

BAB II

LANDASAN TEORI

2.1 Gizi Buruk

Di Indonesia kelompok anak balita menunjukkan prevalensi paling tinggi untuk menderita KKP (Kekurangan Kalori Protein). Kelompok umur ini sulit dijangkau oleh berbagai upaya kegiatan perbaikan gizi dan kesehatan lainnya, karena tidak dapat datang sendiri ke tempat pelayanan kesehatan gizi dan kesehatan[3] .

Penyakit gangguan gizi banyak ditemui pada masyarakat golongan rentan, yaitu golongan yang mudah sekali menderita akibat kekurangan gizi dan juga kekurangan makanan (*dificiency*) misalnya kwashiorkor, busung lapar, marasmus, beri-beri dan lain-lain. Kegemukan (*obesity*), kelebihan berat badan (*over weight*) merupakan tanda gizi salah yang berdasarkan kelebihan dalam makanan.

Penyakit-penyakit atau gangguan kesehatan masyarakat antara lain adalah:

1. Penyakit KKP (Kurang Kalori / KEP) Kurang kalori protein adalah keadaan kurang gizi yang disebabkan oleh rendahnya konsumsi energi dan protein dalam makanan sehari-hari sehingga tidak mencukupi angka kecukupan gizi.
2. Penyakit kegemukan (*obesitas*) *Obesitas* adalah kelebihan berat badan sebagai akibat dari penimbunan lemak tubuh yang berlebihan. Seseorang yang memiliki berat badan 20% lebih tinggi dari nilai tengah kisaran berat badannya yang normal dianggap mengalami obesitas.

2.2 Faktor – Faktor Yang Mempengaruhi Status Gizi

Faktor – faktor yang mempengaruhi status gizi pada balita banyak sekali, diantaranya adalah pendapatan, pengetahuan gizi ibu, akses pelayanan kesehatan, kejadian diare, pemberian ASI eksklusif, sumber air bersih, pola asuh orang tua, Nutrisi pada masa kehamilan dan berat bayi lahir rendah (BBLR). [4]

2.2.1 Faktor Langsung

1. Keadaan infeksi

Ditekankan bahwa terjadi interaksi yang sinergis antara malnutrisi dengan penyakit infeksi. Mekanisme patologisnya dapat bermacam-macam, baik secara

sendiri-sendiri maupun bersamaan, yaitu penurunan asupan zat gizi akibat kurangnya nafsu makan, menurunnya absorpsi dan kebiasaan mengurangi makan pada saat sakit, peningkatan kehilangan cairan/zat gizi akibat penyakit diare, mual/muntah dan pendarahan terus menerus serta meningkatnya kebutuhan baik dari peningkatan kebutuhan akibat sakit dan parasit yang terdapat dalam tubuh.

2. Konsumsi makan

Pengukuran konsumsi makan sangat penting untuk mengetahui kenyataan apa yang dimakan oleh masyarakat dan hal ini dapat berguna untuk mengukur status gizi dan menemukan faktor diet yang dapat menyebabkan malnutrisi.

2.2.2 Faktor Tidak Langsung

1. Pengaruh budaya

Hal-hal yang perlu diperhatikan dalam pengaruh budaya antara lain sikap terhadap makanan, penyebab penyakit, kelahiran anak, dan produksi pangan. Sikap terhadap makanan seperti terdapat pantangan, tahayul, dan tabu dalam masyarakat menyebabkan konsumsi makanan menjadi rendah. Jarak kelahiran anak yang terlalu dekat dan jumlah anak yang terlalu banyak akan mempengaruhi asupan gizi dalam keluarga. Konsumsi zat gizi keluarga yang rendah, juga dipengaruhi oleh produksi pangan. Rendahnya produksi pangan disebabkan karena para petani masih menggunakan teknologi yang bersifat tradisional.

2. Pola pemberian makanan

Program pemberian makanan tambahan merupakan program untuk menambah nutrisi pada balita, biasanya diperoleh saat mengikuti posyandu. Adapun pemberian makanan tambahan tersebut berupa makanan pengganti ASI yang biasa didapat dari puskesmas setempat.

3. Faktor sosial ekonomi

Data sosial ini meliputi keadaan penduduk di suatu masyarakat, keadaan keluarga, pendidikan, perumahan, penyimpanan makanan, air dan kakus. Data ekonomi meliputi pekerjaan, pendapatan keluarga, kekayaan yang terlihat seperti

tanah, jumlah ternak, perahu, mesin jahit, kendaraan dan sebagainya serta harga makanan yang tergantung pada pasar dan variasi musim. Di Indonesia yang jumlah pendapatan penduduk sebagian besar adalah golongan rendah dan menengah akan berdampak pada pemenuhan bahan makanan terutama makanan yang bergizi

4. Pola Asuh Keluarga

Pola asuh adalah pola pendidikan yang diberikan orang tua kepada anak-anaknya. Setiap anak membutuhkan cinta, perhatian, kasih sayang yang akan berdampak terhadap perkembangan fisik, mental dan emosional.

5. Produksi pangan

Data yang relevan untuk produksi pangan adalah penyediaan makanan keluarga, sistem pertanian, tanah, peternakan dan perikanan serta keuangan.

6. Pelayanan kesehatan dan pendidikan

Pelayanan kesehatan meliputi ketersediaan pusat-pusat pelayanan kesehatan yang terdiri dari kecukupan jumlah rumah sakit, jumlah tenaga kesehatan, jumlah staf dan lain-lain. Fasilitas pendidikan meliputi jumlah anak sekolah, remaja dan organisasi karang tarunanya serta media massa seperti radio, televisi dan lain-lain.

2.3 Metode TOPSIS

TOPSIS (*Technique for Order of Preference by Similarity to Ideal Solution*) adalah salah satu metode pengambilan keputusan multikriteria atau alternatif pilihan yang merupakan alternatif yang mempunyai jarak terkecil dari solusi ideal positif dan jarak terbesar dari solusi ideal negatif dari sudut pandang geometris dengan menggunakan jarak Euclidean. Namun, alternatif yang mempunyai jarak terkecil dari solusi ideal positif, tidak harus mempunyai jarak terbesar dari solusi ideal negatif. Maka dari itu, TOPSIS mempetimbangkan keduanya, jarak terhadap solusi ideal positif dan jarak terhadap solusi ideal negatif secara bersamaan [5]. Solusi optimal dalam metode TOPSIS didapat dengan menentukan kedekatan relatif suatu alternatif terhadap solusi ideal positif. TOPSIS akan merangking alternative berdasarkan prioritas nilai kedekatan relatif suatu alternative terhadap solusi ideal positif. Alternatif-alternatif yang telah dirangking kemudian dijadikan sebagai referensi bagi pengambil keputusan untuk memilih solusi terbaik yang diinginkan.

Berikut adalah langkah langkah dari metode TOPSIS sebagai berikut : [5]

1. Menggambarkan alternatif (m) dan kriteria (n) ke dalam sebuah matriks, dimana X_{ij} adalah pengukuran pilihan dari alternatif ke-i dan kriteria ke-j yang dapat dilihat pada persamaan 1.

$$D = \begin{matrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ x_{m1} & x_{m1} & \cdots & x_{mn} \end{matrix} \quad (1)$$

2. Membuat matriks R yaitu matriks keputusan ternormalisasi. Dimana nilai dari setiap elemen matriks didapat dari persamaan 2.

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^m x_{ij}^2}} \quad (2)$$

3. Membuat pembobotan pada matriks yang telah dinormalisasi dapat dilihat pada persamaan 3.

$$D = \begin{matrix} w_1 r_{11} & w_1 r_{12} & \cdots & w_1 r_{1n} \\ w_1 r_{21} & w_2 r_{22} & \cdots & w_2 r_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ w_j r_{m1} & w_j r_{m1} & \cdots & w_j r_{mn} \end{matrix} \quad (3)$$

4. Menentukan nilai solusi ideal positif dan solusi ideal negatif. Solusi ideal dinotasikan A^+ , sedangkan solusi ideal negatif dinotasikan A^- . Dimana nilai dari setiap solusi didapat dari persamaan 4 dan 5.

$$A^+ = [y_1^+ \quad y_2^+ \quad \cdots \quad y_n^+] \quad (4)$$

$$A^- = [y_1^- \quad y_2^- \quad \cdots \quad y_n^-] \quad (5)$$

5. Menghitung jarak dari suatu alternatif ke solusi ideal positif didapat dengan menggunakan persamaan 6.

$$S_i^+ = \sqrt{\sum_{j=1}^n (y_i^+ - y_{ij})^2} \quad (6)$$

6. dan alternatif solusi ideal negative didapat dengan menggunakan persamaan 7.

$$S_i^- = \sqrt{\sum_{j=1}^n (y_{ij} y_i^+ - y_i^{-+})^2} \quad (7)$$

7. Menghitung nilai preferensi untuk setiap alternatif. Untuk menentukan ranking tiap-tiap alternatif yang ada maka perlu dihitung terlebih dahulu nilai prefensi dari tiap alternative menggunakan persamaan 8.

$$A_i = \frac{s_i^-}{s_i^- + s_i^+} \quad (8)$$

Nilai A_i yang lebih besar menunjukkan bahwa alternative tersebut yang akan dipilih.

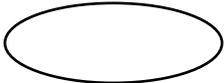
2.4 UML

Unified Modeling Language (UML) adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem [6]. Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut :

1. *Use Case Diagram*

Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *Use Case Diagram* dapat dilihat pada tabel 2.1.

Tabel 2.1 Use Case Diagram

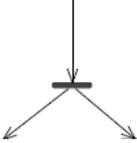
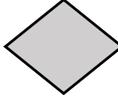
Gambar	Keterangan
	<p><i>Use Case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktif, yang dinyatakan dengan menggunakan kata kerja</p>

Gambar	Keterangan
	<p><i>Actor</i> atau Aktor adalah <i>Abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>Use Case</i>, tetapi tidak memiliki kontrol terhadap <i>use case</i></p>
	<p>Asosiasi antara aktor dan <i>use case</i>, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan data.</p>
	<p>Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem</p>
	<p><i>Include</i>, merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program</p>
	<p><i>Extend</i>, merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi</p>

2. Diagram Aktivitas (*Activity Diagram*)

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity Diagram* dapat dilihat pada tabel 2.2.

Tabel 2.2 Simbol Activity Diagram

Gambar	Keterangan
	<i>Start Point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktivitas
	<i>End Point</i> , akhir aktivitas
	<i>Activities</i> , menggambar kan suatu proses/kegiatan bisnis
	<i>Fork/percabangan</i> , digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu
	<i>Join</i> (penggabungan) atau <i>rake</i> , digunakan untuk menunjukkan adanya dekomposisi
	<i>Decision Points</i> , menggambar kan pilihan untuk pengambilan keputusan, <i>true</i> atau <i>false</i>
	<i>Swimlane</i> , pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa

3. Diagram Urutan (*Sequence Diagram*)

Sequence Diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *Sequence Diagram* dapat dilihat pada tabel 2.3.

Tabel 2.3 Simbol Sequence Diagram

Gambar	Keterangan
	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data

Gambar	Keterangan
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interfaces</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan form entry dan form cetak
	<i>Control class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek
	<i>Message</i> , simbol mengirim pesan antar <i>class</i>
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri
	<i>Activation</i> , mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivasi sebuah operasi
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i>

4. Diagram Kelas (*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. *Class Diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan.

Class Diagram secara khas meliputi : Kelas (*Class*), Relasi *Associations*, *Generalization* dan *Aggregation*, atribut (*Attributes*), operasi (*operation/method*) dan *visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *Multiplicity* atau *Cardinality*

2.5 Website

Secara teknis, Web adalah sebuah sistem dengan informasi yang disajikan dalam bentuk teks, gambar, suara dan lain-lain yang tersimpan dalam sebuah server Web Internet yang disajikan dalam bentuk hiperteks. Informasi Web dalam bentuk teks umumnya ditulis dalam format HTML (Hypertext Markup Language). Informasi lainnya disajikan dalam bentuk grafis (dalam format GIF, JPG, PNG), suara (dalam format AU, WAV), dan objek multimedia lainnya (seperti MIDI, Shockwave, Quicktime Movie, 3D World) [7].

2.6 PHP

Hypertext Preprocessor (PHP) adalah bahasa *server-side scripting* yang menyatu dengan HTML untuk membuat halaman web yang dinamis PHP banyak dipakai untuk pemrograman situs WEB dinamis. Karena PHP merupakan *server-side scripting* maka sintaks dan perintah-perintah PHP akan dieksekusi di server kemudian hasilnya dikirim ke *browser* dalam format HTML[8]. Dengan demikian kode program yang ditulis dalam PHP tidak akan terlihat oleh user sehingga keamanan halaman web lebih terjamin. PHP dirancang untuk membentuk suatu tampilan berdasarkan permintaan terkini, seperti menampilkan isi basis data ke halaman web. Beberapa kelebihan PHP dari bahasa pemrograman web, antara lain:

1. Bahasa pemrograman PHP adalah sebuah bahasa script yang tidak melakukan sebuah kompilasi dalam penggunaannya.
2. PHP memiliki tingkat akses yang lebih cepat.
3. PHP memiliki tingkat *lifecycle* yang cepat sehingga selalu mengikuti perkembangan teknologi internet.
4. PHP juga mendukung akses ke beberapa database yang sudah ada baik yang bersifat free/gratis ataupun komersial. Database itu antara lain : MySQL, PostgreSQL, infomix, dan MicrosoftSQL Server. Web server yang mendukung PHP dapat ditemukan dimana mana dari mulai Apache, IIS, AOServer, phttp, Fhttp, PWS, Lighttpd hingga Xitami dengan konfigurasi yang relative mudah.

Dalam sisi pengembangan lebih mudah, karena banyaknya milis-milis dan developer yang siap membantu dalam pengembangan.

2.7 MySQL

MySQL (*MY Structure Query Language*) adalah salah satu Basis Data *Management Sistem* (DBMS) dari sekian banyak DBMS seperti Oracle, MS SQL, Postagre SQL, dan lainnya. MySQL berfungsi untuk mengolah Basis Data menggunakan bahasa SQL. MySQL bersifat *open source* sehingga kita bisa menggunakannya secara gratis. Pemrograman PHP juga sangat mendukung atau mensupport dengan Basis Data MySQL[9].

Sedangkan MySQL merupakan Basis Data yang paling digemari dikalangan programmer web, dengan alasan bahwa program ini merupakan Basis Data yang sangat kuat dan cukup stabil untuk digunakan sebagai media penyimpanan data. Sebagai sebuah Basis Data server yang mampu untuk memenajemen Basis Data dengan baik, mysql terhitung merupakan Basis Data yang paling digemari dan paling banyak digunakan dibanding Basis Data lainnya. Selain mysql masih terdapat beberapa jenis Basis Data server yang juga memiliki kemampuan yang juga tidak bisa dianggap enteng, Basis Data itu adalah Oracle dan PostgreSQL.

MySQL dibuat oleh TcX dan telah dipercaya mengelola sistem dengan 40 buah database berisi 10,000 tabel dan 500 di antaranya memiliki 7 juta baris (kirakira 100 gigabyte data). Database ini dibuat untuk keperluan sistem database yang cepat, handal dan mudah digunakan. Walaupun memiliki kemampuan yang cukup baik, MySQL untuk sistem operasi Unix bersifat freeware, dan terdapat versi shareware untuk sistem operasi windows. Menurut pembuatnya, MySQL disebut seperti "myessqueell".

Sebagaimana database sistem yang lain, dalam SQL juga dikenal hierarki server dengan database – database. Tiap – tiap database memiliki tabel – tabel, tiap – tiap tabel memiliki field – field. Umumnya informasi tersimpan dalam tabel – tabel yang secara logic merupakan struktur dua dimensi terdiri atas baris dan kolom. Field – field tersebut dapat berupa data seperti int , realm char, date, time dan lainnya. SQL tidak memiliki fasilitas pemrograman yang lengkap, tidak ada looping ataupun percabangan ,misalnya. Sehingga untuk menutupi kelemahan ini perlu digabung dengan bahasa pemrograman semisal C.

2.8 Laravel

Laravel dirilis dibawah lisensi MIT dengan kode sumber yang sudah disediakan oleh Github, sama seperti *framework-framework* yang lain, Laravel dibangun dengan konsep MVC (*Model-Controller-View*), kemudian Laravel dilengkapi juga *command line tool* yang bernama “**Artisan**” yang bisa digunakan untuk *packaging bundle* dan *instalasi bundle* melalui command prompt. Maka tidak heran jika *Framework* Laravel berbasis PHP ini digadang-gadang menjadi primadona *framework* bagi programmer PHP untuk membuat aplikasi aplikasi yang lebih elegan dan dinamis. Karena *framework* ini menekankan kesederhanaan dan fleksibilitas pada desain-nya, maka dari itu pengguna *framework* ini meningkat dari tahun ke tahun[10]. Kelebihan *Framework* Laravel antara lain adalah :

1. *Expressif*

Laravel adalah framework PHP yang *expressif*, artinya ketika melihat suatu sintaks Laravel, seorang programmer diharaplan akan langsung tahu kegunaan dari sintaks tersebut meskipun belum pernah mempelajarinya apalagi menggunakannya.

2. *Simple*

Salah satu yang membuat Laravel begitu *simple* adalah dengan adanya Eloquent ORM. Misalnya kita ingin mengambil semua data yang ada di dalam table users, maka yang diperlukan hanya membuat sebuah class model bernama user.

3. *Accessible*

Laravel dibuat dengan dokumentasi yang selengkap mungkin. *Code Developernya* dari Laravel sendiri berkomitmen untuk selalu menyertakan dokumentasi yang lengkap setiap kali rilis versi terbarunya.

Beberapa fitur yang dimiliki oleh *framework* Laravel adalah sebagai berikut:

4. *Bundles* yaitu sebuah itur dengan sistem pengemasan modular dan berbagai *bundle* telah tersedia untuk di gunakan dalam aplikasi Anda.
5. *Eloquent ORM* merupakan penerapan PHP lanjutan dari pola “*active record*” menyediakan metode internal untuk mengatasi kendala hubungan antara objek database. Pembangun *query* Laravel *Fluent* didukung *Eloquent*.

6. *Application Logic* merupakan bagian dari aplikasi yang dikembangkan, baik menggunakan *Controllers* maupun sebagai bagian dari deklarasi *Route*. Sintaks yang digunakan untuk mendefinisikannya mirip dengan yang digunakan oleh *framework Sinatra*.
7. *Reverse Routing*, mendefinisikan hubungan antara *Link* dan *Route*, sehingga jika suatu saat ada perubahan pada *route* secara otomatis akan tersambung dengan link yang relevan. Ketika *Link* yang dibuat dengan menggunakan nama-nama dari *Route* yang ada, secara otomatis *Laravel* akan membuat *URI* yang sesuai.
8. *Restful Controllers*, memberikan sebuah option (pilihan) untuk memisahkan logika dalam melayani *HTTP GET* dan permintaan *POST*.
9. *Class Auto Loading*, menyediakan otomatis loading untuk *class-class PHP*, tanpa membutuhkan pemeriksaan manual terhadap jalur masuknya. Fitur ini mencegah loading yang tidak perlu.
10. *View Composers* adalah kode unit logical yang dapat dijalankan ketika sebuah *View* di load.
11. *IoC Container* memungkinkan untuk objek baru yang dihasilkan dengan mengikuti prinsip control pembalik, dengan pilhan contoh dan referensi dari objek baru sebagai *Singletons*.
12. *Migrations* menyediakan versi sistem control untuk skema database, sehingga memungkinkan untuk menghubungkan perubahan adalah basis kode aplikasi dan keperluan yang dibutuhkan dalam merubah tata letak database, mempermudah dalam penempatan dan memperbaiki aplikasi.
13. *Unit Testing* mempunyai peran penting dalam *framework Laravel*, dimana unit testing ini mempunyai banyak tes untuk mendeteksi dan mencegah regresi. Unit testing dapat dijalankan melalui fitur "*artisan command -line*".
14. *Automatic pagination* menyederhanakan tugas dari penerapan halaman, menggantikan penerapan yang manual dengan metode otomatis yang terintegrasi ke *Laravel*.

2.9 Visual Studio Code

Visual Studio Code adalah *Software* yang sangat ringan, namun kuat editor kode sumbernya yang berjalan dari desktop. Muncul dengan built-in dukungan

untuk *JavaScript*, naskah dan Node.js dan memiliki *array* beragam ekstensi yang tersedia untuk bahasa lain, termasuk C ++, C # , Python, dan PHP [11]. Hal ini didasarkan sekitar Github ini Elektron, yang merupakan versi *cross-platform* dari Atom komponen kode-editing, berdasarkan *JavaScript* dan HTML5. Editor ini adalah fitur lengkap lingkungan pengembangan terpadu (IDE) dirancang untuk pengembang yang bekerja dengan teknologi cloud yang terbuka Microsoft. Visual Studio Code menggunakan open source NET perkakas untuk memberikan dukungan untuk ASP.NET C # kode, membangun alat pengembang Omnisharp NET dan *compiler Roslyn*. Antarmuka yang mudah untuk bekerja dengan, karena didasarkan pada gaya explorer umum, dengan panel di sebelah kiri, yang menunjukkan semua file dan folder Anda memiliki akses ke panel editor di sebelah kanan, yang menunjukkan isi dari file yang telah dibuka. Dalam hal ini, editor telah dikembangkan dengan baik, dan menyenangkan pada mata. Ia juga memiliki fungsi yang baik, dengan intellisense dan autocomplete bekerja dengan baik untuk JSON, CSS, HTML, {kurang}, dan Node.js.

Visual Studio Code telah dirancang untuk bekerja dengan alat-alat yang ada, dan Microsoft menyediakan dokumentasi untuk membantu pengembang bersama, dengan bantuan untuk bekerja dengan ASP.NET 5, Node.js, dan Microsoft naskah, serta alat-alat yang dapat digunakan untuk membantu membangun dan mengelola aplikasi Node.js. Visual Studio Code benar-benar sedang ditargetkan pada pengembang JavaScript yang ingin alat pengembangannya lengkap untuk scripting server-side mereka dan yang mungkin ingin usaha dari Node.js untuk kerangka berbasis NET. Visual Studio Code, adalah belum solid, lintas platform kode Editor ringan, yang dapat digunakan oleh siapa saja untuk membangun aplikasi untuk Web.