

The background features a light gray gradient with several circular, mandala-like patterns. These patterns consist of concentric rings divided into segments of various colors including red, orange, yellow, green, and blue. Some patterns are large and prominent, while others are smaller and faded, creating a layered, artistic effect.

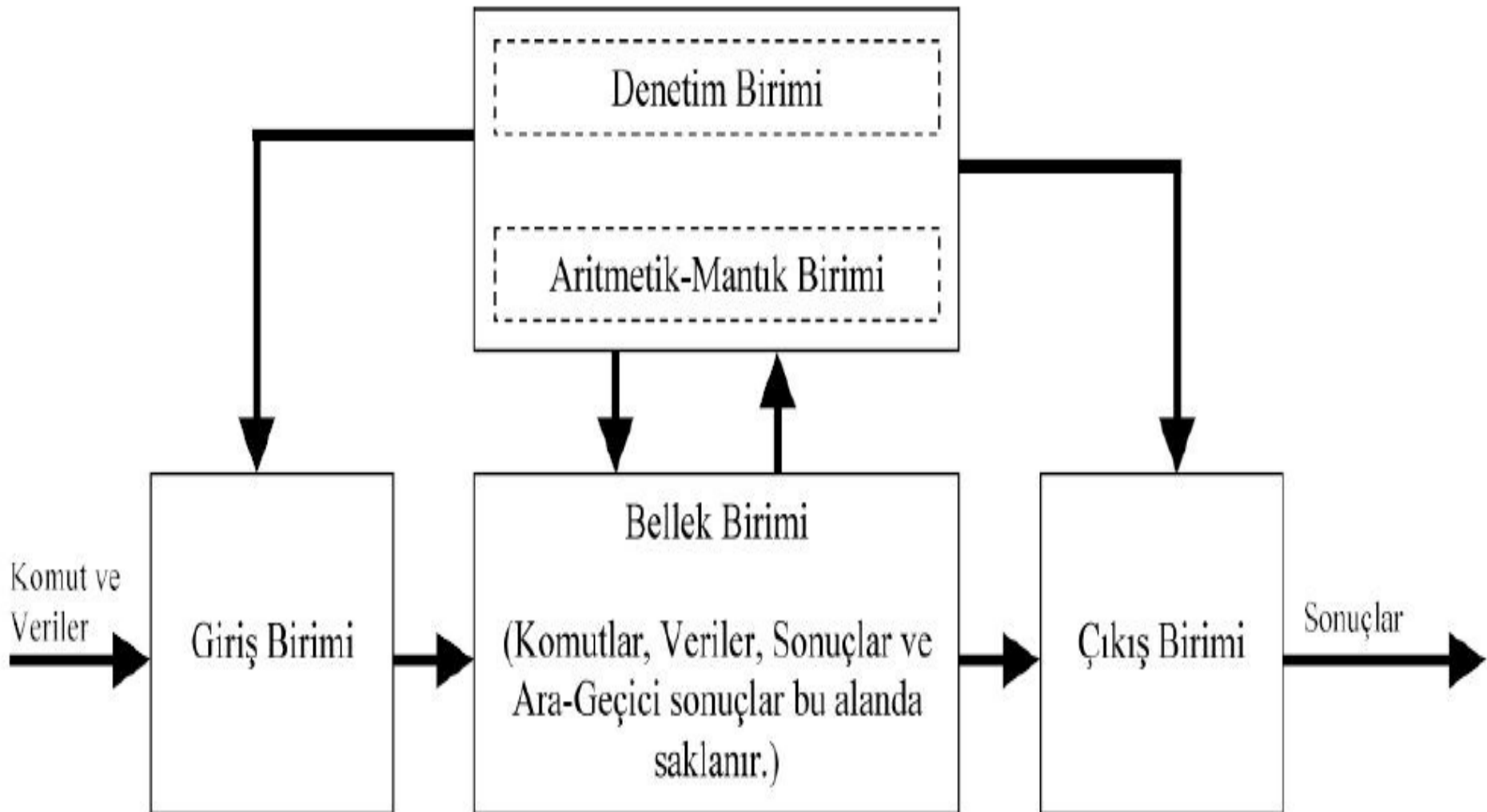
# **PROGRAMLAMA TEMELLERİ**

# Bilgisayar Nedir?

Verileri İşleme  
Verileri saklama  
Verileri aktarma

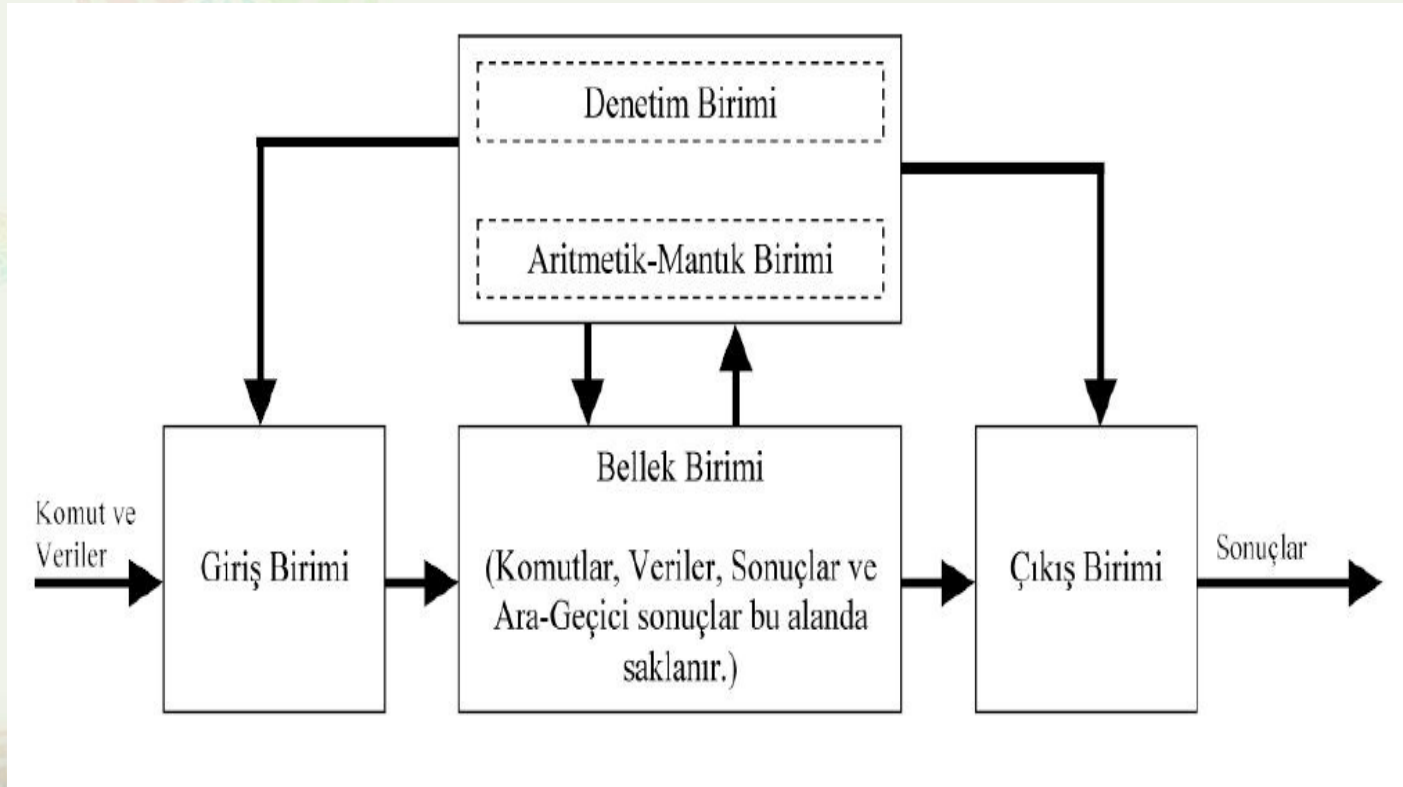


# Bilgisayar Organizasyonu



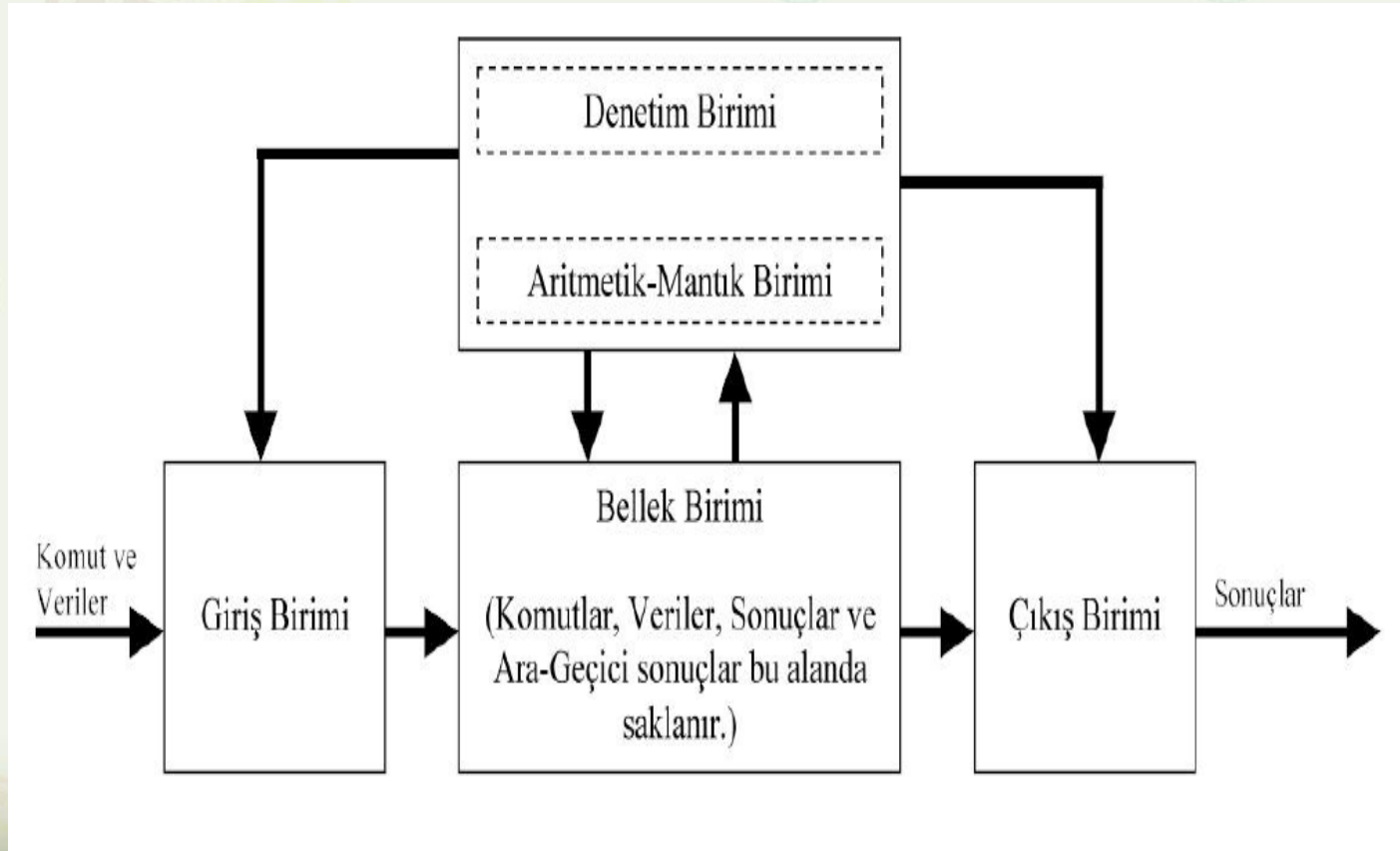
# Giriş Birimi

- Klavye, Fare, mikrofon, kamera vb. cihazlardan biri veya daha fazlasından oluşur. Veri ve komutların bilgisayara yollanmasını sağlar.



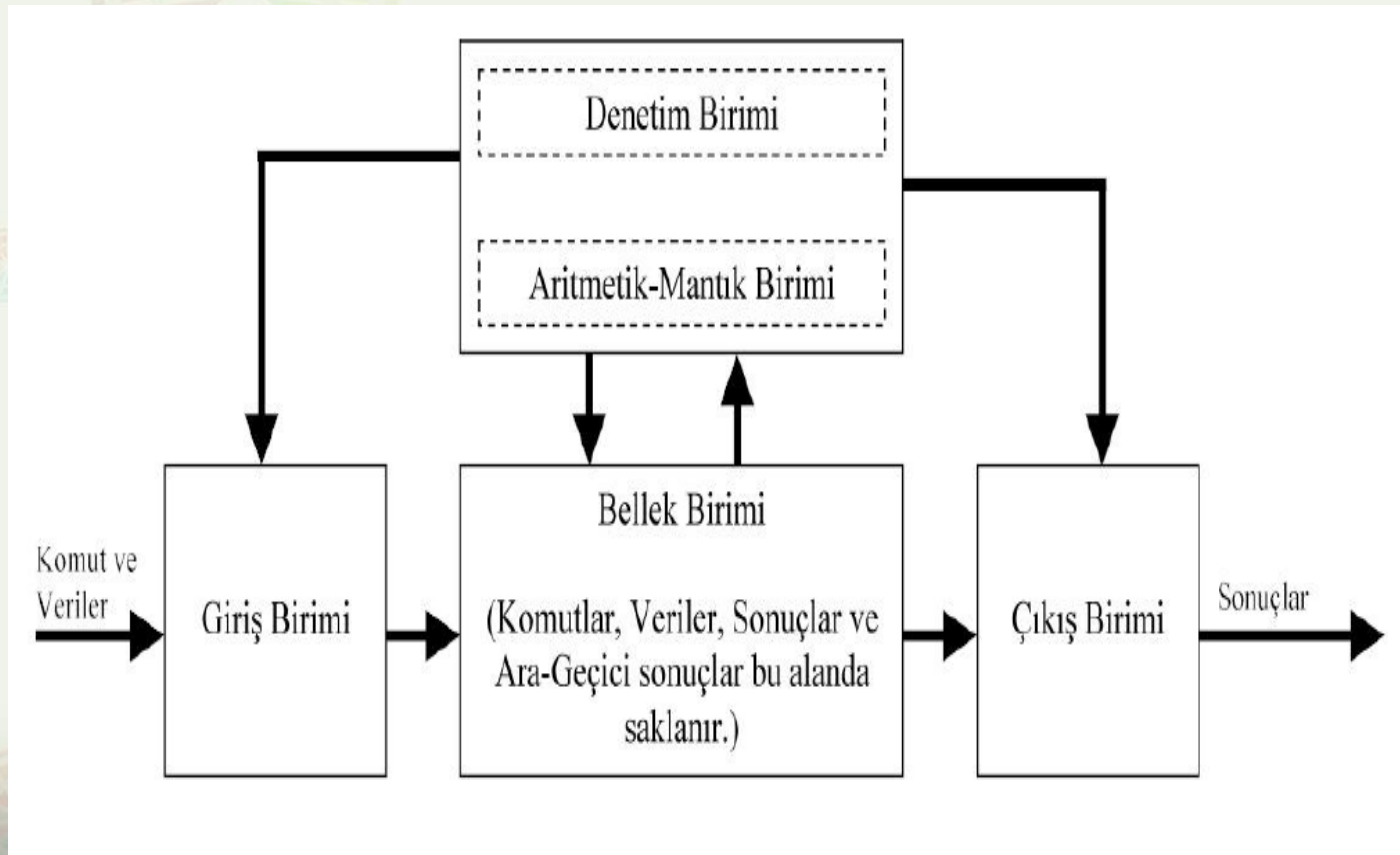
# Bellek Birimi

- Veri, komut ve programların saklandığı donanımları temsil eder.



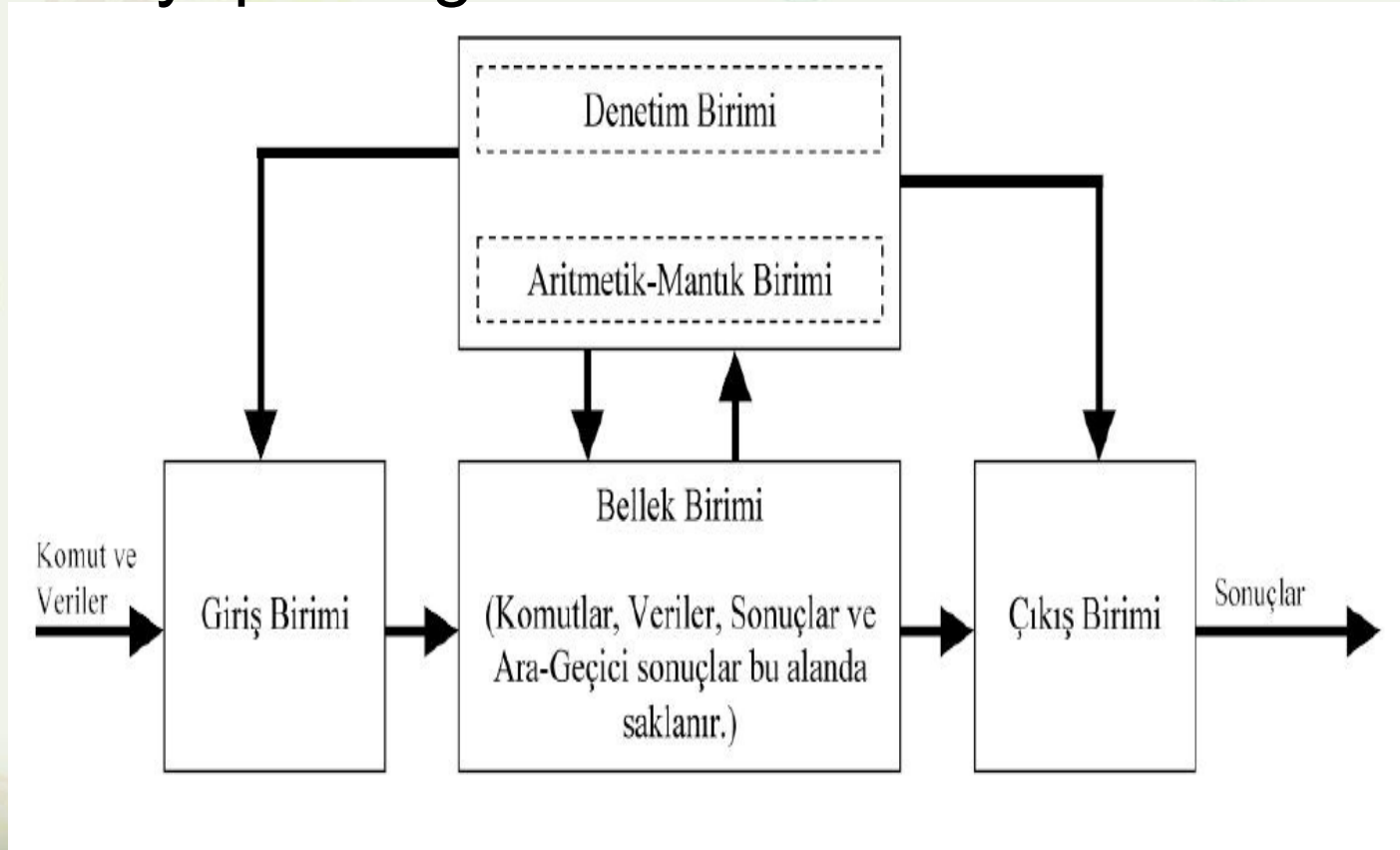
# CPU: Aritmetik-Mantık Birimi

- Bilgisayardaki tüm Aritmetik ve Mantıksal işlemlerin yapıldığı birimdir. Tüm işlemleri yapan ana birimdir.



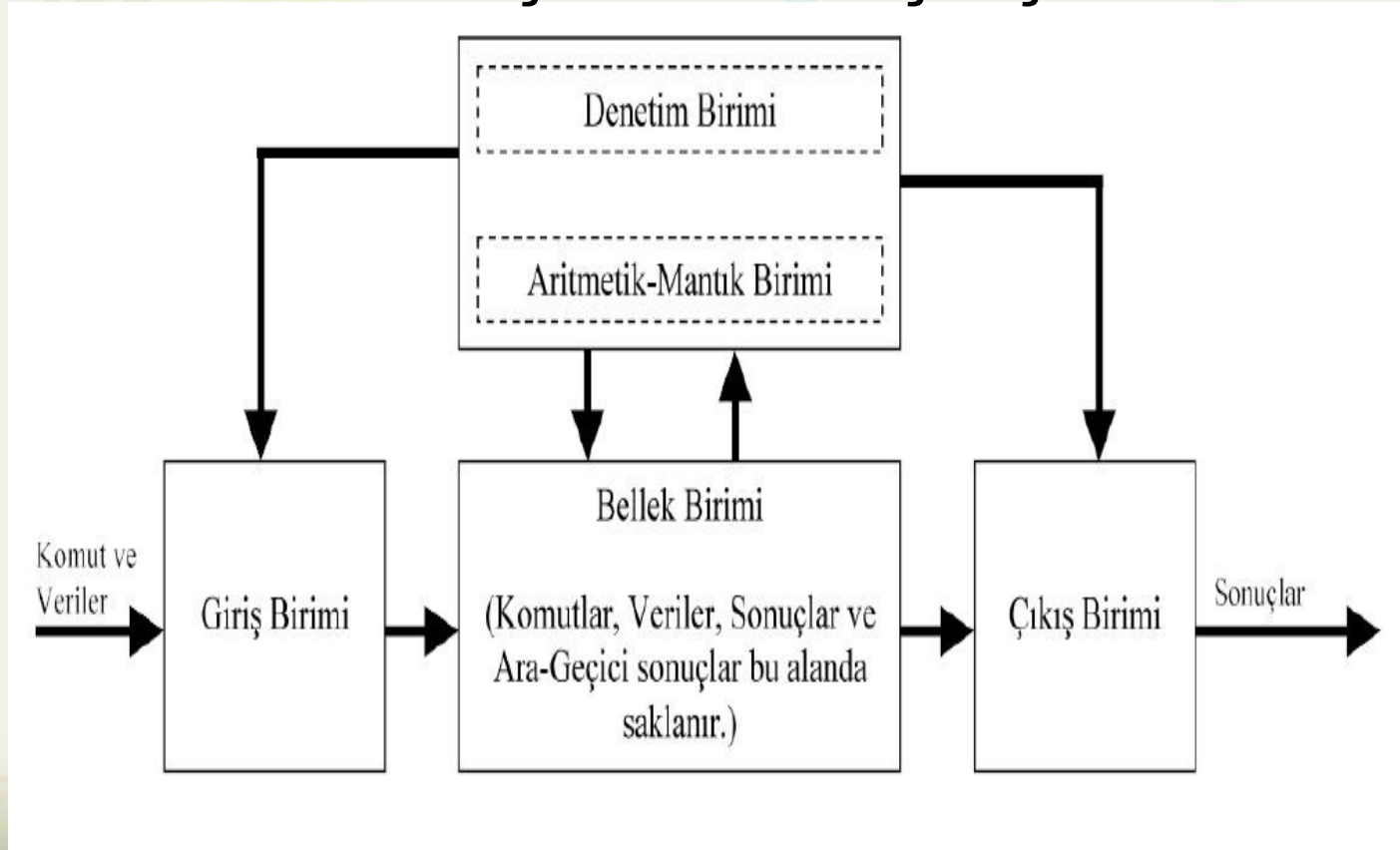
# CPU: Denetim Birimi

- Bilgisayar programının çalışmasını denetleyen, programda yapılan hesaplamalara göre hangi işlerin yapılacağına karar veren birimdir.



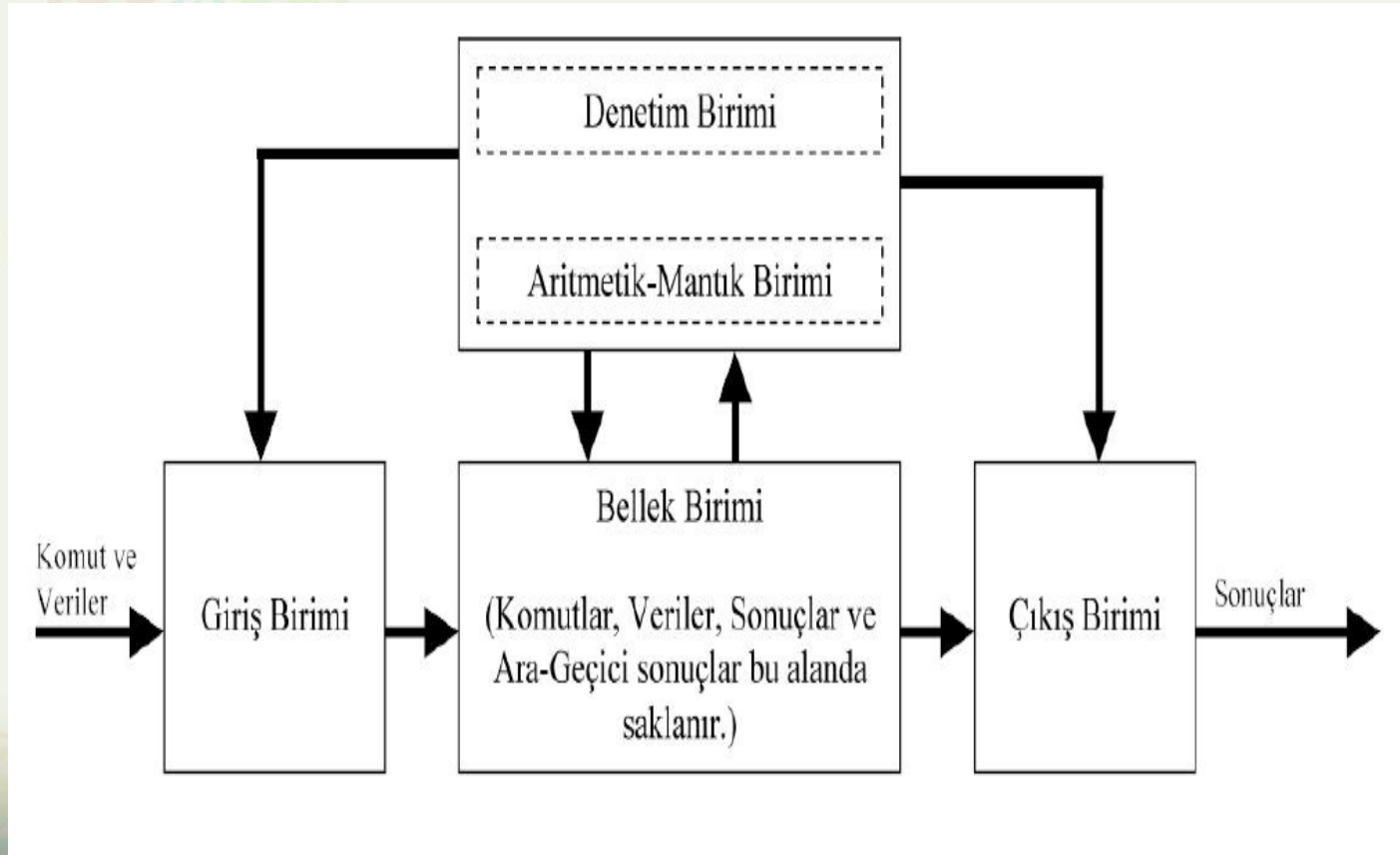
# CPU: Denetim Birimi

- Makine diline çevrilmiş bir programda komutları teker teker ve sırayla Ana işlem birimine getirip yorumlar ve sonuçta komutu çalıştırır.



# Çıkış Birimi

- Bilgisayar ortamında oluşan verilerin dış ortama verilmesini sağlayan birimlerdir. Ekran, yazıcı gibi birimlerdir.



# Bilgisayarın Tarihçesi

M.Ö. 500	Abaküs
1642	Pascalın Mekanik Toplama Makinesi
1827	Babbage'in çıkarma makinesi
1941	İkili>İkili Mekanik Hesaplayıcı (Zuse)
1944	Ondalık Elektromekanik Hesaplayıcı (Aiken)
1945-54	İlk Kuşak Vakum Tüpler ve Işıklar
1955-64	İkinci Kuşak Transistörler ve Manyetik Bellekler
1965-71	Üçüncü Kuşak Tümlşik Devreler
1971-90	Dördüncü Kuşak VLSI Devreler
1982	IBM Pc & MS-DOS
1984	MAC
1990'lar	Paralel İşlemciler, Yapay Zeka, İnternet ve WWW

# Problem Nedir?

- Bir işlemin,
- otomasyonun
- bilimsel hesaplamanın

bilgisayarla çözülmesi fikrinin ortaya çıkmasına problem denir.

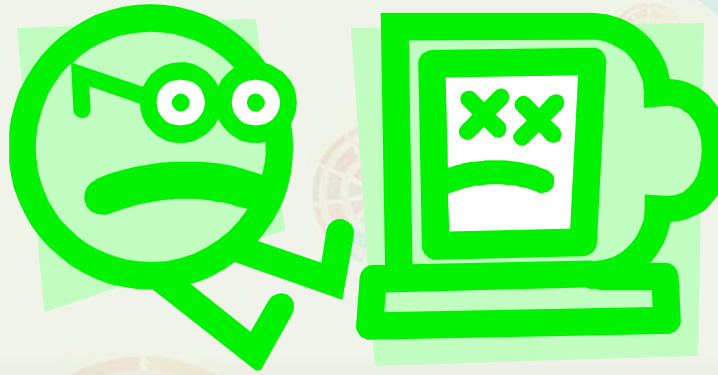
# Problem Çözümü

- Soruyu anla
- İhtiyaç ve istekleri belirle
- Zihinsel alıştırmalar; bilgisayara yakın olan çözüm yolları
- Oluşturulan adımlar algoritmaya dökülmeli
- Akış Çizgesi oluşturulmalı
- Programlama dili seç
- Bilgisayar ortamına aktar
- Test Et



# Program Nedir?

- Problem Çözümü kısmında anlatılan adımlar uygulandıktan sonra ortaya çıkan ve sorunumuzu bilgisayar ortamında çözen ürüne Program denir.
- Bazı durumlarda bu ürüne yazılım denebilir



# Programlama Nedir?

- Çok iyi tanımlanmış bir sorunun
- çözümüne dair adımlar ile çözümün oluşturulup
- bunun bir programlama dili ile bilgisayar ortamına aktarılması

Programlama diye adlandırılabilir.



# Akış Çizgesi Nedir?

- Bir algoritmanın daha görsel gösterimidir. Çizgiler, Dörtgen, daire vb. geometrik şekillerle algoritmanın gösterilmesini sağlar.
- Doğal dille yazılmadığı için daha formal olduğu düşünülebilir.



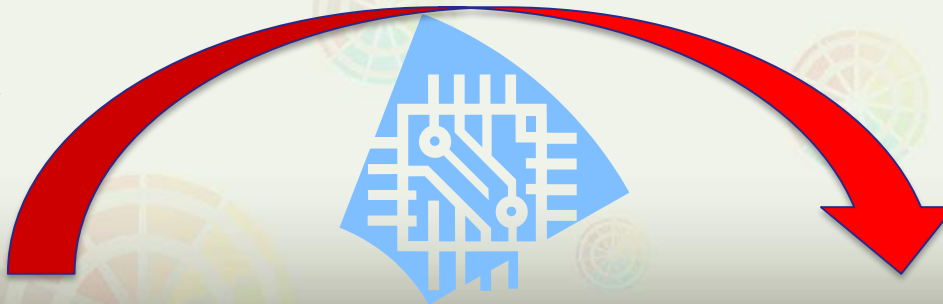
# Programlama Dili Nedir?

- Bir Problemin Algoritmik çözümünün Bilgisayara anlatılmasını sağlayan, son derece sıkı-sıkıya kuralları bulunan kurallar dizisidir.



# Derleyici Nedir?

- Bir programlama dili ile bilgisayara aktarılan programın bilgisayarın anlayabileceği Makine Diline çevirmeyi sağlayan ve yazılan programda söz dizim hatalarının olup olmadığını bulan olup olmadığını bulan yazılımlardır.
- Her Programlama dili için bir derleyici olması gerekmektedir.



# Yorumlayıcı Nedir?

- Derleyici gibi çalışan ancak yazılmış programları o anda Makine diline çeviren yazılımlardır.
- Bu tür bir yazılımda Programın Makine dili ile oluşturulmuş kısmı bilgisayarda tutulmaz. Programın her çalıştırılmasında her adım için Makine dili karşılıkları oluşturulur ve çalıştırılır.



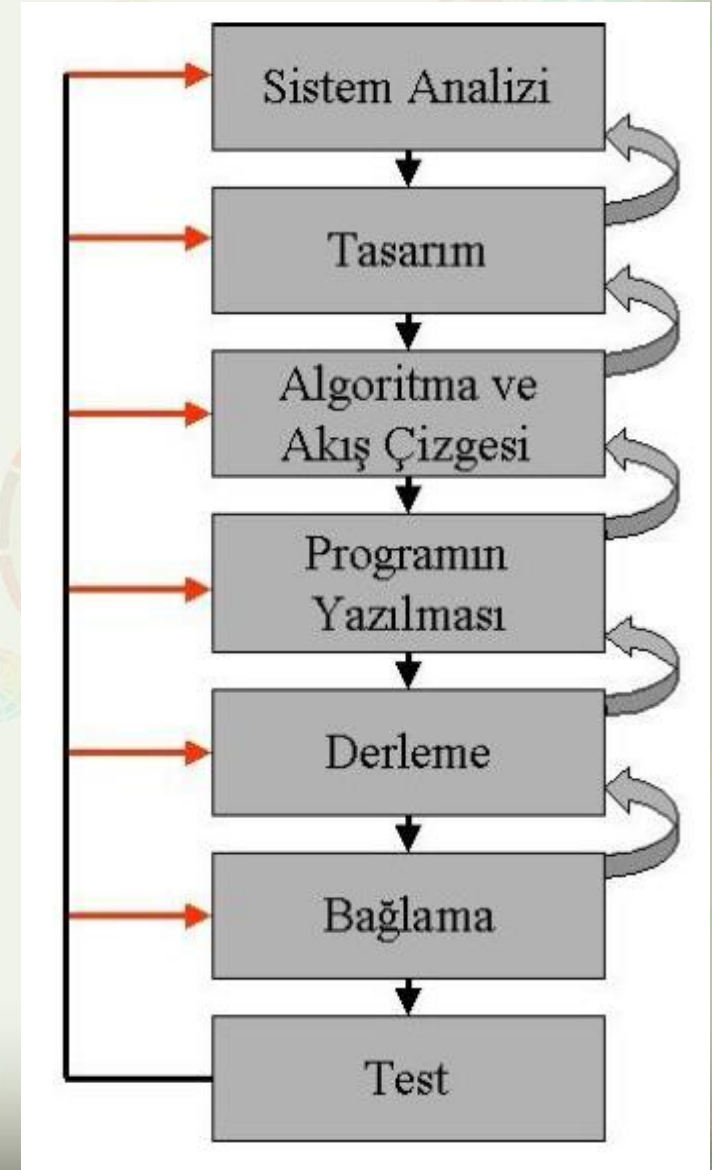
The background features a light gray gradient with several overlapping circular patterns. These patterns consist of concentric rings divided into segments of various colors including red, orange, yellow, green, and blue. The patterns are arranged in a way that some are more prominent than others, creating a layered effect.

# **PROGRAMLAMA TEMELLERİ**

## Yazılım Geliştirme

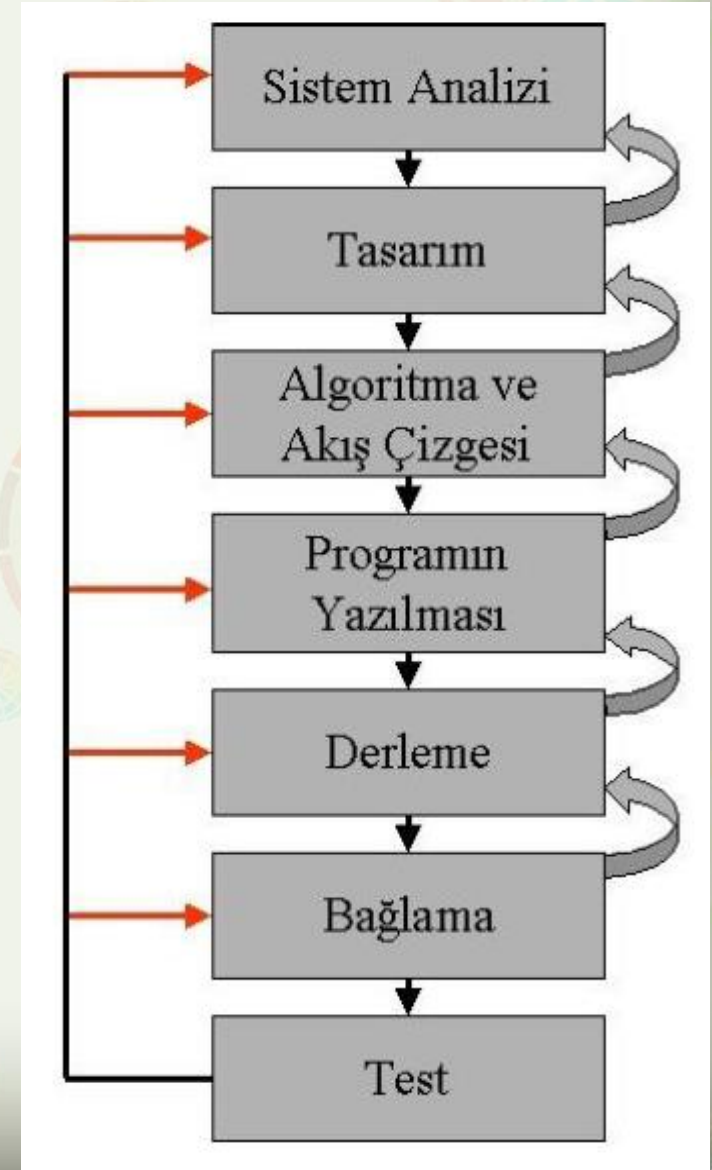
# YAZILIM GELİŞTİRME

- Bu çizgeden anlaşılacağı gibi adımlardan birinde bir sorunla karşılaşılsa bu sorunu çözebilmek için bir önceki adıma geri dönmek gerekecektir.
- Bu geri dönüş bazen bir kaç adım olabilir.



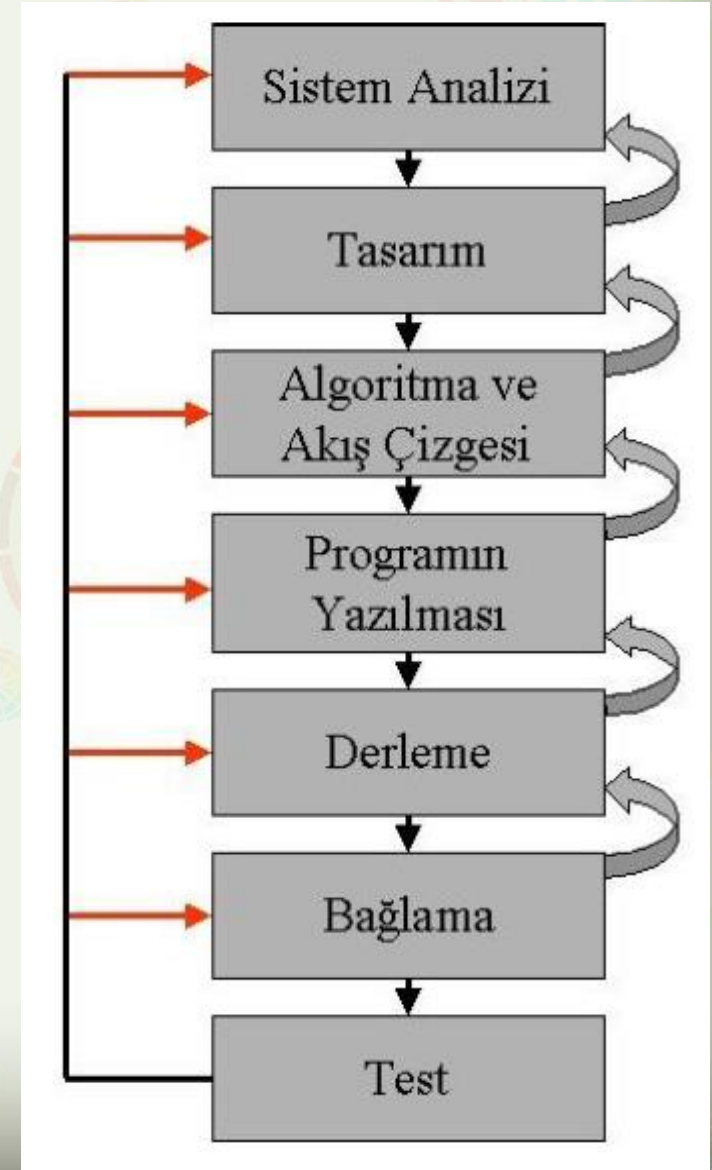
# YAZILIM GELİŞTİRME

- **Sistem Analizi** : Sorunun çözülebilmesi için tamamen anlaşılmasını sağlayan çalışmalardır.
- **Tasarım** : İsteklerle ilgili olarak belirlenen bir takım çözümlerin tanımlanmasıdır.



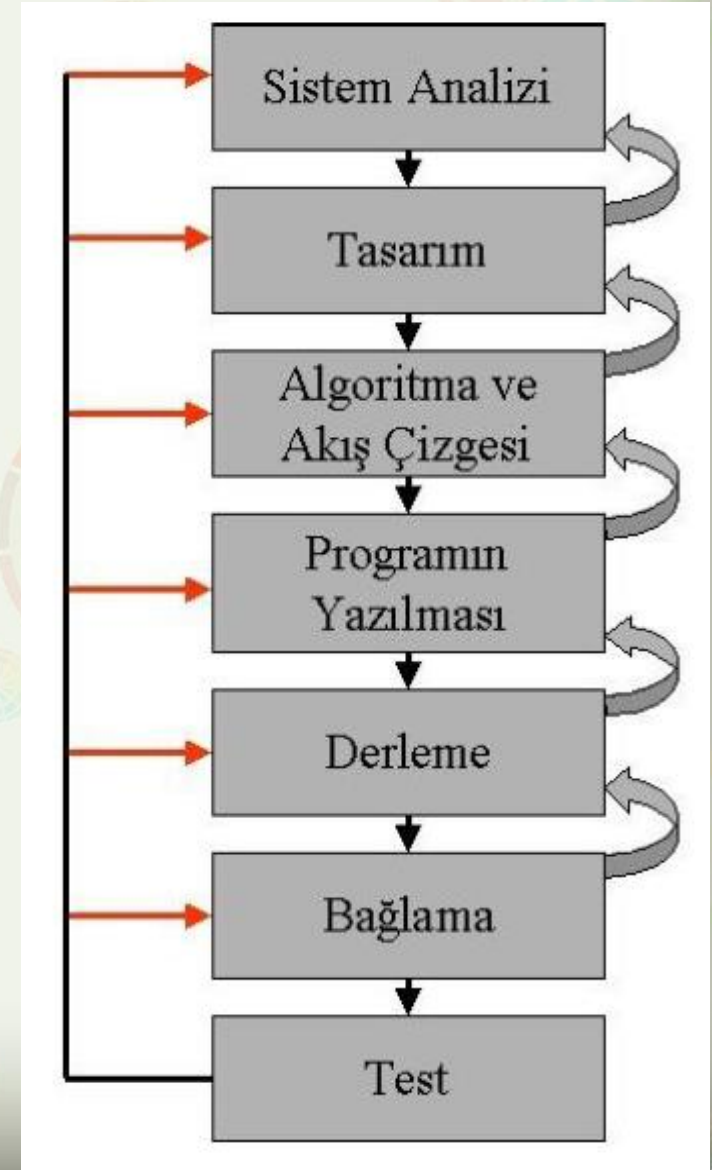
# YAZILIM GELİŞTİRME

- **Algoritma** : Çözümün adımlarla ifade edilmesidir.
- **Akış Çizgesi** : Algoritmanın şekillerle ifade edilmesidir.
- **Programlama Dili Seçimi** : Çözümün netleşmesinden sonra yapılacak işlemleri kolay bir şekilde bilgisayar ortamına aktaracak dilin seçilmesidir. Önemli olan bu dilin özelliklerinin programcı tarafından iyi bilinmesidir.



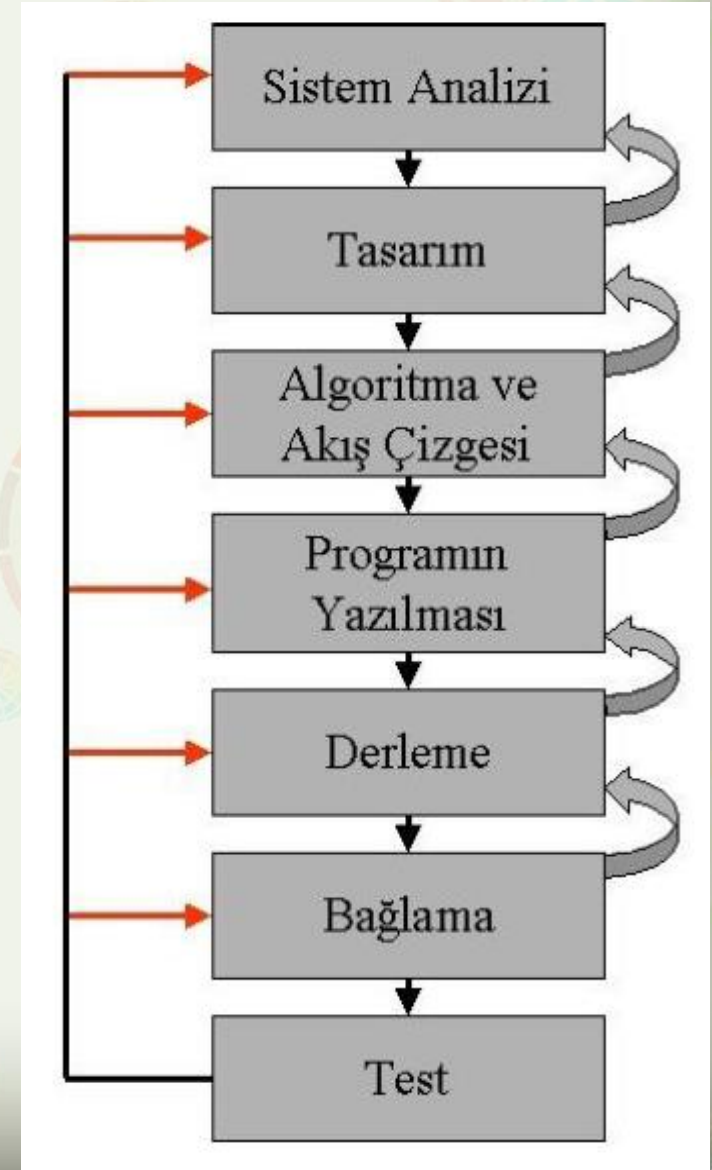
# YAZILIM GELİŞTİRME

- **Programın Yazılması :** Seçilen Programlama dilinin kuralları kullanılarak program yazılmaya başlanır. bu amaçla çoğunlukla sade bir metin editörü kullanılır.
- Bazı durumlarda Syntax highlighting denilen bir özelliğe sahip olan daha akıllı editörler de kullanılabilir.
- Bazen de editör ile Programlama dilinin derleyicisinin, bağlayıcısının hatta hata ayıklayıcısının iç içe bulunduğu IDE (Integrated Development Environment) denilen türde derleyiciler kullanılır.



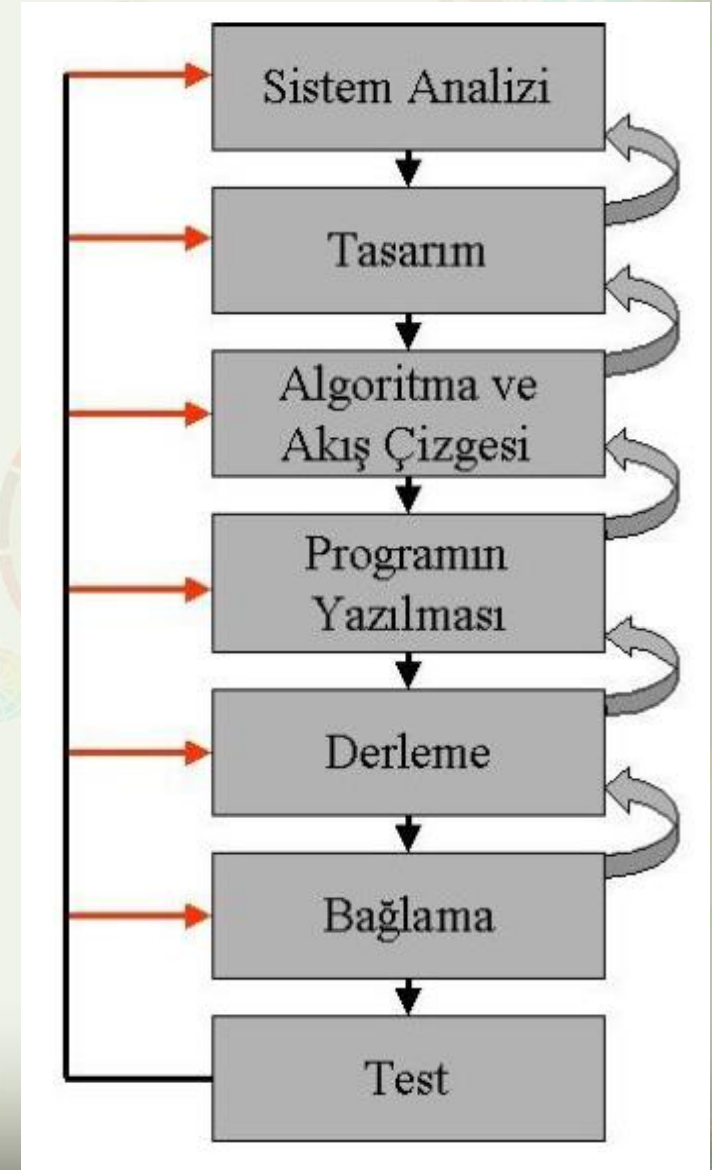
# YAZILIM GELİŞTİRME

- **Derleme** : Programlama Dili ile yazılmış programın yazım hatalarının olup olmadığının kontrol edilmesini ve ara kod olarak Obje kodun üretilmesini sağlama adıımıdır.
- **Bağlama** : Derlenmiş ara kod diğer kütüphane ve parça programlarla birleştirilerek Makine dilinde programın oluşturulması adıımıdır. Ancak bazı IDE ortamlarda ve derleyicilerde Derleme ve Bağlama bir bütündür ve beraberce halledilirler. Programcının ayrıca bir bağlama işlemi yapması gerekmez işlemi yapması gerekmez.



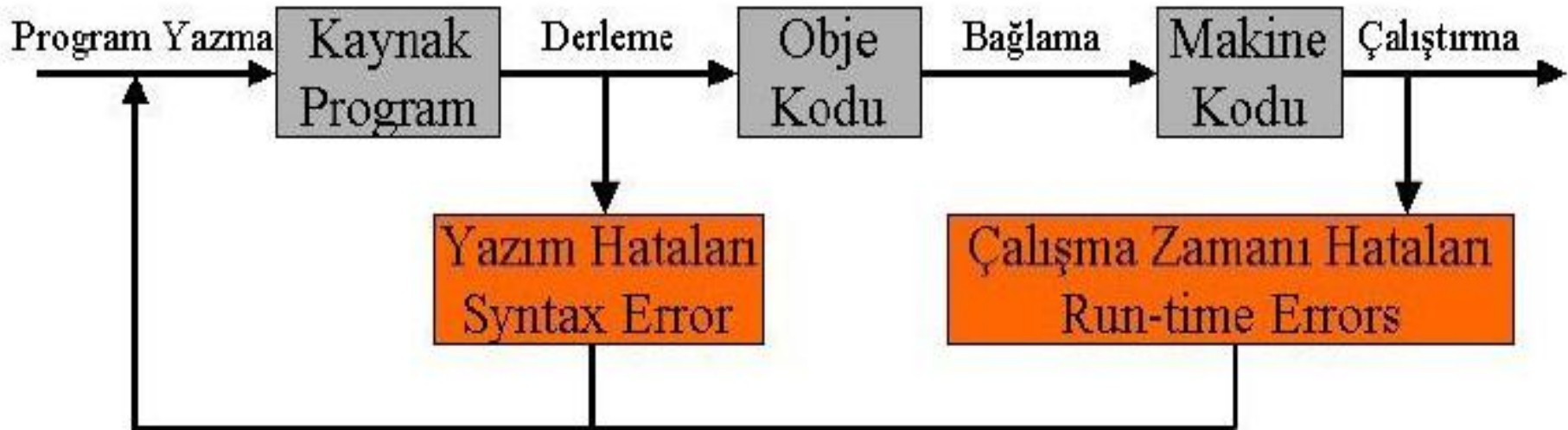
# YAZILIM GELİŞTİRME

- **Çalıştırma** : Oluşturulan Makine dili Programının çalıştırılması adımıdır. Yukarıdaki adımların hepsi yolunda gittiyse program sorunsuz olarak çalışabilmelidir.
- **Test** : Programın Mantıksal olarak test edilmesini sağlar ve içerik olarak her ihtimal için doğru sonuçlar üretip üretmediğini kontrol etmenizi sağlar.



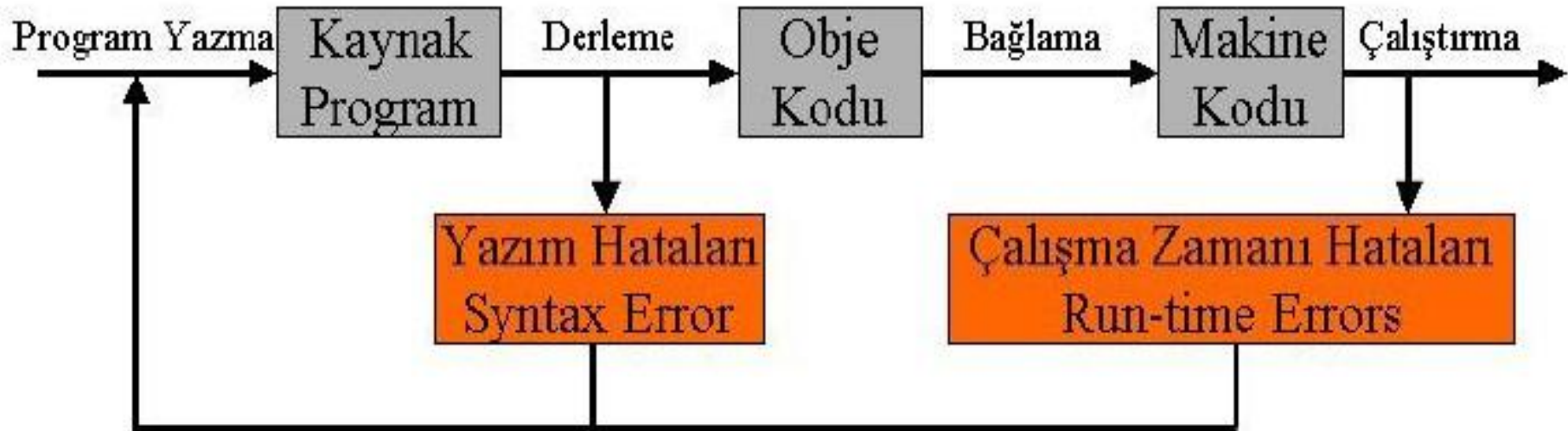
# YAZILIMIN YAŞAM DÖNGÜSÜ

- **Syntax Error** : Yazılan programda programlama dili kurallarına aykırı bir takım ifadelerden dolayı karşılaşılabilecek hatalardır.
- Düzeltilmesi son derece basit hatalardır. Hatanın bulunduğu satır derleyici tarafından rapor edilir. Hatta bazı derleyiciler hatanın ne olduğunu ve nasıl düzeltilmesi gerektiğini dahi bildirebilirler. Bazen Syntax Error tipi hataları Bağlama zamanında da ortaya çıkabilir. Eğer bir derlemede Syntax Error alındı ise obje kod üretilmemiştir demektir.



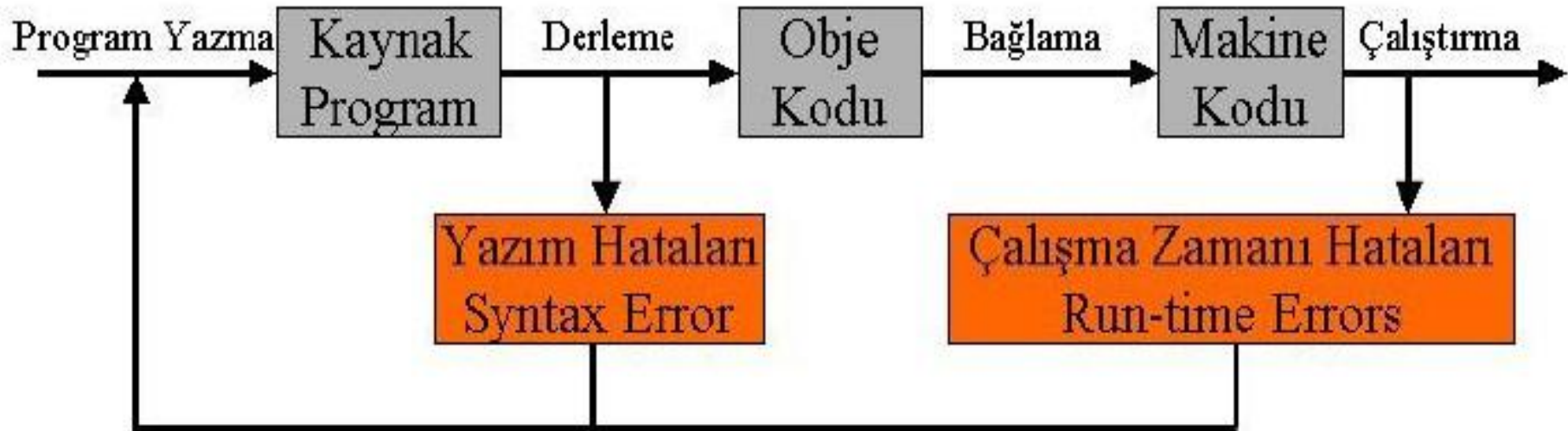
# YAZILIMIN YAŞAM DÖNGÜSÜ

- **Run-time Error** : Programın çalıştırılması sırasında karşılaşılan hatalardır. Programcının ele almadığı bir takım aykırı durumlar ortaya çıktığında programın işletim sistemi tarafından kesilmesi ile ortaya çıkar. Bu tip hatalarda hata mesajı çoğunlukla çalışan işletim sisteminin dili ile verilir. Eğer bu tip hataları kullanıcı ele almışsa, program programcının vereceği mesajlarla ve uygun şekilde sonlandırılabilir.



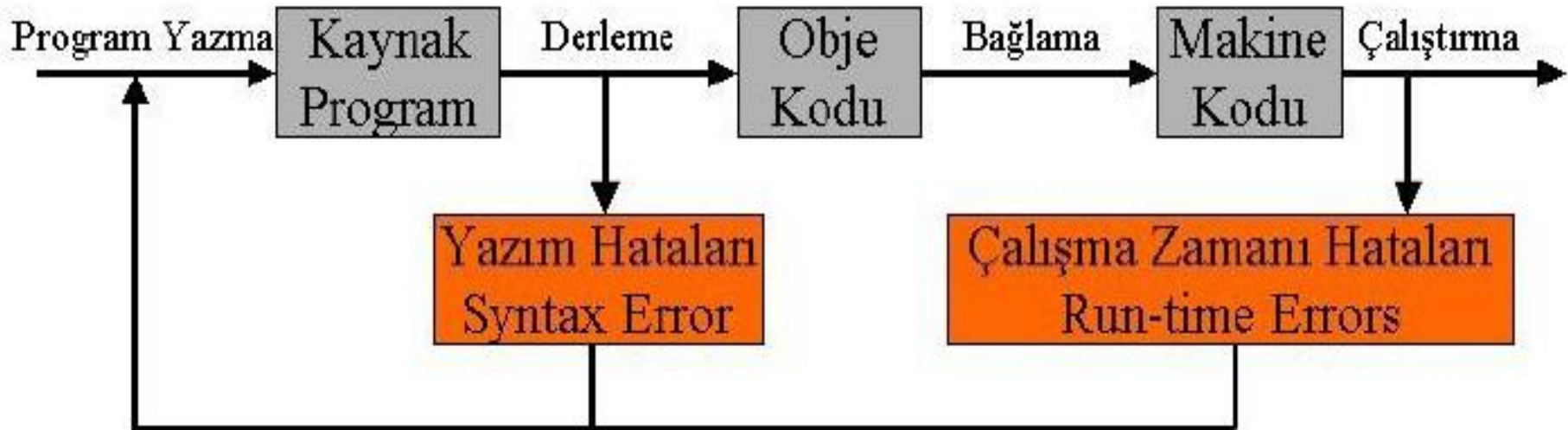
# YAZILIMIN YAŞAM DÖNGÜSÜ

- **Run-time Error** : Bu tip hataların nerelerde ve hangi şartlarda ortaya çıkabileceğini bazen kestirmek zor olabilir. Çoğunlukla işletim sistemi ve donanım kaynakları ile ilgili sorunlarda bu tip hatalar ortaya çıkar demiştik. Örneğin olamayan bir dosya açmaya çalışmak, var olan bir dosyanın üzerine yazmaya çalışmak, olmayan bir bellek kaynağından bellek ayırtmaya çalışmak, olmayan bir donanıma ulaşmaya çalışmak vs. vs. vs.



# YAZILIMIN YAŞAM DÖNGÜSÜ

- **Logical Error** : Karşılaşılabileceğiniz en tehlikeli hatadır. Programlama mantığında bir takım şeylerin yanlış düşünülmesinden kaynaklanır. Hata test aşamasında ortaya çıkar. Hesaplanması gereken veya bulunması gereken eğerlerin eksik veya yanlış hesaplanması ile tespit edilir. Bu sorunun giderilebilmesi için Tasarım hatta çözümleme aşamasına geri dönülmesi gerekebilir. Bazen bu hatanın nereden kaynaklandığını bulabilmek çok zor olmaktadır.



# YAZILIMIN YAŞAM DÖNGÜSÜ

## Bug :

- Logical Error diyebileceğimiz Mantıksal hatalara verilen adlar bug yani böcek diye de tanımlanmış olabilir.
- Bu tip hatalar eğer çok net değil ve zamanla ortaya çıkabiliyor ise veya nedeni çok net olarak anlaşılamamışsa bug diye adlandırılır.
- Gerek serbest yazılım gerek ticari yazılımların tümünde bug dediğimiz mantıksal hatalar bulunur. Çünkü hatasız program yazabilmek çok zordur. İlk seferde yazılan bir programın tamamen hatasız olmasını beklemek son derece hatalıdır.
- Günümüzde en meşhur yazılım firmaları bile yazılımlarında bug olduğunu kabul eder ve zaman zaman bu bugları giderebilmek için ya yazılımlarına yama yazılımı üretirler yada o yazılımın yeni bir versiyonunu piyasaya sürerler.

# YAZILIMIN YAŞAM DÖNGÜSÜ

## Debug :

- Mantıksal hataları giderebilmek ve yazılımdaki bug'ları bulabilmek için yapılan işlemin adıdır.
- Genellikle yazılan programın adım adım ve denetim altında çalıştırılmasıdır.
- Programın her adımında ilgili değişkenlerin hangi değere sahip olduğunu görmeyi sağlar ve anormal bir durumu daha kolay izleyip bulmanızı sağlar.
- Bu işlemi gerçekleştirebilmek için bazı IDE ortamlarında debugger dediğimiz yardımcı komut veya yaz



HAZIRLAYAN  
ÖĞRETİM GÖREVLİSİ  
**NACİYE**  
MACİT SEZİKLİ

