuk pemodelan dan komunikasi rancangan pengembangan sistem dengan menggunakan diagram atau simbol simbol tertentu. Dengan menggunakan *Unified Modelling Language (UML)* menyediakan beberapa diagram visual untuk menunjukkan beberapa sudut pandang dalam sistem antara lain :

1. Use Case Diagram

“*Use case diagram* digunakan untuk mendeskripsikan tipikal interaksi antara pengguna dengan sistem informasi”

(Maulana, 2014).

Berdasarkan pendapat ahli di atas, dapat disimpulkan bahwa *use case diagram* merupakan diagram UML yang berfungsi sebagai alat bantu 23 pemodelan untuk menggambarkan tingkah laku (*behavior*) dari sudut pandang luar sistem untuk menjelaskan interaksi dan peran antara aktor dengan sistem yang dirancang.

2. Activity Diagram

Menurut Meilinda *activity diagram* didefinisikan sebagai berikut;

“*Activity diagram* merupakan diagram yang menerangkan tentang aktifitas-aktifitas yang dapat dilakukan oleh seorang *entity* atau pengguna yang akan diterapkan pada aplikasi.”

(Meilinda, 2016).

Sedangkan menurut Sukamto & Shalahuddin activity diagram mendefinisikan bahwa;

“Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak.”

(Sukamto & Shalahuddin 2015:161)

Dapat disimpulkan bahwa *activity diagram* merupakan diagram yang menggambarkan aktifitas-aktifitas sistem dimana setiap urutan aktifitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.

3. Class Diagram

Menurut Hendini *Class diagram* didefinisikan sebagai berikut;

“*Class diagram* merupakan hubungan antar kelas dan penjelasan detail tiap- tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan- aturan dan tanggung jawab entitas yang menentukan perilaku sistem.”

(Hendini, 2016).

“Diagram kelas atau class diagram menggambarkan struktur sistem dari segi pendefinisian kelas- kelas yang akan dibuat untuk membangun sistem”

Sukamto & Shalahuddin (2015:141),

Maka dari itu, dapat disimpulkan bahwa class diagram merupakan diagram UML yang menggambarkan struktur sistem dari segi pendefinisian kelas-kelas dan memperlihatkan aturan serta tanggung jawab entitas serta berisikan metode-metode atau aturan tertentu.

4. Sequence Diagram

Menurut Meilinda *Sequence Diagram* didefinisikan sebagai berikut;

“*Sequence diagram* merupakan UML yang menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa message yang digambarkan terhadap waktu.”

(Meilinda, 2016)

Menurut Sukamto & Shalahuddin *Sequence Diagram* (2015:165) mengemukakan bahwa:

“Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima oleh objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.”

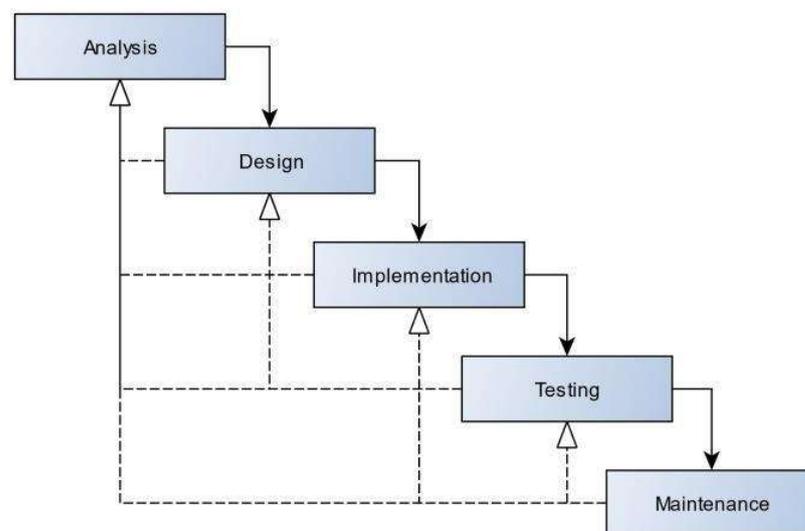
(Sukamto & Shalahuddin 2015:165)

Dapat disimpulkan bahwa *sequence diagram* dapat diartikan sebagai alat pemodelan rancangan sistem yang menggambarkan alur atau urutan sistem yang bersinkronisasi dengan *use case diagram* untuk mendeskripsikan waktu hidup objek dan message yang dikirm atau diterima oleh objek tersebut.

2.1.2 Metode Pengembangan Waterfall

Penelitian ini dilakukan menggunakan metode *Waterfall*, Metode *Waterfall* adalah model pengembangan perangkat lunak yang mengikuti alur seperti air terjun, di mana setiap tahapan harus selesai sepenuhnya sebelum beralih ke tahap berikutnya. Metode ini sangat terstruktur dan linier, sehingga sering dianggap

sebagai pendekatan tradisional dalam pengembangan perangkat lunak. Metode Waterfall adalah pendekatan pengembangan perangkat lunak yang dilakukan secara berurutan dan sistematis, dimulai dari analisis kebutuhan pengguna, perancangan sistem, implementasi kode, pengujian menyeluruh, hingga pemeliharaan berkelanjutan, di mana setiap tahap harus selesai sepenuhnya sebelum beralih ke tahap berikutnya. Desain penelitian menggunakan Software Development Life Cycle (SDLC) model *waterfall*, terdapat empat tahap penelitian pada gambar di bawah ini



Gambar 2. 1 SDLC Model *Waterfall*

Penjelasan tahapan-tahapan perancangan sistem yaitu:

1. *Analysis* (analisis)

Fase ini merupakan awal dari proyek pengembangan perangkat lunak. Pada tahap ini, tim pengembangan berfokus untuk memahami persyaratan perangkat lunak yang akan dikembangkan. Analisis sistem bekerja sama

dengan pemangku kepentingan untuk mengumpulkan informasi dan mengidentifikasi kebutuhan bisnis yang spesifik.

2. *Design* (desain)

Setelah analisis, langkah berikutnya adalah fase desain. Pada tahap ini, tim pengembangan merencanakan struktur sistem dan merinci bagaimana sistem akan dibangun. Desain sistem mencakup perencanaan struktur data, arsitektur sistem, antarmuka pengguna, serta komponen perangkat lunak dan perangkat keras yang dibutuhkan.

3. *Implementation* (implementasi)

Setelah desain selesai, tahap implementasi dimulai. Pada fase ini, perangkat lunak sebenarnya dikembangkan menggunakan bahasa pemrograman yang dipilih. Program-program yang dikodekan diuji dan diintegrasikan untuk memastikan bahwa mereka berfungsi sesuai dengan desain yang telah ditetapkan.

4. *Testing* (pengujian)

Fase pengujian adalah langkah kritis dalam model *Waterfall*. Sistem secara menyeluruh diuji untuk memastikan bahwa perangkat lunak berfungsi dengan benar dan memenuhi persyaratan. Pengujian mencakup pengujian unit, integrasi, dan sistem. Kesalahan yang ditemukan selama pengujian harus diperbaiki sebelum melanjutkan ke tahap berikutnya.

5. *Maintenance* (Pemeliharaan)

Tahap pemeliharaan terjadi setelah perangkat lunak diterapkan dan dijalankan di lingkungan produksi. Pemeliharaan mencakup pemecahan *bug*, peningkatan fungsionalitas, dan perubahan-perubahan lainnya untuk memastikan bahwa perangkat lunak tetap sesuai dengan kebutuhan bisnis yang berkembang.

2.1.3 Tahapan Analisis dan Perancangan Berorientasi Objek

Analisis dan perancangan berorientasi objek (OOAD) adalah metodologi pengembangan perangkat lunak yang berfokus pada pemodelan sistem perangkat lunak sebagai kumpulan objek yang saling berinteraksi. Metodologi ini membantu dalam membangun sistem yang mudah dipahami, dipelihara, dan dimodifikasi (Boucher, G., 2022). Berikut adalah tahapan – tahapannya:

1. Analisis Kebutuhan

Mengidentifikasi dan memahami kebutuhan sistem dari sudut pandang pengguna dan pemangku kepentingan.

2. Desain Konseptual

Membangun model konseptual sistem yang menggambarkan arsitektur sistem secara keseluruhan dan komponen-komponen utama.

3. Desain Detail

Merancang implementasi terperinci dari kelas-kelas dan komponen sistem.

4. Implementasi

Membangun kode sumber sistem berdasarkan desain yang telah dibuat.

5. Pengujian dan Pemeliharaan

Memastikan sistem memenuhi kebutuhan dan berjalan dengan benar, serta memeliharanya dari waktu ke waktu.

2.1.4 Sistem Penerimaan Peserta Didik Baru (PPDB) dan Proses Seleksi Akademik

Sistem Penerimaan Peserta Didik Baru (PPDB) dan Proses Seleksi Akademik adalah proses yang digunakan untuk menyeleksi dan menerima peserta didik baru di sekolah. Sistem ini dirancang untuk memastikan bahwa semua peserta didik yang memenuhi syarat memiliki kesempatan yang sama untuk diterima di sekolah yang mereka inginkan. Tujuan utama PPDB adalah:

1. Memastikan pemerataan akses Pendidikan
Memberikan kesempatan yang sama bagi semua peserta didik yang memenuhi syarat untuk diterima di sekolah yang mereka inginkan.
2. Meningkatkan kualitas Pendidikan
Menerima peserta didik yang memiliki potensi dan kemampuan untuk belajar dengan baik, sehingga dapat meningkatkan kualitas pendidikan di sekolah.
3. Menjaga akuntabilitas dan transparansi
Melakukan proses PPDB yang transparan dan akuntabel, sehingga dapat terhindar dari praktik nepotisme dan korupsi.

Proses PPDB umumnya terdiri dari beberapa tahap, yaitu:

1. Pendaftaran

Peserta didik mendaftarkan diri ke sekolah yang mereka inginkan dengan membawa dokumen yang diperlukan.

2. Seleksi

Sekolah menyeleksi peserta didik berdasarkan jalur yang dipilih dan kriteria yang telah ditetapkan.

3. Pengumuman

Sekolah mengumumkan hasil seleksi kepada peserta didik.

4. Daftar Ulang

Peserta didik yang diterima melakukan pendaftaran ulang dan mengikuti proses selanjutnya.

2.1.5 Teknologi Web dan Model Pengembangan Sistem

Teknologi web adalah teknologi yang digunakan untuk membuat, mengembangkan, dan memelihara aplikasi web. Aplikasi web adalah aplikasi yang dapat diakses melalui internet dengan menggunakan browser. Teknologi web dapat mempermudah akses dan penggunaan aplikasi, karena aplikasi web dapat diakses dari mana saja dan kapan saja selama terhubung dengan internet.

Model pengembangan sistem adalah suatu pendekatan atau metode yang digunakan dalam pengembangan sistem. Terdapat beberapa model pengembangan sistem, seperti model *waterfall*, model spiral, dan model *agile*. Model pengembangan sistem yang dipilih harus sesuai dengan kebutuhan dan karakteristik dari sistem yang akan dikembangkan. Sistem Basis Data

Sistem basis data adalah sistem yang digunakan untuk mengelola dan menyimpan data secara terorganisir. Basis data dapat digunakan untuk menyimpan

data dari berbagai jenis aplikasi, seperti sistem informasi manajemen, sistem informasi akademik, dan sistem penerimaan siswa baru. Sistem basis data dapat menggunakan berbagai jenis sistem manajemen basis data (DBMS), seperti MySQL, Oracle, dan SQL Server.

2.1.6 Implementasi Sistem

Pada implementasi *system*, peneliti menggunakan Vercel sebagai *cloud service* untuk melakukan *deployment* aplikasi web yang dikembangkan. Vercel adalah *platform cloud* yang digunakan untuk meng-host, membangun, dan mengelola aplikasi web dan situs web. Platform ini dikenal karena dukungan terintegrasi yang kuat untuk aplikasi yang dibangun dengan teknologi web seperti React, Next.js, Vue.js, Angular, dan banyak lagi. Vercel memungkinkan pengembang untuk dengan mudah melakukan *deployment* aplikasi web mereka ke server mereka dengan cepat dan mudah.

2.2. Tinjauan Teori Khusus

2.2.1 Penerimaan Peserta Didik Baru (PPDB) Berbasis Web

Penerimaan Peserta Didik Baru (PPDB) berbasis web adalah sistem penerimaan siswa baru yang menggunakan teknologi web. Dalam PPDB berbasis web, siswa dan orang tua siswa dapat melakukan pendaftaran dan mengunggah berkas persyaratan melalui *website* yang disediakan oleh sekolah. PPDB berbasis web mempermudah siswa dan orang tua dalam melakukan pendaftaran, karena dapat dilakukan dari mana saja dan kapan saja selama terhubung dengan internet. Algoritma *Deferred Acceptance* adalah algoritma pencocokan yang telah terbukti efisien dan stabil. Efisiensi berarti bahwa tidak ada siswa yang lebih suka program

studi yang dia terima daripada program studi yang dia tidak terima. Stabilitas berarti bahwa tidak ada dua siswa yang dapat saling bertukar program studi tanpa membuat satu pihak menjadi lebih buruk. (Guillaume Haeringer, 2021)

2.2.2 Model Pengembangan Sistem Berbasis Web

Model pengembangan sistem berbasis web adalah model pengembangan sistem yang khusus digunakan untuk mengembangkan aplikasi web. Terdapat beberapa model pengembangan sistem berbasis web, seperti model Waterfall, model Spiral, dan model Agile. Model pengembangan sistem yang paling sesuai untuk pengembangan PPDB berbasis web adalah model Agile, karena model ini lebih fleksibel dan adaptif terhadap perubahan kebutuhan pengguna.

2.2.3 Bahasa Pemrograman Web

Bahasa pemrograman web adalah bahasa pemrograman yang digunakan untuk mengembangkan aplikasi web. Beberapa bahasa pemrograman web yang populer adalah PHP, Python, dan JavaScript. Bahasa pemrograman yang digunakan dalam pengembangan PPDB berbasis web di SMK Eben Haezer adalah PHP dan JavaScript.

2.2.3.1 MySQL

MySQL adalah sistem manajemen basis data relasional yang banyak digunakan untuk mengelola basis data aplikasi web. MySQL merupakan salah satu DBMS (Database Management System) open source yang populer. MySQL digunakan dalam pengembangan PPDB berbasis web di SMK Eben Haezer untuk menyimpan data siswa dan orang tua siswa yang mendaftar.

2.2.3.2 React js

React.js, atau lebih dikenal sebagai React, adalah sebuah pustaka JavaScript yang digunakan untuk membangun antarmuka pengguna (UI) pada aplikasi web. Dikembangkan oleh Facebook, React menyajikan konsep pengembangan UI yang bersifat deklaratif, memungkinkan pengembang untuk dengan mudah mengelola dan memperbarui tampilan aplikasi secara efisien. Poin utama React melibatkan penggunaan komponen, yaitu blok-blok membangun yang dapat digunakan kembali yang merepresentasikan bagian-bagian dari UI. Setiap komponen dapat dianggap sebagai unit mandiri yang dapat diatur, dikelola, dan diinteraksi dengan komponen lainnya. React menggunakan konsep "virtual DOM" untuk mempercepat proses pembaruan tampilan dengan hanya merender perubahan yang terjadi, bukan seluruh halaman.

2.2.3.3 Framework Laravel

Laravel adalah sebuah framework aplikasi web berbasis PHP yang populer dan kuat. Ini dikembangkan oleh Taylor Otwell dan pertama kali dirilis pada tahun 2011. Laravel dirancang untuk memudahkan pengembangan aplikasi web dengan menyediakan berbagai fitur dan alat yang kuat untuk mengatasi tugas-tugas umum dalam pengembangan web. Laravel digunakan dalam pengembangan PPDB berbasis web di SMK Eben Haezer untuk mempercepat proses pengembangan dan mempermudah pemeliharaan aplikasi.

2.2.3.4 Bootstrap

Bootstrap adalah framework CSS yang digunakan untuk mempercepat proses pengembangan tampilan aplikasi web. Bootstrap menyediakan berbagai

komponen tampilan, seperti tombol, formulir, tabel, dan lain-lain yang siap pakai, sehingga dapat mempercepat proses pengembangan tampilan aplikasi web. Bootstrap digunakan dalam pengembangan PPDB berbasis web di SMK Eben Haezer untuk mempercepat proses pengembangan tampilan dan mempermudah pemeliharaan aplikasi.

2.2.3.5 TailwindCSS

TailwindCSS adalah suatu *framework* CSS yang dirancang untuk mempermudah pengembangan tata letak dan desain antarmuka pengguna. Tailwind tidak seperti kebanyakan *framework* CSS tradisional yang memberikan sejumlah besar komponen siap pakai. Sebaliknya, Tailwind menyediakan sejumlah besar kelas utilitas kecil yang dapat diterapkan langsung pada elemen HTML.

2.2.3.6 Shadcn/UI

Shadcn/UI adalah gabungan antara Radix UI dan TailwindCSS. Radix UI adalah kumpulan komponen UI yang dirancang untuk dibangun di atas dasar desain Tailwind CSS. Ini menyediakan komponen-komponen yang dapat digunakan dan dikustomisasi untuk membuat antarmuka pengguna yang dapat diakses dan menarik.

2.2.4 Deployment dan Keamanan Aplikasi Web

Netlify, sebagai platform *hosting* dan penerbitan aplikasi web, memberikan beberapa fitur keamanan untuk melindungi aplikasi web yang dihosting di sana. Berikut adalah beberapa aspek keamanan yang umumnya terkait dengan penggunaan Netlify:

1. HTTPS *by Default*

Netlify menyediakan HTTPS gratis dengan sertifikat SSL yang otomatis dikeluarkan menggunakan Let's Encrypt. Ini membantu menjaga data yang ditransmisikan antara pengguna dan situs web tetap aman.

2. *Global Content Delivery Network (CDN)*

Netlify menggunakan CDN global untuk mendistribusikan konten ke server yang paling dekat dengan pengguna. Hal ini tidak hanya meningkatkan kecepatan pemuatan halaman tetapi juga memberikan lapisan keamanan tambahan dengan menyediakan cadangan data dari lokasi server yang berbeda.

3. *Continuous Deployment Security*

Proses penerbitan berkelanjutan (*continuous deployment*) di Netlify memastikan bahwa aplikasi selalu diperbarui dengan versi terbaru dari kode sumber yang diunggah ke repositori Git. Ini membantu mengurangi risiko eksploitasi keamanan dengan memastikan bahwa aplikasi menjalankan versi terkini dari perangkat lunak dan tidak memiliki kerentanan keamanan yang diketahui.

4. *Automated Backups*

Netlify secara otomatis mencadangkan konten dan data aplikasi. Jika ada masalah atau perlu untuk mengembalikan ke versi sebelumnya, Netlify menyediakan opsi untuk merestorasi dari cadangan ini.

5. *Role-Based Access Control (RBAC)*

Netlify memberikan kontrol akses berbasis peran yang memungkinkan pengaturan tingkat akses untuk anggota tim yang berbeda. Ini membantu melindungi aplikasi dari akses yang tidak sah dan memastikan bahwa setiap orang hanya memiliki akses yang diperlukan.

6. *Distributed Denial of Service (DDoS) Protection*

Dengan menggunakan arsitektur CDN dan layanan keamanan, Netlify dapat memberikan perlindungan terhadap serangan DDoS dengan mendistribusikan lalu lintas di seluruh jaringan server mereka.

7. *Identity and Authentication Services*

Netlify menyediakan layanan identitas dan otentikasi yang memungkinkan pengguna untuk mengelola dan mengamankan identitas pengguna mereka. Ini dapat digunakan untuk melindungi bagian tertentu dari aplikasi atau mengamankan akses ke sumber daya tertentu.